



Universidade Federal de Pernambuco

Mestrado em Ciência da Computação

**COMPARAÇÃO DE ARQUITETURAS DE
REDES NEURAIS PARA SISTEMAS DE
RECONHECIMENTO DE PADRÕES EM
NARIZES ARTIFICIAIS**

por

Aida Araújo Ferreira

Dissertação de Mestrado

RECIFE-PE
Março de 2004



Universidade Federal de Pernambuco

Mestrado em Ciência da Computação

COMPARAÇÃO DE ARQUITETURAS DE REDES NEURAIIS PARA SISTEMAS DE
RECONHECIMENTO DE PADRÕES EM NARIZES ARTIFICIAIS

por

Aida Araújo Ferreira

Dissertação de Mestrado

Orientadora: Prof.^a Dr.^a TERESA BERNARDA LUDERMIR

RECIFE-PE
Março de 2004

COMPARAÇÃO DE ARQUITETURAS DE REDES NEURAIIS PARA SISTEMAS DE RECONHECIMENTO DE PADRÕES EM NARIZES ARTIFICIAIS¹

Autora: Aida Araújo Ferreira

Orientadora: Dra. Teresa Bernarda Ludermir

RESUMO

Um nariz artificial é um sistema modular composto de duas partes principais: um sistema sensor, formado de elementos que detectam odores e um sistema de reconhecimento de padrões que classifica os odores detectados. Redes neurais artificiais têm sido utilizadas como sistema de reconhecimento de padrões para narizes artificiais e vêm apresentando resultados promissores.

Desde os anos 80, pesquisas para criação de narizes artificiais, que permitam detectar e classificar odores, vapores e gases automaticamente, têm tido avanços significativos. Esses equipamentos podem ser utilizados no monitoramento ambiental para controlar a qualidade do ar, na área de saúde para realizar diagnóstico de doenças e nas indústrias de alimentos para o controle de qualidade e o monitoramento de processos de produção.

Esta dissertação investiga a utilização de quatro técnicas diferentes de redes neurais para criação de sistemas de reconhecimento de padrões em narizes artificiais. O trabalho está dividido em quatro partes principais: (1) introdução aos narizes artificiais, (2) redes neurais artificiais para sistema de reconhecimento de padrões, (3) métodos para medir o desempenho de sistemas de reconhecimento de padrões e comparar os resultados e (4) estudo de caso.

Os dados utilizados para o estudo de caso, foram obtidos por um protótipo de nariz artificial composto por um arranjo de oito sensores de polímeros condutores, expostos a nove tipos diferentes de aguarrás. Foram adotadas as técnicas Multi-Layer Perceptron (MLP), Radial Base Function (RBF), Probabilistic Neural Network (PNN) e Time Delay Neural Network (TDNN) para criar os sistemas de reconhecimento de padrões. A técnica PNN foi investigada em detalhes, por dois motivos principais:

esta técnica é indicada para realização de tarefas de classificação e seu treinamento é feito em apenas um passo, o que torna a etapa de criação dessas redes muito rápida. Os resultados foram comparados através dos valores dos erros médios de classificação utilizando o método estatístico de Teste de Hipóteses.

As redes *PNN* correspondem a uma nova abordagem para criação de sistemas de reconhecimento de padrões de odor. Estas redes tiveram um erro médio de classificação de **1.1574%** no conjunto de teste. Este foi o menor erro obtido entre todos os sistemas criados, entretanto mesmo com o menor erro médio de classificação, os testes de hipóteses mostraram que os classificadores criados com *PNN* não eram melhores do que os classificadores criados com a arquitetura *RBF*, que obtiveram um erro médio de classificação de **1.3889%**. A grande vantagem de criar classificadores com a arquitetura *PNN* foi o pequeno tempo de treinamento dos mesmos, chegando a ser quase imediato. Porém a quantidade de nodos na camada escondida foi muito grande, o que pode ser um problema, caso o sistema criado deva ser utilizado em equipamentos com poucos recursos computacionais. Outra vantagem de criar classificadores com redes *PNN* é relativa à quantidade reduzida de parâmetros que devem ser analisados, neste caso apenas o parâmetro relativo à largura da função *Gaussiana* precisou ser investigado.

Palavras-chave: Narizes Artificiais, Redes Neurais Artificiais, Reconhecimento de Padrões, Multi-Layer Perceptron (MLP), Radial Base Function (RBF), Probabilistic Neural Network (PNN), Time Delay Neural Network (TDNN), Teste de Hipóteses.

ⁱ Dissertação de Mestrado em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, 2004.

COMPARING ARCHITECTURE OF NEURAL NETWORKS FOR PATTERN RECOGNATION SYSTEM ON ARTIFICIAL NOSESⁱⁱ

Author: Aida Araújo Ferreira

Adviser: Dra. Teresa Bernarda Ludermir

ABSTRACT

An electronic nose is a modular system composed by two principal parts: a sensor system composed of elements to detect odors and a pattern recognition system to classify the odors. Artificial neural networks have been used to pattern recognition system in artificial nose and have been achieved promising results.

Since 80 years, research to create artificial nose, to consent detect odors, vapors and gas automatic have been expressive advance. These machines can be utilized for monitoring the environment to controlling the air quality, in healthy to help diagnose of diseases and food industry to quality control and process production monitoring.

This dissertation investigates the use of four different neural networks to create pattern recognition systems on artificial noses. The work is dividing on four principal parts: (1) artificial nose introduction, (2) artificial networks to pattern recognition system, (3) methods to evaluate pattern recognition system performance and to compare the results and (4) case study.

The data utilized in case study was obtained for a prototype of artificial nose compound by an array of eight conducting polymer sensors, exposed to nine different aguarrás. There are adopting the technical of Multi-layer Perceptron (MLP), Radial Base Function (RBF), Probabilistic Neural Network (PNN) and Time Delay Neural Networks (TDNN) to create the pattern recognition systems. The PNN technical was investigating for two principal causes: this technical is indicating for classify tasks and his training has doing in one step, become this step very fast. The results were obtained through the classify median errors and were compared with the statistical method hypothesis testing.

The *PNN* networks are a new technical to create pattern recognition system of odors. These networks gain classify median error of **1.1574%** on test set. This was the smallest error gain between all system created, although even with the smallest classify median error, the hypothesis tests showing that the classifier created with *PNN* wasn't better than the classifiers created with *RBF* architecture, they gain a classify median error of **1.3889%**. The principal advantage of create classifier with *PNN* architecture was the little time to training, almost immediate. Although if the number of nodes in the hide layer was very big, this can be a trouble, if the system was used for machines with less computation resources. Another advantage to create classifier with *PNN* networks is related to less quantity of adjust parameters to be analyzed, this case only the parameter related with *Gaussian* function width needs be investigated.

Keywords: Artificial Noses, Artificial Neural Networks, Pattern Recognition, Multi-Layer Perceptron (MLP), Radial Base Function (RBF), Probabilistic Neural Network (PNN), Time Delay Neural Network (TDNN), Hypothesis Tests.

ⁱⁱ Master of Science dissertation in Computer Science, Information Center, Federal University of Pernambuco, Recife, PE, 2004.

ÍNDICE ANALÍTICO

CAPÍTULO 1	1
INTRODUÇÃO	1
1 INTRODUÇÃO.....	1
1.1 <i>Motivação</i>	1
1.2 <i>Objetivos</i>	2
1.3 <i>Organização da dissertação</i>	3
CAPÍTULO 2	5
NARIZ ARTIFICIAL	5
2 INTRODUÇÃO.....	5
2.1 <i>O sistema de olfato humano e os odores</i>	6
2.1.1 Anatomia do sistema olfativo humano.....	7
2.1.2 Fisiologia do sistema olfativo humano.....	9
2.1.3 Identificação dos cheiros.....	11
2.2 <i>Nariz Artificial</i>	11
2.2.1 Sistema sensor.....	15
2.2.2 Sistema de reconhecimento de padrões.....	17
2.3 <i>Desenvolvimento de narizes artificiais</i>	20
2.3.1 Diag-Nose.....	20
2.3.2 Nariz artificial para diagnosticar tuberculose.....	20
2.3.3 Menor nariz artificial.....	21
2.4 <i>Sumário do capítulo</i>	22
CAPÍTULO 3	23
REDES NEURAIS ARTIFICIAIS PARA CRIAÇÃO DE SISTEMAS DE RECONHECIMENTO DE PADRÕES	23
3 INTRODUÇÃO.....	23
3.1 <i>Fundamentos de redes neurais</i>	23
3.2 <i>Redes neurais Perceptron multicamadas</i>	25
3.3 <i>Redes neurais com processamento temporal</i>	27
3.4 <i>Redes RBF</i>	29
3.5 <i>Redes PNN</i>	31
3.6 <i>Parâmetros para treinamento de redes neurais</i>	33
3.7 <i>Revisão de trabalhos relacionados</i>	36
3.8 <i>Sumário do capítulo</i>	39
CAPÍTULO 4	40
MÉTODOS PARA MEDIR O DESEMPENHO DE SISTEMAS DE RECONHECIMENTO DE PADRÕES E PARA COMPARAR OS RESULTADOS	40
4 INTRODUÇÃO.....	40
4.1 <i>Tipos de erro</i>	40
4.2 <i>Métodos para estimar a taxa de erro</i>	45
4.2.1 Estimação da taxa de erro através de treinar-e-testar.....	45
4.2.2 Métodos Resampling.....	46
4.3 <i>Metodologia para avaliação dos classificadores</i>	53
4.3.1 Teoria da amostragem.....	54
4.3.2 Hipóteses estatísticas.....	55

4.3.3	Teste de hipóteses e significância	56
4.3.4	Testes que envolvem a distribuição normal	57
4.3.5	Testes unilaterais e bilaterais	58
4.3.6	Comparando duas alternativas	59
4.4	<i>Sumário do capítulo</i>	59
CAPÍTULO 5.....		61
REALIZAÇÃO DOS EXPERIMENTOS PARA ESCOLHA DE ARQUITETURA.....		61
5	INTRODUÇÃO.....	61
5.1	<i>Descrição do problema e da base de dados</i>	61
5.2	<i>Pré-processamento dos dados</i>	65
5.3	<i>Divisão do conjunto de dados</i>	66
5.4	<i>Aspectos observados para escolha de arquitetura</i>	67
5.5	<i>Realização dos experimentos para escolha da arquitetura</i>	69
5.5.1	Experimentos com redes RBF	69
5.5.2	Experimentos com redes PNN	74
5.5.3	Experimentos com redes MLP.....	76
5.5.4	Experimentos com TDNN	80
5.6	<i>Sumário do capítulo</i>	84
CAPÍTULO 6.....		87
EXPERIMENTOS FINAIS		87
6	INTRODUÇÃO.....	87
6.1	<i>Resultado dos experimentos finais</i>	87
6.2	<i>Comparação dos resultados finais</i>	92
6.3	<i>Conclusões</i>	94
CAPÍTULO 7.....		97
CONCLUSÃO		97
7	INTRODUÇÃO.....	97
7.1	<i>Considerações finais</i>	97
7.2	<i>Contribuições deste trabalho</i>	100
7.3	<i>Proposta de trabalhos futuros</i>	101
REFERÊNCIA BIBLIOGRÁFIA.....		102

LISTA DE FIGURAS

Figura 2-1: Localização dos componentes do sistema olfativo nos seres humanos [Yamazaki, 2001].	6
Figura 2-2: Principais componentes do epitélio olfativo.	8
Figura 2-3: Representação esquemática (a) um nariz biológico e (b) um nariz eletrônico.	13
Figura 2-4: Composição de uma nariz artificial.	14
Figura 2-5: Arquitetura básica de um nariz artificial	15
Figura 2-6: Menor nariz artificial.	21
Figura 3-1: Estrutura geral de uma rede neural multicamada, alimentada a diante.	24
Figura 3-2: Um nodo de uma rede neural é mostrado nesta figura. Cada conexão de entrada do nodo tem um valor de peso associado, e o nodo produz uma saída simples que é transmitida para outros nodos.	25
Figura 3-3: Função sigmóide logística.	27
Figura 3-4: Função tangente hiperbólica.	27
Figura 3-5: Exemplo de rede MLP utilizando a técnica de janelas de tempo para processamento temporal. A janela de tempo desta rede tem tamanho igual à p .	28
Figura 3-6: Diagrama básico de uma rede RBF.	30
Figura 3-7: Arquitetura típica de uma rede PNN.	33
Figura 3-8: Saída x Entrada com fator de “alisamento” de 0.05 .	34
Figura 3-9: Saída x Entrada com fator de “alisamento” de 0.08 .	35
Figura 3-10: Saída x Entrada com fator de “alisamento” de 0.1 .	35
Figura 4-1: Esquema para o método 4-fold cross-validation	49
Figura 4-2: Esquema do método leave-one-out cross-validation	50
Figura 4-3: Exemplo do método bootstrapping.	52
Figura 4-4: Distribuição normal reduzida.	57
Figura 5-1: Formação dos $k=10$ conjuntos	67
Figura 5-2: Diagrama de uma rede criada com <i>newrb</i> .	70
Figura 5-3: Diagrama de uma rede criada com <i>newrbe</i> .	73
Figura 5-4: Diagrama das redes <i>PNN</i> .	74
Figura 5-5: Diagrama de uma rede <i>MLP</i> .	78
Figura 5-6: Diagrama de uma rede <i>TDNN</i> com <i>MLP</i> .	80

LISTA DE TABELAS

Tabela 2-1: Métodos utilizados em narizes eletrônicos para criação de sistemas de reconhecimento de padrões.....	18
Tabela 4-1: Exemplo de matriz de confusão para duas classes.....	42
Tabela 4-2: Tipos de erro para classificação com duas classes.....	42
Tabela 4-3: Medidas de desempenho de classificação mais comuns.	43
Tabela 4-4: Exemplo de matriz de custo.....	44
Tabela 4-5: Comparação entre Holdout e Random Subsampling.....	48
Tabela 4-6: Comparação entre métodos cross-validation.....	50
Tabela 5-1: Composição do arranjo de sensores.....	62
Tabela 5-2: Descrição dos tipos de aguarrás <i>Cera Inglesa</i> e quantidade de análises para cada um dos tipos...63	
Tabela 5-3: Quantidade de padrões obtidos por análise/classe de aguarrás <i>Cera Inglesa</i> na fase de coleta de dados.....	64
Tabela 5-4: Lista das base de dados criadas com <i>10-fold cross-validation</i>	67
Tabela 5-5: Resultados com <i>newrb</i> para seleção de <i>GOAL</i> e <i>SPREAD</i>	71
Tabela 5-6: Resultados obtidos com <i>newrbe</i> e para seleção de <i>SPREAD</i>	73
Tabela 5-7: Resultados obtidos com <i>newpnn</i> para seleção de <i>SPREAD</i>	76
Tabela 5-8: Resultados obtidos com <i>MLP</i> para seleção da quantidade de nodos na camada escondida na <i>base de dados 1</i>	79
Tabela 5-9: Resultados obtidos com <i>MLP</i> para seleção da quantidade de nodos na camada escondida <i>base de dados 2</i>	79
Tabela 5-10: Resultados obtidos para <i>TDNN com MLP</i> e <i>base de dados 3</i> para seleção da quantidade de nodos na camada escondida.....	81
Tabela 5-11: Resultados obtidos para <i>TDNN com MLP</i> e <i>base de dados 4</i> para seleção da quantidade de nodos na camada escondida.....	81
Tabela 5-12: Resultados obtidos para <i>TDNN com RBF</i> para seleção de <i>SPREAD</i>	83
Tabela 5-13: Resultados obtidos para <i>TDNN com PNN</i> para seleção de <i>SPREAD</i>	84
Tabela 6-1: Formação dos conjuntos de treinamento, validação e teste para o treinamento das redes <i>RBF com newrb</i> , <i>RBF com newrbe</i> e <i>PNN</i>	88
Tabela 6-2: Resultados obtidos para <i>RBF com newrb</i> , <i>RBF com newrbe</i> e <i>PNN</i>	88
Tabela 6-3: Formação dos conjuntos de treinamento, validação e teste para o treinamento de redes <i>MLP</i> .90	
Tabela 6-4: Resultados finais para as redes <i>MLP</i> com 16 nodos na camada escondida.	90
Tabela 6-5: Resultados obtidos para <i>TDNN com RBF</i> e, <i>TDNN com PNN</i>	91
Tabela 6-6: Resultados finais para as redes <i>TDNN com MLP</i> com 26 nodos na camada escondida.....	92
Tabela 6-7: Comparação estatística entre os classificadores.....	93

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus, que sempre esteve presente iluminando todos os momentos da minha vida.

Agradeço a meu querido marido Ronaldo que foi o meu grande incentivador, seria quase impossível terminar o mestrado sem o seu apoio.

Agradeço à minha mãe Maria José e às minhas filhas Amanda e Ana Clara que sonharam junto comigo com esta dissertação e que souberam compreender os muitos momentos em que estive ausente.

Quero agradecer especialmente a Prof^a Dr^a Teresa Bernarda Ludermir pela orientação no mestrado, pela atenção e pelo incentivo durante a escrita da dissertação.

Agradeço ao colega Akio que me apresentou à minha orientadora e que me ajudou em muitos momentos respondendo vários questionamentos.

Agradeço a todos os meus professores do mestrado por terem fornecido conhecimentos valiosos nas disciplinas cursadas.

Agradeço às minhas irmãs, Ermelinda, Idalina e Fátima que sempre foram exemplos a serem seguidos de dedicação aos estudos.

Agradeço aos meus amigos, especialmente à Ana Cláudia e à Paula Renata, por estarem comigo em muitos momentos importantes do mestrado.

INTRODUÇÃO

1 Introdução

Este capítulo apresenta a motivação, os objetivos traçados para o trabalho e uma visão geral dos capítulos desta dissertação.

1.1 Motivação

Desde os anos 80 pesquisas para criação de sistemas olfativos artificiais, que permitam detectar e classificar odores, vapores e gases automaticamente, têm tido avanços significativos. Esses sistemas, conhecidos como narizes artificiais ou narizes eletrônicos, podem ser utilizados em várias áreas como no monitoramento ambiental, para detectar vazamentos, controlar a qualidade do ar e controlar a quantidade de emissão de gases pelas indústrias. Na área de saúde, para diagnosticar doenças, uma vez que os odores da respiração podem indicar problemas gastrintestinais, infecções, diabetes e doenças do fígado. Na área industrial esses equipamentos podem ser utilizados no controle de qualidade e no monitoramento de processos de produção de alimentos, bebidas e cosméticos.

Os narizes artificiais já foram utilizados em várias áreas com sucesso [Keller et al., 1995a] e seu uso apresenta muitas vantagens quando comparado com a utilização do olfato humano. Os seres humanos estão sujeitos à fadiga, mudanças no estado mental, alergias e doenças que ocasionam mudanças da sensibilidade olfativa, e, além disso, em vários processos existe o risco de inalação de substâncias tóxicas que podem causar problemas à saúde.

De uma maneira geral, um nariz artificial é um sistema modular composto de duas partes: um sistema sensor, formado de elementos que detectam odores e que traduzem a quantidade química detectada em sinais elétricos e um sistema de reconhecimento de padrões que classifica os odores conhecidos ou identifica odores desconhecidos. O sistema sensor pode ser construído a partir de várias tecnologias, como óxido metálico semicondutor ou polímeros condutores. Da mesma forma que os sistemas sensores, os sistemas de reconhecimento de padrões dos narizes artificiais podem ser construídos utilizando diferentes tecnologias. Sistemas de reconhecimento de padrões que utilizam técnicas estatísticas como PCA (Principal

Components Analysis) [Gardner e Barlett, 1992], análise discriminante, análise de clusters, ou técnicas de redes neurais como MLP (Multi Layer Perceptron) [Santos, 2000; Yamazaki, 2004], Redes de função de base radial (RBF) [Santos, 2000; Almeida et al., 2004], Time Delay Neural Network (TDNN) já foram utilizadas [Yamazaki, 2001; Zanchettin, 2004] com sucesso.

A importância dos sistemas de reconhecimento de padrões para os narizes artificiais e a grande variedade de técnicas que podem ser aplicadas para o desenvolvimento dos mesmos são as principais motivações para realização deste trabalho.

1.2 Objetivos

O principal objetivo deste trabalho é estudar a utilização de redes neurais para construção de sistemas de reconhecimento de padrões para narizes artificiais.

Inicialmente é apresentado o sistema olfativo humano com o objetivo de dar uma visão geral da anatomia e a fisiologia do mesmo. Da mesma forma, são apresentados os principais componentes de um nariz artificial com o objetivo de mostrar a arquitetura e o funcionamento desses equipamentos.

No universo de redes neurais existem diversas técnicas que podem ser utilizadas de acordo com o tipo do problema. Para narizes artificiais, várias técnicas já foram adotadas, mas até o momento não existe um consenso sobre qual a técnica mais adequada para o problema de reconhecimento de padrões de odor. Assim, temos o objetivo de apresentar algumas técnicas de redes neurais que foram utilizadas com sucesso.

Uma parte importante na criação de redes neurais é a preparação dos dados em conjuntos de treinamento e teste para os algoritmos de aprendizados. Esta dissertação tem o objetivo de apresentar diversas técnicas para criação desses conjuntos e utilizar uma técnica denominada de *cross-validation* que apesar de ser referenciada pelo relatório Proben [Prechelt, 1994] não é muito utilizada nos trabalhos de redes neurais.

Outra questão sobre a realização dos experimentos de redes neurais é a maneira de comparar os resultados obtidos por diferentes técnicas. Esta dissertação tem também o objetivo de apresentar uma técnica estatística, conhecida como *teste de hipóteses*, para comparar o resultados de diferentes aplicações.

Outro objetivo deste trabalho é validar técnicas de redes neurais utilizadas em outros trabalhos em uma nova base de dados [Santos, 2000; Yamazaki, 2001; Zanchettin, 2004]. A base de dados utilizada nesta dissertação foi obtida pela equipe do projeto AROMA, da Unicap, através da utilização de um protótipo de nariz artificial. A refinaria Gabriel Passos (REGAP/MG) da Petrobrás cedeu amostras de gases correspondentes a nove tipos de aguarrás *Cera Inglesa* para que a base de dados fosse criada. Este é o primeiro trabalho que utiliza redes neurais para criar classificadores para esta base de dados.

Analisar sistemas de reconhecimento de padrões para uma base de dados mais complexa também é um objetivo deste trabalho. Os trabalhos relacionados criaram sistemas que reconhecem no máximo cinco classes diferentes e este trabalho criou sistemas que reconhecem nove classes diferentes.

Esta dissertação propõe a utilização das redes Probabilistic Neural Networks (PNN) para criação de sistemas de reconhecimento de padrões. Essas redes apresentam algumas vantagens em relação a outras técnicas de redes neurais que são: treinamento rápido e pequena quantidade de parâmetros a serem investigados.

1.3 Organização da dissertação

O capítulo 2 mostrar a anatomia e a fisiologia do sistema olfativo humano, o histórico dos narizes artificiais, os principais componentes de um nariz artificial e apresenta alguns protótipos de narizes artificiais encontrados através de pesquisa à literatura.

O capítulo 3 apresenta as técnicas de redes neurais que serão adotadas para criação dos sistemas de reconhecimento de padrões desta dissertação. Dentre as abordagens apresentadas, as redes *PNN* são destacadas por dois motivos principais: técnica indicada para realização de tarefas de classificação e treinamento realizado em apenas um passo, o que torna a etapa de criação dessas redes muito rápida. Este capítulo faz uma revisão de cinco trabalhos que criaram sistemas de reconhecimento de padrões com redes neurais.

O capítulo 4 apresenta medidas que podem ser utilizada para avaliar sistemas de reconhecimento de padrões, métodos utilizados para criação dos conjuntos de treinamento e teste e um método estatístico para comparação dos resultados obtidos com estes sistemas.

O capítulo 5 descreve a base de dados utilizada no estudo de caso, a preparação dos dados, a metodologia utilizada para realização dos experimentos e os resultados obtidos nos experimentos para escolha das melhores arquiteturas de redes MLP, RBF, PNN e TDNN.

O capítulo 6 apresenta os resultados finais que foram obtidos nas melhores arquiteturas de redes MLP, RBF, PNN e TDNN.

O capítulo 7 apresenta as considerações finais do trabalho bem com a descrição das contribuições desta dissertação e algumas propostas de trabalhos futuros.

NARIZ ARTIFICIAL

2 Introdução

O nariz humano ainda é o principal instrumento usado para cheirar ou sentir o sabor em várias indústrias, como nas indústrias de cosméticos, alimentos ou de bebidas. Este processo é muito caro, pois é necessário treinar especialistas e estes só podem trabalhar por curtos períodos de tempo. A utilização de pessoas especializadas para realização dessas atividades apresenta uma série de desvantagens, pois as pessoas estão sujeitas à fadiga, mudanças no estado mental, alergias e doenças que ocasionam mudanças da sensibilidade olfativa e, além disso, há o risco de inalação de substâncias tóxicas que podem causar problemas a saúde dos especialistas[Santos, 2000].

As propriedades físico-químicas dos produtos podem ser medidas usando equipamentos convencionais como cromatografia de gás e espectrômetro de massa. Esses métodos mecânicos consomem muito tempo e também produzem resultados inadequados. Por exemplo, existem alguns componentes no sabor de uma cerveja que estão abaixo do limite de detecção da maioria desses equipamentos. A maior parte do relacionamento entre as propriedades físico-químicas das moléculas odorantes e seu impacto sensorial ainda não está clara e ainda é necessária muita pesquisa para que este relacionamento possa ser completamente definido. Conseqüentemente existe uma demanda muito grande por instrumentos eletrônicos que possam imitar o olfato humano e prover informações sensoriais rápidas e com custo baixo [Gardner e Barlett, 1994].

Desde os anos 1980 pesquisas para criação de sistemas olfativos artificiais, que permitam detectar e classificar odores, vapores e gases automaticamente, têm tido avanços significativos. Esses sistemas podem ser utilizados em várias áreas como no monitoramento ambiental, para detectar vazamentos, controlar a qualidade do ar e controlar a quantidade de emissão de gases pelas indústrias. Na área de saúde, equipamentos capazes reconhecer odores automaticamente podem ser utilizados para diagnóstico de doenças, uma vez que os odores da respiração podem indicar problemas gastrintestinais, infecções, diabetes e doenças do fígado. Na indústria de alimentos, esses equipamentos podem ser utilizados no controle de qualidade e no monitoramento de processos de produção [Yamazaki, 2001].

Este capítulo tem o objetivo de apresentar uma visão geral dos narizes artificiais. A seção 2.1 fala sobre a anatomia e a fisiologia do olfato humano, pois os narizes artificiais foram criados baseados no sistema olfativo humano. A seção [2.2] apresenta um histórico da criação dos narizes artificiais, define o conceito de nariz artificial e identifica os principais componentes de um nariz artificial.

2.1 O sistema de olfato humano e os odores

O olfato é um componente importante nas sensações de cheiro e sabor dos seres humanos, muitas vezes ele é usado sozinho para definir o sabor de vários produtos. A sensação de cheiro chega, através de estímulos, ao sistema olfativo humano por moléculas odorantes emitidas por objetos, como por exemplo, uma rosa. Quando cheiramos uma rosa, as moléculas odoríferas que evaporam e estão fluando no ar entram através das fossas nasais e se encontram com as células receptoras localizadas em uma pequena região entre as sobrancelhas, chamada de epitélio olfativo ou mucosa olfativa clássica [Santos, 2000]. A **Figura 2-1** exibe a localização dos principais componentes do sistema olfativo dos seres humanos.

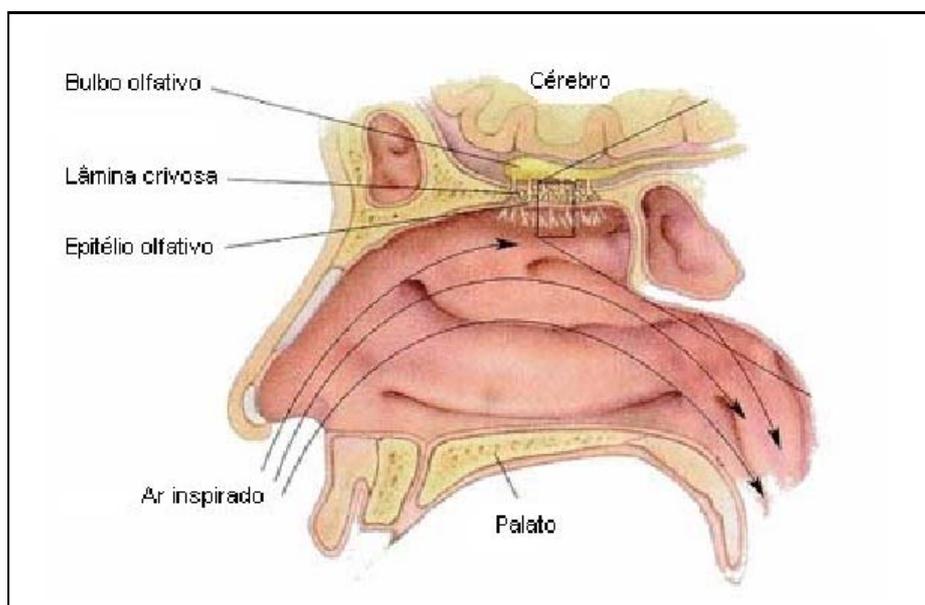


Figura 2-1: Localização dos componentes do sistema olfativo nos seres humanos [Yamazaki, 2001].

No epitélio olfativo há uma fina camada de muco aquoso na qual se estendem os cílios das células olfativas. Células receptoras G, construídas de proteínas, são localizadas na superfície dos cílios e atuam

como receptores químicos. Acredita-se que existe um pequeno número de receptores de proteínas, mas que estas células têm sensibilidade múltipla. As moléculas odorantes são tipicamente hidrofóbicas¹ e polares² com massa molecular de cerca de 300 Da³. Um odor simples é formado de uma única molécula. Na verdade, a maior parte dos cheiros, sabores e perfumes é formada de uma mistura de espécies químicas que contem centenas ou milhares de componentes. Frequentemente a diferença na quantidade desses componentes determina o cheiro do produto.

2.1.1 *Anatomia do sistema olfativo humano*

Alguns cientistas acreditam que existem duas regiões fisicamente distintas no nariz que são responsáveis pela detecção de cheiros: o epitélio olfativo e o órgão vômero nasal. A função do epitélio olfativo é detectar cheiros, enquanto que a função do órgão vômero nasal está relacionada com o comportamento social e com a reprodução [Santos, 2000]. Este trabalho dará ênfase à função do epitélio olfativo, pois esta função está relacionada com os narizes artificiais. Para poder realizar a detecção dos cheiros o sistema olfativo humano utiliza três componentes principais: o epitélio olfativo, o bulbo olfativo e córtex cerebral.

O epitélio olfativo é formado por várias células, entre as quais se destacam: os *neurônios receptores*, as *células de sustentação* e as *células basilares*. O tamanho do epitélio pode ser um indicador da intensidade do sentido do cheiro do animal. Existe muita controvérsia a respeito do tamanho exato desta área, mas estima-se que nos seres humanos ela ocupa de 1 a 2 cm², podendo variar entre indivíduos [Santos, 2000].

Os neurônios receptores são bipolares e medem entre 5 e 7 μ m. Eles possuem a função de detectar odorantes. Quando as moléculas odorantes entram em contato com os neurônios receptores, um fluxo básico de informações, sobre a intensidade e a duração do estímulo, é enviado por ele para o cérebro. O único dendrito⁴ de um neurônio receptor se projeta para a superfície do epitélio olfativo formando uma vesícula na extremidade. Desta vesícula partem vários cílios que se espalham na camada de muco. Esses

¹ Moléculas hidrofóbicas têm afinidade por água.

² Moléculas polares têm separação natural de cargas elétricas.

³ Unidade de pesos molecular, significa “Dalton”.

⁴ Dendritos – São responsáveis por receber as informações ou impulsos oriundos de outros neurônios e conduzi-los até o corpo celular.

cílios formam uma vasta rede no muco fornecendo uma ampla superfície de contato para interagir com as moléculas odorantes. A **Figura 2-2** ilustra os principais componentes do epitélio olfativo.

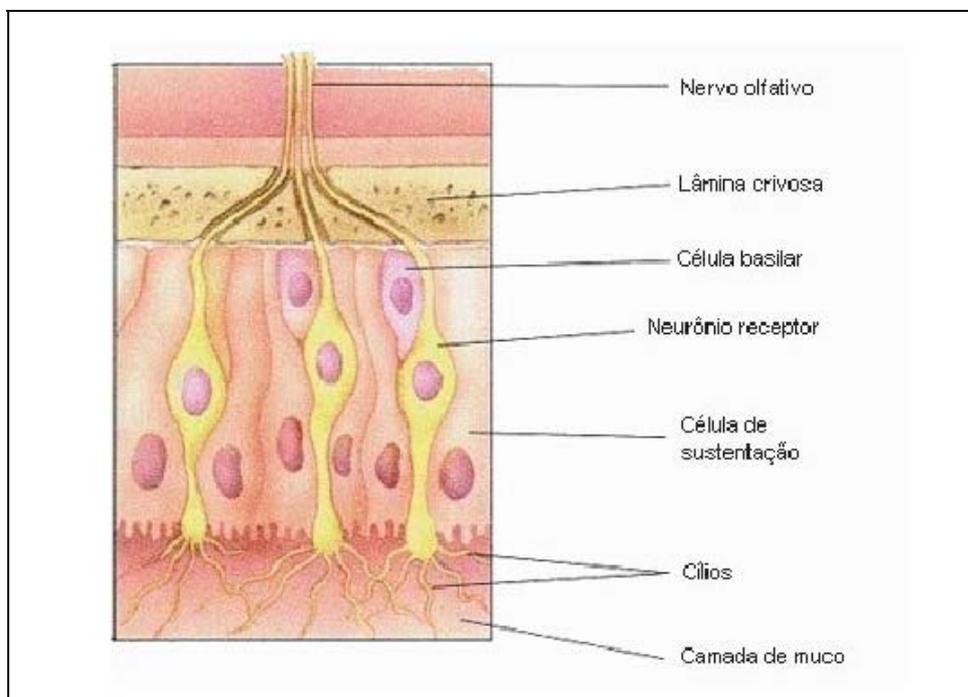


Figura 2-2: Principais componentes do epitélio olfativo.

Os axônios⁵ de vários neurônios receptores são unidos e envolvidos por células de *Schwann* formando os *nervos olfativos* que se projetam diretamente para o bulbo olfativo sem haver ramificações ou conexões sinápticas antes do bulbo. Os axônios terminam na camada superficial do bulbo em aglomerações esféricas chamadas de *glomérulos* [Yamazaki, 2001].

O *bulbo olfativo* é uma expansão do cérebro que tem a forma aproximada de uma pequena uva alongada. Ele está localizado em cada narina na parte da frente do cérebro. O bulbo olfativo é o local onde ocorrem às primeiras sinapses⁶ no caminho percorrido pelo sinal olfativo. Os quatro tipos principais de células do bulbo olfativo são: *células mitrais*, *células periglomerulares*, *células granulares* e *células do axônio curto*. Essas células e os glomérulos ficam distribuídos no bulbo olfativo nas seguintes camadas: camada do nervo olfativo, *camada glomerular*, *camada externa plexiforme*, *camada de células mitrais* e *camada de células granulares*.

⁵ Axônios – São responsáveis por enviar informações ou impulsos de um neurônio para outros neurônios.

⁶ Sinapses – É o ponto de contato entre o axônio de um neurônio e o dendrito de outro neurônio.

Dentro dos glomérulos, os axônios dos neurônios receptores fazem conexões com os dendritos das células mitrais, cujos axônios formam a saída do bulbo olfativo. As células mitrais possuem conexões internas com outras partes do bulbo. As células periglomerulares, granulares e do axônio curto formam circuitos locais no bulbo.

O estágio seguinte ao bulbo olfativo é na região do córtex olfativo. O córtex olfativo combina as informações do bulbo com informações de outros sistemas resultando em uma percepção de odor. O córtex olfativo é dividido em três camadas principais. A *camada I* é uma camada que possui os dendritos das células de camadas mais profundas. Os axônios vindos do bulbo olfativo terminam na parte superficial desta camada. A *camada II* é uma camada compacta que contém predominantemente o corpo celular das *células piramidais*. A *camada III* é menos compacta e possui o corpo celular das células piramidais mais profundas e os dendritos basais das células piramidais mais superficiais. Existem outros tipos de células no córtex olfativo, além das células piramidais.

Para explicar a maneira pela qual o sistema olfativo reconhece os odores foram criadas várias teorias, entre elas podemos citar o modelo de localização espacial do epitélio olfativo, o modelo de mapeamento dos glomérulos e o modelo de conexões cruzadas. O modelo de localização espacial do epitélio olfativo considera que o que define a qualidade do odor é a localização espacial dos neurônios receptores e a ativação dos mesmos em locais específicos. O modelo de mapeamento dos glomérulos define que os neurônios que levam um certo tipo de receptor se posicionam de forma aleatória no epitélio olfativo, mas seus axônios convergem para uma determinada área do cérebro. O modelo de conexões cruzadas considera que os neurônios e suas projeções para o cérebro são escolhidos de forma aleatória e que o cérebro utiliza um algoritmo sofisticado para decodificar os sinais aleatórios.

2.1.2 *Fisiologia do sistema olfativo humano*

Estudos em unidades receptoras isoladas indicam que os odores não ativam neurônios altamente específicos, que respondem a apenas um dado composto. O estímulo de um odor ativa um conjunto de células receptoras, cada uma tendo uma faixa ampla de sensibilidade. O estímulo de um odor distinto ativará um outro conjunto de células receptoras, que poderá conter algumas células que responderam ao odorante anterior [Yamazaki, 2001]. Para tentar explicar como os odores são detectados, identificados e reconhecidos pelo sistema olfativo, várias teorias foram elaboradas.

A *teoria dos odores primários* considera que odores distintos são na verdade combinação de um pequeno número de odores primários. Várias classificações de odores primários foram sugeridas na literatura. A classificação mais antiga possui uma lista com sete cheiros denominados de básico ou primários e foi sugerida pelo botânico Lineu⁷ [Santos, 2000]. Os sete cheiros identificados por Lineu são: aromático, fragrante, ambrosíaco, aliáceo, hircino, fétido, nauseabundo e repulsivo.

A *teoria estereoquímica do odor* incorpora o conceito de odores primários. Ela considera que todas as sensações de odor são baseadas em várias combinações de um número limitado de odores primários, sendo cada odor primário detectado por um receptor diferente. Existiria um número limitado de diferentes tipos de receptores no nariz e cada um deles reconheceria uma forma particular de moléculas [Santos, 2000]. A principal objeção a esta teoria reside na existência de exemplos de substância que têm a mesma forma molecular e diferentes odores.

A *teoria vibracional* define que os receptores olfativos entram seletivamente em ressonância com as moléculas de odor. Para ela o cérebro é capaz de interpretar padrões de ativação de detectores sintonizados vibracionalmente com odores [Santos, 2000]. Existem algumas objeções a esta teoria, por exemplo, os isômeros ópticos, que polarizam o plano de luz para direita ou para esquerda com a mesma frequência de vibração e com cheiros diferentes.

Para a *teoria bioquímica*, quando as moléculas odorantes entram na cavidade nasal e se dissolvem no muco do epitélio, elas se ligam às proteínas receptoras específicas que estão nos cílios dos neurônios receptores. É nesse momento que se inicia o processo de transdução⁸ ou detecção do odor.

O *modelo de Li e Hertz* é composto do bulbo e do córtex olfativo e não considera a parte do processamento realizada pelos neurônios receptores. O modelo proposto tem como objetivo fazer a segmentação do odor. A segmentação do odor é necessária, pois muitas vezes temos no ambiente dois ou mais odores e o sistema deve identificá-los separadamente. Muitas vezes esses odores podem ser uma mistura que o ser humano não consegue distinguir. Esta proposta não contempla o processamento dos diferentes padrões de atividades dos neurônios com as moléculas de odor que serve como entrada para o bulbo olfativo [Santos, 2000].

⁷ Lineu foi o botânico que fez a primeira classificação do reino vegetal.

⁸ Transdução – É a transformação de um sinal em outro.

2.1.3 *Identificação dos cheiros*

O sentido do olfato percebe apenas um odor de cada vez. Esse odor pode ser uma combinação de vários odores. Por exemplo, quando sentimos um odor pútrido e um odor floral, o odor que sentimos é o que possuir maior intensidade. Se ambos possuem a mesma intensidade, então um odor será percebido em um momento e o outro em outro momento, não sentiremos a combinação dos dois.

Existe um limiar para detectar um cheiro e um limiar para identificá-lo. A intensidade da sensação aumenta lentamente em função da intensidade do estímulo. Quando há uma pequena concentração de substância odorante é possível perceber a sensação de estar cheirando algo, mas não é possível identificar o cheiro. Para reconhecer um odor é necessária uma concentração maior da substância.

A sensação de cheiro está relacionada também ao período de tempo em que os estímulos são mantidos. Quando o período é longo, a intensidade da sensação diminui e se instala a adaptação ao cheiro. A capacidade de adaptação do olfato humano é muito grande. Ela pode se dar totalmente quando na presença de estímulos prolongados de alta intensidade e após a saturação a sensação de odor desaparece.

2.2 **Nariz Artificial**

O trabalho mais antigo no desenvolvimento de um instrumento específico para detectar odores data aproximadamente de 1961 [Gardner e Barlett, 1994], este instrumento era mecânico e foi proposto por Moncrieff [Moncrieff, 1961]. Os primeiros narizes eletrônicos foram desenvolvidos por Wilkens e Hatman [Wilkens e Hatman, 1964] em 1964, Buck et al. [Buck et al., 1965] e Dravnieks e Trotter [Dravnieks e Trotter, 1965] ambos em 1965. O conceito de nariz artificial como um sistema inteligente de sensores químicos para classificação de odores só apareceu cerca de vinte anos mais tarde, nos trabalhos de Persaud e Dodd [Persaud e Dodd, 1982] em 1982 e nos trabalhos de Ikegami e Kaneyasu [Ikegami e Kaneyasu, 1985] em 1985.

O termo nariz eletrônico surgiu no final da década de 80 [Gardner e Barlett, 1994], e foi especificado no 8th Int. Congress of European Chemoreception Research Organization em 1987 [Gardner, 1987]. Em 1989, uma sessão do Advanced Workshop on Chemosensory Information Processing da OTAN foi dedicada ao olfato artificial e o desenho de um sistema de olfato artificial foi estabelecido [Gardner et al.,

1990b]. Em 1990 foi realizada a primeira conferência dedicada ao tópico de narizes eletrônicos em Berlim [Gardner e Barlett, 1994].

Em 1994 foi realizado o primeiro International Symposium Olfaction & Electronic Nose em Toulouse na França. Este evento ocorre anualmente e em 2003 aconteceu o décimo em Riga, Latvia [Endereço 1].

O nariz humano contém cerca de 50 milhões de células no epitélio olfativo que atuam como receptores primários das moléculas de odores. Existem cerca de 10.000 neurônios primários associados com esses receptores que ligam com neurônios secundários que alimentam o córtex olfativo no cérebro [Gardner et al., 1994]. Esta arquitetura paralela sugere um arranjo que poderia ser representado por instrumentos análogos capazes de imitar o sistema biológico. A **Figura 2-3** mostra uma representação de um nariz biológico e uma analogia dos narizes eletrônicos que foi investigada por Warwick [Shurmer et al., 1987]. Os receptores do nariz biológico são substituídos nos narizes artificiais por arranjos de sensores que respondem a uma larga faixa de vapores e odores. A resposta, que é caracterizada por uma mudança na resistência elétrica, é processada pelo nível secundário que converte o sinal analógico para sinal digital e finalmente alimenta o sistema de reconhecimento de padrões. Para modelar o sistema do olfato humano é necessário combinar o desenvolvimento de equipamentos sensores com técnicas de processamento do arranjo de sensores. Desde os anos 1980, o uso de arranjo de sensores para detecção de vapores tóxicos e para discriminação de odores tem sido proposto e investigado. A maior vantagem oferecida pelo uso do arranjo de sensores integrados é melhorar o problema associado com a fraca especificidade de sensores individuais [Gardner et al., 1994].

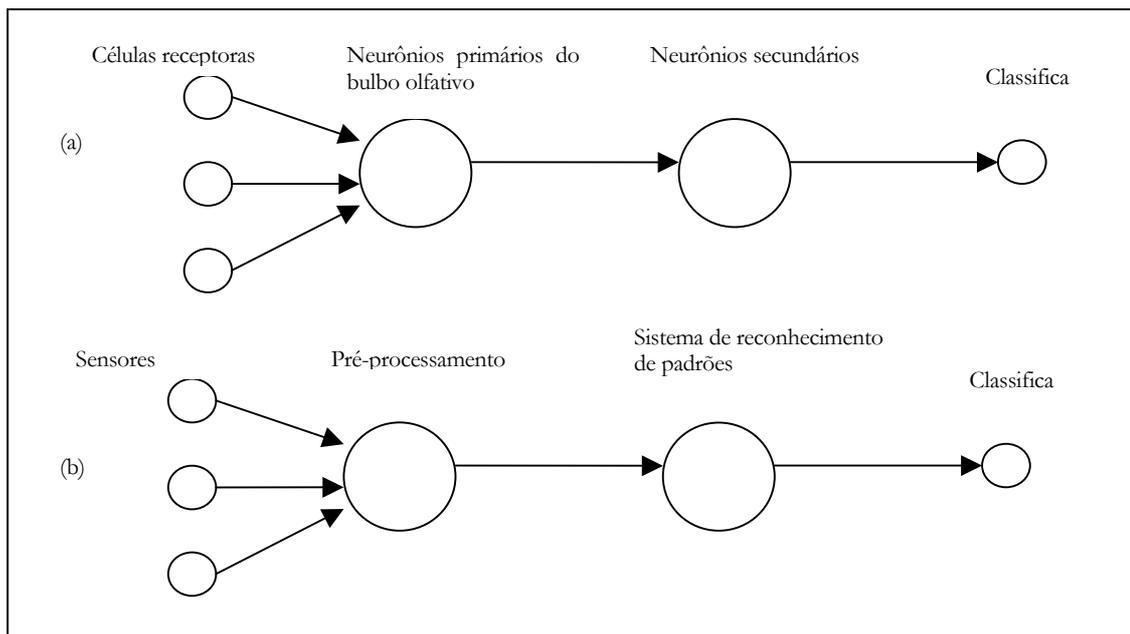


Figura 2-3: Representação esquemática (a) um nariz biológico e (b) um nariz eletrônico.

O termo nariz artificial ou nariz eletrônico foi definido por Gardner como um instrumento composto de duas partes: um arranjo de sensores químicos eletrônicos distintos, com especificidade parcial, e um sistema de reconhecimento de padrões apropriado, capaz de reconhecer odores simples ou complexos. Esta definição restringe o termo nariz eletrônico àqueles tipos de sistemas de arranjo de sensores químicos que são usados especificamente para detectar moléculas odorantes em analogia ao nariz humano. Entretanto a arquitetura de um nariz artificial também se aplica à detecção de componentes individuais ou misturas de gases ou vapores [Gardner e Barlett, 1994].

Neste trabalho utilizaremos a definição de nariz artificial dada por [Santos, 2000]. Para ela um nariz artificial é um dispositivo composto de um conjunto de sensores construídos de polímeros condutores que constituem a parte seletiva e também de um sistema de classificação e caracterização dos aromas, onde podem se utilizadas técnicas estatísticas de reconhecimento de padrões ou então Redes Neurais. Esses odores podem ser simples ou complexos.

De uma maneira geral, um nariz artificial é um sistema modular composto de duas partes: um sistema sensor, formado de elementos que detectam odores e que traduzem a quantidade química detectada

em sinais elétricos e um sistema de reconhecimento de padrões que classifica os odores conhecidos ou identifica odores desconhecidos. A **Figura 2-4** ilustra esta composição.

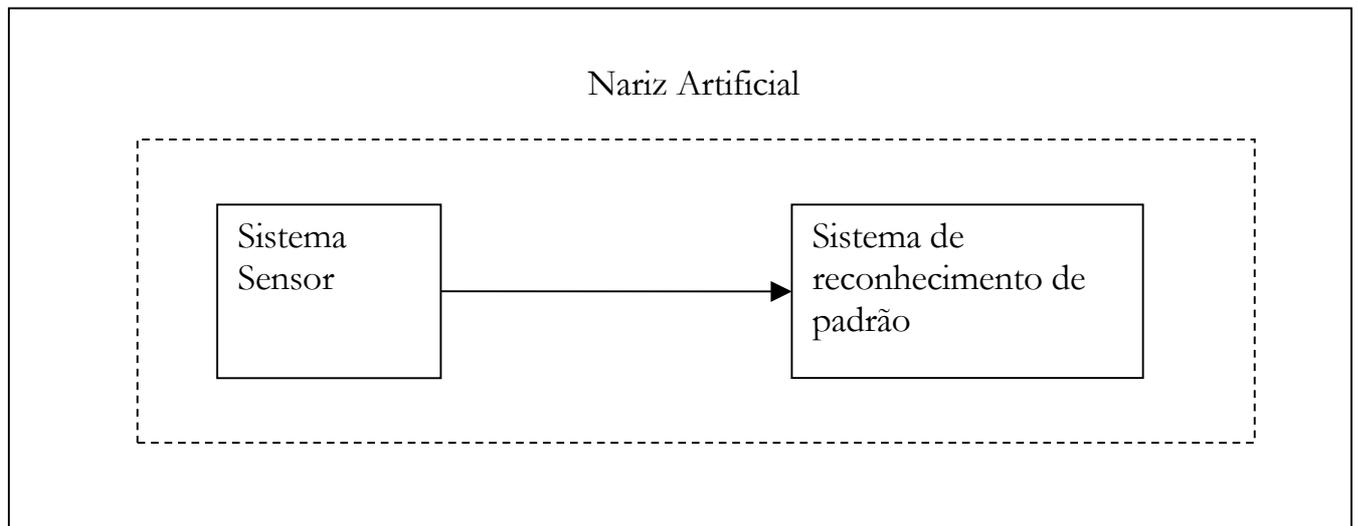


Figura 2-4: Composição de uma nariz artificial

O sistema sensor pode ser formado de um único dispositivo (por exemplo, espectrômetro) que produz um conjunto de medições para cada odorante ou por um conjunto de sensores distintos, em que cada elemento mede uma propriedade diferente do composto odorante ou por uma combinação das duas estruturas.

A Figura 2-5 ilustra a arquitetura básica de narizes artificiais. Um odor j é apresentado aos materiais ativos dos sensores, que convertem uma entrada química em um sinal elétrico, produzindo então um padrão característico. Este padrão é apresentado ao sistema de reconhecimento de padrões. Normalmente é necessário realizar um pré-processamento dos padrões gerados pelo sistema sensor antes da fase de reconhecimento dos padrões [Gardner e Hines, 1997]. Como muitos odores diferentes são apresentados ao arranjo de sensores, é construído também um banco de dados com as classes de odores e seus respectivos padrões. Este banco de dados é utilizado para treinar o sistema de reconhecimento de padrões. O objetivo deste treinamento é configurar o sistema de reconhecimento de padrões para produzir um único mapeamento de cada odor de forma a permitir que uma identificação automática seja implementada.

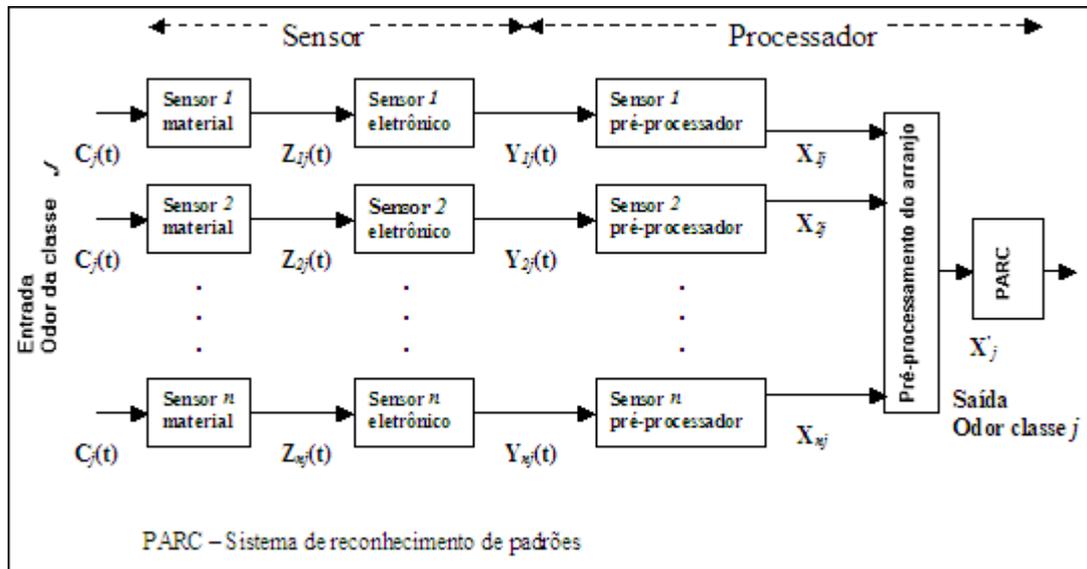


Figura 2-5: Arquitetura básica de um nariz artificial

2.2.1 Sistema sensor

Os sensores de gases são os principais componentes dos narizes artificiais e têm como função transformar em sinal físico a ação química do componente orgânico volátil ocorrida sobre o material sensível. A medida física utilizada para cada sensor depende da tecnologia escolhida, influenciando assim o sinal de saída [Santos, 2000]. É necessário que os sensores nos narizes eletrônicos tenham sensibilidade parcial, isto é, que eles respondam em linhas gerais a uma faixa ou classe de gases ao invés de responder a um gás específico. É claro que é o oposto de um sensor ideal de gás, ou seja, um sensor que responda a um único gás, por exemplo, metano e produza uma única saída. Entretanto, em um nariz eletrônico, como no nariz humano, o que se deseja é identificar muitos odores que podem conter centenas de componentes químicos individuais.

Existem duas categorias principais de sensores com características de terem seletividade ampla e parcialmente superposta: semicondutores quimicamente sensíveis e sensores de absorção [Yamazaki, 2001].

2.2.1.1 *Semicondutores quimicamente sensíveis*

Existem duas classes principais de sensores quimicamente sensíveis: sensores de semicondutores de óxido metálico e sensores de polímeros condutores. Esses sensores apresentam mudanças nas resistências quando são expostos a compostos orgânicos voláteis [Kress-Rogers, 1997].

Semicondutores de Óxido metálico

O composto mais utilizado nos semicondutores de óxido metálico é o SnO_2 , sendo também encontrados ZnO , In_2O_3 , WO_3 , Fe_2O_3 , e Ga_2O_3 . Esses sensores operam em temperaturas altas, entre 100 e 600°C e têm um consumo de energia relativamente alto, pois necessitam de um elemento aquecedor. Eles medem a mudança na condutividade na superfície catalisadora onde ocorre a decomposição de moléculas de gás a altas temperaturas.

Esses sensores apresentam um baixo custo e uma disponibilidade comercial em larga escala e são os mais utilizados na construção de narizes eletrônicos. Eles foram os primeiros sensores utilizados na construção de narizes eletrônicos [Santos, 2000].

Polímeros Condutores

Os sensores de polímero condutores foram identificados pela primeira vez na década de 1970 e pertencem a uma classe de materiais orgânicos que podem ser sintetizados eletroquimicamente [Gardner e Barlett, 1995]. Estes sensores sofrem mudanças em sua condutividade quando um gás se difunde dentro do polímero. Os materiais ativos desses sensores são os polímeros condutores das famílias do polipirrol, politiofeno, polianilina, etc [Santos, 2000].

Os sensores de polímero condutores apresentam vantagens na sua utilização em narizes artificiais quando comparados com os sensores de óxido metálico, pois eles são altamente sensíveis a muitas substâncias diferentes e também porque eles atuam em temperatura ambiente, não necessitando de elemento aquecedor e conseqüentemente consomem pouca energia. As desvantagens desses sensores são: susceptíveis a variações de umidade, pois são sensíveis à água e também são de difícil limpeza ou purga de um gás que esteja impregnado, pois eles são formados de material poroso [Santos, 2000].

2.2.1.2 *Sensores de absorção*

Um sensor de absorção é formado por uma camada de material absorvente quimicamente seletivo aplicado em um dispositivo de base acústico ou óptico [Grate et al., 1997]. Quando a camada absorvente entra em contato com odorantes, ela sofre mudanças em certas propriedades ópticas, como índice de refração, ou mecânicas, como espessura, densidade ou elasticidade. Estas mudanças são detectadas pelos dispositivos de base que podem ser ópticos, como o SPR (surface plasmon resonance) [Lawrence e Geddes, 1997], ou acústicos, como o SAW (surface acoustic wave) [D'Amico et al., 1997]. Esses dispositivos de base produzem um sinal relacionado à concentração do odorante em contato com a superfície [Yamazaki, 2001].

2.2.2 *Sistema de reconhecimento de padrões*

Para criação de sistemas de reconhecimento de padrões em narizes artificiais, várias técnicas diferentes são utilizadas, como por exemplo, técnicas estatísticas ou técnicas de Redes Neurais Artificiais. Alguns dos métodos utilizados em reconhecimento de padrões são paramétricos, e outros são não paramétricos. Os métodos paramétricos assumem que os dados podem ser descritos por uma função de densidade de probabilidade. Normalmente assume-se que os dados têm uma distribuição normal, com média e variância conhecida. Para os métodos não paramétricos as funções de densidade de probabilidade são desconhecidas.

Para analisar os vetores de resposta gerados pelo arranjo de sensores dos narizes artificiais, em geral são necessários dois estágios. No primeiro estágio, a saída do arranjo de sensores é treinada pelo método de reconhecimento de padrões escolhido, usando regras matemáticas que relacionam a saída com um odor conhecido (classe) armazenado na base de conhecimento. Este processo é conhecido como aprendizado supervisionado. No segundo estágio, um odor desconhecido é testado e o sistema determina a classe que o odor pertence. Alguns métodos de aprendizado não-supervisionado também podem ser utilizados.

Os principais métodos utilizados para reconhecimento de padrões em narizes artificiais são listados na **Tabela 2-1**.

Tabela 2-1: Métodos utilizados em narizes eletrônicos para criação de sistemas de reconhecimento de padrões.

Método para criação de sistemas de reconhecimento de padrões	Linear	Paramétrica	Supervisionado
Análise de componentes principais	Sim	Não	Sim
Análise de função discriminante	Sim	Sim	Sim
Redes Neurais MLP	Não	Não	Sim
Redes Neurais RBF	Não	Não	Sim
Redes Neurais de Kohonen	Não	Não	Não
Análise de clusters Euclidiana	Sim	Não	Não
Outras análises de clusters	Não	Não	Não

Análise de componentes principais

Em um nariz artificial, os dados de um odor são obtidos por vários sensores, em que cada um deles é dependente de uma ou muitas variáveis. Este tipo de dado é chamado geralmente de multivariado. Um grande problema que deve ser considerado é que estes dados representam um espaço multidimensional com dimensão igual ao número de sensores utilizado no nariz artificial. Para representar esses dados é necessário preservar em escala a dimensionalidade da informação. A análise de componentes principais - PCA é uma ferramenta útil para esta operação [Yamazaki, 2001].

A análise de componentes principais se baseia no fato de que os sensores de dados freqüentemente não são uniformemente distribuídos, mas que existe uma certa correlação entre os sensores, eles tendem a se arrumar em subconjuntos do espaço original. PCA opera projetando os dados linearmente em um subconjunto de dimensionalidade reduzida. Se a dimensionalidade é menor ou igual a 3, uma representação direta pode ser feita e os resultados das análises dos narizes eletrônicos podem ser lidos em 2D ou 3D.

A análise de componentes principais é uma técnica não paramétrica, supervisionada e linear que tem sido utilizada para classificar odores simples e complexos. Ela expressa os vetores de resposta dos sensores em termos de combinações lineares de vetores ortogonais. Cada vetor ortogonal (principal) contribui na variância dos dados com um grau descendente de importância.

Redes Neurais Artificiais

A quantidade e complexidade dos dados coletados pelo arranjo de sensores podem ser analisadas quimicamente. Uma abordagem na identificação dos odores pode ser feita através da construção de sensores, em que cada sensor do arranjo é designado para responder a um odor específico. Com esta abordagem, o número de sensores únicos deve ser pelo menos igual ao número de odores que estão sendo monitorados. É difícil e caro construir sensores com alta seletividade química [Keller et al., 1995b].

Redes neurais artificiais são utilizadas para analisar dados complexos e para reconhecimento de padrões [Barros Filho et al., 2004; Santos, 2000; Yamazaki, 2001; Zanchettin, 2004]. Para o reconhecimento de vapores químicos, redes neurais artificiais têm mostrado resultados promissores. Quando uma rede neural artificial é combinada com um arranjo de sensores, o número de odores detectados é geralmente maior que o número de sensores [Hoffheins, 1989]. Dessa forma, menos sensores seletivos são necessários para construir o sistema o que o torna mais barato. Depois que a rede neural artificial é treinada para o reconhecimento de odores, a operação do sistema consiste de propagar os dados dos sensores pela rede. Como ela é simplesmente uma série de multiplicações de matrizes, odores desconhecidos podem ser rapidamente identificados [Keller et al., 1995b].

Narizes artificiais que incorporam redes neurais artificiais já foram utilizados em várias aplicações, como no monitoramento de odores de alimentos e de bebidas [Gardner et al., 1990a], controle do cozimento dos alimentos [Keller et al., 1995b], controle automático do sabor [Moriizumi et al., 1992], análise de mistura de fluidos [Lauf e Hoffheins, 1991], discriminação da fumaça de diferentes marcas de cigarros [Shurmer, 1990], detecção de vazamento de óleo, identificação de diferentes tipos de álcool, identificar odores químicos domésticos [Keller et al., 1994], quantificação de componentes individuais em mistura de gases [Sundgren et al., 1991] [Nakamoto et al., 1992], e classificação de safras de vinho [Yamazaki, 2001; Santos, 2000]. Algumas configurações de redes neurais artificiais foram usadas em narizes artificiais incluindo redes *Multi-layer Perceptrons* (MLP), *Time Delay Neural Networks* (TDNN), *Kobonen's self-organizing maps* (SOM), *Learning Vector Quantizers* (LVQ), redes *Hamming*, máquina de *Boltzmann* e redes *Hopfield* [Keller et al., 1995a].

Devido às limitações de tecnologia atual, muitos narizes eletrônicos baseados em redes neurais têm menos de 20 elementos sensores e menos de 100 neurônios. Esses sistemas são projetados para serem utilizados em aplicações específicas com uma faixa limitada de odores. Sistemas que imitem o olfato humano

vão necessitar de um conjunto muito maior de elementos sensores e de redes neurais artificiais bem maiores[Keller et al., 1995b].

2.3 Desenvolvimento de narizes artificiais

Atualmente vários grupos de pesquisa e companhias estão estudando, desenvolvendo e comercializando narizes eletrônicos. A seguir vamos listar alguns protótipos de narizes eletrônicos encontrados através de pesquisa à Internet.

2.3.1 *Diag-Nose*

Os doutores *Selly Saini* e *Jan Leiferkus* da universidade *Cranfield* em *Bedfordshire* na *Inglaterra* criaram um nariz artificial capaz de diagnosticar infecções. Eles se basearam na longa experiência da medicina chinesa com o sentido do olfato. Os chineses possuem muitos trabalhos utilizando o sentido do olfato para diagnosticar doenças.

O *Diag-Nose* é equipado com uma faixa de sensores eletrônicos ajustados para uma substância particular. Se traços da substância estão presentes, o sensor envia a mensagem para o sistema de reconhecimento de padrões do equipamento, que é uma rede neural treinada para associar certos odores com eventos particulares, neste caso com a presença de infecções por bactérias.

O sistema de reconhecimentos de padrões do *Diag-Nose* foi treinado para o diagnóstico de infecções urinárias, mas também pode ser adaptado para o diagnóstico de tuberculose, certos cânceres de intestino e infecções em ferimentos. Mais detalhes sobre o *Diag-nose* podem ser encontrados em [Endereço 2]

2.3.2 *Nariz artificial para diagnosticar tuberculose*

Cientistas da universidade *Cranfield* em *Bedfordshire* na *Inglaterra* também criaram um nariz artificial para diagnosticar tuberculose causada por bactéria. O equipamento pode produzir o resultado em quatro horas. Atualmente os métodos laboratoriais levam dois dias para confirmar a infecção.

O equipamento também é composto de um arranjo de sensores e um sistema de reconhecimento de padrões implementado por redes neurais. O equipamento analisa a saliva e o muco dos pacientes

convertidos na forma gasosa. Quando os pacientes estão contaminados, a saliva e o muco possuem vários ácidos e outros produtos do pulmão infectado. Mais detalhes são encontrados em [Endereço 3]

2.3.3 *Menor nariz artificial*

Cientistas das universidades de *Warwick, Leicester e Edinburgh* desenvolveram um sistema que imita o olfato humano. Da mesma forma que os anteriores, o nariz artificial desenvolvido por eles é formado por um arranjo de sensores e um sistema de reconhecimento de padrões. Entretanto este nariz artificial é portátil, como é mostrado na **Figura 2-6**, pois eles combinaram o arranjo de sensores com o sistema de reconhecimento de padrões em um único chip de silício. O equipamento necessita de pouca energia e cabe confortavelmente na palma da mão. Mais detalhes sobre este equipamento são encontrados em [Endereço 4].



Figura 2-6 Menor nariz artificial

2.4 Sumário do capítulo

Atualmente vários grupos de pesquisa e companhias estão estudando, desenvolvendo e comercializando narizes eletrônicos. Todos os narizes eletrônicos encontrados na Internet têm basicamente a mesma implementação com o mesmo princípio de trabalho, as diferenças se concentram nos sensores escolhidos e nas técnicas implementadas nos sistemas de reconhecimento de padrões. É possível falar de uma primeira geração de narizes artificiais, que é basicamente um sistema aberto de medição com uma separação clara entre os sensores e a avaliação da saída do sensor. Considerando como um sistema de medição, o nariz artificial é um sistema complexo, e o desempenho final do instrumento depende do desempenho de uma variedade de componentes.

A próxima geração de narizes artificiais está centrada na construção de modelos de olfação, plausíveis, baseados na biologia e implementados em eletrônica. Esta abordagem já está sendo desenvolvida e é denominada de abordagem *neuromorfica* [Santos, 2000].

O nariz artificial pode ser uma ferramenta útil para diagnóstico. Usando um equipamento de mão equipado com sensores, um médico poderia pesquisar os odores do corpo, da respiração e de seus fluidos para identificar possíveis problemas. Por exemplo, odores da respiração podem indicar problemas gastrintestinais, problemas de sinusite, infecções, diabetes e problemas do fígado. Feridas e tecidos infectados emitem odores que podem ser detectados por narizes artificiais. Odores dos fluidos do corpo, como do sangue e da urina, podem indicar problema no fígado ou na bexiga.

Este capítulo procurou mostrar a anatomia e a fisiologia do sistema olfativo humano, o histórico dos narizes artificiais, os principais componentes de um nariz artificial e apresentou alguns protótipos de narizes artificiais encontrados através de pesquisa à Internet. As informações mostradas neste capítulo contribuem para se ter noção sobre o funcionamento do sistema olfativo humano e bem como permitem conhecer as principais etapas envolvidas no funcionamento de um nariz artificial.

No próximo capítulo serão apresentadas algumas técnicas de redes neurais utilizadas em sistemas de reconhecimento de padrões de odor e será feita uma revisão sobre estes sistemas.

REDES NEURAIS ARTIFICIAIS PARA CRIAÇÃO DE SISTEMAS DE RECONHECIMENTO DE PADRÕES

3 Introdução

As pesquisas de redes neurais artificiais iniciaram em 1943 com o trabalho pioneiro de *McCulloch* e *Pitts* [McCulloch e Pitts, 1943]. Eles descrevem, neste trabalho, um cálculo lógico das redes neurais que unificava os estudos da neurofisiologia e da lógica matemática. Até o final da década de 1960 a área de redes neurais recebeu uma enorme atenção da comunidade científica, porém após a publicação de *Perceptrons* [Minsky e Papert, 1969] em 1969, pelos professores *Marvin Minsky* e *Seymour Papert*, o interesse nesta área esmoreceu. A análise de *Minsky* e *Papert* revelou que existem limites fundamentais para o que pode ser calculado pelos perceptrons de camada única, além disso, eles argumentaram que embora existisse um algoritmo de aprendizado que garantia a convergência para modelos com camada única de nodos, o mesmo não acontecia para redes perceptron com mais de uma camada [Braga et al., 2000]. O artigo de *Hopfield* em 1982 [Hopfield, 1982] e o livro em dois volumes de *Rumelhart* e *McClelland* em 1986 [Rumelhart et al., 1986] são considerados as publicações mais influentes, responsáveis pelo ressurgimento do interesse em redes neurais nos anos 1980 [Haykin, 1999].

A retomada das pesquisas de redes neurais provocou um grande impacto na pesquisa e desenvolvimento de sistemas de reconhecimento de padrões. Embora a função das redes neurais artificiais não seja limitada ao reconhecimento de padrões, não há dúvida que o progresso em reconhecimento de padrões e suas aplicações agora dependem fortemente de redes neurais [Chen, 1991]. Nas seções seguintes serão descritos os fundamentos de redes neurais, serão apresentadas quatro abordagens de redes neurais e será feita uma revisão de cinco trabalhos que criaram sistemas de reconhecimento de padrões para narizes artificiais com redes neurais.

3.1 Fundamentos de redes neurais

Em geral, uma rede neural é uma coleção de unidades de processamento simples (também conhecidas com *neurônios* ou *nodos*), conectadas através de ligações conhecidas como *conexões*. Uma rede neural é representada na forma de um grafo, onde os *nodos* representam as unidades de processamento, os

arcos representam as conexões e as setas dos arcos indicam a direção normal do fluxo do sinal. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de *aprendizagem*. Como mostrado na Figura 3-1, os *nodos* são agrupados em camadas estruturadas, onde cada *nodo* de cada camada, aplica uma *função de ativação* nos sinais de entrada do mesmo que irá determinar o valor de ativação do *nodo* (ou valor de saída).

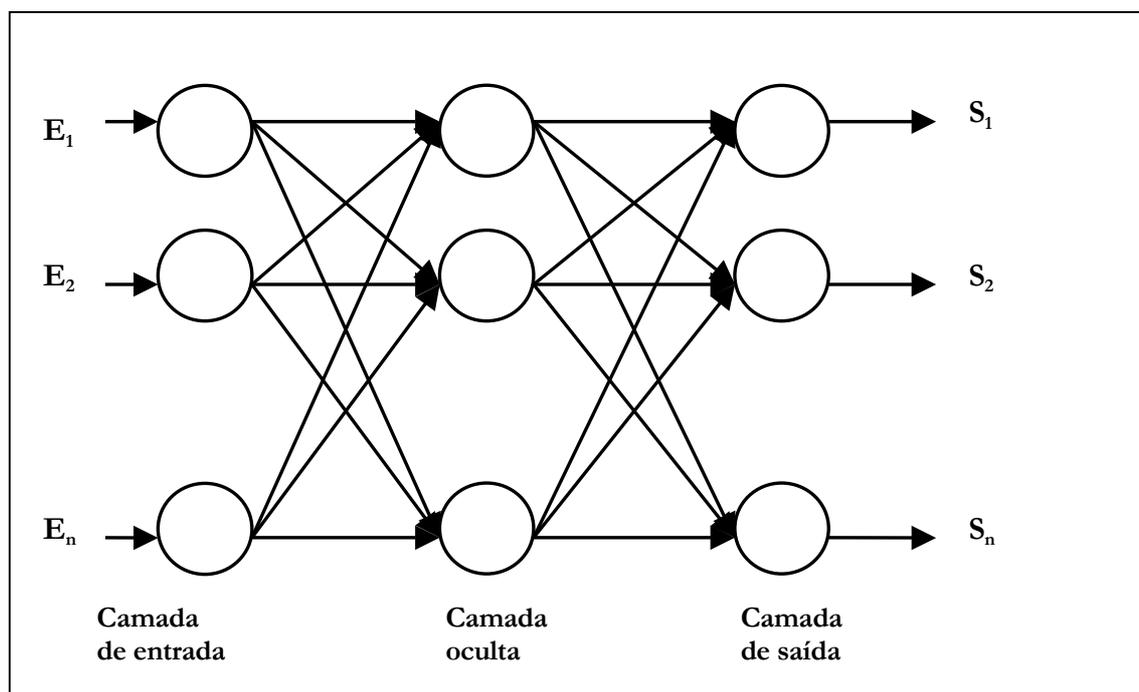


Figura 3-1 Estrutura geral de uma rede neural multicamada, alimentada a diante.

As conexões entre os nodos, conhecidas como *pesos sinápticos* ou simplesmente *pesos*, são utilizadas para armazenar o conhecimento adquirido. A **Figura 3-2** mostra um nodo com seus *pesos*. O procedimento utilizado para realizar o processo de aprendizagem é chamado de *algoritmo de aprendizagem*, cuja função é modificar os *pesos* da rede de uma forma ordenada para alcançar um objetivo de projeto desejado. A modificação dos *pesos* é o método tradicional para o projeto de redes neurais, entretanto é possível também para uma rede neural modificar sua própria topologia [Haykin, 1999].

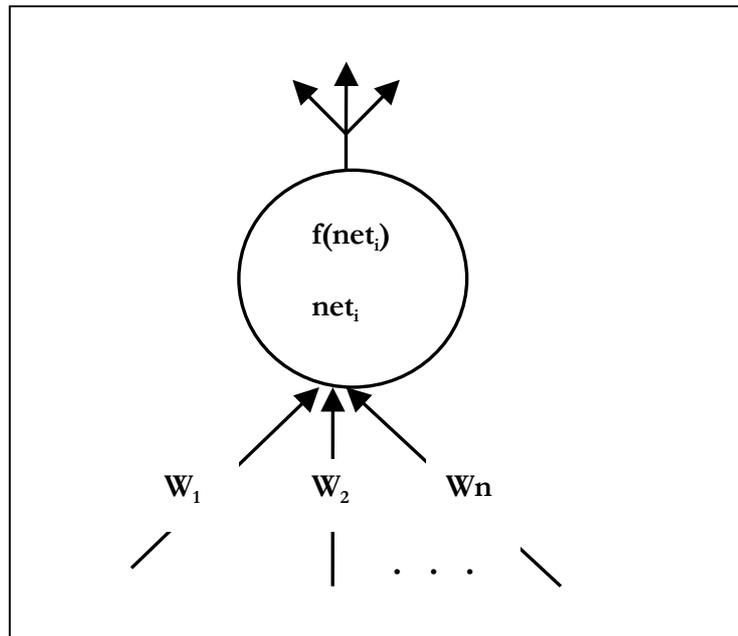


Figura 3-2 Um nodo de uma rede neural é mostrado nesta figura. Cada conexão de entrada do nodo tem um valor de peso associado, e o nodo produz uma saída simples que é transmitida para outros nodos.

O processamento de uma rede neural começa com toda a rede em um estado inativo. Um padrão externo, composto por um conjunto de sinais, é aplicado a camada de entrada, em que cada sinal estimula um dos nodos da camada de entrada. Cada nodo da camada de entrada gera um sinal de saída simples, com uma magnitude que é função do estímulo total recebido pela unidade. As saídas produzidas pelos nodos de uma camada são passadas como padrão de entrada para os nodos das camadas subseqüentes. Este processo se repete até que os nodos da camada de saída produzam uma saída para o vetor de entrada corrente.

3.2 Redes neurais Perceptron multicamadas

As redes neurais *Perceptron multicamadas* ou *Multilayer Perceptrons* (MLP) são uma importante classe de redes neurais. Elas são compostas tipicamente de uma *camada de entrada* (que não possui pesos ajustáveis), uma ou mais *camadas ocultas* e uma *camada de saída*. O sinal de entrada se propaga para frente através da rede, camada por camada.

As redes MLP têm sido aplicadas com sucesso para resolver diversos problemas difíceis, através de seu treinamento de forma supervisionada com o algoritmo conhecido como *algoritmo de retropropagação do erro*

(error backpropagation). Este algoritmo é baseado na *regra de aprendizagem por correção do erro* [Rumelhart et al., 1986].

A aprendizagem por retropropagação do erro consiste de dois passos: um passo para frente, a *propagação*, e um passo para trás, a *retropropagação*. No passo para frente, um padrão de entrada (vetor de entrada) é aplicado aos *nodos* e seu efeito se propaga através da rede camada por camada. Finalmente um conjunto de saídas é produzido com a resposta real da rede. Durante o passo de propagação, os *pesos* da rede são todos fixos. Durante o passo para trás, os *pesos* são todos ajustados de acordo com uma regra de correção de erros. Especificamente, a resposta real da rede é subtraída de uma resposta desejada (alvo) para produzir um sinal de erro. Esse sinal de erro é então propagado para trás através da rede, contra a direção das conexões sinápticas. Os *pesos* são ajustados de forma que a resposta real da rede se mova para mais perto da resposta desejada.

Desde a criação do algoritmo *Backpropagation* padrão [Rumelhart et al., 1986], várias alterações têm sido proposta visando acelerar o tempo do treinamento, bem como melhorar seu desempenho na classificação de padrões. As variações mais utilizadas são *backpropagation* com *momentum* [Rumelhart et al., 1986], *Quickprop* [Fahlman, 1988], *momentum* de segunda ordem [Pearlmutter, 1992], *Rprop* [Riedmiller, 1994] e *Levenberg-Marquardt* [Hagan e Menhaj, 1994]. O algoritmo *Levenberg-Marquardt* utiliza uma aproximação do método de Newton e foi utilizado como algoritmo de treinamento de vários experimentos desta dissertação, mais detalhes sobre este algoritmo podem ser encontrados em [Hagan e Menhaj, 1994].

Uma rede Perceptron multicamadas tem quatro características distintas:

- A rede contém uma ou mais *camada ocultas*, que não são parte da camada de entrada ou da camada de saída.
- A rede exibe um alto grau de *conectividade*, determinada pelas conexões da rede.
- O modelo de cada nodo da rede inclui uma *função de ativação*. Entre as funções de ativação utilizadas pelos neurônios das redes MLP podemos citar: a função sigmóide logística, cuja forma da curva é ilustrada na **Figura 3-3**, e a função tangente hiperbólica, demonstrada pela **Figura 3-4**.

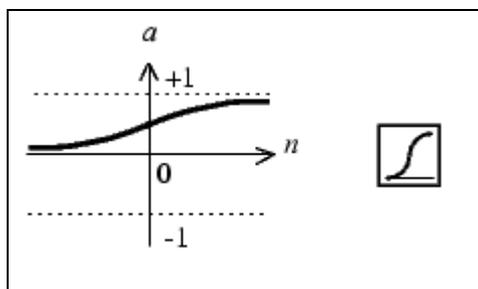


Figura 3-3: Função sigmóide logística.

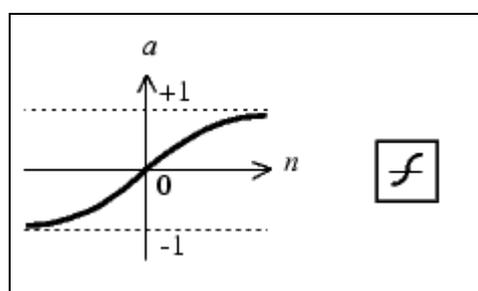


Figura 3-4: Função tangente hiperbólica.

- A rede utiliza um algoritmo de retropropagação do erro.

Nesta dissertação temos interesse em redes *MLP*, pois em geral essas redes são utilizadas como referência para comparação dos resultados obtidos com outras técnicas.

3.3 Redes neurais com processamento temporal

O *tempo* constitui um ingrediente essencial do processo de aprendizagem. Ele pode ser contínuo ou discreto. Independente de sua forma, o tempo é uma entidade ordenada importantíssima para muitas tarefas de reconhecimento de padrões, tais como, reconhecimento de voz, detecção de movimentos, visão, processamento de sinais, controle motor, etc. É através da incorporação do tempo na operação de uma rede neural que ela é capacitada a trabalhar com problemas de reconhecimento de padrões dinâmicos como, por exemplo, o reconhecimento da fala ou detecção de movimentos. Para que uma rede neural seja considerada dinâmica, ela precisa possuir memória [Braga et al., 2000].

Alguns algoritmos de treinamento de redes neurais não são capazes de implementar mapeamentos dinâmicos, como, por exemplo, o algoritmo *backpropagation*, que aprende apenas mapeamentos estáticos. Um artifício utilizado para processamento temporal é a utilização de *janelas de tempo*, em que a entrada da rede é formada por trechos das seqüências temporais dos dados como se fossem padrões estáticos. A **Figura 3-5** ilustra o diagrama de uma rede MLP, completamente conectada, que introduziu o processamento temporal através da formatação dos dados de entrada. A entrada desta rede é definida de maneira que para gerar uma saída $y(n)$, foram fornecidos os sinais de entrada correspondentes ao tempo n , $(n-1)$, $(n-2)$, ..., $(n-p)$. A janela de tempo desta rede tem tamanho igual à p .

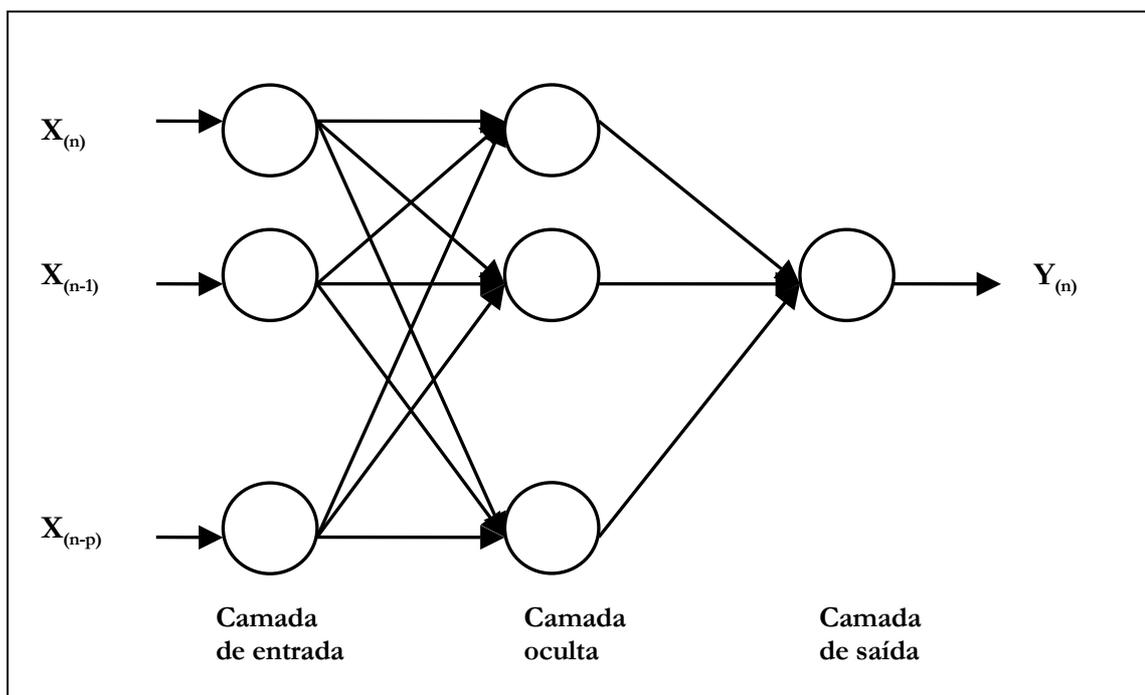


Figura 3-5 Exemplo de rede MLP utilizando a técnica de janelas de tempo para processamento temporal. A janela de tempo desta rede tem tamanho igual à p .

Existem duas maneiras básicas de incluir memória a uma rede neural: introduzir atraso no tempo ou utilizar redes recorrentes⁹, tais como redes de *Elman* [Elman, 1990] e redes de *Jordan* [Jordan, 1986].

⁹ Redes recorrentes são aquelas redes em que a saída de algum neurônio na i -ésima camada da rede é utilizada como entrada de nodos em camadas de índice menor ou igual a i .

A técnica TDNN (Time Delay Neural Networks) [Waibel et al., 1989] é uma técnica que introduz atraso no tempo. As redes TDNN são redes alimentadas a diante, com múltiplas camadas, cujos nodos das camadas escondidas e da camada de saída são duplicados ao longo do tempo. A topologia de uma rede TDNN está embutida em uma rede MLP em que cada conexão é representada por um filtro de impulso de resposta com duração finita, conhecida como FIR (Finite Duration Impulse Response). Este tipo de rede, conhecida como FIR MLP é treinada através da criação de uma rede estática equivalente. Esta rede é obtida desdobrando a FIR MLP no tempo. Após isso pode ser utilizado o algoritmo backpropagation padrão [Braga et al., 2000]. Consideramos nesta dissertação que a técnica de *janelas de tempo* é uma técnica TDNN, pois esta técnica introduz atraso de tempo através da formatação dos dados de entrada e também por esta técnica ser chamada de TDNN em vários trabalhos da área.

As redes neurais recorrentes são outra técnica capaz de lidar com variações temporais. Há dois tipos de redes recorrentes: aquelas em que o padrão de entrada é fixo e a saída caminha, dinamicamente, para um estado estável, e aquelas em que tanto a entrada quanto a saída variam com o tempo; estas últimas são mais gerais e são utilizadas com mais frequência.

3.4 Redes RBF

O projeto de uma rede neural supervisionada pode ser executado de várias formas. O algoritmo de retropropagação do erro que foi apresentado para redes *MLP* é uma dessas maneiras. Outra maneira de projetar redes neurais supervisionadas é através da utilização de redes de função de base radial [He e Lapedes, 1991].

O problema de uma rede *RBF* é visto como um problema de ajuste de curva (aproximação) em um espaço de alta dimensionalidade. De acordo com este ponto de vista, aprender é equivalente a encontrar uma superfície, em um espaço multidimensional, que forneça o melhor ajuste para os dados de treinamento.

As redes RBF, como as redes MLP, são redes multicamadas, alimentadas adiante. A principal diferença entre os dois tipos de rede é que a função de ativação sigmóide, dos nodos da camada oculta, é alterada por outra classe de função. A construção de uma rede RBF, em sua forma mais básica, envolve três camadas com papéis totalmente diferentes. A camada de entrada é constituída de nós fontes que conectam a rede a seu ambiente e não executam nenhuma função sobre os dados de entrada. A segunda camada, a única camada oculta da rede, aplica uma transformação não linear do espaço de entrada para o espaço oculto; na

maioria das aplicações, o espaço oculto é de alta dimensionalidade. Esta camada transforma o conjunto de padrões não linearmente separáveis em conjunto de saídas linearmente separáveis. A camada de saída é linear, ela procura classificar os padrões recebidos da camada anterior. A **Figura 3-6** ilustra o diagrama básico de uma rede RBF.

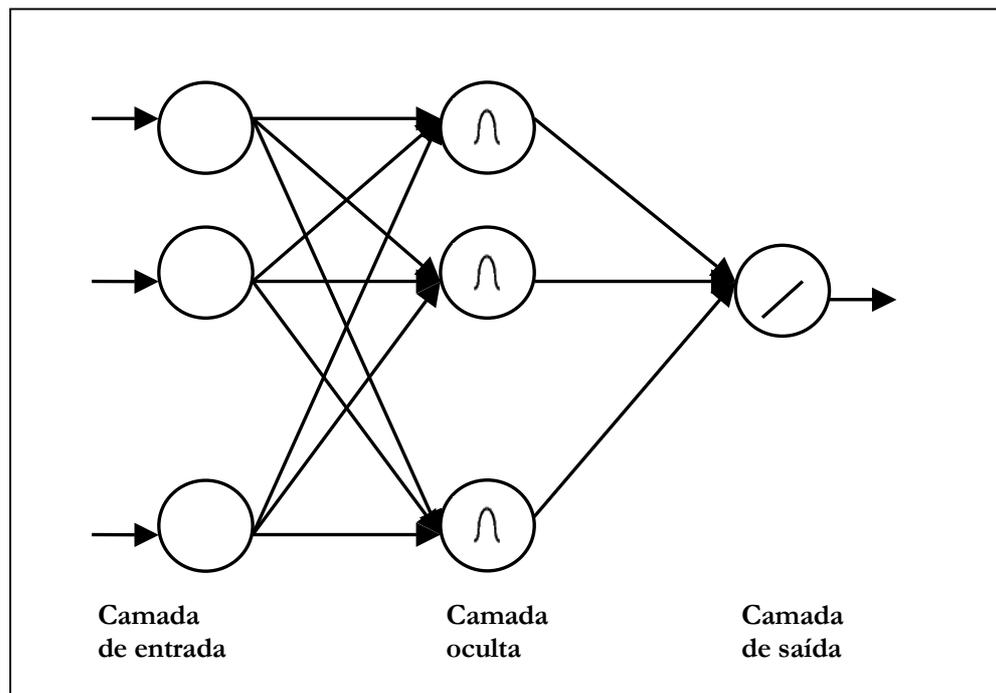


Figura 3-6 Diagrama básico de uma rede RBF.

As funções de base radiais representam uma classe especial de funções cujo valor diminui ou aumenta em relação à distância de um ponto central. Essas funções são utilizadas nos nodos da camada oculta das redes *RBF*. Diferentes funções de base radiais têm sido utilizadas em redes *RBF*. As mais comuns são:

- Função *Gaussiana*: $f(u) = \exp\left(-\frac{v^2}{2\sigma^2}\right)$
- Função *multiquadrática*: $f(u) = \sqrt{(v^2 + \sigma^2)}$
- Função *thin-plate-spline*: $f(u) = v^2 \log(v)$

Em que $v = \|x - \mu\|$, é geralmente a distância euclidiana, x é o vetor de entrada, μ e σ são respectivamente o centro e a largura da função base radial. As funções radiais mais utilizadas possuem dois parâmetros: o centro e o raio. A distância euclidiana $\|x - \mu\|$, do vetor de entrada x ao vetor de centro μ_i serve de entrada para função, que retorna o valor de ativação dos nodos da camada intermediária.

As redes *RBF* e as redes *MLP* são exemplos de redes multicamadas, alimentadas a diante, não-lineares. As redes *RBF* podem necessitar de mais neurônios do que redes *MLP*, mas freqüentemente elas são criadas em uma fração do tempo necessário para treinar redes *MLP*. Elas trabalham melhor quando muitos vetores de treinamento estão disponíveis. Ambas são aproximadores universais de função, portanto sempre existe uma rede *RBF* capaz de imitar uma rede *MLP* e vice-versa [Haykin, 1999]. Existem, entretanto, várias diferenças:

1. Cada nodo da camada intermediária de uma rede *RBF* define uma hiper-elipsóide do espaço de padrões de entrada, enquanto que nas redes *MLP* eles particionam o espaço de entrada em hiperplanos;
2. Uma rede *RBF* constrói aproximadores locais, isto é, apenas as regiões do espaço de entrada que apresentam dados de treinamento terão resposta da rede. As redes *MLP* constroem aproximadores globais com maior capacidade de generalização para regiões onde não há dados de treinamento.
3. Nas redes *RBF*, o valor de ativação dos nodos da camada intermediária é dado em função da distância euclidiana entre o vetor de entrada e o vetor centro da unidade. Nas redes *MLP* o valor de ativação dos nodos da camada intermediária é uma função do produto escalar entre o vetor de entrada e o vetor de pesos deste nodo.

3.5 Redes PNN

As redes *PNN* (Probabilistic Neural Networks) são uma variação das redes *RBF*. Estas redes separam os dados em um número específico de categorias de saída e são conhecidas por sua habilidade de treinamento rápido. Esta dissertação tem interesse nas redes *PNN*, pois segundo Specht [Specht, 1990] as mesmas são indicadas para tarefas de classificação, também porque elas possuem vantagens em relação aos métodos de retropropagação do erro e, além disso, elas não foram utilizadas para criar sistemas de

reconhecimento de padrões de odor em outros trabalhos. As principais vantagens das redes *PNN* são: treinamento é realizado em apenas um passo e garantia de que a superfície de decisão se aproxima dos limites de uma decisão Bayseana ótima à medida que o número de exemplos de treinamento cresce [Specht, 1990]. Outra grande motivação para a utilização das redes *PNN* é o curto tempo necessário para treiná-las quando comparado com o tempo necessário para treinar redes *MLP*. A principal desvantagem das redes *PNN* é que todos os exemplos de treinamento devem ser armazenados e usados para classificar novos padrões. Entretanto, esta desvantagem não é um problema atualmente, pois o custo de memória é baixo e os computadores dispõem de grande capacidade de armazenamento.

As redes *PNN* implementam a estratégia de decisão *Bayseana* para classificar os vetores de entrada [Skapura, 1996]. A **Figura 3-7** mostra a arquitetura básica de uma rede *PNN* para classificação dos padrões de entrada X em duas classes. As redes *PNN* são divididas em quatro camadas: *camada de unidades de entrada*, *camada de unidades de padrões*, *camada de unidades de soma* e *camada de unidades de saída*. Como nas redes *MLP*, as unidades da *camada de entrada* não realizam operações sobre os vetores de entrada. As unidades da *camada de padrões* executam o produto interno do vetor de entrada X pelo vetor de pesos W_i , $Z_i = X \cdot W_i$, e então executam uma operação não linear em Z_i antes de enviar suas saídas para as unidades da *camada de soma*. A operação não linear utilizada é $\exp[(Z_i - 1)/\sigma^2]$. As unidades da *camada de soma* executam simplesmente a soma de suas entradas. As entradas das unidades da *camada de soma* têm origem nas correspondentes unidades da camada de padrão de uma determinada classe. As unidades da *camada de saída* produzem uma saída binária, que é igual a 1 em apenas uma das unidade e igual a 0 nas demais [Specht, 1990].

A rede é treinada criando-se uma unidade na *camada de padrões* para cada um padrões do conjunto de treinamento e fazendo com que o vetor de peso dessas unidades seja igual ao correspondente padrão do conjunto de treinamento. Depois disso, as saídas das unidades da *camada de padrão* são conectadas com as unidades da *camada de soma* da classe correspondente.

Em essência, a rede opera pela soma das saídas dos nodos da *camada de padrões*, que devido à utilização de funções de ativação Gaussianas, agrupam os padrões de entrada em classes. Assim, a rede *PNN* calcula a função de distribuição da probabilidade a posteriori para uma classe simples, avaliando o ponto definido pelo padrão de entrada. Este ambiente é útil para aplicações de classificação, onde se deseja estimar a probabilidade de um novo padrão ser membro de alguma classe pré-definida.

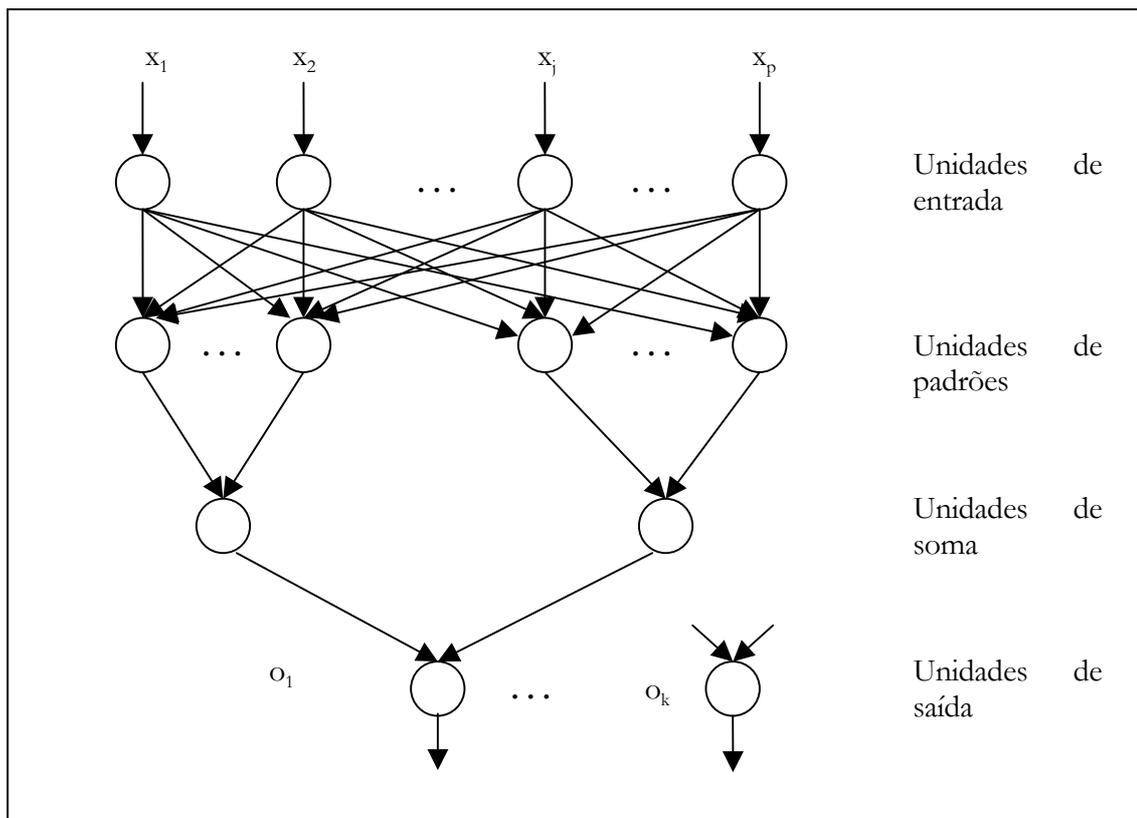


Figura 3-7 Arquitetura típica de uma rede PNN.

3.6 Parâmetros para treinamento de redes neurais

O treinamento de uma rede neural é um processo empírico. Para definir a topologia de uma rede *MLP* existem vários parâmetros que precisam ser especificados. Embora existam algumas heurísticas para auxiliar na definição da topologia, este processo é realizado geralmente a partir de experimentos sistemáticos. Os parâmetros envolvidos na definição de uma rede *MLP* vão desde o número de camada intermediárias, passando pelo número de nodos nessas camadas até a definição de funções de ativação e algoritmo de treinamento. De acordo com o algoritmo de treinamento escolhido, um conjunto de parâmetros relacionados com o mesmo também deve ser definido empiricamente. O valor de cada parâmetro pode gerar alterações significativas no desempenho do sistema.

Além dos parâmetros que definem a topologia de uma rede *MLP*, também é necessário especificar critérios para parada do treinamento e verificação da capacidade de generalização. Para os métodos baseados

no gradiente descendente é interessante também especificar o critério do número máximo de épocas¹⁰, o critério de progresso de treinamento e o critério de perda de generalização.

Para definir a topologia de uma rede *RBF* ou de uma rede *PNN* também é necessário especificar o valor de alguns parâmetros, porém a quantidade de parâmetros é bem inferior. Para as redes *RBF* necessitamos basicamente definir o algoritmo de treinamento e um fator de *alisamento* ou *suavidade*. Para as redes *PNN*, necessitamos apenas definir o fator de *alisamento*.

O fator de *alisamento* das redes *RBF* e *PNN* afeta a forma da superfície de decisão da rede. Ela pode ser tão complexa quanto for necessária ou simples se for desejado. Para explicar o funcionamento do fator de *alisamento*, imaginemos que para entradas iguais a 25, 50 e 75, a saída desejada seja 1 e para os demais valores, a saída desejada seja igual a 0. O fator de *alisamento* pode aumentar ou diminuir a diferença das saídas dessas redes. Na **Figura 3-8** podemos observar que com o fator de alisamento de 0.05, os padrões 25, 50 e 75 têm um valor de saída igual a 1 e para os outros padrões os valores de saída estão próximos a zero. À medida que este fator vai sendo aumentado, a diferença entre os valores de saída vai diminuindo. Na **Figura 3-9**, o fator foi aumentado para 0.08, dessa forma os valores de saída ficaram na faixa de 0.5 e 1. Na **Figura 3-10**, o fator foi aumentado novamente, ficando em 0.1 e todos os valores de saída ficaram próximos a 1.

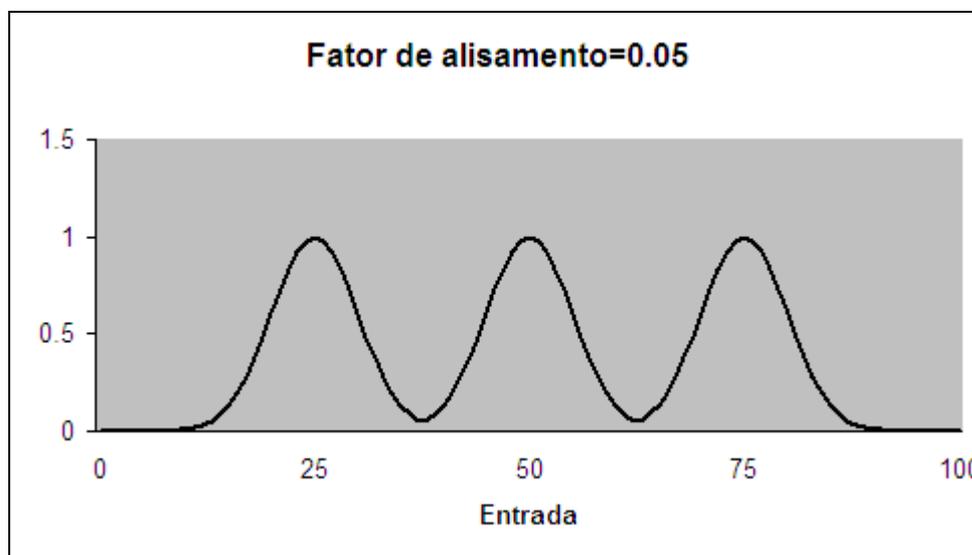


Figura 3-8 Saída x Entrada com fator de “alisamento” de 0.05.

¹⁰ Número máximo de épocas para treinamento de redes neurais é o número máximo de iterações que será executado pelo algoritmo de aprendizagem. Caso não se defina o número máximo de épocas, o algoritmo pode ficar sendo executado indefinidamente.

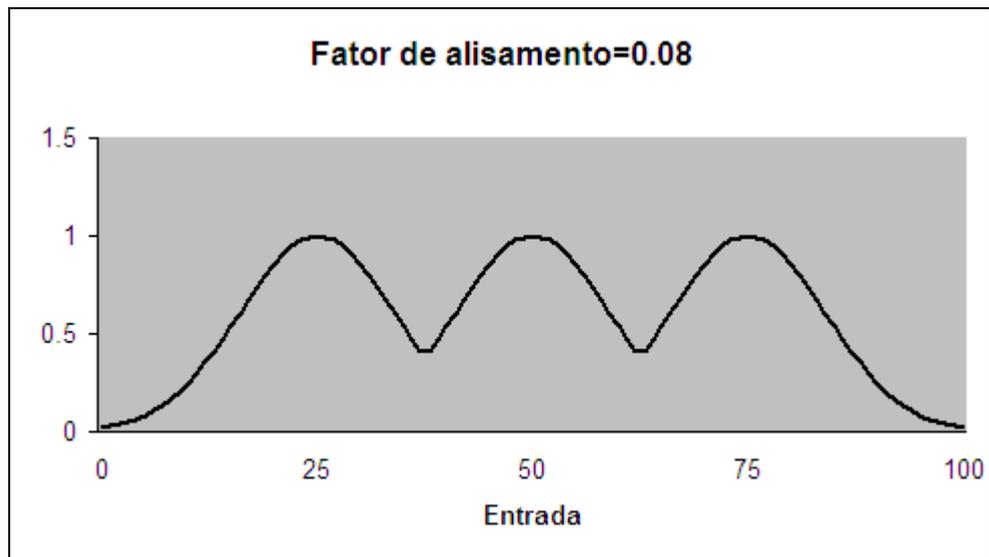


Figura 3-9 Saída x Entrada com fator de “alisamento” de 0.08.

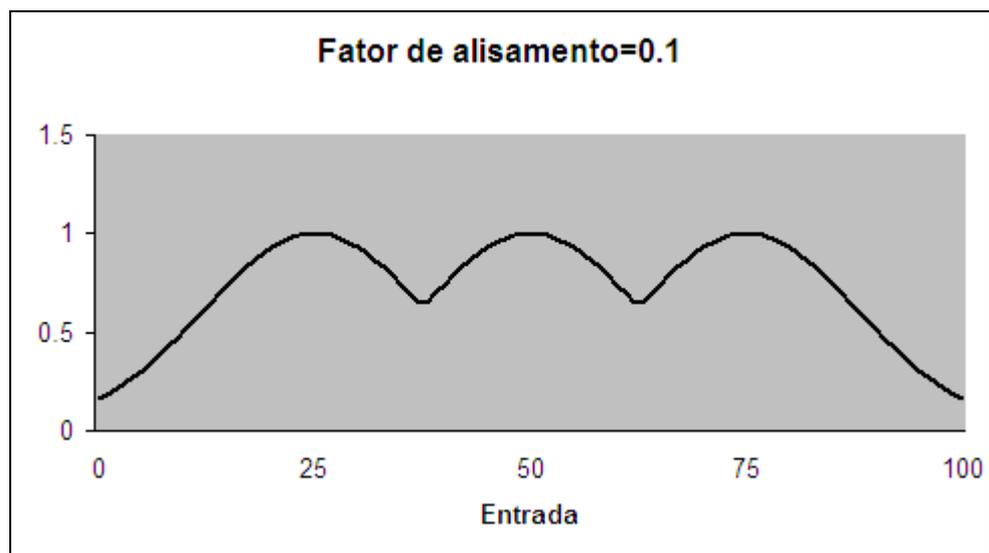


Figura 3-10 Saída x Entrada com fator de “alisamento” de 0.1.

Apesar do processo de criação de redes *MLP*, *RBF* e *PNN* ser um processo empírico, a quantidade reduzida de parâmetros para redes *RBF* e *PNN* é uma vantagem em relação às redes *MLP*.

3.7 Revisão de trabalhos relacionados

O objetivo desta seção é fazer uma revisão sobre os sistemas de reconhecimento de padrões de odor criados nos trabalhos de [Santos, 2000; Yamazaki, 2001; Yamazaki, 2004; Zanchettin, 2004; Almeida et al., 2004]. A seleção desses trabalhos se deve ao fato deles terem utilizados bases de dados obtidas pelos protótipos de nariz artificial criados por Santos [Santos, 2000].

Em Santos [Santos, 2000] foram criados quatro protótipos diferentes de narizes artificiais. Foram construídos para estes protótipos, diferentes modelos de sistemas de reconhecimento de padrões de odor com redes *MLP*, *RBF* e *Elman*. Várias bases de dados foram criadas para treinar e testar esses sistemas.

O objetivo do primeiro protótipo foi verificar a viabilidade da construção de narizes artificiais. Os sistemas de reconhecimento de padrões criados para este protótipo utilizaram redes *MLP* e obtiveram percentual de erro próximo a zero na classificação de três substâncias diferentes, o que viabilizou a construção dos outros protótipos.

Para o segundo protótipo foram criadas redes *MLP* para classificar vinho tinto, rosé e branco e também para classificar vinho branco e whisky. Uma outra base de dados composta de três safras de um mesmo vinho tinto foi gerada por este protótipo e foi utilizada para criar três sistemas de reconhecimento de padrões diferentes utilizando redes *MLP*, *RBF* e *Elman*. Era esperado que a rede *Elman* apresentasse vantagens em relação às redes *MLP* e *RBF*, pois ela trabalha com processamento temporal e como os dados são obtidos em determinados momentos do tempo ela seria mais adequada para o problema em questão. Porém, não foram observadas diferenças significativas entre os resultados apresentados pelos modelos *MLP*, *RBF* e *Elman*. Foi apontada uma desvantagem na utilização de redes *MLP* em relação às outras redes que foi o longo tempo gasto no treinamento das mesmas. Outra contribuição importante dos experimentos desta etapa foi a identificação da influência do pré-processamento dos dados para o desempenho das redes.

O terceiro protótipo criou bases de dados formadas de substâncias simples e mistura dessas substâncias. Os sistemas criados para este protótipo foram capazes de distinguir entre as substâncias puras e entre as misturas dessas substâncias.

Para o quarto protótipo foram criadas redes neurais baseadas no funcionamento do nariz biológico. Essas redes obtiveram bons resultados e foi indicado que se poderia chegar a bons resultados a partir da otimização das mesmas.

Em 2001, Yamazaki [Yamazaki, 2001] criou sistemas de reconhecimento de padrões para classificação de odores provenientes de três safras distintas de um mesmo vinho tinto com redes *MLP* e redes *TDNN*. O protótipo de nariz artificial utilizado para obtenção da base de dados era composto de seis sensores.

As redes *MLP* foram criadas com 6 nodos na camada de entrada, 4, 8, 12, 16 e 20 nodos na camada escondida e 3 nodos na camada de saída. O algoritmo para treinamento das redes *MLP* foi o algoritmo *Levenberg-Marquardt*. Esses experimentos mostraram que o erro de classificação diminuía com o aumento da quantidade de nodos na camada escondida.

A possível relação temporal entre as aquisições de odores por narizes artificiais que foi levantada por Santos [Santos, 2000] foi investigada novamente no trabalho de Yamazaki [Yamazaki, 2001]. Para realizar esta investigação foram criados sistemas com rede do tipo *TDNN*. Essas redes utilizaram o artifício de *janelas do tempo* com delay igual a 1. A rede criada foi uma rede *MLP* com 12 nodos na camada de entrada, 4, 8, 12, 16 e 20 nodos na camada intermediária e 3 nodos na camada de saída. O algoritmo para treinamento foi o algoritmo *Levenberg-Marquardt*. A camada de entrada foi formatada para receber o sinal de dois padrões simultaneamente. Um padrão correspondente à leitura dos sensores no tempo t e outro padrão correspondente à leitura dos sensores no tempo $t-1$. Os resultados obtidos na classificação dos padrões pelas redes *TDNN* mostraram que o uso do processamento temporal melhora sensivelmente a classificação dos odores. O erro médio de classificação com as redes *MLP* foi 34,56%, enquanto que o erro obtido com as redes *TDNN* foi 4,51%, além disso foram necessários menos nodos na camada intermediária das redes *TDNN* para obter bons resultados.

Em 2004, Yamazaki [Yamazaki, 2004] continuou a investigar técnicas de redes neurais para criação de sistemas de reconhecimento de padrões de odor para a mesma base de dados que foi utilizada no trabalho de 2001. A definição da arquitetura é um ponto importante na criação de sistemas com redes neurais artificiais. Dependendo do problema abordado, uma rede neural com poucas conexões pode não ser capaz de resolver a tarefa, devido à quantidade insuficiente de parâmetros ajustáveis. Por outro lado, se a rede possuir conexões demais, pode haver um ajuste excessivo aos dados de treinamento, que pode prejudicar a

capacidade de generalização da rede. Procurando resolver este problema, Yamazaki investigou a otimização simultânea de arquitetura e pesos.

A técnica definida para otimização utiliza uma abordagem automática que é capaz de gerar redes com poucas conexões e alto desempenho de generalização para uma base de dados qualquer. O processo de otimização definido por Yamazaki combina características de *simulated annealing* e *tabu search*. A técnica apresentou bons resultados na base de dados investigada.

O objetivo dos experimentos realizados por Zanchettin [Zanchettin, 2004] foi verificar o desempenho na classificação de odores captados por um nariz artificial utilizando diferentes técnicas para criação dos classificadores, bem como investigar a influência do pré-processamento dos dados no resultado final da classificação. Foram utilizadas redes *MLP*, redes com atraso no tempo, redes Feature-weighted detector e Evolving Fuzzy Neural Networks para criação dos classificadores.

As redes *MLP* obtiveram um erro médio de classificação de 14,05%, porém com a utilização do pré-processamento com transformada de *Wavelets* este erro caiu para 11,5%. Estes resultados foram obtidos após o teste de várias topologias diferentes com uma camada intermediária, variando o número de nodos nesta camada.

As redes neurais com atraso de tempo obtiveram um erro de classificação de 11.34%. Com a utilização do pré-processamento *Wavelet* o erro caiu para 0.75%. O uso da técnica de análise de Wavelet, para redução de ruído e para compressão de sinal proporcionou às redes com atraso no tempo um dos melhores resultados na classificação dos odores.

As redes Feature-weighted Detector apresentaram bons resultados para seleção dos sensores mais importantes do nariz artificial. Sua utilização permitiu uma diminuição de 50% no tamanho da base de dados e por conseqüência na complexidade do reconhecimento de odores, porém o erro médio de classificação dessas redes foi considerado alto.

As redes evolutivas Evolving Fuzzy Neural Networks obtiveram ótimos resultados de classificação. O erro médio foi de 0.88%. Além da boa generalização na classificação de odores essas redes possuem diversas funcionalidades que as tornam interessantes, como o aprendizado incremental e on-line e a inserção, extração e agregação de conhecimento.

Em 2004, Almeida [Almeida et al., 2004] criou sistemas de reconhecimento de padrões para a mesma base de dados utilizada por Yamazaki [Yamazaki, 2001]. Esta base de dados é composta por três safras diferentes de um mesmo vinho tinto. Neste trabalho foram investigadas redes *MLP* treinadas com o algoritmo *backpropagation* e *RPROP*. Foram criadas também redes com a técnica de *MLP* com *Tabu Search* e redes *RBF*. As redes *MLP* treinadas com *RPROP* e com quatro nodos na camada escondida, e as redes *RBF* com 640 nodos na camada escondida foram consideradas as melhores redes para implementação de sistemas de reconhecimento de padrões de odor para esta base de dados. Por outro lado, sistemas de reconhecimento de padrões, implementados em hardware, usando *MLP* com *Tabu search* podem ser mais indicados, pois esta técnica produz redes com menor complexidade.

3.8 Sumário do capítulo

Este capítulo teve o objetivo de apresentar as técnicas de redes neurais que serão adotadas para criação dos sistemas de reconhecimento de padrões desta dissertação. Dentre as abordagens apresentadas, as redes *PNN* são destacadas por dois motivos principais: técnica indicada para realização de tarefas de classificação e treinamento realizado em apenas um passo, o que torna a etapa de criação dessas redes muito rápida. Outro objetivo deste capítulo foi a revisão de trabalhos que criaram sistemas de reconhecimento de padrões com redes neurais, utilizando dados obtidos por protótipos de narizes artificiais criados por Santos [Santos, 2000].

O próximo capítulo apresenta métodos para medir o desempenho de sistemas de reconhecimento de padrões.

MÉTODOS PARA MEDIR O DESEMPENHO DE SISTEMAS DE RECONHECIMENTO DE PADRÕES E PARA COMPARAR OS RESULTADOS

4 Introdução

Nos narizes artificiais, os sistemas de reconhecimento de padrão são responsáveis pela implementação automática do mapeamento de cada odor obtido em uma determinada classe. O objetivo principal desses sistemas é classificar corretamente casos nunca vistos. Quando existem poucos exemplos para treinamento desses sistemas, esse objetivo se torna mais difícil de ser atingido. Nessa situação, não se sabe se o número de exemplos é suficiente ou não para extrair conceitos genéricos.

Existem várias formas de estimar o desempenho de sistemas de reconhecimento de padrões. Algumas formas são melhores que outras. Em termos estatísticos alguns métodos são considerados tendenciosos. Alguns apresentam estimativas de desempenho otimista e outros pessimistas. Encontrar um índice capaz de estimar verdadeiramente o desempenho de um sistema de reconhecimento de padrões é fundamental para que seja possível realizar uma comparação entre diferentes sistemas.

4.1 Tipos de erro

Quando um padrão de entrada é fornecido a um classificador, este responde com a classe a que supostamente este padrão pertence. Algumas vezes o classificador responde corretamente outras vezes não. A taxa de erro é certamente a forma mais utilizada para medir o desempenho de um classificador. Ela pode ser calculada através da **equação (4.1)**.

$$taxa_{erro} = \frac{número_{erros}}{número_{total}} \quad (4.1)$$

A *taxa de erro aparente* ou *taxa de erro de re-substituição* de um classificador é a taxa de erro calculada utilizando somente os exemplos de treinamento, ou seja, quando o sistema é treinado com um conjunto de exemplos e utiliza-se posteriormente o mesmo conjunto para verificar quantos erros o classificador cometeu. A *taxa de erro verdadeira* pode ser entendida como sendo a taxa de erro obtida sobre toda a população, exceto os exemplos utilizados para o treinamento. A taxa de erro verdadeira difere da taxa de erro aparente, pois a taxa de erro verdadeira exige um grande número de exemplos independentes dos exemplos de treinamento para testar o classificador.

A taxa de erro verdadeira fornece uma excelente medida de desempenho de um classificador. Porém em casos reais, conjuntos com grande número de exemplos são raros, o que inviabiliza seu cálculo na maioria das vezes. Resta então saber se é possível aproximar a taxa de erro verdadeira através de taxas de erros empíricos, calculados com pequenos conjuntos de exemplos.

Uma tentativa inicial seria estimar a taxa de erro verdadeira através da taxa de erro aparente. Porém é muito comum o classificador ter uma taxa de erro aparente menor que a taxa de erro verdadeira, pois durante a criação de um classificador pode ocorrer o processo de *overfitting*¹¹. Se a taxa de erro aparente fosse utilizada, a estimativa seria pouco confiável e extremamente otimista se comparada com a taxa de erro verdadeira [Batista, 1997].

Através da utilização de técnicas estatísticas, como por exemplo *cross-validation* [Batista, 1997], para criação dos conjuntos que serão utilizados pelos métodos de aprendizado, é possível obter taxas de erro verdadeira bem mais confiáveis que a taxa de erro aparente.

Se todos os erros são igualmente importantes, o desempenho do classificador pode ser calculado pela **equação (4.1)**. Porém, muitas aplicações necessitam fazer distinção entre os diversos tipos de erro. Por exemplo, o erro cometido no diagnóstico de uma pessoa como sendo sadia, quando na verdade ela está correndo risco de vida (conhecido como decisão *falsa negativa*) é muito mais grave do que o tipo oposto de erro, ou seja, diagnosticar como doente quando na verdade esta pessoa está sadia (conhecido como decisão *falsa positiva*). Possivelmente, com testes futuros e com o passar do tempo, o falso diagnóstico da pessoa sadia será corrigido e nenhum dano sério será causado a sua saúde. Entretanto, a pessoa doente que foi encaminhada para casa, poderá piorar e até mesmo morrer, o que pode tornar o erro muito sério.

¹¹ Overfitting é um termo utilizado para se dizer que um classificador memorizou os dados de treinamento, ou seja, que ele se especializou ou super-ajustou aos dados de treinamento.

A *matriz de confusão* pode ser utilizada quando é necessário fazer distinções entre os tipos de erros. O número de predições correta para cada classe está representado ao longo da diagonal principal da matriz. Todos os outros números são erros de diferentes tipos. A **Tabela 4-1** possui um exemplo de matriz de confusão para duas classes. A classe de *pessoa sadia* foi classificada corretamente 190 vezes, mas foi erroneamente classificada 10 vezes.

Tabela 4-1: Exemplo de matriz de confusão para duas classes.

	Classes preditas	
Classe verdadeira	Pessoa sadia	Pessoa doente
Pessoa sadia	190	10
Pessoa doente	5	195

A **Tabela 4-2** lista os tipos de erros de classificação que podem ser calculados para duas classes.

Tabela 4-2: Tipos de erro para classificação com duas classes.

	Predição positiva	Predição negativa
Classe positiva	Verdadeiramente positivo (a)	Falso negativo (b)
Classe negativa	Falso positivo (c)	Verdadeiramente negativo (d)

É comum encontrar na literatura algumas medidas de precisão de desempenho para classificadores. A **Tabela 4-3** mostra algumas medidas muito comuns; essas medidas se referem ao número de acertos do classificador, ao contrário da taxa de erro que mede o número de erros. Para descrever as medidas de desempenho de classificação mais comuns vamos inicialmente mostrar as medidas que podem ser calculadas em uma matriz de confusão para duas classes.

- (a): Classificados como positivos e pertencentes à classe positiva;
- (b): Classificados como negativos, mas pertencentes à classe positiva;
- (c): Classificados como positivo, mas pertencentes à classe negativa;
- (d): Classificados como negativos e pertencentes à classe negativa;
- a + b: Pertencentes a classe positiva;

- $c + d$: Pertencentes à classe negativa;
- $a + c$: Classificados como positivos;
- $b + d$: Classificados como negativos.

Tabela 4-3: Medidas de desempenho de classificação mais comuns.

Sensibilidade	$\frac{a}{a + b}$
Especificidade	$\frac{d}{c + d}$
Valor preditivo +	$\frac{a}{a + c}$
Valor preditivo -	$\frac{d}{b + d}$
Precisão	$\frac{a + d}{a + b + c + d}$

Sensibilidade: O percentual de casos da classe positiva corretamente classificada;

Especificidade: O percentual de casos da classe negativa corretamente classificada;

Valor preditivo +: O percentual dos casos reconhecidos pelo classificador como pertencentes à classe positiva que realmente pertencem a esta classe;

Valor preditivo -: O percentual dos casos reconhecidos pelo classificador como pertencentes à classe negativa que realmente pertencem a esta classe;

Precisão: O percentual de casos de ambas as classes corretamente classificadas.

O *custo de classificação incorreta* é uma alternativa natural à taxa de erro quando é necessário distinguir entre os diferentes tipos de erros. Neste caso, ao invés de projetar um classificador que minimize a taxa de erro, o objetivo está em minimizar os custos de classificação incorreta. O custo de classificação incorreta é simplesmente uma penalização pela ocorrência de um erro. Por exemplo, no caso com duas classes, um custo pode ser associado ao erro falso positivo, e outro custo pode ser associado ao erro falso negativo. Um custo médio de classificação incorreta pode ser obtido multiplicando cada custo pela respectiva taxa de erro. Mais formalmente, para qualquer matriz de confusão, seja $E_{ij}, i \neq j$, o número de erros da matriz de

confusão e $C_{ij}, i \neq j$, o custo para cada classificação incorreta, o custo total de classificação incorreta. C_{total} é dados pela **equação (4.2)**, onde o custo de classificação correta ($C_{ij}, i = j$) é 0.

$$C_{total} = \sum_{i=1}^n \sum_{j=1}^n E_{ij} C_{ij} \quad (4.2)$$

Para exemplificar a utilização do custo para medir o desempenho de um classificador, será utilizado o exemplo do classificador para predizer ou não uma dada doença. A matriz de confusão é a matriz apresentada na **Tabela 4-1**. A taxa de erro da matriz apresentada na **Tabela 4-1** é:

$$Taxa_{erro} = \frac{10 + 5}{190 + 10 + 5 + 195} = 0.0375 \text{ ou } 3,75\%$$

Como foi dito anteriormente, a decisão falsa negativa é considerada mais grave para este problema e, dessa forma foi elaborada uma função de custo. A **Tabela 4-4** fornece um custo dez vezes maior a decisão falsa negativa sobre a decisão falsa positiva.

Tabela 4-4: Exemplo de matriz de custo.

Classes verdadeiras	Classes preditas	
	Positivo	Negativo
Positivo	0	10
Negativo	1	0

Dessa forma, considerando os valores apresentados nas **Tabela 4-1** e **Tabela 4-4**, o custo total e o custo médio do classificador são:

$$C_{total} = (190 * 0) + (10 * 1) + (5 * 10) + (195 * 0) = 60$$

$$C_{médio} = \frac{60}{400} = 0,15 \text{ ou } 15\%$$

Vale enfatizar que apesar do classificador apresentar uma taxa de erro relativamente baixa (3,75%), este classificador pode ser considerado não apropriado para esta aplicação, pois o classificador não apresentou um custo médio satisfatório (15%).

Na análise de risco ou análise de decisão, ambos custo e benefícios são usados para avaliar o desempenho do classificador. Neste tipo de análise, os custos são chamados de riscos. Se os custos de classificação incorretas estão associados a números negativos e os ganhos para decisões corretas são associados a números positivos, então a **equação (4.1)** pode ser reescrita em termos de riscos da seguinte forma:

$$R_{total} = \sum_{i=1}^n \sum_{j=1}^n E_{ij} C_{ij} \quad (4.3)$$

Porém, no caso de aplicações reais, é muito difícil estabelecer os valores correspondentes a custos e riscos para avaliar o desempenho de um classificador. Por isto a taxa de erro é mais utilizada para medir o desempenho de classificadores.

4.2 Métodos para estimar a taxa de erro

Uma vez que a taxa de erro aparente não é uma boa estimativa para a taxa de erro verdadeira, alguma estimativa deve ser encontrada. Um dos requisitos para estimar a taxa de erro verdadeira é manter a amostra de exemplos em ordem aleatória. Isso significa que a amostra de exemplos não deve ser pré-selecionada, de forma a evitar que seja feita qualquer decisão sobre exemplos mais ou menos representativos.

4.2.1 *Estimação da taxa de erro através de treinar-e-testar*

O princípio básico que envolve os métodos de treinar-e-testar está em dividir a amostra de exemplos em dois grupos mutuamente exclusivos. O primeiro, chamado de *conjunto de treinamento* é usado para treinar o sistema de classificação. O segundo, chamado de *conjunto de teste* é usado exclusivamente para medir a taxa de erro do classificador. A idéia é que o desempenho de um classificador em um conjunto de teste seja seu desempenho em uso real. Isto significa que nenhuma informação do conjunto de teste pode estar disponível durante o treinamento [Prechelt, 1994].

Em muitos casos, o conjunto de treinamento é dividido. Alguns exemplos de treinamento são colocados no *conjunto de validação*. O conjunto de validação é usado como um *pseudoteste* que serve para avaliar o desempenho da rede durante o treinamento.

Os dois conjuntos devem ser amostras aleatórias de alguma população. Em adição, os exemplos dos dois conjuntos de amostras devem ser independentes. Isto significa que não há relação entre os conjuntos, a não ser que ambos são amostras de uma mesma população.

Se nenhuma suposição for feita a respeito da verdadeira distribuição da população, os resultados não são muito satisfatórios, pois é necessário um número muito grande de casos. Entretanto se conjuntos independentes de treinamento e teste são utilizados, e assumimos que eles são amostras representativas da população, excelentes resultados práticos são obtidos. Com esta suposição surge a seguinte questão: *Quantos exemplos de teste são necessários para uma estimação de erro precisa?*

Diferentes métodos estatísticos para criação dos conjuntos de treinamento e teste podem ser utilizados com o objetivo de responder esta pergunta. Na prática o conjunto de exemplos é freqüentemente restrito. Desta forma, é muito importante ter um número suficiente de exemplos e definir qual o método que será utilizado para criação dos conjuntos de treinamento e teste.

Existem vários métodos capazes de dividir a amostra em conjunto de treinamento e conjunto de teste. O método *Holdout* divide fixamente a amostra de exemplos, geralmente 2/3 para treinamento e 1/3 para teste. Deve-se salientar que o método *Holdout* apresenta uma taxa de erro no conjunto de teste que aproxima a taxa de erro verdadeira para grandes amostras de exemplos, o que torna este método viável nestas situações. Entretanto para pequenas amostras, os conjuntos de treinamento e de teste ficam com tamanhos reduzidos, a aproximação se torna pessimista e o método passa a não ter resultados satisfatórios. Dessa forma quando o conjunto de exemplos não é grande, podem ser aplicadas variações dos métodos de treinar-e-testar conhecidas como *resampling*. Esses métodos serão discutidos a seguir.

4.2.2 *Métodos Resampling*

O método *Holdout* é a técnica mais simples de estimativa de erro conhecida e consiste em realizar somente um experimento de treinar-e-testar. Porém uma única partição pode levar a resultados imprecisos, principalmente para pequenas e médias amostras. Os métodos de *resampling* são variações dos métodos de

treinar-e-testar nos quais são realizados vários experimentos de treinar-e-testar com diferentes partições de exemplos. Como resultado de cada experimento de treinar-e-testar é obtida uma taxa de erro, e a média dessas taxas de erro é a estimativa para a taxa de erro verdadeira. Procura-se encontrar um estimador não tendencioso, ou seja, que não estime a taxa de erro de forma pessimista ou otimista. O conceito chave de um estimador não tendencioso é a convergência para o valor verdadeiro quando aplicado a uma amostra de exemplos suficientemente grande. O estimador pode variar de amostra para amostra, mas a média sobre todas as amostras é correta. Um exemplo de estimador não tendencioso é a estimação da taxa de erro sobre o conjunto de teste. Enquanto que a estimação para um conjunto de teste pode diferir do valor verdadeiro, o valor médio das estimativas para todos os possíveis conjuntos de teste é o mesmo que o verdadeiro. A seguir são descritos os seguintes métodos de resampling:

- Random subsampling
- Cross-validation
- Bootstrapping

Random Subsampling

Quando múltiplos experimentos de treinar-e-testar são realizados, um novo classificador é projetado com cada conjunto de treinamento. A taxa de erro estimada é a média das taxas de erro dos classificadores, calculada a partir dos conjuntos de teste independentes e aleatórios. O método que realiza tal procedimento é conhecido como Random Subsampling e pode produzir melhores resultados que uma simples partição treinar-e-testar.

A **Tabela 4-5** mostra uma comparação entre o tamanho do conjunto de treinamento, o tamanho do conjunto de teste e o número de iterações dos métodos de *Holdout* e *Random Subsampling* para uma amostra de n exemplos. O método Random Subsampling soluciona um problema típico do método *Holdout*, a escolha de uma partição dos exemplos que não é representativa. Com o método *Random Subsampling* este problema é contornado, pois são criadas várias partições e o sistema é treinado e testado em cada iteração, com uma partição diferente. No caso de uma partição ser pouco representativa, este problema é amenizado, pois a taxa de erro final é calculada através das médias das taxas de erro de todas as partições.

Tabela 4-5: Comparação entre Holdout e Random Subsampling.

	Holdout	Random Subsampling
Casos de treinamento	j	j
Casos de teste	$n - j$	$n - j$
Iterações	1	num. iterações $\ll n$

Cross-validation

O método de *Cross-validation* não é um único método, mas sim uma classe de métodos para estimar a taxa de erro verdadeira. Esses métodos também são conhecidos como *k-fold cross-validation*, onde k representa o número de partições geradas aleatoriamente a partir da amostra de exemplos para treinar e testar o sistema. Nesses métodos as amostras são divididas em k partições mutuamente exclusivas. A cada iteração uma partição diferente é utilizada para testar o sistema e todas as outras $k - 1$ partições são utilizadas para treinar o sistema. A taxa de erro é a média das taxas de erro calculadas para as k iterações. Por exemplo, no caso de *4-fold cross-validation*, todo o conjunto de exemplos é dividido em 4 partições diferentes, aleatórias e mutuamente exclusivas. São realizadas 4 iterações, a cada iteração uma partição diferente é utilizada para testar o sistema as outras 3 são utilizadas para treinar o sistema. A estimativa da taxa de erro verdadeira é resultado da média das taxas de erro nas 4 partições. A **Figura 4-1** mostra um esquema para o método 4-fold cross-validation.

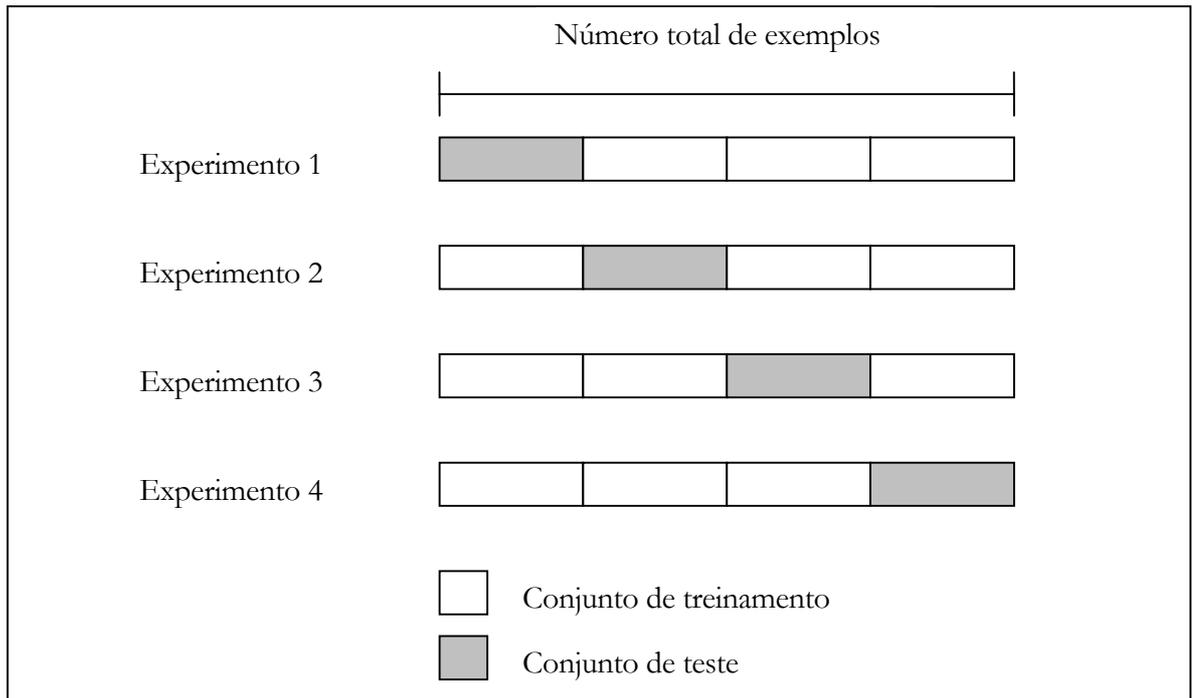


Figura 4-1: Esquema para o método 4-fold cross-validation

Leave-one-out cross-validation

O método *leave-one-out cross-validation* é um caso especial da classe de métodos *cross-validation*. Nesse método, o número de partições é igual ao número de elementos da amostra. Para uma amostra com n elementos, são executadas n iterações. A cada iteração um elemento diferente é retirado da amostra e é utilizado para testar o classificador. Os outros $n - 1$ elementos são utilizados para treinar o sistema. A taxa de erro é calculada dividindo-se o número de erros observados por n . A **Figura 4-2** mostra um esquema do método *leave-one-out cross-validation*, onde a parte destacada é um único elemento a ser testado.

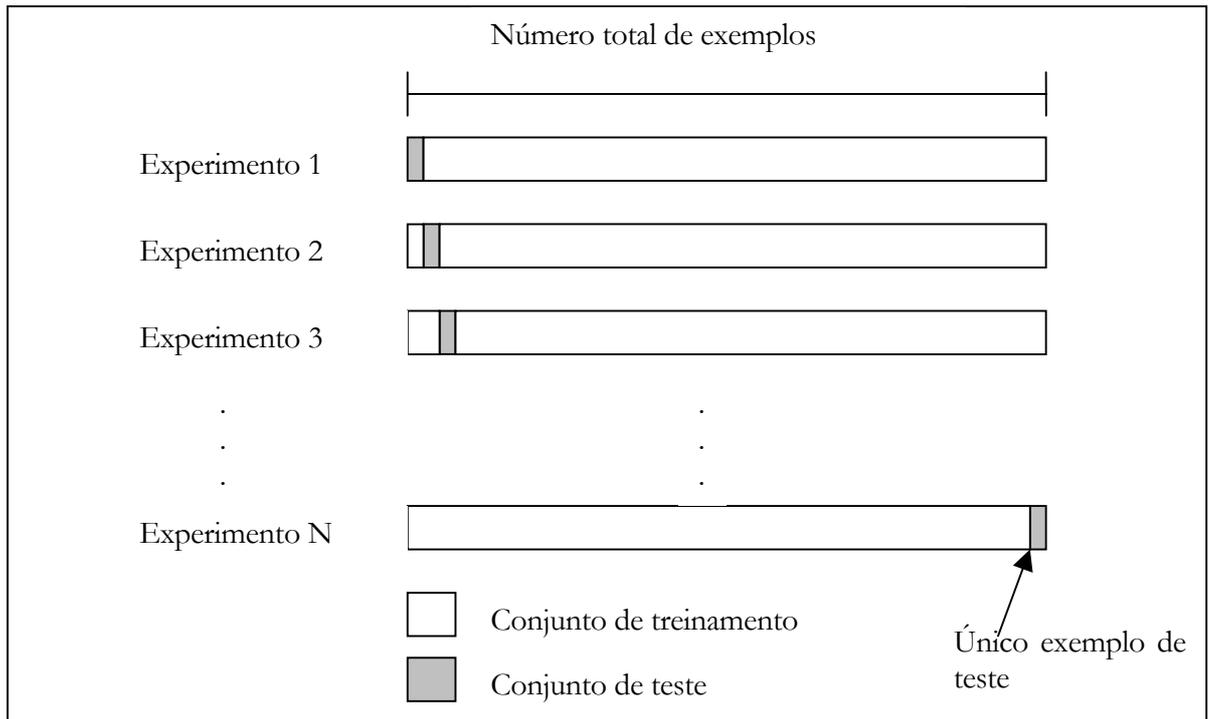


Figura 4-2: Esquema do método leave-one-out cross-validation

A estimativa de erro do método *leave-one-out cross-validation* é particularmente não tendenciosa, isto é, quando usados vários conjuntos de exemplos, a média das estimativas tenderá a taxa de erro verdadeira.

Por ser computacionalmente caro, o método *leave-one-out cross-validation* é usado geralmente em pequenos conjuntos de exemplos. Para conjuntos maiores, o método *k-fold cross-validation*, com $k = 10$, apresenta uma aproximação quase tão apurada quanto o *leave-one-out*, a custo muito inferior [Witten e Frank, 2000]. A **Tabela 4-6** mostra uma comparação entre o tamanho do conjunto de treinamento, o tamanho do conjunto de teste e o número de iterações dos métodos *k-fold cross-validation* e *leave-one-out cross-validation*.

Tabela 4-6: Comparação entre métodos cross-validation.

	Leave-one-out	10-fold cross-validation
Casos de treinamento	$n - 1$	90%
Casos de teste	1	10%
Iterações	n	10

Uma das vantagens de utilização dos métodos de *cross-validation* é que todos os casos da amostra de exemplos são usados para testar e todos os casos são usados para treinar o sistema.

Bootstrapping

Por muito tempo o método *leave-one-out* foi à técnica recomendada para avaliar o desempenho de classificadores para pequenas amostras. Tradicionalmente consideram-se pequenas amostras, aquelas que possuem 30 ou menos elementos. Uma vez que o método *leave-one-out* é não tendencioso, ele pode ser aplicado em amostras muito maiores, apresentando resultados precisos.

O método *Bootstrapping* tem mostrado resultados promissores como estimador da taxa de erro em pequenas amostras. Embora o método *leave-one-out* seja um estimador não tendencioso da taxa de erro, ainda existem dificuldades com esta técnica. Bias e variância de um estimador de taxa de erro contribuem para imprecisão da taxa de erro estimada. Enquanto que o *leave-one-out* é praticamente não tendencioso, sua variância para pequenas amostras é alta. Não tendencioso significa que o estimador após várias aplicações tende para taxa de erro verdadeira. Porém a variância do *leave-one-out* é alta. Variância mede o grau de dispersão dos resultados obtidos pelo estimador em aplicações diferentes.

O efeito da variância tende a dominar em pequenas amostras. Dessa forma, um estimador com baixa variância, mesmo que tendencioso, é potencialmente superior ao *leave-one-out* para pequenas amostras [Witten e Frank, 2000].

Para a técnica de *bootstrapping*, dada uma amostra de tamanho n , o conjunto para treinamento é criado pela seleção aleatória n elementos, com reposição, do conjunto total. Com reposição significa que os elementos são retirados do conjunto inicial, mas cópias dos elementos retirados, permanecem neste conjunto, de forma que um mesmo elemento pode ser selecionado várias vezes. O conjunto de teste é composto de todos os elementos não pertencentes ao conjunto de treinamento. A **Figura 4-3** ilustra um exemplo da aplicação do método *bootstrapping* para construção do conjunto de treinamento e do conjunto de teste a partir de um conjunto total de tamanho igual a 12. O tamanho do conjunto de treinamento gerado pela técnica de *bootstrapping* também é igual a 12. Alguns dados no conjunto de treinamento estão repetidos. O conjunto de teste é formado de todos os elementos do conjunto de dados original que não foram selecionados para o conjunto de treinamento.

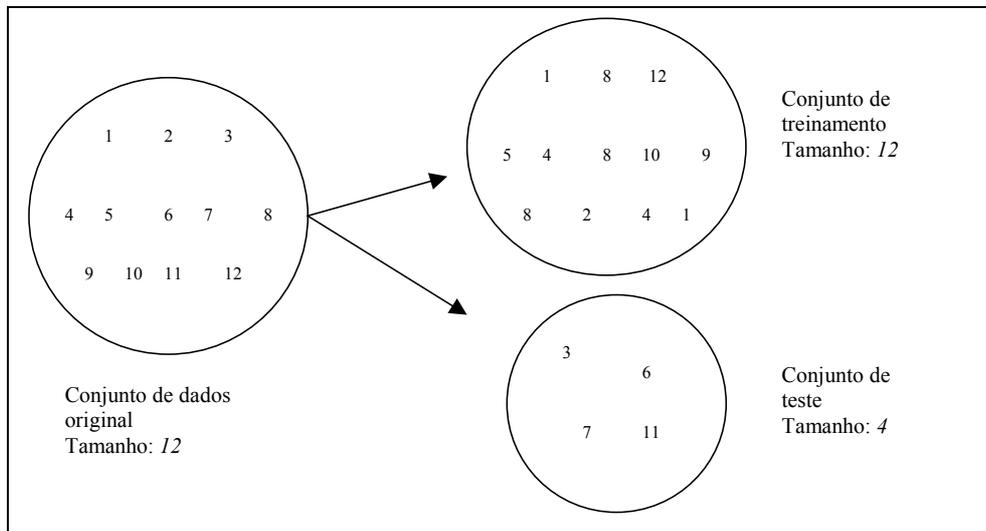


Figura 4-3 Exemplo do método bootstrapping.

Desde que os dados sejam amostrados com reposição, a probabilidade de uma dada instância qualquer não ser escolhida depois de n amostras é $(1 - 1/n)^n \approx e^{-1} \approx 0,368$. O conjunto de teste terá cerca de 36,8% dos dados e o conjunto de treinamento 63,2%. Algumas instâncias serão repetidas no conjunto de treinamento para chegar ao total de n instâncias, que é o mesmo número de instâncias do conjunto de dados original. A estimativa de precisão e_{teste} nos dados de teste será uma estimativa pessimista da taxa de erro real, pois o conjunto de teste tem aproximadamente 37% das instâncias. Para compensar, a taxa de erro no conjunto de teste é combinada com a taxa de erro aparente e_{treino} para dar a estimativa final da taxa de erro do classificador [Witten e Frank, 2000]. Assim a estimativa de erro é:

$$Erro = 0,632.e_{teste} + 0,368.e_{treino}$$

O processo é repetido várias vezes com diferentes sobreposições e os resultados são obtidos através da média dos valores de $Erro$ para todos os experimentos. Cerca de 200 iterações são consideradas necessárias para obter uma boa estimativa com o bootstrap [Batista, 1997]

$$Erro_{bootstrap} = \frac{1}{b} \sum_{i=1}^b (0,632.e_{teste} + 0,368.e_{treino})$$

4.3 Metodologia para avaliação dos classificadores

O principal objetivo de um sistema de classificação está em aprender a partir de amostras de exemplos e generalizar para novos casos. O desempenho de um classificador pode ser medido facilmente a partir da taxa de erro. É muito fácil criar classificadores que cometam poucos erros nos exemplos de treinamento, porém é muito mais difícil obter esses resultados em novos casos. Os sistemas de classificação devem ter seus resultados analisados por uma técnica capaz de estimar a taxa de erro verdadeira, a fim de prever qual será seu desempenho em novos casos. Estas técnicas fornecem a base para comparação do desempenho dos sistemas de aprendizado sobre uma mesma amostra.

As técnicas de estimação da taxa de erro são simples no conceito e precisas na prática. Elas estão baseadas na teoria estatística de *resampling*. O que diferencia basicamente as técnicas apresentadas é a forma como são divididos os dados para treinamento e teste.

Para validação dos sistemas de reconhecimento de padrões propostos neste trabalho, o método *k-fold cross-validation* foi escolhido, em virtude de este método possuir a vantagem de utilizar todos os dados tanto para treinamento quanto para teste e também por ele possuir um menor gasto computacional em relação ao método *leave-one-out cross-validation*.

Ao optar pelo o método *k-fold cross-validation*, é importante decidir quantas partições serão utilizadas, pois um número grande de partições pode levar a uma taxa de erro menor e a um tempo computacional maior, enquanto que um número pequeno de partições pode levar a uma taxa de erro maior e um tempo computacional menor.

Testes extensivos em diferentes conjuntos de dados, com diferentes técnicas de aprendizado, têm mostrado que 10 é o número de partições que permite a melhor estimativa de erro. Dessa forma, *10-fold cross-validation* tem se tornado um método padrão em termos práticos [Witten e Frank, 2000]. O uso de estratificação¹² também é importante, pois a estratificação melhora os resultados significativamente. Assim o método *10-fold cross-validation com estratificação* foi o método escolhido para criação dos dados para treinamento e teste dos sistemas e a taxa de erro de classificação estimada por este método será utilizada para realizar a comparação entre os sistemas.

¹² Estratificação é utilizada aqui para indicar que as partições geradas terão aproximadamente a mesma proporção de elementos de todas as classes.

Um ponto importante neste trabalho é a definição do método para comparar o desempenho dos classificadores criados e determinar qual é o melhor classificador para a aplicação em questão. O que se deseja saber é se os classificadores criados produzem ou não resultados equivalentes. Não existe um consenso a respeito de qual método estatístico deve ser utilizado para comparar sistemas de aprendizado. Em muitos trabalhos na área de aprendizado de máquina, os resultados dos experimentos são mostrados na forma de ranking. Entretanto, uma pergunta muito freqüente é:

Existe diferença estatisticamente significativa entre os resultados apresentados?

Ou seja,

Dados os resultados das medidas realizadas é possível afirmar com segurança que um certo sistema de aprendizado é superior a outro em um determinado conjunto de exemplos?

Para responder a essas perguntas deve-se utilizar algum método estatístico que calcule a significância da diferença entre as médias. Esses métodos ajudam a decidir se diferenças observadas entre duas amostras são realmente devidas a uma variação casual ou se são verdadeiramente significativas. Os conceitos estatísticos necessários para realização dos testes serão descritos a seguir.

4.3.1 Teoria da amostragem

A teoria da amostragem é um estudo das relações existentes entre uma *população*¹³ e as *amostras*¹⁴ dela extraídas. Ela é útil para avaliação das grandezas desconhecidas da população (como média e variância), freqüentemente denominadas de *parâmetros populacionais* ou simplesmente *parâmetros*. As grandezas que podem ser obtidas das amostras (média, variância, etc) são chamadas de *estatísticas amostrais* ou abreviadamente de *estatísticas* [Spiegel, 1979].

A teoria da amostragem também é útil para determinar se as diferenças observadas entre duas amostras são realmente devidas a uma variação casual ou se são verdadeiramente significativas. Essas questões surgem, por exemplo, quando precisamos decidir se um classificador é melhor que o outro. As

¹³ Uma população consiste na totalidade das observações em que estamos interessados.

¹⁴ Uma amostra é um subconjunto de observações selecionadas a partir de uma população.

respostas implicam no uso dos denominados *testes de significância* e *testes de hipóteses*, que são importantes na teoria das decisões [Spiegel, 1979].

Um estudo de inferências feito a respeito de uma população, mediante a utilização de amostras dela extraídas, juntamente com a precisão dessas inferências, obtidas por meio da teoria da probabilidade é denominada em geral de *inferência estatística*.

Para que as conclusões da *teoria da amostragem* e da *inferência estatística* sejam válidas, as amostras devem ser escolhidas de modo a serem representativas da população. É freqüentemente tentador selecionar uma amostra com as observações que sejam mais convenientes ou exercer julgamento na seleção da amostra. Esses procedimentos podem freqüentemente introduzir alguma tendência na amostra e, como resultado, o parâmetro de interesse será consistentemente subestimado (ou superestimado) por tal amostra. Uma das maneiras de se obter uma amostra representativa é o processo denominado *amostragem aleatória*, de acordo com o qual, cada elemento de uma população tem a mesma probabilidade de ser incluído na amostra.

Outra questão em relação às amostras é se ela foi obtida com ou sem reposição. Amostras em que cada elemento de uma população pode ser escolhido mais de uma vez são denominadas de *amostra com reposição*, enquanto que, se cada elemento não pode ser escolhido mais de uma vez, elas são denominadas de *amostras sem reposição*.

Para cada amostra pode-se calcular uma grandeza estatística, como a média, o desvio padrão, etc., que variam de amostra para amostra. Desse modo, obtém-se uma distribuição da grandeza que é denominada de *distribuição amostral*. Se a grandeza adotada for a média da amostra, então a distribuição é denominada *amostral das médias*, ou da *média*. Para cada distribuição amostral pode-se calcular a média, o desvio padrão, etc. Em conseqüência, pode-se falar da média e do desvio padrão da distribuição amostral das médias ou distribuição amostral de alguma outra estatística.

4.3.2 *Hipóteses estatísticas*

Na prática, somos chamados com muita freqüência a tomar decisões a cerca de populações, baseadas nas informações das amostras. Nesses momentos é conveniente que formulemos hipóteses acerca dessas populações. Essas suposições, que podem ser ou não verdadeiras, são denominadas de *hipóteses estatísticas* e,

em geral, consistem em considerações acerca das distribuições de probabilidade das populações. O procedimento de tomada de decisão sobre a hipótese é chamado de *teste de hipóteses*.

Em alguns casos, formula-se uma hipótese estatística com o único propósito de rejeitá-la ou invalidá-la. Por exemplo, se deseja decidir que um classificador é melhor do que outro, então se formula a hipótese de que não há diferença entre eles (isto é, que quaisquer diferenças observadas sejam devidas meramente a flutuações das amostras provenientes da mesma população). Essas hipóteses são denominadas *nulas* e são representadas por H_0 . Qualquer hipótese diferente de uma prefixada é denominada *hipótese alternativa*. Uma hipótese alternativa da hipótese nula é representada por H_1 .

4.3.3 *Teste de hipóteses e significância*

Ao testar uma hipótese há dois tipos de erros que podem ser cometidos:

- Erro do tipo I – Rejeitar a hipótese quando ela for verdadeira.
- Erro do tipo II – Aceitar a hipótese quando ela for falsa.

Em ambos os casos ocorreram decisões erradas ou erros de julgamento. Para que quaisquer testes de hipótese sejam bons, eles devem ser planejados de modo que os erros de decisão sejam reduzidos ao mínimo. Isso não é uma tarefa simples, pois para um dado tamanho de amostra, a tentativa de diminuir um certo tipo de erro é acompanhada, em geral, pelo acréscimo de outro tipo. Na prática, um tipo de erro pode ser mais importante do que outro, de modo que se deve procurar uma acomodação que favoreça a limitação do erro mais sério. O único caminho para redução de ambos os tipos de erros consiste em aumentar o tamanho da amostra, o que pode ou não ser possível.

Ao testar uma hipótese estabelecida, a probabilidade máxima com a qual se sujeita a correr o risco de um erro do tipo I é denominada nível *de significância do teste*. Essa probabilidade, freqüentemente representada por α , deve ser especificada antes da extração de quaisquer amostras, de modo que os resultados obtidos não influenciem na sua escolha.

Na prática, é usual a adoção de um nível de significância *0.05* ou *0.01*, embora possam ser usados outros valores. Se por exemplo é escolhido um nível de significância *0.05* ou *5%*, no planejamento de um teste de hipótese, há então cerca de *5* em *100*, da hipótese ser rejeitada, quando deveria ser aceita, isto é uma

confiança de 95% de que se tome uma decisão acertada. Nesses casos, diz-se que a hipótese é rejeitada no nível de significância 0.05, o que significa que a probabilidade do erro seria 0.05.

4.3.4 Testes que envolvem a distribuição normal

Admita-se que, sob uma certa hipótese, a distribuição amostral de uma estatística S é normal, com média μ_s e desvio padrão σ_s . Então a distribuição da variável reduzida (ou score z), dado por $z = (S - \mu_s)/\sigma_s$ é a distribuição normal reduzida (com média 0 e variância 1) que está representada na **Figura 4-4**.

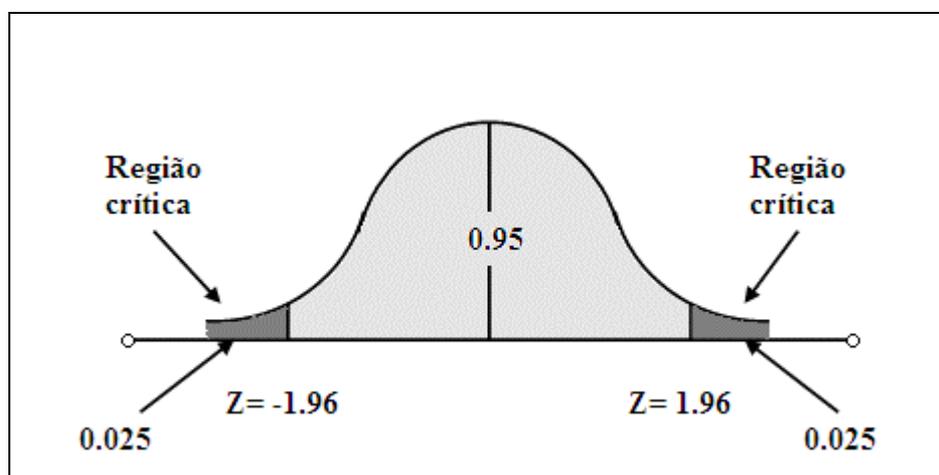


Figura 4-4: Distribuição normal reduzida.

Como está indicado na figura, pode-se estar 95% confiante de que se a hipótese for verdadeira, o escore z de uma estatística amostral real, S , estará compreendido entre -1.96 e 1.96 (visto que a área subentendida pela curva normal, entre esses valores, é 0.95).

Entretanto, se ao escolher uma única amostra aleatória, verifica-se que o escore z dessa estatística cai fora do intervalo de -1.96 e 1.96 , concluir-se-ia que o evento poderia ocorrer com a probabilidade de 0.05 (área total sombreada na figura) se a hipótese estabelecida fosse verdadeira. Dir-se-ia que o escore z difere de modo significativo do que seria esperado daquela hipótese e se estaria propenso a rejeitá-la.

A área total sombreada, de 0.05, é o nível de *significância do teste*. Ela representa a probabilidade de incorrer-se no erro de rejeição da hipótese, isto é, a probabilidade de ter cometido um erro do tipo I. Por esta

razão diz-se que a hipótese é rejeitada no nível de significância 0.05 , ou que o escore z da estatística amostral dada é significativo naquele nível.

Baseada nas observações acima pode ser formulada a seguinte regra de decisão, teste de hipóteses ou significância.

- a. Rejeição da hipótese no nível de significância 0.05 , quando o escore z da estatística J situar-se fora do intervalo de -1.96 a 1.96 . Isto equivale a dizer que a estatística amostral observada é significativa no nível 0.05 .
- b. Aceitação da hipótese (ou se desejado não tomar nenhuma decisão) no caso contrário.

Como o escore z representa papel tão importante nos testes de hipótese e na significância, ele também é denominado de *teste estatístico*.

Podem-se utilizar outros níveis de significância. Por exemplo, se for adotado o nível 0.01 , substitui-se em toda a explicação anterior o valor 1.96 por 2.58 . Se o número de casos for maior que 30 utiliza-se à distribuição normal, porém quando o número de casos for menor ou igual a 30 deve-se utilizar a distribuição de *t student*. Devem-se consultar tabelas específicas de áreas que definem os valores de z para cada nível de significância e distribuição escolhida.

4.3.5 Testes unilaterais e bilaterais

Quando se está interessado nos valores extremos da estatística J , ou nos escores z correspondentes de ambos os lados da média, isto é, em ambas as extremidades da distribuição, utiliza-se os testes *bicaudal* ou *dos dois lados*. Porém quando se está interessado apenas nos valores extremos de um mesmo lado da média, isto é, em uma extremidade da distribuição, utiliza-se os testes *unilaterais* ou *de um lado*.

Por exemplo, quando se deseja saber se um classificador é melhor que outro se deve utilizar os testes unilaterais, porém quando se deseja saber se um classificador é melhor ou pior que outro, deve-se utilizar os testes bilaterais.

4.3.6 Comparando duas alternativas

Quando se realizam n experimentos em dois sistemas (A e B) e quando há correspondência de um a um entre cada um dos enésimos testes do sistema A e do sistema B, as estatísticas obtidas nestes testes são consideradas pareadas. Por exemplo, se x_i e y_i representam a estatística S no teste i , as observações são chamadas *pareadas*.

A análise de observações *pareadas* é simples. As duas amostras são tratadas como amostras com n pares. Para cada par, a diferença estatística d pode ser calculada. O intervalo de confiança pode ser construído. Se o intervalo de confiança inclui zero, os sistemas não são significativamente diferentes.

Outra possibilidade é a análise de observações não-pareadas. Suponha que temos duas amostras de tamanho n_a e n_b para as alternativas A e B respectivamente. As observações são *não-pareadas* no sentido de que não há correspondência entre as observações i das duas amostras. Os passos para determinar o intervalo de confiança na diferença das duas estatísticas necessitam da estimativa da variância e do número de graus de liberdade.

Atualmente o teste de significância conhecido como *t-student pareado* é um dos mais difundidos para comparar o desempenho de sistemas de aprendizado. Ele consiste em aplicar o método random subsampling por n vezes e utilizar o teste de *t-student* para verificar se existe alguma diferença significativa entre as taxas de erro médias.

Podem-se também comparar os classificadores através da construção de intervalo de confiança. O intervalo de confiança é um intervalo de valores, limitado por um valor mínimo e por um valor máximo para estimar um parâmetro desconhecido da população, de forma que permita afirmar que o verdadeiro valor do parâmetro estará contido nesse intervalo.

4.4 Sumário do capítulo

O objetivo deste capítulo foi apresentar medidas que podem ser utilizada para avaliar sistemas de reconhecimento de padrões, alguns métodos utilizado para criação dos conjuntos de treinamento e teste desses sistemas e também apresentar um método estatístico para comparação dos resultados obtidos.

Dentre as medidas apresentadas, foi selecionada como medida para avaliação dos sistemas, o percentual de erro na classificação, pois esta medida foi considerada capaz de representar bem o desempenho do sistema e por ela ser de fácil implementação.

O método escolhido para criação dos conjuntos de treinamento e de teste foi o método *k-fold cross-validation*, com $k = 10$, principalmente pela vantagem deste método utilizar todos os dados tanto para treinamento quanto para teste.

O método estatístico apresentado para comparar os resultados de classificação dos sistemas de reconhecimento de padrões de odor foi o método denominado *Teste de Hipóteses*.

No próximo capítulo será apresentada a base de dados utilizada no estudo de caso, e os resultados obtidos com os experimentos para escolha da arquitetura das redes *RBF*, *PNN*, *MLP* e *TDNN*.

REALIZAÇÃO DOS EXPERIMENTOS PARA ESCOLHA DE ARQUITETURA

5 Introdução

O problema de reconhecimento de padrões trata geralmente da classificação de um conjunto de dados. O número de classes é determinado pelo tipo da aplicação. Para problemas de reconhecimento de padrões podem-se utilizar, entre outras, técnicas estatísticas, redes neurais artificiais e árvores de decisão. Neste trabalho utilizamos a técnica de redes neurais artificiais para criar classificadores que podem ser utilizados como sistema de reconhecimento de padrões em narizes artificiais. Muitas abordagens propostas para sistemas de reconhecimento de padrões em narizes artificiais fazem uso de redes *Multi-Layer Perceptron* (MLP) treinada com o algoritmo *Backpropagation* [Santos, 2000].

Neste capítulo são realizados experimentos iniciais com o objetivo de escolher a melhor arquitetura com várias técnicas de redes neurais artificiais além das redes *Multi-Layer Perceptron* (MLP). Para criação dessas redes, inicialmente vamos explicar o problema proposto e a base de dados. A avaliação dos resultados utiliza a técnica estatística *10-fold cross-validation com estratificação*, que foi descrita no capítulo 4, para criação dos conjuntos de treinamento e teste. As melhores arquiteturas serão selecionadas para a realização dos experimentos finais que estão descritos no capítulo 6.

5.1 Descrição do problema e da base de dados

O problema abordado neste trabalho é a classificação de odores provenientes de um tipo de aguarrás, denominado *Cera Inglesa*. A base de dados utilizada nesta dissertação foi obtida pela equipe do projeto AROMA, da Unicap, através da utilização de um protótipo de nariz artificial. A refinaria Gabriel Passos (REGAP/MG) da Petrobrás cedeu amostras de gases para que a base de dados fosse criada. Este é o primeiro trabalho que cria classificadores com redes neurais para esta base de dados. Para obtenção dos dados foi utilizado um protótipo de nariz artificial composto por um arranjo de oito sensores baseados em filmes de polipirrol preparados com diferentes dopantes por polimerização *in situ* em substratos de vidro condutor (ITO – Óxido de Índio e Estanho – Indium Tin Oxide). A aquisição dos dados foi realizada em três etapas:

- Coleta da linha de base, em que os sensores são expostos a um gás inerte (nitrogênio);
- Coleta dos dados do arranjo de sensores frente ao gás analisado;
- Purga dos sensores com nitrogênio.

Este processo em três etapas foi denominado de análise. Os oito sensores do arranjo de sensores são denominados pelo nome do dopante utilizado, e estão descritos na **Tabela 5-1**.

Tabela 5-1: Composição do arranjo de sensores.

Sensor	Dopante	Nome
01	PER	Perclorato de lítio
02	CAS	Ácido Canforsulfônico
03	NBS	Ácido nitrobenzeno sulfônico
04	PTSA	Ácido p-toluenosulfônico
05	ASA	Antraquinona sulfonato de sódio
06	ASA2	Antraquinona sulfonato de sódio
07	CAS2	Ácido Canforsulfônico
08	PER 2	Perclorato de lítio

A resistência de cada sensor que compõe o arranjo durante cada análise é gravada em um arquivo. Os arquivos gravados são posteriormente importados pelo Excel gerando uma planilha para a análise de cada substância. Cada planilha de dados contém três seções contendo a diferença fracional da resposta do sensor e a resistência do sensor, mas apenas os dados da segunda seção de cada planilha são utilizados nos experimentos.

- A primeira seção contém uma identificação da análise com os dados referentes à linha de base, e tem em média cinco minutos de aquisição;
- A segunda seção contém os dados resultantes da exposição do sensor a um determinado gás, e tem em média dez minutos de aquisição;
- A última seção contém os dados correspondendo à purga do sistema.

As amostras de aguarrás *Cinco Inglesa* estavam divididas em nove tipos diferentes. Para cada tipo de aguarrás foram realizadas várias análises. Cada análise foi obtida utilizando um volume de 30ml da amostra por um de tempo de exposição de dez minutos. Foram realizadas até oito análises de cada substância. A **Tabela 5-2** descreve os tipos de aguarrás e a quantidade disponível de análise para cada um.

Tabela 5-2: Descrição dos tipos de aguarrás *Cera Inglesa* e quantidade de análises para cada um dos tipos.

Descrição dos tipos de aguarrás <i>Cera Inglesa</i>	Quantidade de análises
Amostra de referência (cinref)	8
Amostra contaminada no coque com nível de contaminação de 100ppm (nafta de coque)	8
Amostra contaminada no coque com nível de contaminação de 500ppm (nafta de coque)	6
Amostra contaminada no coque com nível de contaminação de 1000ppm (nafta de coque)	6
Amostra contaminada no craqueamento com nível de contaminação de 100ppm (nafta craqueada)	8
Amostra contaminada no craqueamento com nível de contaminação de 500ppm (nafta craqueada)	8
Amostra contaminada no craqueamento com nível de contaminação de 1000ppm (nafta craqueada)	8
Óleo diesel	8
TBQ46	8

Cada análise¹⁵ foi numerada de 1 à 8 de acordo com a ordem em que foram obtidas. Como existiam apenas seis análises para as amostras contaminadas no coque com nível de contaminação 500ppm e 1000ppm, as análises de número 5 e 6 foram escolhidas aleatoriamente para serem replicadas, dessa forma ficamos com 8 análises de todos os tipo de aguarrás.

Cada conjunto de oito valores, correspondente à leitura dos oitos sensores no mesmo instante de tempo t , foi considerada com um padrão para a base de dados. As análises foram agrupadas em classes de acordo com o tipo de aguarrás. As análises numeradas de 1 à 8 de cada tipo de aguarrás possuíam às seguintes quantidades de padrões obtidos na etapa de coleta de dados.

¹⁵ Uma análise é o processo realizado pelo protótipo de nariz artificial para aquisição dos dados de uma substância.

Tabela 5-3: Quantidade de padrões obtidos por análise/classe de aguarrás *Cera Inglesa* na fase de coleta de dados.

Classe	Descrição dos tipos de aguarrás <i>Cera Inglesa</i>	Quantidade padrões obtidos por análise na fase de coleta de dados							
		1	2	3	4	5	6	7	8
1	Amostra de referência (cinref)	61	31	31	30	31	30	31	31
2	Amostra contaminada no coque com nível de contaminação de 100ppm	31	31	30	31	30	30	31	31
3	Amostra contaminada no coque com nível de contaminação de 500ppm	31	31	30	31	31	30	31 ¹⁶	30 ¹⁷
4	Amostra contaminada no coque com nível de contaminação de 1000ppm	30	13	31	31	31	31	31 ¹⁸	31 ¹⁹
5	Amostra contaminada no craqueamento com nível de contaminação de 100ppm	31	30	31	31	31	30	31	30
6	Amostra contaminada no craqueamento com nível de contaminação de 500ppm	31	31	30	31	31	31	31	31
7	Amostra contaminada no craqueamento com nível de contaminação de 1000ppm	31	30	30	31	31	31	31	30
8	Óleo diesel	31	31	30	31	31	30	31	31
9	TBQ46	31	13	31	31	31	30	31	31

Com o objetivo de utilizar informações balanceadas de todas as análises, foram selecionados os primeiros trinta padrões de cada análise para realização dos experimentos. Como a análise 2 de aguarrás contaminada no coque com nível de contaminação de 1000ppm e a análise 2 de aguarrás TBQ46 possuíam apenas 13 padrões, foi necessário replicar alguns padrões dessas análises para que elas ficassem com o mesmo número de padrões (30) das demais análises. A replicação foi feita seqüencialmente com o objetivo de manter a relação temporal dos dados replicados.

Depois de replicar algumas análises e alguns padrões, cada tipo de aguarrás ficou com 240 padrões, trinta padrões em oito análises. O conjunto composto pelos nove tipos de aguarrás *Cera Inglesa* possui 2160 padrões.

¹⁶ Esta análise é uma cópia da análise de número 5 do mesmo tipo.

¹⁷ Esta análise é uma cópia da análise de número 6 do mesmo tipo.

¹⁸ Esta análise é uma cópia da análise de número 5 do mesmo tipo.

¹⁹ Esta análise é uma cópia da análise de número 6 do mesmo tipo.

5.2 Pré-processamento dos dados

Além de definir a forma de validação do modelo, é necessário definir uma normalização para os padrões obtidos pelo arranjo de sensores e definir uma representação da classe correspondente a cada padrão. A atividade correspondente à normalização dos dados de entrada (padrões) e formatação dos dados de saída (classes) foi denominada de pré-processamento dos dados.

Os padrões foram normalizados de modo a ficar com valores entre 0 e 1. Os valores foram normalizados separadamente para cada sensor. A equação (5.1) apresenta a normalização dos valores gerados por cada sensor.

$$x_{normalizado} = \frac{x - x_{mínimo}}{x_{máximo} - x_{mínimo}} \quad (5.1)$$

Em que $x_{normalizado}$ é o valor normalizado correspondente ao valor original de resistência, e $x_{mínimo}$ e $x_{máximo}$ são, respectivamente os valores mínimo e máximo entre todos os valores do sensor em questão.

A formatação dos dados de saída utilizou a codificação de saída *1-of-m*, onde m é o número de classes. Foram criados conjuntos com saída igual a 0.1 e 0.9 e conjuntos com saída com 0 e 1.

Os padrões do *conjunto 1* e *2* possuem dezessete colunas. As oito primeiras colunas representam a leitura do arranjo de sensores no tempo t e as nove colunas finais representam a classe do padrão. No *conjunto 1*, as nove colunas que representam as classes utilizam os valores 0 e 1, onde 1 indica que é uma determinada classe. No *conjunto 2*, as nove colunas que representam as classes utilizam os valores 0.1 e 0.9, onde 0.9 indica que é uma determinada classe.

Os padrões dos *conjuntos 3* e *4* possuem vinte e cinco colunas. As oito primeiras colunas representam a leitura obtida pelo arranjo de sensores para um odor num tempo t e as próximas oito colunas representam a leitura obtida pelo arranjo de sensores para o mesmo odor num tempo $t+1$. No *conjunto 3*, as nove colunas que representam as classes utilizam os valores 0 e 1, onde 1 indica que é uma determinada classe. No *conjunto 4*, as nove colunas que representam as classes utilizam os valores 0.1 e 0.9, onde 0.9 indica que é uma determinada classe. Neste caso a quantidade total de padrões é menor, pois para o trigésimo padrão, não

existe informação para as colunas de tempo $t+1$. Nestes conjuntos cada classe possui 232 padrões, ou seja, vinte e nove padrões em oito análises, totalizando 2088 padrões.

5.3 Divisão do conjunto de dados

De acordo o método *10-fold cross-validation* com estratificação, que foi escolhido para avaliação dos classificadores no capítulo 4, os padrões devem ser divididos em 10 partições com estratificação. A **Figura 5-1** ilustra o a criação das 10 partições. A **Tabela 5-4** possui a lista de todas as base de dados criadas a partir dos quatro conjuntos disponíveis.

Para criar as bases de dados com as 10 partições, foram codificados dois programas no *Matlab, versão 6.0.0.88, release 12*. Os dois programas recebem os mesmos parâmetros que estão descritos a seguir:

- Número total de padrões.
- Número de padrões por classe.
- Número de classes.
- Número de partições que serão geradas ($k=10$).
- Conjunto de dados (1, 2, 3 ou 4).

O primeiro programa funciona da seguinte maneira: Inicialmente os padrões de cada classe são carregados em matrizes separadas, na mesma ordem em que foram obtidos, pelo protótipo de nariz artificial, durante o processo de aquisição dos dados. Em seguida cada matriz é organizada aleatoriamente. As matrizes são divididas em k partes, em que cada parte possui (*número de padrões por classe/k*) elementos. São criadas então as k partições do método *cross-validation*, onde cada partição i é formada pela parte i de cada uma das matrizes. Dessa forma cada partição tem o mesmo número de padrões de cada classe. No final as k partições são organizadas aleatoriamente.

O segundo programa funciona da seguinte maneira: Inicialmente os padrões de cada classe são carregados em matrizes separadas, na mesma ordem em que foram obtidos, pelo protótipo de nariz artificial, durante o processo de aquisição dos dados. Em seguida as matrizes são divididas em k partes, em que cada parte possui (*número de padrões por classe/k*) elementos. São criadas então as k partições do método *cross-validation*, onde cada partição i é formada pela parte i de cada uma das matrizes. Dessa forma cada partição tem o mesmo número de padrões de cada classe.

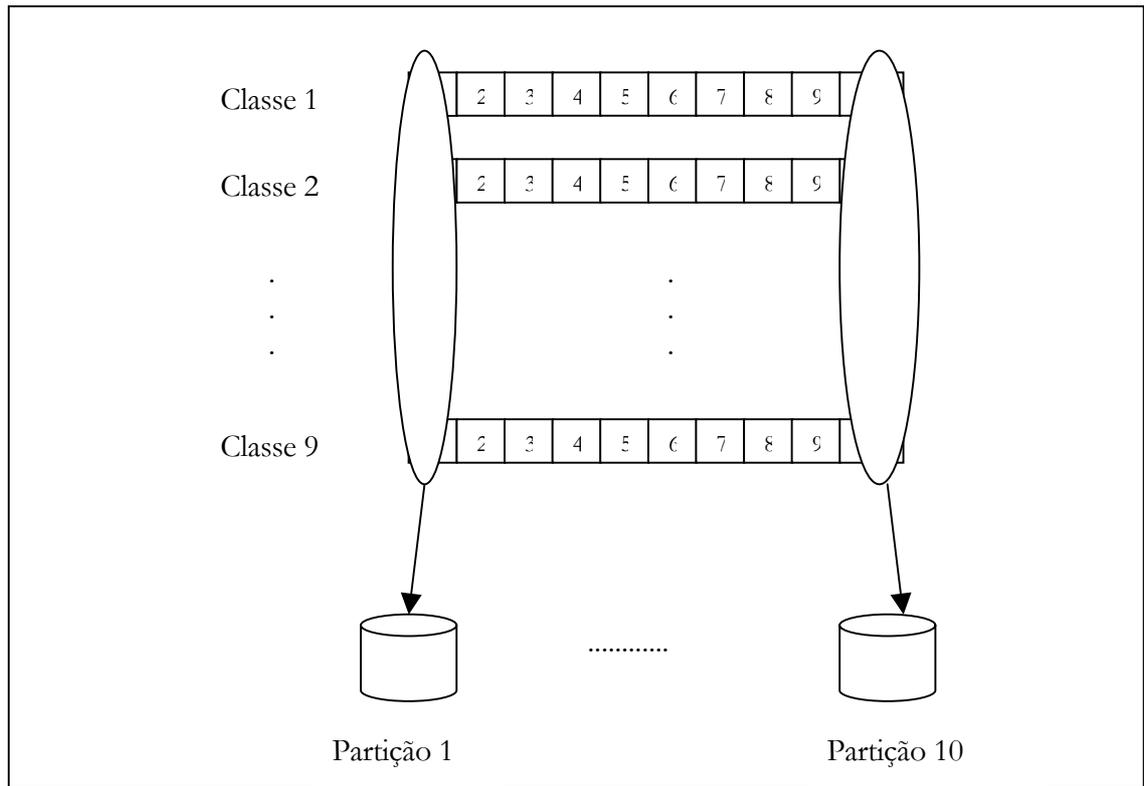


Figura 5-1: Formação dos $k=10$ conjuntos

Tabela 5-4: Lista das base de dados criadas com *10-fold cross-validation*.

Nome	Número de colunas de entrada	Total de colunas de	Formatação das colunas de saída	Organização	Número de partições	Padrões de cada classe por partição
<i>Base de dados 1</i>	8	17	0 e 1	Aleatória	10	24
<i>Base de dados 2</i>	8	17	0.1 e 0.9	Aleatória	10	24
<i>Base de dados 3</i>	16	25	0.1 e 0.9	Seqüencial	10	23
<i>Base de dados 4</i>	16	25	0.1 e 0.9	Aleatória	10	23
<i>Base de dados 5</i>	16	25	0 e 1	Aleatória	10	23

5.4 Aspectos observados para escolha de arquitetura

O objetivo desta etapa é investigar os melhores parâmetros para criação de classificadores com rede *RBF*, *PNN*, *MLP* e *TDNN*. Para cada técnica adotada existem parâmetros e critérios de parada de

treinamento diferentes. Como foi descrito na seção 3.6, os parâmetros e critérios de parada de treinamento produzem resultados com diferenças significativas. Dessa forma é necessário executar vários experimentos para selecionar os parâmetros que se ajustam melhor ao problema.

A medida utilizada para seleção dos parâmetros é o menor percentual de erro médio quadrático (*MSE*) do conjunto de validação. Para o treinamento das redes *RBF* e *PNN* não existe conjunto de validação, neste caso a medida utilizada é o menor erro médio quadrático do conjunto de treinamento. A **equação 5.3** apresenta a fórmula utilizada para calcular a percentagem de erro médio quadrático (*MSE*).

$$E = 100 \frac{O_{\max} - O_{\min}}{N \cdot P} \sum_{p=1}^P \sum_{i=1}^N (o_{pi} - t_{pi})^2 \quad (5.2)$$

Onde O_{\max} e O_{\min} são os valores de mínimo e máximo dos coeficientes de saída, na representação do problema. N é o número de unidades de saída da rede neural, P é o número de padrões (exemplos) da base de dados considerada, o_{pi} é a saída calculada por cada unidade de saída da rede e t_{pi} é a saída desejada da rede para cada unidade. É importante observar que podem produzir uma percentagem de erro médio quadrático (*MSE*) superior a 100% se elas utilizarem função de ativação não restritas aos valores O_{\max} e O_{\min} nos nodos da camada de saída [Prechelt, 1994]

O erro de classificação no conjunto de teste segundo o critério “o vencedor leva tudo” também será observado em todos os experimentos. A **equação 5.3** apresentou a fórmula para cálculo do erro de classificação.

$$E = 100 \cdot \frac{\text{Número}_{\text{erros}}}{\text{Número}_{\text{total}}} \quad (5.3)$$

Dessa forma os aspectos observados ao final de cada experimento são:

- Percentagem de erro médio quadrático no conjunto de treinamento;
- Percentagem de erro médio quadrático no conjunto de validação (quando houver);
- Percentagem de erro médio quadrático no conjunto de teste;
- Erro de classificação no conjunto de teste.

Os critérios de parada de treinamento que foram observados dependem do tipo da rede neural que está sendo criada. Estes critérios estão descritos na metodologia de treinamento de cada rede.

5.5 Realização dos experimentos para escolha da arquitetura

Todos os experimentos utilizaram a ferramenta *Matlab*, versão 6.0.0.88, release 12.

5.5.1 Experimentos com redes RBF

As redes RBF possuem duas camadas que computam funções. A entrada não é considerada como uma camada, pois ela é formada de nodos que não computam função. As redes RBF criadas possuem oito unidades na entrada, pois cada padrão de entrada é formado de oito valores. A quantidade de nodos na camada oculta é definida dinamicamente durante a criação da rede e depende do algoritmo utilizado. Os nodos da camada oculta utilizam função de ativação de base radial. A camada de saída possui nodos que executam uma combinação linear das funções de base radial. Ela é formada por nove nodos, pois foi utilizada a codificação de saída *1-of-m*, onde *m* é o número de classes. A base de dados utilizada foi a *base de dados 1*.

As redes RBF podem ser criadas de duas formas diferentes no *Matlab*. Na primeira forma de criação, o comando utilizado foi o comando *newrb* que cria nodos na camada escondida de acordo com o parâmetro de MSE esperado (*GOAL*). Na segunda forma, o comando utilizado foi o *newrbe*, que cria tantos nodos na camada escondida quanto são os padrões utilizados para o treinamento.

5.5.1.1 Metodologia de treinamento das redes RBF com o comando *newrb*

A função *newrb* cria interativamente uma rede neural. A função é chamada da seguinte maneira:

`net = newrb (P, T, GOAL, SPREAD, MN, DF)`, onde:

- P - Matriz de dados de entrada.
- T - Matriz de dados de saída (target).
- GOAL - Soma erro quadrático. Valor padrão é 0. Foram realizados experimentos com valores de 1000, 800, 600, 400 e 200.
- SPREAD - Largura para função de base radial. O valor default é 1. Foram realizados experimentos com 0.05, 0.08, 0.1, 0.5 e 1.
- MN - Número máximo de nodos na camada oculta. O valor default é o número de padrões.

A **Figura 5-2** ilustra o diagrama de uma das redes *RBF* criadas no *Matlab* com o comando *newrb*.

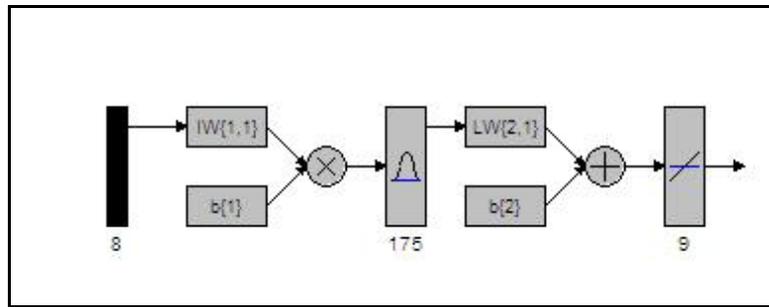


Figura 5-2: Diagrama de uma rede criada com *newrb*.

Inicialmente a camada oculta não tem nodos. Os seguintes passos são repetidos até que o erro médio quadrático seja menor que o valor passado como parâmetro *GOAL* ou que o número máximo de nodos seja atingido. Os valores testados de *GOAL* e de *SPREAD* foram escolhidos em ordem crescente, partindo de valores pequenos até grandes valores.

1. A rede é simulada.
2. O vetor de entrada com maior erro é encontrado.
3. Um nodo *RADBAS*²⁰ é incluído com peso igual ao vetor.
4. Os pesos da camada de saída são ajustados para minimizar o erro.

Com o objetivo de escolher o parâmetro *GOAL* e o parâmetro *SPREAD*, foram criadas cinquenta redes *RBF* para cada *SPREAD* variando-se o *GOAL* e os conjuntos de treinamento e de teste. Da seguinte maneira:

1. Repetir para *SPREAD* (0.05, 0.08, 0.1, 0.5 e 1).
 - a. Repetir para *GOAL* (1000, 800, 600, 400 e 200)
 - i. Repetir k (variando de 1 a 10) vezes.
 - ii. Conjunto de teste = Conjunto k .
 - iii. Conjunto de treinamento = Todos os conjuntos menos o conjunto k .
 - iv. Criar a rede.
 - v. Calcular MSE e erro de classificação para o conjunto de treinamento.
 - vi. Calcular MSE e erro de classificação para o conjunto de teste.

²⁰ Um nodo *RADBAS* do *Matlab* é um nodo cuja função de transferência é uma função de base radial. A entrada desses neurônios é o vetor de distância entre o vetor de peso e o vetor de entrada multiplicada pelo bias. A função de transferência é $\text{radbas}(n) = e^{-n^2}$

5.5.1.2 Resultados obtidos com *newrb*

A **Tabela 5-5** apresenta a quantidade de nodos da camada oculta, o erro médio quadrático do treinamento e do teste e o percentual de erro de classificação no conjunto de teste para cada valor de *SPREAD* e *GOAL* testado. A escolha da melhor topologia de rede *RBF* foi baseada no erro médio quadrático do conjunto de treinamento, pois não existe conjunto de validação na criação de redes *RBF*. Escolher a topologia de uma rede *RBF* criada com o comando *newrb*, é basicamente escolher o valor do parâmetro *GOAL* e o valor do parâmetro *SPREAD*.

Tabela 5-5: Resultados com *newrb* para seleção de *GOAL* e *SPREAD*.

SPREAD	GOAL	Quantidade nodos		MSE treinamento		MSE teste		%Erro classificação teste	
		Média	desvio	Média	Desvio	Média	Desvio	Média	Desvio
0.05	1000	17.4	0.516398	5.63974	0.037083	5.783595	0.145055	48.19444	3.608934
	800	24	0.471405	4.485454	0.052067	4.590477	0.277934	31.2037	3.567452
	600	36.7	0.674949	3.3641	0.036685	3.552861	0.273587	19.07407	2.653244
	400	57.1	1.100505	2.269107	0.015933	2.510031	0.291112	11.57407	2.672918
	200	104.3	5.831905	1.125343	0.016328	1.327297	0.227031	3.703704	1.195365
0.08	1000	17.9	0.316228	5.636365	0.07326	5.773734	0.207518	43.7037	3.141484
	800	26	0	4.486586	0.024352	4.693137	0.240807	31.75926	3.547369
	600	37.8	0.918937	3.386751	0.014472	3.504559	0.24919	18.88889	3.927158
	400	63.4	1.264911	2.260581	0.019278	2.451477	0.269133	10.60185	3.391202
	200	122.4	5.232378	1.135022	0.007108	1.427894	0.272856	4.074074	1.810232
0.1	1000	18.4	0.516398	5.643942	0.056661	5.793821	0.209512	43.05556	4.493895
	800	27.2	0.632456	4.516405	0.034959	4.631151	0.278126	29.53704	3.933218
	600	39.6	0.516398	3.401056	0.027202	3.545403	0.263191	18.42593	3.695979
	400	63.2	1.47573	2.263807	0.013899	2.47196	0.303908	10.37037	3.35413
	200	130.2	3.392803	1.134889	0.006344	1.69189	1.032217	3.564815	1.431946
0.5	1000	23.2	1.229273	5.660176	0.018251	5.893619	0.349491	39.86111	4.262445
	800	40.5	1.354006	4.546466	0.017261	14.79506	28.90986	30.60185	4.378204
	600	61.2	1.619328	3.402624	0.010685	57.53594	105.9626	18.00926	3.467589
	400	95.1	2.766867	2.269591	0.009683	997.9545	2006.682	7.407407	2.236324
	200	168.6	4.115013	1.138623	0.002915	527293.9	1619054	2.685185	0.780809
1	1000	31	1.333333	5.660991	0.029612	30.18903	53.93599	36.52778	3.355905
	800	45.7	1.946507	4.541674	0.021543	211.7566	387.9458	30.27778	2.620732
	600	68.9	2.766867	3.404091	0.018291	3528.665	6096.554	17.87037	3.253209
	400	105.7	3.267687	2.277726	0.006693	103402.9	164044.4	6.990741	2.109743
	200	443.4	528.5065	1.321094	0.577463	4873059	8663518	3.842593	3.632613

Os melhores resultados foram obtidos com *GOAL* igual a 200. Nesses casos, os valores de *SPREAD* iguais a 0.05, 0.08, 0.1 e 0.5 produziram *MSE* do conjunto de treinamento com valores praticamente iguais. A escolha foi então pelo valor de *SPREAD* de 0.5, pois ele garante um maior poder de generalização em relação aos outros três. O *SPREAD* define a suavidade (*alisamento*) para função de aproximação, valores muito grandes significam que poucos nodos serão necessários para ajustar uma função que muda rapidamente. Valores muito pequenos significam que muitos nodos serão necessários para ajustar uma função que muda suavemente e a rede pode não generalizar bem. Foi observado que quanto menor for o valor de *GOAL*, maior será a quantidade de nodos criados na camada oculta e maior será o tempo de treinamento da rede. Para criar redes com *GOAL* menor que 200 foi necessário um longo tempo de treinamento, tornando inviável a execução de experimentos com esses valores. A topologia selecionada para os experimentos finais com *newrb* foi à topologia com *GOAL* igual a 200 e *SPREAD* igual a 0.5.

5.5.1.3 Metodologia de treinamento das redes RBF com o comando *newrbe*

A função *newrbe* cria rapidamente uma rede RBF. Esta função pode criar uma rede com erro zero no vetor de entrada. A função é chamada da seguinte maneira:

`net = newrbe (P, T, SPREAD)`, onde:

- P - Matriz de dados de entrada.
- T - Matriz de dados de saída (target).
- SPREAD - Largura para função de base radial. O valor default é 1.

Esta função cria tantos nodos na camada oculta quanto são os padrões no vetor de entrada e faz os pesos dessa camada iguais ao vetor de entrada transposto. Assim, a camada oculta é formada de nodos *RADBAS*²¹ com cada nodo agindo como um detector para um vetor de entrada diferente. Se há *n* vetores de entrada, então haverá *n* nodos *RADBAS*. A **Figura 5-3** ilustra o diagrama de uma das redes RBF criadas com o comando *newrbe*.

²¹ Um nodo *RADBAS* do *Matlab* é um nodo cuja função de transferência é uma função de base radial. A entrada deste nodo é o vetor de distância entre o vetor de peso e o vetor de entrada. A função de transferência é $\text{radbas}(n)=e^{-n^2}$.

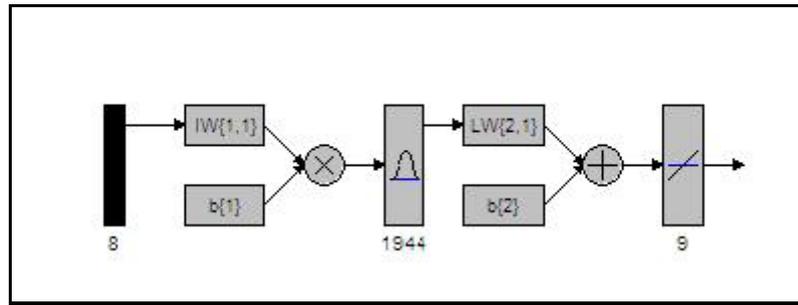


Figura 5-3: Diagrama de uma rede criada com *neurbe*.

O único parâmetro que necessita ser especificado com o comando *neurbe* é o parâmetro *SPREAD*. Com o objetivo de escolher o melhor valor, foram selecionados os mesmos valores de *SPREAD* dos experimentos com *neurbe* e foram criadas dez redes *RBF* para cada *SPREAD* da seguinte maneira:

1. Repetir para *SPREAD* (0.05, 0.08, 0.1, 0.5, 1).
 - a. Repetir k (variando de 1 a 10) vezes.
 - i. Conjunto de teste = Conjunto k .
 - ii. Conjunto de treinamento = Todos os conjuntos menos o conjunto k .
 - iii. Criar a rede.
 - iv. Calcular MSE e erro de classificação para o conjunto de treinamento.
 - v. Calcular MSE e erro de classificação para o conjunto de teste.

5.5.1.4 Resultados obtidos com *neurbe*

A **Tabela 5-6** apresenta a quantidade de nodos da camada oculta, o erro médio quadrático do treinamento e do teste e o percentual de erro de classificação no teste para cada valor de *SPREAD*.

Tabela 5-6: Resultados obtidos com *neurbe* e para seleção de *SPREAD*.

SPREAD	Quantidade nodos		MSE treinamento		MSE teste		%Erro classificação teste	
	Média	desvio	Média	Desvio	Média	Desvio	Média	Desvio
0.05	1944	0	0.0621	0.0216	127.3336	130.1690	1.3889	0.7238
0.08	1944	0	0.1137	0.0289	369.7772	454.1146	1.1574	0.6269
0.1	1944	0	0.2018	0.0357	784.1488	737.1007	1.1574	0.7638
0.5	1944	0	1.4267	0.1756	30069.6966	43278.3553	3.3796	1.1559
1	1944	0	3.0187	0.2865	21337.9373	54639.3091	15.5556	5.3065

Como nas redes *RBF* anteriores, foi utilizado o valor do *MSE* no treinamento para selecionar a melhor arquitetura. Todas as redes criadas possuem 1944 nodos na camada oculta, pois o algoritmo de treinamento do comando *newrbe* cria um nodo para cada padrão do conjunto de treinamento. O menor valor de *MSE* no conjunto de treinamento foi obtido com *SPREAD 0.05*, logo este foi o valor selecionado para realização dos experimentos finais com *newrbe*.

5.5.2 Experimentos com redes PNN

A função *newpnn* cria rapidamente redes *PNN*. O treinamento e a operação das redes *PNN* foi explicado na seção 3.5. As unidades da *camada de soma* que foram referenciadas na seção 3.5 são representadas no *Matlab* pela operação de soma que liga a saída dos nodos da camada oculta com os nodos da camada de saída. A base de dados utilizada foi a *base de dados 1*.

A **Figura 5-4** ilustra o diagrama básico das redes criadas com *newpnn*. Note que a entrada é formada de oito unidades, já que cada padrão de entrada é composto de oito valores. A camada oculta é formada por nodos que computam funções de base radial. A quantidade de nodos nesta camada é igual ao número de padrões utilizado no treinamento. A camada de saída é formada por nove nodos, pois foi utilizada a codificação de saída *1-of-m*, onde *m* é o número de classes. A base de dados utilizada foi a *base de dados 1*.

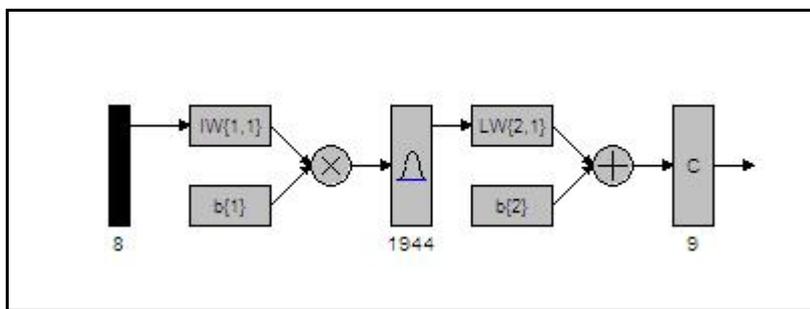


Figura 5-4: Diagrama das redes *PNN*.

As redes *PNN* possuem algumas diferenças em relação às redes *RBF*. As redes *PNN* calculam a probabilidade a posteriori de um dado padrão de entrada ser de uma das classes predefinidas. A saída de uma rede *PNN* é uma saída binária (0 ou 1) e esta saída define a classe mais provável para o padrão de entrada. Os nodos da camada de saída das redes *PNN* utilizam função de ativação “*competitiva*” (esta função produz uma saída igual a 1 para o nodo que define a classe mais provável do padrão de entrada e 0 para os demais

nodos). As redes *RBF* possuem na camada de saída nodos que utilizam função de ativação linear. A saída das redes *RBF* é uma saída linear, logo, para que as redes *RBF* sejam utilizadas em problemas de classificação é necessária à utilização de algum critério para definição da classe do padrão de entrada, como por exemplo, o critério “o vencedor leva tudo”.

5.5.2.1 Metodologia de treinamento das redes PNN

As redes PNN foram criadas com o comando *newpnn*. Ele é chamado da seguinte maneira:

`net = newpnn (P, T, SPREAD)`, onde:

- P - Matriz de dados de entrada.
- T - Matriz de dados de saída (target).
- SPREAD - Largura para função de base radial. O valor default é 0.1.

Com o objetivo de escolher o parâmetro *SPREAD*, foram selecionados em ordem crescente, cinco valores de *SPREAD* e depois foram criadas dez redes PNN para cada *SPREAD* da seguinte maneira:

1. Repetir para SPREAD (0.0001, 0.0005, 0.005, 0.5, 0.1).
 - a. Repetir k (variando de 1 a 10) vezes.
 - i. Conjunto de teste = Conjunto *k*.
 - ii. Conjunto de treinamento = Todos os conjuntos menos o conjunto *k*.
 - iii. Criar a rede.
 - iv. Calcular MSE e erro de classificação para o conjunto de treinamento.
 - v. Calcular MSE e erro de classificação para o conjunto de teste.

5.5.2.2 Resultados obtidos com PNN

A **Tabela 5-7** apresenta a quantidade de nodos da camada oculta, o erro médio quadrático do treinamento e do teste e o percentual de erro de classificação no conjunto de teste para cada valor de *SPREAD* testado.

Tabela 5-7: Resultados obtidos com *neupnn* para seleção de *SPREAD*.

SPREAD	Quantidade nodos		MSE treinamento		MSE teste		%Erro classificação teste	
	Média	desvio	Média	Desvio	Média	Desvio	Média	Desvio
0.0001	1944	0	0.0206	0.0048	5.5247	0.6526	24.8611	2.9366
0.0005	1944	0	0.0206	0.0048	0.2572	0.1306	1.1574	0.5876
0.005	1944	0	0.3418	0.0383	0.5967	0.2692	2.6852	1.2112
0.05	1944	0	7.0725	0.3052	7.9630	0.8569	35.8333	3.8562
0.1	1944	0	9.9303	0.5387	10.8539	0.6309	48.8426	2.8393

Como não existe conjunto de validação para as redes *PNN*, utilizamos o valor do *MSE* no treinamento para selecionar a melhor arquitetura. Os valores de *SPREAD* de *0.0001* e *0.0005* apresentaram os menores valores *MSE* para o treinamento. Esses valores foram iguais, logo o *MSE 0.0005* foi selecionado para realização dos experimentos finais, pois ele permite um maior poder de generalização. O treinamento das redes *PNN* foi o treinamento mais rápido de todos os experimentos executados.

5.5.3 Experimentos com redes *MLP*

Foram criadas redes *MLP* com apenas uma camada intermediária. A camada de entrada é formada de oito nodos, já que cada padrão é composto de oito valores, que não computam função. A camada de saída é formada de nove nodos, pois foi utilizada a codificação de saída *1-of-m*, onde *m* é o número de classes. Os nodos da camada de saída possuem função de ativação *sigmóide logística*. Os nodos da camada intermediária também possuem função de ativação *sigmóide logística*. Foram realizados experimentos aleatórios com *8, 10, 12, 14, 16, 18, 20, 22, 24* e *26* nodos na camada intermediária. A rede contém todas as possíveis conexões *feedforward* entre as camadas adjacentes e não possui conexões que ligam nodos de camadas não-adjacentes.

Foram utilizadas as *bases de dados 1* e *2* nos experimentos *MLP*, pois segundo o relatório do Proben1 [Prechelt, 1994] devem ser realizados experimentos com saída formatada para *0.1* e *0.9* quando se perceber que os neurônios de saída estão saturando, e isto aconteceu com as redes *MLP*. A investigação de outros valores para os parâmetros do algoritmo de aprendizado com o objetivo de evitar o problema de saturação dos neurônios de saída não foi realizada pois a formatação dos dados de saída entre *0.1* e *0.9* produziu uma melhora significativa dos resultados de classificação das redes *MLP*. As redes criadas com a *base de dados 2* (saídas formatadas em *0.1* e *0.9*) se mostraram mais estáveis do que as redes criadas com *base de dados 1*

(saídas formatadas em 0 e 1), e, além disso, foi possível criar redes com menores valores de *MSE* e de erro de classificação.

5.5.3.1 Metodologia de treinamento das redes MLP

O treinamento das redes *Multi-Layer Perceptron* (MLP) pode ser realizado com vários algoritmos além do algoritmo *Backpropagation*. Neste caso, foi utilizado o algoritmo *Levenberg-Marquardt*. Foram realizadas dez execuções com diferentes inicializações de peso para cada quantidade de nodos na camada intermediária. As redes foram criadas com o comando *newff*²². Os seguintes parâmetros foram utilizados para criação das redes:

- PR – Matriz de valores máximos e mínimos da matriz de entrada. O valor 1 corresponde aos valores máximos e o valor 0 aos valores mínimos. Esta matriz define a quantidade de nodos da camada de entrada, que para este caso foi 8.
- Si – Quantidade de nodos na camada intermediária e na camada de saída. Foram utilizados os valores 8, 10, 12, 14, 16, 18, 20, 22, 24 e 26 para camada intermediária e o valor 9 para camada de saída.
- TFi – Função de ativação de cada camada. Para camada intermediária e para camada de saída foi utilizada a função *logsig* (sigmóide logística).
- BTF – Função de treinamento *trainlm* que é o *Levenberg-Marquardt*.
- BLF – Função de aprendizado para os pesos e bias. Foi utilizado o valor padrão *learnqdm*.
- PF – Função de desempenho. Foi utilizado o valor padrão *mse*.

As redes foram treinadas utilizando o comando *train*²³ com os seguintes parâmetros:

- NET – Rede criada anteriormente.
- P – Matriz com os dados de entrada do treinamento.
- T – Matriz com os dados de saída do treinamento (target).
- VV – Estrutura de validação. Composta da matriz de entrada de validação e da matriz de saída de validação.
- epochs – Número máximo de épocas. O valor utilizado foi de 2500.
- goal – Objetivo. O valor utilizado foi 0.
- max_fail – Número máximo de falhas²⁴ no conjunto de validação. Foi utilizado o valor padrão do *Matlab* de 5.

²² Cria uma rede *feed-forward backpropagation*.

²³ Os pesos e bias da rede são atualizados depois de todo o conjunto de treinamento ser aplicado à rede.

²⁴ O número máximo de falhas no conjunto de validação é utilizado pelo *Matlab* para implementar o método de *early stopping* que aumenta a capacidade de generalização da rede. O erro no conjunto de validação é monitorado durante o treinamento. O erro no conjunto de validação geralmente diminui durante a fase inicial do treinamento. Entretanto, quando a rede começa a memorizar os dados de treinamento (*overfitting*), o

- `men_reduc` – Fator de trade off entre o uso de memória e a velocidade de treinamento. O valor utilizado foi 1.
- `min_grad` – Gradiente mínimo de desempenho. O valor utilizado foi 10^{-10} .
- `mu` – Valor inicial de μ . O valor utilizado foi 0.001.
- `mu_dec` – Fator de decréscimo de μ . O valor utilizado foi 0.1
- `mu_inc` – Fator de acréscimo de μ . O valor utilizado foi 10.
- `mu_max` – Valor máximo de μ . O valor utilizado foi 10^{10} .

Para cada base de dados foram treinadas dez redes com diferentes inicializações de peso para 8, 10, 12, 14, 16, 18, 20, 22, 24 e 26 nodos na camada intermediária, com o objetivo de selecionar a melhor arquitetura para realização dos experimentos finais. Como o objetivo desta etapa era investigar diferentes arquiteturas para escolher a melhor, o método cross-validation não foi utilizado nesta etapa. Em todos os experimentos desta fase, a partição 1 foi selecionado para formar o conjunto de teste, as partições 2, 3 e 4 foram selecionadas formar o conjunto de validação e as partições 5, 6, 7, 8, 9 e 10 para formar o conjunto de treinamento. A **Figura 5-5** ilustra o diagrama de uma das redes *MLP* criadas no *Matlab*.

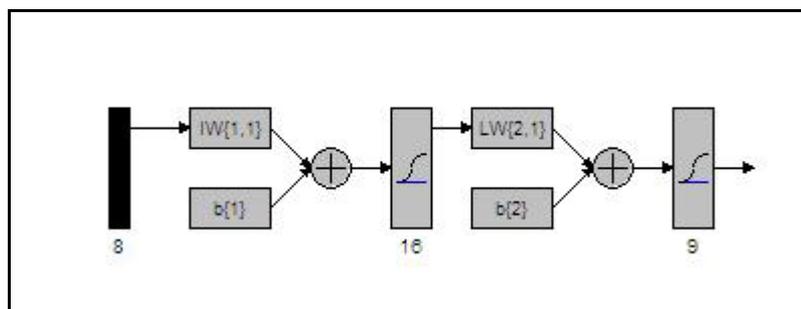


Figura 5-5: Diagrama de uma rede *MLP*.

5.5.3.2 Resultados obtidos com MLP

A **Tabela 5-8** apresenta o erro médio quadrático do treinamento, da validação e do teste e o percentual de erro de classificação no conjunto de teste para as diferentes quantidades de nodos na camada escondida na *base de dados 1*. A **Tabela 5-9** apresenta os mesmos resultados para a *base de dados 2*.

erro no conjunto de validação começa a aumentar. Quando o erro no conjunto de validação aumenta em um número específico de iterações, o treinamento é interrompido e os pesos e bias são retornados para os valores do momento do erro mínimo de validação.

Tabela 5-8: Resultados obtidos com *MLP* para seleção da quantidade de nodos na camada escondida na *base de dados 1*.

Quantidade de nodos	Épocas		MSE treinamento		MSE validação		MSE teste		%Erro classificação teste	
	Média	desvio	Média	desvio	Média	desvio	Média	desvio	Média	desvio
8	118	66.7683	0.1180	0.0716	11.7963	7.1551	11.6577	7.2474	60.8796	29.9703
10	94.4	57.5233	0.0930	0.0737	9.3019	7.3748	9.1081	7.7017	55.6481	31.8872
12	112.3	47.7518	0.0845	0.0599	8.4509	5.9898	8.3761	5.8968	56.7130	31.3786
14	141.5	57.1027	0.0950	0.0664	9.4956	6.6350	9.3711	6.5294	59.2593	27.4327
16	81.5	60.4249	0.0860	0.0710	8.6028	7.1013	8.5432	7.1649	52.0833	36.3046
18	105.5	47.3480	0.0897	0.0557	8.9691	5.5710	8.7823	5.5209	60.7870	28.8808
20	126.8	88.9904	0.1153	0.0876	11.5308	8.7596	11.4790	8.8234	61.9444	32.6780
22	165.5	87.5281	0.0917	0.0650	9.1694	6.5032	9.1569	6.4936	57.3148	27.0795
24	165.3	158.7411	0.1138	0.0896	11.3804	8.9582	11.2873	8.8185	65.9465	29.1787
26	219.4	130.0000	0.0951	0.0662	9.5083	6.6217	9.4014	6.6900	58.2870	28.6193

Tabela 5-9: Resultados obtidos com *MLP* para seleção da quantidade de nodos na camada escondida *base de dados 2*.

Quantidade nodos	Épocas		MSE treinamento		MSE validação		MSE teste		%Erro classificação teste	
	Média	desvio	Média	desvio	Média	desvio	Média	desvio	Média	desvio
8	100.7	35.2264	4.3624	3.0537	4.5318	2.9741	4.3808	3.0387	43.2870	27.6008
10	99.3	39.2939	3.5604	2.0682	3.7502	2.0011	3.6138	2.0611	36.8519	26.9591
12	121.6	46.4284	3.2036	2.3006	3.4160	2.2652	3.2848	2.2499	29.4444	23.6300
14	131.6	47.6100	2.8448	2.6612	3.0839	2.6110	2.9248	2.6439	24.9537	24.1762
16	127.1	51.2108	2.7585	3.8108	3.0029	3.7167	2.8268	3.8000	19.8611	25.4685
18	121.2	34.7141	4.7333	3.5646	4.9552	3.5164	4.7239	3.5076	46.4352	33.8537
20	109.4	46.1693	5.6596	4.3165	5.7727	4.1741	5.7218	4.2560	52.3148	37.0267
22	137.9	27.9064	4.7543	5.1284	4.9145	4.9965	4.7895	5.1209	39.6759	37.6664
24	112.3	55.2933	4.7132	3.8330	4.8695	3.7054	4.7131	3.8394	45.4167	36.3860
26	130.7	130.0000	6.1194	4.6289	6.3156	4.5695	6.2191	4.6283	55.1852	39.7393

O menor valor de MSE para o conjunto de validação utilizando a *base de dados 1* foi obtido com 12 nodos na camada de escondida e foi igual a **8.4509%**. Nos experimento que utilizam a *base de dados 2*, o menor valor de *MSE* para o conjunto de validação foi obtido com 16 nodos na camada escondida e foi igual a **3.0029%**. Na análise feita com a matriz de confusão dessas redes, ficou claro que as redes criadas com a segunda base de dados conseguiram resultados mais equilibrados em todas as classes. Para realização dos experimentos finais foi escolhida a rede *MLP* com 16 nodos na camada intermediária e a *base de dados 2*.

5.5.4 Experimentos com TDNN

Os experimentos TDNN foram realizados utilizando redes MLP, RBF e PNN. A informação temporal está inserida nos dados de entrada independente da arquitetura de rede adotada. As redes TDNN criadas com a arquitetura MLP possuem 16 nodos na camada de entrada, nove nodos na camada de saída, pois foi utilizada a codificação de saída 1-of-m, onde m é o número de classes. Os nodos da camada de saída e da camada intermediária possuem função de ativação *sigmóide logística*. Foram realizados experimentos aleatórios com 8, 10, 12, 14, 16, 18, 20, 22, 24 e 26 nodos na camada intermediária. A rede contém todas as possíveis conexões *feedforward* entre as camadas adjacentes e não possui conexões que ligam nodos de camadas não-adjacentes.

As redes TDNN criadas com a arquitetura RBF e PNN possuem 16 nodos na camada de entrada, 1863 nodos na camada intermediária e 9 nodos na camada de saída.

5.5.4.1 Metodologia de treinamento das redes TDNN com MLP

Os experimentos TDNN com MLP foram realizados utilizando os mesmos comandos do Matlab que foram utilizados nos experimentos MLP. Foram utilizadas as base de dados 3 e 4.

Para cada base de dados foram treinadas dez redes com diferentes inicializações de peso para 8, 10, 12, 14, 16, 18, 20, 22, 24 e 26 nodos na camada intermediária, com o objetivo de selecionar a melhor arquitetura para realização dos experimentos finais. A Figura 5-6 ilustra o diagrama de uma das redes TDNN com MLP criadas no Matlab.

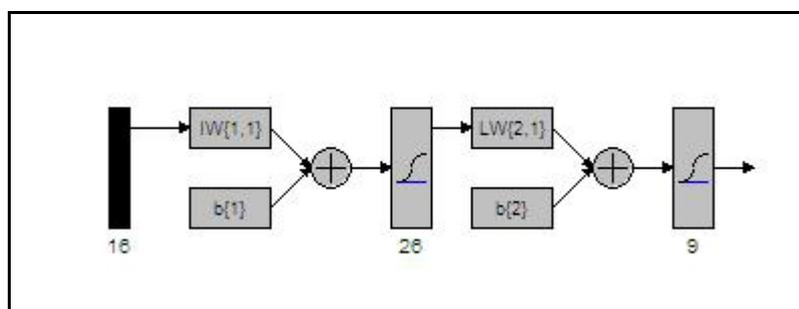


Figura 5-6: Diagrama de uma rede TDNN com MLP.

5.5.4.2 Resultados obtidos para TDNN com MLP

A Tabela 5-10 apresenta o erro médio quadrático do treinamento, da validação e do teste e o percentual de erro de classificação no conjunto de teste para as diferentes quantidades de nodos na camada escondida com a base de dados 3 (seqüencial). A

Tabela 5-11 apresenta os mesmos resultados para a base de dados 4 (aleatória).

Tabela 5-10: Resultados obtidos para TDNN com MLP e base de dados 3 para seleção da quantidade de nodos na camada escondida.

Quantidade nodos	Épocas		MSE treinamento		MSE validação		MSE teste		%Erro classificação teste	
	Média	desvio	Média	desvio	Média	desvio	Média	desvio	Média	desvio
8	103.4	271.5651	9.7635	5.1606	9.4526	4.4944	9.5318	4.8581	87.7778	3.5136
10	17.1	5.8775	10.1330	5.4924	10.5855	4.8366	10.9467	5.0321	89.9034	3.5606
12	17.8	6.8928	10.5691	4.8442	11.2108	4.5721	11.5715	5.1441	88.8889	5.2378
14	15.8	6.3210	10.9648	7.0131	11.1964	5.8437	11.2669	5.8279	87.8261	3.3609
16	17.2	6.3910	8.9607	5.9599	9.8863	5.2157	10.4915	5.0080	88.8889	0.0000
18	12.8	3.9384	12.0616	6.0736	12.0497	5.0440	12.2612	5.0384	87.7778	3.5136
20	19.4	10.5430	8.3955	4.8448	9.0397	4.0703	10.0643	5.2887	88.1643	3.3950
22	19.8	8.3772	10.2725	6.2070	11.9691	6.2019	12.8983	6.8211	88.8889	0.0000
24	15.3	5.8888	9.0326	6.7467	10.2684	5.2519	11.3053	5.6221	88.3575	4.1398
26	14.1	130.0000	13.2024	6.3962	12.3821	5.8417	12.9513	6.2815	88.8406	0.1528

Tabela 5-11: Resultados obtidos para TDNN com MLP e base de dados 4 para seleção da quantidade de nodos na camada escondida.

Quantidade nodos	Épocas		MSE treinamento		MSE validação		MSE teste		%Erro classificação teste	
	Média	desvio	Média	desvio	Média	desvio	Média	desvio	Média	desvio
8	92.9	37.7402	5.3800	4.1936	5.5393	4.0479	5.5942	4.0000	49.6618	33.6448
10	83.6	39.9922	5.2318	3.3721	5.3684	3.3007	5.3740	3.2899	52.8019	32.1401
12	85.8	24.5665	4.3137	5.0071	4.5473	4.9047	4.4646	4.8184	30.2415	32.2353
14	123.2	32.8322	3.6896	3.4548	3.8831	3.3620	3.9175	3.4146	34.2029	33.7387
16	119.9	47.7760	4.9379	4.5632	5.1540	4.4965	5.1664	4.5829	41.8841	35.5160
18	116	47.7563	4.3371	4.1375	4.5362	4.0655	4.6076	4.1944	38.7923	31.4715
20	124	47.2229	3.3306	3.6979	3.5435	3.6519	3.5343	3.6663	29.5169	27.6424
22	146.5	75.3529	4.6126	5.8247	4.8644	5.7755	4.7903	5.8056	36.4251	39.8336
24	141.3	46.1328	2.1226	1.8282	2.3654	1.7414	2.3697	1.7739	22.0773	23.3966
26	171.3	130.0000	1.2078	1.4338	1.5186	1.3778	1.4893	1.4696	11.0628	19.5205

Pelos resultados apresentados observa-se que os valores médios do *MSE* nos conjuntos de validação utilizando a *base de dados 4* (aleatória) são melhores do que os resultados obtidos com a *base de dados 3* (seqüencial). A arquitetura com 26 nodos na camada escondida foi selecionada para os experimentos finais, pois apresentou o menor valor médio de *MSE* no conjunto de validação.

5.5.4.3 Metodologia de treinamento das redes TDNN com RBF

Os experimentos *TDNN com RBF* foram realizados utilizando o comando *newrbe* do *Matlab*, pois este comando criou redes *RBF* mais rapidamente do que o comando *newrb* e como valores menores de erro médio quadrático no conjunto de treinamento. Foi utilizada a *base de dados 5*.

Os mesmos valores de *SPREAD* analisados nos experimentos *RBF* foram analisados para *TDNN com RBF*. Foram criadas cinquenta redes para cada *SPREAD* da seguinte maneira:

1. Repetir para *SPREAD* (0.05, 0.08, 0.1, 0.5, 1).
 - a. Repetir *k* (variando de 1 a 10) vezes.
 - i. Conjunto de teste = Conjunto *k*.
 - ii. Conjunto de treinamento = Todos os conjuntos menos o conjunto *k*.
 - iii. Criar a rede.
 - iv. Calcular *MSE* e erro de classificação para o conjunto de treinamento.
 - v. Calcular *MSE* e erro de classificação para o conjunto de teste.

5.5.4.4 Resultados obtidos para TDNN com RBF

A **Tabela 5-12** apresenta a quantidade de nodos da camada oculta, o erro médio quadrático do treinamento e do teste e o percentual de erro de classificação no conjunto de teste para cada valor de *SPREAD* testado.

Tabela 5-12: Resultados obtidos para TDNN com RBF para seleção de SPREAD.

SPREAD	Quantidade nodos		MSE treinamento		MSE teste		%Erro classificação teste	
	Média	desvio	Média	Desvio	Média	Desvio	Média	Desvio
0.05	1863	0	0.0122	0.0029	1212.2239	1566.1677	7.4396	3.9654
0.08	1863	0	0.0267	0.0069	716.5719	841.2775	2.5121	1.1341
0.1	1863	0	0.0398	0.0035	768.8459	859.1595	2.6087	1.0730
0.5	1863	0	0.8333	0.0923	17046.8132	15960.8119	2.3671	1.1489
1	1863	0	2.2752	0.1735	6007.4488	9368.7373	9.7585	3.9365

Como nas redes *RBF* anteriores, utilizamos o valor do *MSE* no treinamento para selecionar a melhor arquitetura. Para os experimentos *TDNN com RBF*, observamos que o menor valor de *MSE* foi obtido com *SPREAD 0.05*, logo este foi o valor selecionado para realização dos experimentos finais.

5.5.4.5 Metodologia de treinamento das redes TDNN com PNN

Os experimentos *TDNN com PNN* foram realizados utilizando o comando *newpnn* do *Matlab*. Foi utilizada a base de dados 5. Os mesmos valores de *SPREAD* analisados nos experimentos *PNN* foram analisados para *TDNN com PNN*. Para escolher o parâmetro *SPREAD*, foram criadas dez redes *PNN* para cada *SPREAD* da seguinte maneira:

1. Repetir para *SPREAD* (*0.0001*, *0.0005*, *0.005*, *0.5*, *0.1*).
 - a. Repetir *k* (variando de 1 a 10) vezes.
 - i. Conjunto de teste = Conjunto *k*.
 - ii. Conjunto de treinamento = Todos os conjuntos menos o conjunto *k*.
 - iii. Criar a rede.
 - iv. Calcular *MSE* e erro de classificação para o conjunto de treinamento.
 - v. Calcular *MSE* e erro de classificação para o conjunto de teste.

5.5.4.6 Resultados obtidos para TDNN com PNN

A **Tabela 5-13** apresenta a quantidade de nodos da camada oculta, o erro médio quadrático do treinamento e do teste e o percentual de erro de classificação no conjunto de teste para cada valor de *SPREAD* testado.

Tabela 5-13: Resultados obtidos para *TDNN com PNN* para seleção de *SPREAD*.

SPREAD	Quantidade nodos		MSE treinamento		MSE teste		%Erro classificação teste	
	Média	desvio	Média	Desvio	Média	Desvio	Média	Desvio
0.0001	1863	0	0.0107	0.0038	11.2077	0.5730	50.4348	2.5785
0.0005	1863	0	0.0107	0.0038	0.9554	0.2793	4.2995	1.2567
0.005	1863	0	0.1074	0.0203	0.3865	0.2037	1.7391	0.9166
0.05	1863	0	6.3386	0.2860	6.6774	0.7934	30.0483	3.5704
0.1	1863	0	8.2829	0.2604	9.0607	0.8724	40.7729	3.9260

O valor do *MSE* no treinamento também foi utilizado para selecionar a melhor arquitetura. Os valores de *SPREAD* de *0.0001* e *0.0005* apresentaram os menores valores médios de *MSE* para o treinamento. Como as médias de *MSE* para estes dois valores de *SPREAD* foram iguais, selecionamos o *SPREAD 0.0005* para realização dos experimentos finais, pois ele permite um maior poder de generalização.

5.6 Sumário do capítulo

O objetivo deste capítulo foi descrever a base de dados utilizada no estudo de caso, o pré-processamento adotado, o método adotado para formação dos conjuntos de treinamento, validação e teste e investigar os melhores parâmetros para criação de classificadores com as redes *RBF*, *PNN*, *MLP* e *TDNN*. Uma dificuldade encontrada nesta etapa foi a grande quantidade de possibilidades de formatação dos dados, seleção de parâmetros e definição de arquiteturas.

Para as redes *RBF* foi utilizada a *base de dados 1*. Como existem dois treinamentos diferentes para criação de redes *RBF* no *Matlab*, foi necessário investigá-los. Para cada um dos treinamentos foram observadas vantagens e desvantagens. A vantagem de utilizar *newbr* para criar redes *RBF* é que estas redes são criadas com um menor número de nodos na camada oculta do que as redes criadas com *newrbe*. A desvantagem das redes criadas com *newrb* é que o tempo gasto no treinamento é muito superior ao tempo gasto com *newrbe*. Além de testar dois treinamentos diferentes para redes *RBF*, também foi necessário testar parâmetros diferentes de *GOAL* e *SPREAD*. Dentre as cinquenta redes criadas com o comando *newrb*, o valor *200* foi selecionado para o parâmetro *GOAL* e o valor *0.5* foi selecionado para o parâmetro *SPREAD*, pois eles apresentaram menor erro médio quadrático no conjunto de treinamento. Para as redes *RBF* criadas com *newrbe*, o valor *0.05* foi selecionado como para o parâmetro *SPREAD*.

Para as redes *PNN* foram observadas duas vantagens principais: o tempo de treinamento das redes foi muito pequeno, e só foi necessário testar um parâmetro. Dentre as cinquenta redes *PNN* criadas o valor *0.0005* foi selecionado para o parâmetro *SPREAD*, pois ele apresentou menor erro médio quadrático no conjunto de treinamento.

O algoritmo *levenberg-marquardt* foi selecionado para o treinamento das redes *MLP* pois este algoritmo é uma variação do algoritmo *backpropagation* padrão que utiliza a derivada segunda como forma de otimização do mesmo, tornando assim o treinamento das redes mais rápido [Braga et al., 2000]. Durante a realização dos experimentos *MLP*, percebeu-se que as redes não estavam obtendo bons resultados de classificação com a *base de dados 1*. Havia uma forte indicação de que os nodos da camada de saída estavam saturando, pois os erros de classificação eram muito altos. Adotamos então a sugestão [Prechelt, 1994] de formatar os valores de saída entre *0.1* e *0.9* para tentar resolver o problema e realizamos também experimentos com a *base de dados 2*. Esta nova formatação de saída permitiu às redes *MLP* uma melhora significativa no erro médio quadrático no conjunto de validação e no desempenho da classificação. Nesta etapa foram testadas *100* redes com a *base de dados 1* e *100* redes com a *base de dados 2*. O tempo de treinamento das redes *MLP* foi muito superior ao tempo de treinamento das redes *RBF* e *PNN*. Para realização dos experimentos finais com *MLP* foi selecionada a *base de dados 2* e *16* nodos na camada oculta.

Para os experimentos *TDNN* foram utilizadas as *bases de dados 3, 4 e 5*. A idéia foi utilizar a formatação dos dados de entrada para incluir o processamento temporal, então cada padrão de entrada passou a representar a leitura do arranjo de sensores para um determinado odor no tempo *t* e no tempo *t+1*. Como a informação temporal está nos dados e não na arquitetura das redes, e como as redes *RBF* são equivalentes às redes *MLP*, foram realizados experimentos *TDNN* com arquiteturas *RBF, PNN e MLP*. Surgiu uma nova questão nesta etapa que foi relativa a organização dos conjuntos, se ela deveria ser seqüencial ou aleatória. As duas organizações foram investigadas nos experimentos *TDNN com MLP*.

Nos experimentos *TDNN com MLP*, a formatação de saída adotada foi à formatação com *0.1* e *0.9*, pois esta formatação foi melhor nos experimentos *MLP*. Os resultados obtidos com a base organizada aleatoriamente (*base de dados 4*) foram superiores aos resultados da base organizada seqüencialmente seqüencial (*base de dados 3*). Foi selecionada então para os experimentos finais de *TDNN com MLP*, a arquitetura com *26* nodos na camada oculta e a *base de dados 4*.

Nos experimentos *TDNN com RBF*, foram criadas redes apenas com o comando *neurbe*. A base de dados utilizada foi a *base de dados 5*. Foi selecionada então para os experimentos finais de *TDNN com RBF*, o *SPREAD 0.05*.

Nos experimentos *TDNN com PNN* foi utilizada a mesma base de dados dos experimentos *TDNN com RBF*. Foi selecionada então para os experimentos finais de *TDNN com PNN*, o *SPREAD 0.0005*.

No próximo capítulo serão apresentados os resultados dos experimentos finais e estes resultados serão comparados utilizando o teste de hipóteses.

EXPERIMENTOS FINAIS

6 Introdução

Na comparação entre classificadores é comum ocorrer o problema de escolher qual dos classificadores é o melhor. É claro que alguns classificadores podem ser melhores que outros porque tem menor complexidade computacional; outros porque possuem algum conhecimento a priori dos dados. Entretanto em problemas de classificação muitas vezes o que se deseja saber é se eles são equivalentes nos resultados gerados ou se existe alguma razão a favor de um algoritmo sobre os outros. É útil comparar o desempenho para definir se as diferenças observadas são devido a uma variação casual ou se são verdadeiramente significativas.

Neste capítulo vamos exibir os resultados obtidos nos experimentos realizados com as arquiteturas escolhidas no capítulo 5 e realizar comparações estatísticas entre os resultados obtidos para os erros de classificação. O método selecionado para realizar a comparação estatística dos classificadores é o *teste de hipóteses* que foi descrito no capítulo 4. Os aspectos descritos na seção 5.4 serão observados também nos experimentos finais.

6.1 Resultado dos experimentos finais

Os experimentos finais sem processamento temporal com as redes *RBF com newrb*, *RBF com newrbe* e *PNN* utilizaram a *base de dados 1*. Como explicado na seção [5.4] os experimentos com redes *RBF* podem produzir *MSE* superior a 100%, pois a função de ativação dos nodos da camada de saída é linear. As redes *RBF* e *PNN* criadas no Matlab são determinísticas, logo para os mesmos parâmetros e mesmos conjuntos de treinamento e teste as redes são iguais. Como o método adotado para criação dos conjuntos de treinamento e teste foi o método *10-fold cross-validation com estratificação*, foram geradas 10 redes para cada arquitetura. Em cada uma dessas redes, um conjunto foi utilizado para testar a rede e os $n-1$ conjuntos restantes foram utilizados para treinar a rede. A **Tabela 6-1** exibe a numeração de cada rede e configuração do conjunto de treinamento e teste. A **Tabela 6-2** exibe os resultados obtidos pelas redes *RBF* e *PNN*.

Tabela 6-1: Formação dos conjuntos de treinamento, validação e teste para o treinamento das redes *RBF com newrb*, *RBF com newrbe* e *PNN*.

Experimento	Rede	Partições utilizadas no treinamento	Partições utilizadas no teste
1	1	2, 3, 4, 5, 6, 7, 8, 9, 10	1
2	2	1, 3, 4, 5, 6, 7, 8, 9, 10	2
3	3	1, 2, 4, 5, 6, 7, 8, 9, 10	3
4	4	1, 2, 3, 5, 6, 7, 8, 9, 10	4
5	5	1, 2, 3, 4, 6, 7, 8, 9, 10	5
6	6	1, 2, 3, 4, 5, 7, 8, 9, 10	6
7	7	1, 2, 3, 4, 5, 6, 8, 9, 10	7
8	8	1, 2, 3, 4, 5, 6, 7, 9, 10	8
9	9	1, 2, 3, 4, 5, 6, 7, 8, 10	9
10	10	1, 2, 3, 4, 5, 6, 7, 8, 9	10

Tabela 6-2: Resultados obtidos para *RBF com newrb*, *RBF com newrbe* e *PNN*.

Modelo	Rede	Quantidade de nodos	MSE treinamento	MSE teste	%Erro classificação teste
<i>RBF com newrb</i> GOAL = 200 SPREAD = 0.5	1	175	1.1391	22097.4780	1.3889
	2	168	1.1403	22797.8210	2.7778
	3	168	1.1385	1.2754	2.3148
	4	172	1.1406	82341.4660	2.7778
	5	168	1.1428	141.8074	1.3889
	6	162	1.1393	9818.5201	3.2407
	7	165	1.1405	5134638.6000	2.7778
	8	169	1.1326	1.9034	3.7037
	9	174	1.1355	2.1331	3.2407
	10	165	1.1370	1098.1492	3.2407
	Média	168.6000	1.1386	527293.9154	2.6852
	Desvio	4.1150	0.0029	1619053.5852	0.7808
<i>RBF com newrbe</i> SPREAD = 0.05	1	1944	0.0351	228.7185	2.3148
	2	1944	0.0586	17.7258	1.3889
	3	1944	0.0589	0.2130	0.4630
	4	1944	0.0460	198.0080	0.9259
	5	1944	0.0412	30.8125	1.3889
	6	1944	0.0968	268.7139	2.3148
	7	1944	0.0597	335.0389	1.3889
	8	1944	0.0825	0.3077	0.4630
	9	1944	0.0932	0.4263	0.9259
	10	1944	0.0489	193.3719	2.3148
	Média	1944.0000	0.0621	127.3336	1.3889
	Desvio	0.0000	0.0216	130.1690	0.7238

Modelo	Rede	Quantidade de nodos	MSE treinamento	MSE teste	%Erro classificação teste
PNN SPREAD = 0.0005	1	1944	0.0229	0.3086	1.3889
	2	1944	0.0229	0.3086	1.3889
	3	1944	0.0229	0.0000	0.0000
	4	1944	0.0229	0.1029	0.4630
	5	1944	0.0229	0.2058	0.9259
	6	1944	0.0229	0.4115	1.8519
	7	1944	0.0229	0.3086	1.3889
	8	1944	0.0114	0.2058	0.9259
	9	1944	0.0229	0.4115	1.8519
	10	1944	0.0114	0.3086	1.3889
	Média	1944.0000	0.0206	0.2572	1.1574
	Desvio	0.0000	0.0048	0.1306	0.5876

Os experimentos finais das redes *MLP* sem processamento temporal foram realizados com a arquitetura com 16 nodos na camada escondida e com a base de dados 2. Como as redes *MLP* não são determinísticas é necessário que se executem vários experimentos com inicializações de peso diferentes e com a mesma configuração de rede e a mesma base de dados para obtermos uma média do desempenho da rede esta configuração. Dessa forma foram criadas 100 redes *MLP* nesta etapa. Para cada conjunto de treinamento, validação e teste, formados a partir das partições criadas pelo método *10-fold cross-validation com estratificação*, foram criadas 10 redes. Com relação à formação dos conjuntos de treinamento e teste, os experimentos com redes *MLP* possuem a seguinte diferença quando comparados com os experimentos realizados com as outras redes: como os experimentos *MLP* utilizam o conjunto de validação para verificação do critério de parada “*Early Stop*”, foi necessário dividir o conjunto de treinamento em conjunto de treinamento e conjunto de validação. Os experimentos com as outras redes não utilizam o critério de parada “*Early Stop*”, logo eles não dividem o conjunto de treinamento. A **Tabela 6-3** descreve a formação dos conjuntos de treinamento, validação e teste para as redes *MLP*, bem como identifica as redes que utilizaram os respectivos conjuntos. As médias obtidas nas 10 redes criadas para cada conjuntos de treinamento, validação e teste, são consideradas como o resultado esperado para redes *MLP*. Os resultados estão descritos na **Tabela 6-4**.

Tabela 6-3: Formação dos conjuntos de treinamento, validação e teste para o treinamento de redes *MLP*.

Experimento	Rede	Partições utilizadas no treinamento	Partições utilizadas na validação	Partições utilizadas no teste
1	De 1 a 10	5, 6, 7, 8, 9, 10	2, 3, 4	1
2	De 11 a 20	1, 6, 7, 8, 9, 10	3, 4, 5	2
3	De 21 a 30	1, 2, 7, 8, 9, 10	4, 5, 6,	3
4	De 31 a 40	1, 2, 3, 8, 9, 10	5, 6, 7	4
5	De 41 a 50	1, 2, 3, 4, 9, 10	6, 7, 8	5
6	De 51 a 60	1, 2, 3, 4, 5, 10	7, 8, 9	6
7	De 61 a 70	1, 2, 3, 4, 5, 6	8, 9, 10	7
8	De 71 a 80	2, 3, 4, 5, 6, 7	9, 10, 1	8
9	De 81 a 90	3, 4, 5, 6, 7, 8	1, 2, 10	9
10	De 91 a 100	4, 5, 6, 7, 8, 9	1, 2, 3	10

Tabela 6-4: Resultados finais para as redes *MLP* com 16 nodos na camada escondida.

Experimento	Épocas		MSE treino		MSE validação		MSE teste		%Erro classificação teste	
	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
1	98.3	57.21	5.9143	4.7810	6.0697	4.6778	5.9655	4.7618	47.8241	33.3691
2	157.6	34.72	1.6238	1.4453	1.9139	1.3526	1.8459	1.3779	13.9352	19.8018
3	134.9	55.64	3.9894	4.1174	4.2561	3.9846	4.0391	4.0650	33.3796	31.1005
4	174.7	79.10	3.1423	3.4725	3.3199	3.4191	3.6481	3.3029	27.5926	26.9087
5	137.1	56.63	4.1044	5.4135	4.3117	5.3323	4.3735	5.2923	30.9259	27.5054
6	138.6	43.47	4.9831	4.4034	5.1145	4.3130	5.0600	4.3661	43.4722	34.1990
7	161.2	70.84	2.9451	1.9518	3.2074	1.8524	3.1225	1.9251	31.0185	25.8708
8	137.1	67.38	5.1152	4.2876	5.2602	4.2170	5.3267	4.1340	47.1759	33.8484
9	146.6	42.62	2.1985	1.8732	2.3683	1.7761	2.4027	1.8699	20.0000	21.5551
10	143.7	130.00	3.7816	4.3493	3.8839	4.2440	4.0513	4.2205	30.1852	32.5019
Média	142.9800		3.7798		3.9705		3.9835		32.5509	
Desvio	20.3181		1.3412		1.3075		1.2898		11.0873	

Os experimentos finais com processamento temporal com as redes TDNN com RBF, TDNN com PNN utilizaram a base de dados 5. É importante destacar que estas redes também são determinísticas, logo para os mesmos parâmetros e mesmos conjuntos de treinamento e teste as redes são iguais. Foram criadas então 10 redes para cada arquitetura. Foi utilizada a mesma configuração descrita na **Tabela 6-1** nesses experimentos. A **Tabela 6-5** exibe os resultados obtidos por estas redes.

Tabela 6-5: Resultados obtidos para *TDNN com RBF* e, *TDNN com PNN*.

Modelo	Rede	Quantidade de nodos	MSE treinamento	MSE teste	%Erro classificação teste
TDNN com RBF SPREAD = 0.05	1	1863	0.0139	227.2762	2.8986
	2	1863	0.0148	3838.2141	9.1787
	3	1863	0.0048	5.8893	5.7971
	4	1863	0.0105	691.9043	4.8309
	5	1863	0.0142	244.1416	2.8986
	6	1863	0.0119	296.3100	8.2126
	7	1863	0.0133	2600.1505	11.1111
	8	1863	0.0117	3799.3233	14.9758
	9	1863	0.0138	406.5590	4.3478
	10	1863	0.0132	12.4703	10.1449
		Média	1863.0000	0.0122	1212.2239
	Desvio	0.0000	0.0029	1566.1677	3.9654
TDNN com PNN SPREAD = 0.0005	1	1863	0.0119	0.7515	3.3816
	2	1863	0.0119	1.0735	4.8309
	3	1863	0.0000	1.1809	5.3140
	4	1863	0.0119	0.8588	3.8647
	5	1863	0.0119	0.5368	2.4155
	6	1863	0.0119	1.5030	6.7633
	7	1863	0.0119	0.9662	4.3478
	8	1863	0.0119	0.9662	4.3478
	9	1863	0.0119	1.0735	4.8309
	10	1863	0.0119	0.6441	2.8986
		Média	1863.0000	0.0107	0.9554
	Desvio	0.0000	0.0038	0.2793	1.2567

Os experimentos finais com processamento temporal e com a rede *TDNN com MLP*, a arquitetura escolhida foi à arquitetura com 26 nodos na camada escondida e a *base de dados 4* foi utilizada treinar estas redes. Da mesma forma que as redes MLP, são necessários vários experimentos com a mesma configuração da rede e os mesmos dados, mas com inicializações de pesos aleatória para se obter um desempenho médio dessas redes. A formação do conjunto de treinamento, validação e teste é igual a descrita na **Tabela 6-3** para os experimentos *MLP*. Foram criadas 100 redes *TDNN com MLP*, e as médias obtidas nas 10 redes criadas para cada conjuntos de treinamento, validação e teste, são considerados como os resultados esperados para as redes. Estes resultados e estão descritos na **Tabela 6-6**.

Tabela 6-6: Resultados finais para as redes *TDNN com MLP* com 26 nodos na camada escondida.

Exp.	Épocas		MSE treino		MSE validação		MSE teste		%Erro classificação teste	
	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
1	136.9	70.2716	4.1998	5.2601	4.4230	5.1239	4.4105	5.1095	32.5604	39.6333
2	126.7	67.7611	7.1597	8.2675	7.3252	8.0640	7.3561	8.2318	43.1884	39.5855
3	142.8	37.9614	3.5820	3.6837	3.9205	3.6380	3.7127	3.6492	34.5411	32.7912
4	134.3	50.9118	5.4667	7.4396	5.7528	7.2675	5.8805	7.2519	33.9130	39.4409
5	156.6	44.4677	2.8884	3.2313	3.1384	3.1074	3.4216	3.1508	29.5169	31.4067
6	152.8	55.4873	4.3753	6.0397	4.6172	5.9436	4.5728	5.9323	32.1739	35.8127
7	157.7	43.0789	1.6714	1.3176	2.0282	1.3172	2.1067	1.3271	17.8261	17.7725
8	163.9	61.1000	3.4114	3.9355	3.6578	3.8759	3.7590	3.9018	32.8986	35.7157
9	174.2	29.2681	1.5134	1.1952	1.7924	1.2052	1.7331	1.1107	13.9130	13.8258
10	174.7	130.0000	3.8614	4.9415	4.0099	4.8182	4.1988	4.6004	32.8986	39.8123
Média	152.0600		3.8130		4.0665		4.1152		30.3430	
Desvio	16.5664		1.6782		1.6422		1.6466		8.4591	

6.2 Comparação dos resultados finais

O objetivo dessa seção é comparar o desempenho obtido pelos classificadores nos experimentos finais. A comparação realizada entre os classificadores é uma comparação idônea, pois os classificadores foram criados e testados com os mesmos dados originais. Esses dados são formados por 240 amostras de odor de nove classes de aguarrás, totalizando 2160 padrões. Quando nos referimos nesta dissertação, à base de dados diferentes estamos nos referindo às diferentes formatações dos dados de saída ou a inclusão de informações temporais ou mesmo a organização dos dados dentro das partições, nunca de dados originais diferentes.

O parâmetro de comparação utilizado é o erro médio de classificação obtido por cada um dos classificadores. O erro de classificação é dado pela razão da quantidade de padrões classificados de forma incorreta pela quantidade total de padrões testados. Estes resultados serão comparados estatisticamente pelos *testes de hipóteses* que foram descritos na seção [4.3].

O teste de hipótese que será realizado entre os classificadores é o teste *t-Student pareado*. A **Tabela 6-7** apresenta os resultados obtidos com estes testes a um nível de significância de 5%. As médias de erros de

classificação foram comparadas, e a tabela exibe os classificadores, o tipo do teste de hipótese utilizado, o *p-value*²⁵ gerado, o intervalo de confiança e a decisão obtida em cada comparação.

Tabela 6-7: Comparação estatística entre os classificadores.

H ₀ : Classificador 1 = Classificador 2 H ₁ : Classificador 1 < Classificador 2 Significância do teste: 5% Tipo do teste: t-Student pareado					
Classificador 1	Classificador 2	Decisão	p-value	Intervalo de confiança	
<i>PNN</i>	<i>RBF com newrb</i>	Rejeita H ₀	0.000052	-inf	-0.99
<i>PNN</i>	<i>RBF com newrbe</i>	Aceita H ₀	0.221290	-inf	0.28
<i>PNN</i>	<i>MLP</i>	Rejeita H ₀	0.000000	-inf	-25.31
<i>PNN</i>	<i>TDNN com PNN</i>	Rejeita H ₀	0.000001	-inf	-2.38
<i>PNN</i>	<i>TDNN com RBF</i>	Rejeita H ₀	0.000051	-inf	-4.08
<i>PNN</i>	<i>TDNN com MLP</i>	Rejeita H ₀	0.000000	-inf	-24.54
<i>RBF com newrb</i>	<i>RBF com newrbe</i>	Aceita H ₀	0.999410	-inf	1.88
<i>RBF com newrb</i>	<i>MLP</i>	Rejeita H ₀	0.000000	-inf	-23.77
<i>RBF com newrb</i>	<i>TDNN com PNN</i>	Rejeita H ₀	0.001427	-inf	-0.80
<i>RBF com newrb</i>	<i>TDNN com RBF</i>	Rejeita H ₀	0.000783	-inf	-2.54
<i>RBF com newrb</i>	<i>TDNN com MLP</i>	Rejeita H ₀	0.000000	-inf	-23.00
<i>RBF com newrbe</i>	<i>MLP</i>	Rejeita H ₀	0.000000	-inf	-25.07
<i>RBF com newrbe</i>	<i>TDNN com PNN</i>	Rejeita H ₀	0.000003	-inf	-2.12
<i>RBF com newrbe</i>	<i>TDNN com RBF</i>	Rejeita H ₀	0.000081	-inf	-3.84
<i>RBF com newrbe</i>	<i>TDNN com MLP</i>	Rejeita H ₀	0.000000	-inf	-24.30
<i>MLP</i>	<i>TDNN com PNN</i>	Aceita H ₀	1.000000	-inf	34.37
<i>MLP</i>	<i>TDNN com RBF</i>	Aceita H ₀	1.000000	-inf	31.57
<i>MLP</i>	<i>TDNN com MLP</i>	Aceita H ₀	0.688660	-inf	9.86
<i>TDNN com PNN</i>	<i>TDNN com RBF</i>	Rejeita H ₀	0.014080	-inf	-0.86
<i>TDNN com PNN</i>	<i>TDNN com MLP</i>	Rejeita H ₀	0.000000	-inf	-21.35
<i>TDNN com RBF</i>	<i>TDNN com MLP</i>	Rejeita H ₀	0.000000	-inf	-17.78

²⁵ O *p-value* é a probabilidade de observar um resultado, dado que a hipótese nula é verdadeira. Valores pequenos para o *p-value* lançam dúvidas sobre a validade da hipótese nula.

6.3 Conclusões

O objetivo desse capítulo foi realizar a comparação estatística, através do teste de hipóteses *t-Student pareado*, do desempenho de classificação de sistemas de reconhecimento de padrões de odor criados com redes *RBF*, *PNN*, *MLP* e *TDNN*. As arquiteturas de redes neurais foram escolhidas por dois critérios principais. O primeiro critério foi investigar arquiteturas utilizadas anteriormente em outras bases de dados nesta nova base de dados. De acordo com este critério foram escolhidas as arquiteturas *MLP*, *TDNN* e *RBF*. O segundo critério foi investigar uma arquitetura que não houvesse sido utilizada para criar sistemas de reconhecimento de padrões de odor. A arquitetura *PNN* foi escolhida por este motivo.

As redes *PNN* correspondem a uma nova abordagem para criação de sistemas de reconhecimento de padrões de odor. Estas redes tiveram um erro médio de classificação de **1.1574%** no conjunto de teste. Este foi o menor erro obtido entre todos os sistemas criados, entretanto mesmo com o menor erro médio de classificação os testes de hipóteses mostraram que os classificadores criados com *PNN* não eram melhores do que os classificadores criados com a arquitetura *RBF com newrbe*, que obtiveram um erro médio de classificação de **1.3889%**. A grande vantagem de criar classificadores com a arquitetura *PNN* foi o pequeno tempo de treinamento dos mesmos, chegando a ser quase imediato. Porém a quantidade de nodos na camada escondida foi muito grande (1944) o que pode ser um problema, caso o sistema criado deva ser utilizado em equipamentos com poucos recursos computacionais. Outra vantagem de criar classificadores com *PNN* é relativa à quantidade reduzida de parâmetros que devem ser analisados, neste caso apenas o parâmetro relativo à largura da função *Gaussiana (SPREAD)* precisou ser investigado.

As redes *RBF* foram treinadas de duas maneiras diferentes. Na primeira maneira, foi utilizado o treinamento incremental (comando *newrb* do *Matlab*), ou seja, os nodos na camada escondida são incluídos um a um até que o critério de parada seja alcançado. Neste caso, o critério utilizado foi à soma do erro quadrático (GOAL). O erro médio de classificação no conjunto de teste foi **2.6852%** e a quantidade média de nodos na camada escondida foi 168.6, que é muito menor do que a quantidade de nodos da camada escondida das redes *PNN*. O desempenho dessas redes foi superior ao desempenho das redes *MLP* e também de todas as redes *TDNN*. A vantagem observada nessas redes foi que elas criaram classificadores com menos nodos na camada escondida, porém o tempo de treinamento foi bem superior ao tempo de treinamento das redes *PNN* e *RBF com newrbe*.

A segunda maneira utilizada para treinar redes *RBF* foi através do comando *newrbe* do *Matlab*. Este treinamento cria um nodo na camada escondida para cada padrão do conjunto de treinamento. As mesmas vantagens e desvantagens, observadas para as redes *PNN* também foram observadas para as redes *RBF com newrbe*. O erro médio de classificação no conjunto de teste foi **1.3889%**. A quantidade de nodos na camada escondida foi igual à quantidade das redes *PNN* (1944). Embora o erro médio de classificação tenha sido maior do que o das redes *PNN*, o teste de hipótese mostrou que os classificadores criados com *PNN* e os classificadores criados com *newrbe* são equivalentes a um nível de significância de 5%.

As redes *MLP* são umas das redes mais utilizadas para classificação de odor. Sua utilização em narizes artificiais já foi bastante investigada e seus resultados são muitas vezes melhores do que os resultados obtidos com outras técnicas tradicionais de classificação de odor. Para a classificação dos nove tipos de aguarrás, as redes *MLP* apresentaram um erro médio de classificação de **32.5019%**. Nos testes de hipóteses realizados o desempenho dessas redes não foi superior ao desempenho de nenhuma outra rede. A grande vantagem de utilizar redes *MLP* é o número reduzido de nodos na camada escondida, que neste caso foi 16, porém o tempo de treinamento dessas redes é muito maior do que o tempo necessário para treinar redes *RBF* ou *PNN*, e a quantidade de parâmetros que podem ser ajustados também é bem maior.

As redes *TDNN* foram criadas através do artifício de incluir a informação temporal nos dados de entrada. Dessa forma foi possível investigar três arquiteturas diferentes para treinamento de redes *TDNN*. A primeira arquitetura investigada para criar as redes *TDNN* foi à arquitetura *PNN*. Essas redes obtiveram um erro médio de classificação no teste de **4.2995%**. A quantidade de nodos criados na camada escondida dessas redes foi muito grande (1863). A inclusão da informação temporal não melhorou o desempenho das redes *TDNN com PNN* em relação às redes *PNN*. Como a dimensão dos dados de entrada das redes *TDNN* dobrou, acreditamos que seriam necessários mais padrões de treinamento para que as redes *TDNN* obtivessem melhores resultados. Outra possibilidade para tentar melhorar o desempenho das redes *TDNN* seria a utilização de alguma técnica para reduzir a dimensionalidades dos padrões de entrada. Fica então como sugestão para trabalhos futuros a tentativa de otimizar as redes *TDNN* pela utilização de conjuntos de treinamento maiores ou pela redução da dimensionalidade dos padrões de entrada.

A segunda arquitetura investigada foi *TDNN com RBF*. O erro médio de classificação dessas redes foi de **7.4396%**. Foram criados 1863 nodos na camada escondida. O desempenho dessas redes foi superior ao desempenho das redes *MLP* e das redes *TDNN com MLP*.

A terceira arquitetura investigada foi *TDNN com MLP*. O número de nodos na camada escondida foi 26 e o erro médio de classificação do teste foi **30.3430%**. O erro médio de classificação das redes *TDNN com MLP* foi inferior ao erro médio das redes *MLP*, ou seja, o processamento temporal ajudou a melhorar o desempenho de classificação dos padrões de odor, como já havia sido proposto por Yamazaki [Yamazaki, 2001]. Porém os testes de hipóteses mostraram que esta melhora não foi suficiente para afirmar com segurança que os classificadores *TDNN com MLP* desta dissertação são melhores do que os classificadores *MLP* a um nível de significância de 5%.

CONCLUSÃO

7 Introdução

Este capítulo apresenta as considerações finais desta dissertação, a lista de contribuições e as propostas de trabalhos futuros.

7.1 Considerações finais

A importância da olfação para a humanidade remonta desde a pré-história, em que o homem fazia uso dos sentidos do olfato com mais frequência do que na atualidade. Isto acontecia principalmente porque a procura por alimentos e a necessidade de proteção dos animais era muito maior do que atualmente.

Com a evolução da espécie humana, o homem foi diminuindo a utilização do olfato e conseqüentemente sua capacidade de perceber os odores a longas distancias diminuiu. Porém o olfato continua sendo muito importante para os seres humanos. O sentido do olfato pode ser aplicado em várias áreas tais como: monitoramento de processos industriais, diagnósticos de doenças e monitoramento ambiental. Existem situações onde a detecção de odores executada por seres humanos pode ser inconveniente ou mesmo perigosa.

As pesquisas para desenvolvimento de equipamentos que simulem o olfato humano têm se desenvolvido bastante. Dentro deste contexto, este trabalho se concentra na investigação de diferentes técnicas de redes neurais artificiais, que podem ser utilizadas para criação de sistemas de reconhecimento de padrões de odor para narizes artificiais.

Foram criados sistemas de reconhecimento de padrões de odor com redes *RBF* [Santos, 2000; Almeida et al., 2004], *MLP* [Santos, 2000; Yamazaki, 2001; Zanchettin, 2004; Almeida et al., 2004], *TDNN* [Yamazaki, 2001; Zanchettin, 2004] e *PNN*. A base de dados utilizada no estudo de casos desta dissertação é

composta por odores de nove tipos diferentes de aguarrás e foram obtidas por um protótipo de nariz artificial.

As redes *MLP* são as redes mais utilizadas para criação de sistemas de reconhecimento de padrões de odor [Santos, 2000; Yamazaki, 2001; Zanchettin, 2004], dessa forma elas foram escolhidas para que seus resultados sirvam de referência para comparação com as demais redes. As redes *RBF* e as redes *TDNN* foram utilizadas em alguns trabalhos [Yamazaki, 2001; Zanchettin, 2004; Almeida et al., 2004] e mostraram resultados promissores, elas foram escolhidas para comprovação de sua aplicação em uma nova base de dados. Como as redes *PNN* não haviam sido utilizadas anteriormente para construção de sistemas de reconhecimento de padrões de odor e como o treinamento dessas redes é muito rápido, elas foram escolhidas para investigação de seus resultados.

Quanto à base de dados utilizada neste trabalho, temos dois pontos principais que justificam a sua escolha: este é o primeiro trabalho que cria sistemas de reconhecimento de padrões de odor com redes neurais artificiais para esta base de dados e esta base de dados possui uma quantidade de classes bem superior a quantidade de classes das bases de dados utilizadas nos trabalhos investigados por esta dissertação [Santos, 2000; Yamazaki, 2001; Zanchettin, 2004; Almeida et al., 2004; Yamazaki, 2004]. Quanto maior o número de classes, maior é a complexidade do sistema criado. É importante verificar se as arquiteturas propostas anteriormente mantêm bons resultados em bases de dados mais complexas.

As redes *PNN* apresentaram um erro médio de classificação de **1.1574%** no conjunto de teste. Este foi o menor erro obtido entre todos os sistemas criados, entretanto mesmo com o menor erro médio de classificação, os testes de hipóteses mostraram que os classificadores criados com *PNN* não são melhores do que os classificadores criados com a arquitetura *RBF com neurbe*. Além de ter apresentado o menor erro médio no conjunto de teste, as redes *PNN* apresentaram a vantagem do tempo de treinamento ser muito pequeno (quase imediato) e a quantidade reduzida de parâmetros que foram investigados. Porém as redes *PNN* apresentaram uma desvantagem, quando comparamos com outras redes, que é a necessidade de criar um nodo na camada escondida para cada padrão de entrada. Esta desvantagem pode vir a ser um problema, caso o sistema criado deva ser utilizado em equipamentos com poucos recursos computacionais.

As redes *RBF* foram treinadas de duas maneiras diferentes. A primeira maneira adotada foi o treinamento incremental. Este treinamento apresentou uma vantagem, em relação à segunda opção de treinamento de redes *RBF*, que foi a criação de redes com uma quantidade relativamente pequena de nodos

na camada escondida. Porém estas redes apresentaram uma desvantagem que foi o longo tempo de treinamento das mesmas, pois os nodos na camada escondida são incluídos um a um até que o critério de parada seja alcançado. Neste caso, o critério utilizado foi à soma do erro quadrático (GOAL). O erro médio de classificação no teste foi **2.6852%** e a quantidade média de nodos na camada escondida foi *168.6*, que é muito menor do que a quantidade de nodos da camada escondida das redes *PNN*. O desempenho dessas redes só não foi superior ao desempenho das redes *PNN* e das redes *RBF com newrbe*.

A segunda maneira utilizada para treinar redes *RBF* foi através do comando *newrbe* do *Matlab*. Este treinamento cria um nodo na camada escondida para cada padrão do conjunto de treinamento. As redes *RBF com newrbe* foram equivalentes às redes *PNN*, inclusive foram observadas as mesmas vantagens e desvantagens das redes *PNN*. O erro médio de classificação no conjunto de teste foi **1.3889%**. A quantidade de nodos na camada escondida foi igual à quantidade das redes *PNN* (1994).

As redes *MLP* apresentaram um erro médio de classificação no teste de **32.5019%**. O desempenho dessas redes não foi superior ao desempenho de nenhuma outra rede. A grande vantagem encontrada para se utilizar redes *MLP* é que os sistemas criados apresentaram o menor número de nodos na camada escondida, porém a quantidade de parâmetros que podem ser investigados e o tempo de treinamento são desvantagens quando comparamos com redes *RBF* ou *PNN*. Durante o treinamento das redes *MLP*, verificamos a grande importância da normalização dos dados na qualidade dos resultados. A alteração da formatação dos padrões de saída da rede da faixa entre 0 e 1 para a faixa 0.1 e 0.9, permitiu criar sistemas mais estáveis e com erro médio bem inferiores.

As redes *TDNN* criadas utilizaram o artifício de *janelas de tempo*. Para isto foi criada uma base de dados com a informação dos odores obtidos pelo protótipo de nariz artificial no tempo t e no tempo $t+1$. Como a informação temporal está nos dados de entrada e não na arquitetura da rede, foram investigadas três arquiteturas diferentes para criar as redes *TDNN*. A primeira arquitetura utilizada para criar redes *TDNN* foi à arquitetura *PNN*. O erro médio de classificação, dessas redes, no conjunto de teste foi **4.2995%**. A inclusão da informação temporal não melhorou o desempenho das redes *TDNN com PNN* em relação às redes *PNN*. Uma das possíveis explicações para que os resultados dessas redes não tenham sido superiores aos resultados das redes *PNN* é que como a dimensão dos dados de entrada dobrou seriam necessários mais padrões de treinamento, ou então seria necessária a utilização de alguma técnica para reduzir a dimensionalidades dos padrões de entrada.

A segunda arquitetura investigada para criação de redes *TDNN* foi à arquitetura *RBF*. Os resultados obtidos por estas redes também não superaram os resultados obtidos pelas redes sem processamento temporal. O erro médio de classificação dessas redes foi de **7.4396%** e foram criados *1.863* nodos na camada escondida.

A terceira arquitetura investigada para criação de redes *TDNN* foi à arquitetura *MLP*. Assim como as redes *MLP* sem processamento temporal, as redes *TDNN com MLP* apresentam a vantagem de utilizar uma quantidade reduzida de nodos na camada escondida, porém o erro médio de classificação no conjunto de teste foi **30.3430%**. Este erro só foi inferior ao erro médio das redes *MLP*, ou seja, o processamento temporal ajudou a melhorar o desempenho de classificação dos padrões de odor, com a arquitetura *MLP*, mas isto não aconteceu para as redes *PNN* e *RBF*. Apesar de apresentar um erro médio de classificação, no conjunto de teste, inferior ao erro apresentado pelas redes *MLP*, os testes de hipóteses mostraram que esta melhora não foi suficiente para afirmar com segurança que os classificadores *TDNN com MLP* são melhores do que os classificadores *MLP* a um nível de significância de 5%.

A criação de sistemas de reconhecimento de padrões de odor, para um nariz artificial, com redes *PNN* ou *RBF*, permitiria uma rápida atualização do equipamento para novas bases de dados, já que o treinamento dessas redes é quase imediato e poucos parâmetros ajustáveis precisam ser investigados. Porém se o equipamento possuir poucos recursos computacionais, seria mais indicado à utilização da arquitetura *MLP*, pois esta arquitetura permite a criação de redes com menos nodos na camada escondida, embora seja necessário um tempo bem maior para treinamento das mesmas.

7.2 Contribuições deste trabalho

Os resultados obtidos por este trabalho servem de base para futuras comparações com novos sistemas de reconhecimento de padrões, além disso, outras contribuições podem ser destacadas como:

A utilização da técnica de *10-fold cross-validation com estratificação* para criação dos conjuntos de treinamento e teste mostrou que diferentes partições da mesma base de dados apresentam resultados variáveis e que para termos uma estimativa mais precisa do erro de classificação verdadeiro é necessário calcular uma média do erro de classificação em diferentes partições. Outro fator importante da utilização desta técnica se refere às redes *PNN* e *RBF*. Como estas redes são determinísticas, os resultados obtidos no treinamento das mesmas é igual para um mesmo conjunto de treinamento e mesmos parâmetros ajustáveis,

o que não acontece com as redes *MLP*. Os resultados das redes *MLP* para um mesmo conjunto de treinamento e mesmos parâmetros ajustáveis dependem também da inicialização dos pesos, logo as redes *MLP* não são determinísticas.

Utilização de diferentes arquiteturas para criação de redes *TDNN* com *janelas de tempo*. Como a informação temporal foi inserida nos padrões de entrada e não na arquitetura das redes, foi possível investigar a criação de redes *TDNN* com as arquiteturas *RBF* e *PNN* além da arquitetura *MLP* que já havia sido utilizada em outros trabalhos [Yamazaki, 2001; Zanchettin, 2004].

Apresentação das potencialidades das redes *PNN* para criação de sistemas de reconhecimento de padrões de odor. As redes *PNN* não haviam sido utilizadas anteriormente em problemas de reconhecimento de padrões de odor em narizes artificiais.

Criação de redes *RBF* com diferentes algoritmos de treinamento para criação de sistemas de reconhecimento de padrões em narizes artificiais e apresentação das vantagens e desvantagens das mesmas.

7.3 Proposta de trabalhos futuros

Os dados utilizados neste trabalho não foram utilizados anteriormente para criação de sistemas de reconhecimento de padrões de odor com redes neurais, um trabalho futuro poderia ser a utilização de outras abordagens de redes neurais artificiais com a base de dados deste trabalho.

Considerando o desempenho apresentado pelas redes *PNN* e a simplicidade das mesmas, a investigação dessas redes em outras bases de dados é interessante.

Outra possibilidade é aplicação de técnicas para compressão e diminuição de ruído inerente aos estímulos odorantes, como por exemplo a Transformada de Wavelet, que foi utilizada por Zanchettin [Zanchettin, 2004] aos dados utilizados nesta dissertação para verificar o resultado dos classificadores.

Reduzir a dimensionalidade dos dados de entrada, utilizando, por exemplo, a técnica de *PCA*, para verificar se o desempenho das redes *TDNN* para esta base de dados pode ser melhorado.

Investigação da possibilidade de utilização de técnicas de otimização, como a técnica que foi proposta por Yamazaki [Yamazaki, 2004], para reduzir o tamanho das redes *PNN* e das redes *RBF*.

REFERÊNCIA BIBLIOGRÁFIA

- [Endereço 1] <http://www.cfi.lu.lv/isoen/secretariat.html>
Acesso em: 01 fev. 2004
- [Endereço 2] <http://www.wired.com/news/technology/0,1282,31556,00.html>
Acesso em: 11 out. 2003
- [Endereço 3] <http://news.bbc.co.uk/1/hi/health/1492338.stm>
Acesso em: 11 out. 2003
- [Endereço 4] <http://www.spacedaily.com/news/future-02a.html>
Acesso em: 11 out. 2003
- [Almeida et al., 2004] Almeida, M. B., Souto, M. C. P., Ludermir, T. B. Comparative study of connectionist techniques for implementing the pattern recognition system of an artificial nose. A ser publicado no *IJCCN*, 2004.
- [Barros Filho et al., 2004] Barros Filho, V. N. A., Aquino, R. R. B., Ferreira, A. A., Souza, L. T. A. Monitoramento e diagnóstico de estado de grandes transformadores, com ênfase para o diagnóstico pela cromatografia com o uso de redes neurais. A ser publicado no *IX Simpósio de Especialistas em Planejamento da Operação e Expansão Elétrica*, 2004.
- [Batista, 1997] Batista, G. E. A. P. A. Um Ambiente de Avaliação de Algoritmos de Aprendizado de Máquina. Dissertação de Mestrado, USP –São Carlos, 1997.
- [Braga et al., 2000] Braga, A. P., Ludermir, T. B., Carvalho, A. C. P. L. F. *Redes Neurais Artificiais Teoria e Aplicações*, LTC, 2000.
- [Buck et al., 1965] Buck, T. M., Allen, F. G. e Dalton, M. Detection of chemical species by surface effects on metals and semiconductors. T. Bregman e A. Dravnieks (Eds.), *Surface Effects in Detection*, Spartan Books Inc., USA, 1965.
- [Chen, 1991] Chen, C. H. *Neural Networks in Pattern Recognition and Their Applications* (pp. iii), World Scientific Publishing Co. Pte. Ltd., 1991.
- [D'Amico et al., 1997] D'Amico, A., Di Natale, C. e Verona, E. Acoustic Devices. E. Kress-Rogers (Ed.), *Handbook of Biosensors and Electronic Noses: Medicine, Food and the Environment* (pp. 197-223), CRC Press, 1997.
- [Dravnieks e Trotter, 1965] Dravnieks, A. e Trotter, P. J. Polar vapors detection based on thermal

- modulation of contact potentials. *Journal of Scientific Instruments*, vol. 42, (pp. 624), 1965.
- [Elman, 1990] Elman, J. L. Finding structure in time. *Cognitive Science*, vol. 14, (pp. 179-211), 1990.
- [Fahlman, 1988] Fahlman, S. E. An empirical study of learning speed in backpropagation networks, *Technical report*, Carnegie Mellow University, 1988.
- [Gardner, 1987] Gardner, J. W. Pattern recognition in the Warwick Electronic Nose, 8th *Int. Congress of European Chemoreception Research Organization*, University of Warwick, UK., 1987.
- [Gardner e Barlett, 1992] Gardner, J. W. e Bartlett, P. N. Sensors and Sensory Systems for an Electronic Nose, *NATO ASI Series E: Applied Sciences*, vol. 212, Kluwer, Dordrecht, 1992.
- [Gardner e Barlett, 1994] Gardner, J. W. e Bartlett, P. N. A brief history of electronic noses. *Sensors and Actuators B*, vol. 18-19, (pp. 211-220), 1994.
- [Gardner e Barlett, 1995] Gardner, J. W. e Bartlett, P. N. Application of conducting polymer technology in Microsystems. *Sensors and Actuators A*, 51, (pp. 57-66), 1995.
- [Gardner e Hines, 1997] Gardner, J. W. e Hines, E. L. Pattern Analysis Techniques, E. Kress-Rogers (Ed.), *Handbook of Biosensors and Electronic Noses: Medicine, Food and Environment*, CRC Press., (pp. 633-652), 1997.
- [Gardner et al., 1990a] Gardner, J. W., Hines, E. L. e Wilkinson, M. Application of artificial neural networks to an electronic olfactory system. *Measurement Science Technology*, vol. 1, (pp. 446-451), 1990.
- [Gardner et al., 1990b] Gardner, J. W., Bartlett, P. N., Dodd, G. H., Shormer, H. V. The design of an artificial olfactory system. Schild (Ed.), *Chemosensory Information Processing, NATO ASI Series H: Cell Biology*, vol. 39, Springer, Berlin, (pp. 131-173), 1990.
- [Gardner et al., 1994] Gardner, J. W., Pearce, T. C., Friel, S., Bartlett, P. N. e Blair, N. A Multisensor system for beer flavour monitoring using an array of conducting polymers and predictive classifiers. *Sensors and Actuators B*, vol. 18-19, (pp. 240-243), 1994.
- [Grate et al., 1997] Grate, J. W., Abraham, M. H. e McGill, R. A. Sorbent Polymer

- Materials for Chemical Sensors and Array. E. Kress-Rogers (Ed.), *Handbook of Biosensors and Electronic Noses: Medicine, Food and the Environment*, (pp. 593-612), CRC Press, 1997.
- [Hagan e Menhaj, 1994] Hagan, M. e Menhaj, M. Training feedforward networks with the Marquardt algorithm. In *IEEE Transactions on Neural Networks*, vol. 5, (pp. 989-993), 1994.
- [Haykin, 1999] Haykin, S. *Redes Neurais: Princípios e Prática*, ARTMED Editora S. A., Brasil, (pp 27-69, 283-348), 1991.
- [He e Lapedes, 1991] He, X. e Lapedes, A. Nonlinear modeling and prediction by successive approximations using radial basis functions. *Technical Report*, Los Alamos National Laboratory, 1991.
- [Hildebrand, 1986] Hildebrand, D. *Statistical Thinking for Behavioral Scientists*. Duxbury Press, Boston, MA, 1986.
- [Hoffheins, 1989] Hoffheins, B. S. Using Sensor Arrays and Pattern Recognition to Identify Organic Compounds. MS-Thesis, The University of Tennessee, Knoxville, TN, 1989.
- [Hopfield, 1982] Hopfield, J. J. Neural networks and physical system with emergent collective properties, *Proc. Nat. Acad. Sci.*, vol. 79, (pp. 2554-2558), 1982.
- [Ikegami e Kaneyasu, 1985] Ikegami, A. e Kaneyasu, M. Olfactory detection using integrated sensors, *Proc. 3rd Int. Conf. Solid-State Sensors and Actuators (Transducers'85)*, Philadelphia, PA, USA, (pp. 136-139), 1985.
- [Jordan, 1986] Jordan, M. I. Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of Eighth Annual Conference of the Cognitive Science Society*, (pp. 531-546), Amherst, Hillsdale: Erlbaum, 1986.
- [Keller et al., 1994] Keller, P. E., Kouzes, R. T. e Kangas, L. J. Three neural network based sensor system for environmental monitoring. *IEEE Electro 94 International Conference*, Boston, MA, (pp. 377-382), 1994.
- [Keller et al., 1995a] Keller, P. E., Kangas, L. J., Liden, L. H., Hashem, S. e Kouzes, R. T. Electronic noses and their applications. *IEEE North on Technical Application Conference (TAC '95)*, Portland, OR, (pp. 116-119), 1995.
- [Keller et al., 1995b] Keller, P. E., Kouzes, R. T, Kangas, L. J., Hashem, S. Transmission of Olfactory Information for Telemedicine. *Interactive Technology and the New Paradigm for Healthcare*, (pp. 168-172), 1995.
- [Kress-Rogers, 1997] Kress-Rogers, E. Biosensors and Electronic Noses for Practical Applications. E. Kress-Rogers (Ed.), *Handbook of Biosensors and*

Electronic Noses: Medicine, Food and the Environment, CRC Press, (pp. 3-39), 1997.

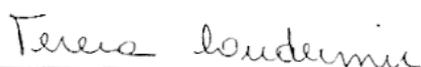
- [Lauf e Hoffheins, 1991] Lauf, R. J. e Hoffheins, B. S. Analysis of Liquid Fuels Using a Gas Sensor Array, *Fuel*, vol. 70, (pp. 935-940), 1991.
- [Lawrence e Geddes, 1997] Lawrence, C. R. e Geddes, N. J. Surface Plasmon Resonance (SPR) for Biosensing. E. Kress-Rogers (Ed.), *Handbook of Biosensors and Electronic Noses: Medicine, Food and the Environment*, CRC Press, (pp. 149-168), 1997.
- [McCulloch e Pitts, 1943] McCulloch, W. S. e Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol. 5, (pp.115-133), 1943.
- [Minsky e Papert, 1969] Minsky, M., Papert, S. *Perceptrons: an introduction to computational geometry*, MIT Press, Massachusetts, 1969.
- [Moncrieff, 1961] Moncrieff, R. W. An instrument for measuring and classifying odors, *Journal Applied Physiology*, vol. 16, (pp. 742), 1961.
- [Moriizumi et al., 1992] Moriizumi, T., Nakamoto, T. e Sakuraba Y. Pattern Recognition in Electronic Noses by Artificial Neural Network Models. J. W. Gardner e P. N. Bartlett (Eds). *Sensors and Sensory Systems for an Electronic Nose*. Kluwer Academic Publishers, Amsterdam, Netherlands, (pp. 217-236), 1992.
- [Nakamoto et al., 1992] Nakamoto, T., Takagi, H., Utumi, S. e Moriizumi, T. Gas/Odor Identification by Semiconductor Gas Sensor Array and an Analog Artificial Neural Network Circuit. *International Conference on Microelectronics*, vol. 4, Bandung, Indonesia, (pp. 1-9), 1992.
- [Pearlmutter, 1992] Pearlmutter, B. Gradient Descent: second order momentum and saturation error. J. E. Moody, S. Hanson e R. Lippman (Eds.). *Advances in Neural Information Processing Systems 2*, (pp. 887-894), Morgan Kaufmann, 1992.
- [Persaud e Dodd, 1982] Persaud, K. e Dodd, G. H. Analysis of discrimination mechanisms of mammalian olfactory system using a model nose, *Nature*, vol. 299, (pp. 352-355), 1982.
- [Prechelt, 1994] Prechelt, L. Proben1 – A Set of Neural Network Benchmark Problems and Benchmarking Rules. *Technical Report 21/94*, Fakultät für

Informatik, Universität Karlsruhe, Germany, 1994.

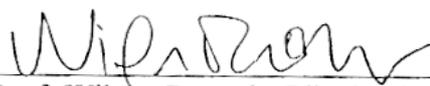
- [Riedmiller, 1994] Riedmiller, M. Rprop – description and implementation details. *Technical report*, University of Karlsruhe, 1994.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hilton, G. E. e Williams, R. J. Learning internal representations by error propagation. D. E. Rumelhart e J. L. McClelland (Eds.). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol. 1: Foundations*, Cambridge, MA, MIT Press, (pp. 318-362), 1986.
- [Santos, 2000] Santos, M. S. Construção de um nariz artificial usando redes Neurais, Tese de Doutorado, Centro de Informática, UFPE, 2000.
- [Shurmer et al., 1987] Shurmer, H. V., Fard, A., Barker J., Bartlett P. N., Dodd G. H. e Hayat U. *Phys Technology, vol. 18*, (pp. 359-369), 1987.
- [Shurmer, 1990] Shurmer, H. V. An electronic nose: a sensitive and discriminating substitute for a mammalian olfactory system. *IEE Proceedings 137 G*, (pp. 197-204), 1990.
- [Skapura, 1996] Skapura, D. M. Gordon, P. S. (Ed.), *Building Neural Networks*, ACM Press, (pp. 41-42), 1996.
- [Specht, 1990] Specht, D. F. Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification. *IEEE Transactions on Neural Networks, vol. 1*, (pp. 111-121), 1990.
- [Spiegel, 1979] Spiegel, Murray Ralph. Ed. e rev. por Pereira, C. J. McGraw-Hill do Brasil (Ed.), Coleção Schaum. *Estatística: resumo da teoria, 875 problemas resolvidos, 619 problemas propostos*, 1979.
- [Sundgren et al., 1991] Sundgren, H., Winqvist, F., Lukkari, I. e Lundstrom, I. Artificial Neural Networks and Gas Sensor Arrays: Quantification of Individual Components in a Gas Mixture. *Measurement Science and Technology, vol. 2*, (pp. 464-469), 1991.
- [Waibel et al., 1989] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. e Lang, K. Phonemic Recognition using Time Delay Neural Networks. *IEEE Trans. Acoust., Speech, Signal Processing, vol. 37*, n. 3, (pp. 328-339), 1989.

- [Wilkens e Hatman, 1964] Wilkens, W. F. e Hatman, A.D. An electronic analog for olfactory process, *Ann. NY Academy of Science*, vol. 116, (pp. 608), 1964.
- [Witten e Frank, 2000] Witten, I. H. e Frank, E. Cerra, D. D. (Ed.) *Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.
- [Yamazaki, 2001] Yamazaki, A. Reconhecimento de padrões em um nariz artificial por redes neurais, Dissertação de Mestrado, Centro de Informática, UFPE, 2001.
- [Yamazaki, 2004] Yamazaki, A. Uma metodologia para otimização de arquiteturas e pesos de redes neurais. Tese de doutorado, Centro de Informática, UFPE, 2004.
- [Zanchettin, 2004] Zanchettin, C. Sistema Neural Híbrido Para Reconhecimento de Padrões em um Nariz Artificial. Dissertação de Mestrado, Centro de Informática, UFPE, 2004.

Dissertação de Mestrado apresentada por **Aida Araújo Ferreira** a Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título, "**Comparação de Arquitetura de Redes Neurais para Sistemas de Reconhecimento de Padrões em Narizes Artificiais**", orientada pela **Profa. Teresa Bernarda Ludermir** e aprovada pela Banca Examinadora formada pelos professores:



Profa. Teresa Bernarda Ludermir
Centro de Informática / UFPE



Prof. Wilson Rosa de Oliveira Junior
Departamento de Matemática e Física / UFRPE



Prof. Marcilio Carlos Pereira de Souto
Departamento de Informática e Matemática Aplicada / UFRN

Visto e permitida a impressão.
Recife, 1 de março de 2004.



Prof. JAELSON FREIRE BRELAZ-DE CASTRO

Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.