

**UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE INFORMÁTICA**

**Autenticação On-line de assinaturas utilizando
Redes Neurais**

Milton Roberto Heinen

**Prof. Dr. Fernando Santos Osório
Orientador**

*Monografia submetida como requisito
parcial para a obtenção do título de
Bacharel em Informática.*

São Leopoldo, novembro de 2002

*Dedico este trabalho aos meus
pais e a minha noiva,
Alessandra Huther.*

Sumário

LISTA DE FIGURAS	6
LISTA DE TABELAS	8
LISTA DE ABREVIATURAS.....	9
AGRADECIMENTOS	10
RESUMO.....	11
ABSTRACT	12
1 INTRODUÇÃO	13
2 RECONHECIMENTO DE ASSINATURAS	15
2.1 Conceitos gerais.....	15
2.1.1 Identificação de assinaturas	15
2.1.2 Autenticação de assinaturas	15
2.1.3 Detecção de fraudes	15
2.2 Formas de autenticação:	16
2.2.1 <i>Off-line</i>	16
2.2.2 <i>On-line</i>	16
2.3 Aplicações possíveis.....	17
2.3.1 Caixas eletrônicos	18
2.3.2 Cartões de crédito [5]	18
2.3.3 Serviços de <i>home-banking</i>	18
2.3.4 Sistemas de gestão	18
2.3.5 Acesso a instalações de segurança	19
2.4 Sistemas de Reconhecimento de Assinaturas.....	19
2.5 Etapas da autenticação de assinaturas.....	22
2.5.1 Obtenção e armazenamento das assinaturas	22
2.5.2 Pré processamento	22
2.5.3 Processamento	23
3 REDES NEURAIS ARTIFICIAIS	24
3.1 Reconhecimento de Padrões.....	24
3.2 Inteligência Artificial e Aprendizado de Máquinas.....	25

3.3	Redes Neurais Artificiais.....	25
3.3.1	<i>Perceptron.....</i>	28
3.3.2	<i>Backpropagation.....</i>	31
3.3.3	Problemas e limitações do <i>Backpropagation.....</i>	33
3.3.4	<i>Cascade-Correlation.....</i>	34
3.4	Redes Neurais aplicadas a autenticação de assinaturas	35
3.5	Problema de classes fechadas.....	36
3.6	Simulador Neusim.....	38
4	SISTEMA NEURALSIGNX.....	40
4.1	Módulo de entrada	40
4.2	Pré-processamento	41
4.2.1	Montagem da assinatura.....	42
4.2.2	Ajuste de posição.....	43
4.2.3	Ajuste de escala	44
4.2.4	Extração de atributos.....	45
4.2.4.1	Tempo de duração da assinatura.....	45
4.2.4.2	Número de vezes em que a caneta foi levantada	45
4.2.4.3	Velocidade média da assinatura	45
4.2.4.4	Velocidade máxima da assinatura	45
4.2.4.5	Amostragem sequencial da assinatura	46
4.2.4.6	Simetria da assinatura.....	47
4.2.4.7	Densidade da assinatura com informações de grade.....	48
4.2.4.8	Interseções de linhas verticais em relação a assinatura.....	49
4.2.4.9	Interseções de linhas horizontais em relação a assinatura.....	49
4.2.4.10	Número de vetores apontando para cada um dos pontos geográficos	49
4.2.4.11	Soma do comprimento dos vetores para cada ponto geográfico.....	50
4.2.4.12	Número de pontos coletados pelo <i>tablet</i>	50
4.2.4.13	Comprimento total da assinatura.....	51
4.2.4.14	Número de trocas de sentido da caneta nos eixos x e y	51
4.2.4.15	Densidade de pontos da assinatura por quadrantes.....	51
4.2.4.16	Número de cruzamentos	51
4.2.4.17	Número de laços fechados da assinatura.....	51
4.2.4.18	Pressão média da caneta	52
4.2.4.19	Pressão máxima da caneta.....	52
4.2.4.20	Número de interseções da assinatura em relação a linhas diagonais.....	52
4.2.4.21	Trajetória da assinatura em relação a grade	52
4.3	Processamento.....	52
4.4	Protótipo do Sistema NeuralSignX	52
4.4.1	Módulo de entrada	53
4.4.2	Módulo de pré-processamento.....	54
4.4.3	Processamento	58
4.4.4	Fase de reconhecimento	59
5	RESULTADOS OBTIDOS.....	60
5.1	Base de dados de assinaturas	60
5.2	Configuração dos parâmetros do sistema	61
5.3	Determinação dos atributos a serem utilizados	62

5.3.1	Análise de Componentes Principais (ACP)	62
5.3.2	Matriz de correlação.....	64
5.3.3	Utilização do Sistema NeuralSignX para a seleção de atributos.....	65
5.3.3.1	Identificação de usuários.....	66
5.3.3.2	Atributos similares	66
5.3.4	Árvores de Decisão	68
5.4	Validação do modelo proposto	70
5.4.1	Identificação de usuários	70
5.4.2	Autenticação de assinaturas.....	71
5.4.2.1	Primeira simulação – Somente assinaturas autênticas	72
5.4.2.2	Segunda simulação – assinaturas autênticas e pictográficas	75
5.4.2.3	Terceira simulação – utilização da Nuvem de Pontos.....	78
5.4.3	Utilização de assinaturas falsificadas	80
6	CONCLUSÃO	84
	ANEXO A EXEMPLOS DE ASSINATURAS COLETADAS	86
	ANEXO B EXEMPLOS DE ARQUIVOS DO SISTEMA NEURALSIGNX	87
B.1	Arquivo de assinaturas do Sistema NeuralSignX	87
B.2	Arquivo de configuração do Sistema NeuralSignX.....	87
B.3	Arquivo de configuração do Neusim (CFG).....	88
B.4	Arquivo de Aprendizado do Neusim (LRN).....	89
B.5	Arquivo de saída do Neusim (OUT).....	89
	BIBLIOGRAFIA.....	90

Lista de figuras

Figura 2.1 - Exemplo de um <i>tablet</i> [9]	16
Figura 2.2 - Subdivisão dos tipos de reconhecimento de escrita	20
Figura 2.3 - Etapas de um sistema de autenticação de assinaturas	22
Figura 3.1 - Neurônio biológico humano [32].....	26
Figura 3.2 - Esquema de um neurônio artificial [32]	27
Figura 3.3 - Gráfico da função de transferência Sigmoides	27
Figura 3.4 - Descida do gradiente de uma superfície de erro	29
Figura 3.5 - Esquema de um Neurônio artificial com duas entradas e uma saída	30
Figura 3.6 - Reta que divide as classes de saída da função AND	30
Figura 3.7 - Retas que dividem as classes de saída da função XOR.....	31
Figura 3.8 - Esquema de uma Rede Neural do tipo MLP [32]	31
Figura 3.9 - Curvas de erro no aprendizado e na generalização [32].....	33
Figura 3.10 - Topologia incremental do <i>Cascade-Correlation</i>	35
Figura 3.11 - Problema de classificação envolvendo classes fechadas.....	36
Figura 3.12 - Retas de divisão das classes A, B e C	36
Figura 3.13 - Separação da classe fechada B em relação as demais classes	37
Figura 3.14 - Execução do simulador de Redes Neurais Neusim.....	38
Figura 4.1 - Módulos do Sistema NeuralSignX.....	40
Figura 4.2 - Atividades do módulo de pré-processamento	41
Figura 4.3 - Pontos de uma assinatura coletados pelo <i>tablet</i>	42
Figura 4.4 - Pontos coletados sendo interligados através de retas	42
Figura 4.5 - Forma final de uma assinatura depois da montagem	42
Figura 4.6 - Ajuste de posição pelo canto superior esquerdo	43
Figura 4.7 - Centro de massa de uma assinatura.....	43
Figura 4.8 - Ajuste de posição pelo centro de massa de uma assinatura.....	44
Figura 4.9 - Assinatura após o ajuste de posição pelo centro de massa	44
Figura 4.10 - Ajuste de escala (assinaturas centralizadas)	44
Figura 4.11 - Ajuste de escala (assinaturas ajustadas pelo canto superior esquerdo)	45
Figura 4.12 - Exemplo de assinatura com uma trajetória não usual	46
Figura 4.13 - Assinatura com os pontos inicial e final destacados	46
Figura 4.14 - Assinatura com 8 pontos da trajetória amostrados.....	47

Figura 4.15 - Assinatura com maior simetria vertical	47
Figura 4.16 - Assinatura com maior simetria horizontal	48
Figura 4.17 - Densidade de pontos com informações de grade	48
Figura 4.18 - Linhas verticais e horizontais que interseccionam uma assinatura.....	49
Figura 4.19 - Forma estrutural de uma assinatura.....	49
Figura 4.20 - Analogia de uma assinatura com os pontos cardeais	50
Figura 4.21 - Divisão de uma assinatura a partir do centro de massa	51
Figura 4.22 - Caixa de mensagem para informar o nome do usuário	53
Figura 4.23 - Módulo de entrada do Sistema NeuralSignX.....	53
Figura 4.24 - Tela principal do módulo de pré-processamento	54
Figura 4.25 - Tela de configuração do protótipo	55
Figura 4.26 - Tela de escolha dos atributos serão utilizados nas simulações	57
Figura 4.27 - Tela principal do programa que realiza a autenticação das assinaturas ...	59
Figura 5.1 - Árvore de decisão para as assinaturas do usuário Arno Heinen	68
Figura 5.2 - Árvore de decisão para as assinaturas do usuário Jorge Felipe	69
Figura 5.3 - Árvore de decisão para as assinaturas do usuário Paulo César Flores	69
Figura 5.4 - Faixa de valores de saída para uma assinatura autêntica.....	72
Figura 5.5 - Exemplo de uma assinatura do usuário Milton Heinen.....	73
Figura 5.6 - Exemplo de uma assinatura da Marilei da Costa.....	74
Figura 5.7 - Assinatura indevidamente aceita como sendo da Marilei da Costa	74
Figura 5.8 - Exemplo de uma assinatura do usuário Carlos Schneider	75
Figura 5.9 - Exemplo de uma assinatura da Lori Huther	76
Figura 5.10 - Assinatura indevidamente aceita como sendo da Lori Huther.....	76
Figura 5.11 - Árvore de decisão gerada para as assinaturas da Lori Huther	76
Figura 5.12 - Regras de produção geradas para as assinaturas da Lori Huther	77
Figura 5.13 - Regras de produção geradas para as assinaturas de Miguel de Oliveira ..	77
Figura 5.14 - Árvore de decisão criada utilizando a Nuvem de Pontos	78
Figura 5.15 - Exemplo de uma assinatura da Tatiane Boll corretamente aceita.....	79
Figura 5.16 - Exemplo de uma assinatura da Tatiane Boll indevidamente rejeitada	79
Figura 5.17 - Assinatura original, falsificação traçada e falsificação especializada das assinaturas do usuário Milton Heinen.....	81
Figura 5.18 - Assinatura original, falsificação traçada e falsificação especializada das assinaturas do usuário Jalton Heinen	82
Figura 5.19 - Assinatura original, falsificação traçada e falsificação especializada das assinaturas da Alessandra Huther	83

Lista de tabelas

Tabela 3.1 - Valores de entrada e saída para a função AND	29
Tabela 3.2 - Valores das entradas, dos pesos e das saídas obtidos após o aprendizado da função AND	31
Tabela 5.1 - Valores obtidos para os parâmetros genéricos do Sistema NeuralSignX ..	61
Tabela 5.2 - Valores obtidos para os parâmetros específicos de alguns atributos do Sistema NeuralSignX.....	61
Tabela 5.3 - <i>Eigenvalues</i> obtidos para os vinte primeiros fatores da Análise de Componentes Principais.....	63
Tabela 5.4 - Entradas que mais contribuíram na formação do primeiro fator da Análise de Componentes Principais	64
Tabela 5.5 - Exemplos de saídas desejadas para um sistema de identificação de usuários com cinco usuários cadastrados	66
Tabela 5.6 - Resultados obtidos com a identificação de 20 usuários diferentes	70
Tabela 5.7 - Resultados obtidos na identificação de 46 usuários diferentes.....	71
Tabela 5.8 - Resultados obtidos na autenticação das assinaturas do usuário Milton Heinen	72
Tabela 5.9 - Resultados obtidos na autenticação de assinaturas de cinco usuários	73
Tabela 5.10 - Resultados obtidos para as assinaturas de Carlos Schneider	75
Tabela 5.11 - Resultados obtidos na autenticação de assinaturas de cinco usuários (assinaturas reais e pictográficas)	75
Tabela 5.12 - Resultados obtidos na autenticação de assinaturas da Tatiane Boll.....	79
Tabela 5.13 - Resultados obtidos na autenticação de assinaturas de cinco usuários utilizando a Nuvem de Pontos.....	80
Tabela 5.14 - Resultados obtidos utilizando a Nuvem de Pontos apenas no arquivo de aprendizado	80
Tabela 5.15 - Resultados obtidos na autenticação de assinaturas do usuário Milton Heinen (com fraudes envolvidas)	81
Tabela 5.16 - Resultados obtidos na autenticação de assinaturas de cinco usuário (com fraudes envolvidas)	82

Lista de abreviaturas

IA	Inteligência Artificial
RNA	Redes Neurais Artificiais
ACP	Análise de Componentes Principais
CBR	<i>Case Based Reasoning</i>
ILP	<i>Inductive Logic Programming</i>
IDT	<i>Induction of Decision Trees</i>
AND	Função Lógica E
XOR	Função Lógica OU Exclusivo
MLP	<i>Multi Layer Perceptron</i>
MS-DOS	<i>Microsoft Disk Operation System</i>
AI	Aceites Indevidos
RI	Rejeições Indevidas
DP	Desvio Padrão

Agradecimentos

Agradeço a minha noiva, Alessandra Huther, que sempre esteve ao meu lado quando eu mais precisei, e aos meus pais e irmãos, que me deram todo o apoio necessário para que eu pudesse atingir meus objetivos.

Gostaria de agradecer também a todas as pessoas que contribuíram para a realização deste trabalho, fornecendo as assinaturas para a montagem da base de dados, e principalmente ao meu orientador, Fernando Santos Osório, pela sua paciência, apoio e confiança em mim e no meu trabalho.

Agradeço também a todas as pessoas que estiveram ao meu lado, aos meus amigos, aos meus colegas de graduação e de trabalho, e todas as pessoas que conviveram comigo durante esta longa jornada.

Obrigado a todos vocês !

Resumo

Nos dias atuais, o uso cada vez mais freqüente de sistemas de informação traz a necessidade de se autenticar os usuários de forma segura. Na maioria dos sistemas, a autenticação de usuários ocorre através de senhas alfanuméricas, que representam um sério problema de segurança quando acabam parando em mãos erradas. Para evitar este problema, várias formas de autenticação de usuários baseadas em características biométricas vem sendo desenvolvidas, mas estas implicam em um custo elevado dos equipamentos de *hardware* e em um alto grau de intrusão. O objetivo deste trabalho é apresentar a proposta de uma metodologia, e a sua implementação em um protótipo, que permita realizar a autenticação de usuários através do uso de assinaturas manuscritas.

Abstract

Nowadays, the increasing utilization of information systems brings forward the need for a secure way to authenticate users. In most systems, user authentication consists of recognizing alphanumeric passwords, which represent a serious security problem when unauthorized people get access to them. To avoid this problem, several forms of user authentication based on biometrics characteristics are being developed, but these imply in highly expensive hardware equipment and in a high intrusion level. The purpose of this paper is to present a proposal of methodology, and further development into a prototype, that allows accomplishing the user authentication through handwritten signatures.

1 Introdução

Nos dias atuais, a tecnologia da informação está cada vez mais presente em nossas vidas. Nas empresas, os sistemas de gestão são utilizados para controlar as mais diversas atividades, como os estoques, a contabilidade, as finanças, entre outras atividades. Nos bancos, caixas eletrônicos permitem que as pessoas retirem dinheiro de suas contas em qualquer hora do dia ou da noite, até mesmo nos fins de semana. Com o advento da Internet, os serviços de *home banking* permitiram que os correntistas pudessem acessar seus saldos, fazer transferências e pagamento de contas sem sair de casa. A Internet propiciou também o surgimento do comércio eletrônico, onde as pessoas podem fazer compras sem sair de casa e fazer os pagamentos via cartão de crédito.

Todas estas inovações trouxeram muitos benefícios para a sociedade, pois as empresas puderam reduzir seus custos e assim se tornarem mais competitivas, os bancos puderam propiciar um maior conforto para seus clientes, e o comércio eletrônico permitiu que os consumidores tivessem mais liberdade de escolha e comodidade nas suas compras. Mas um dos problemas que surgiram com todas estas tecnologias é a autenticação de usuários, que implica em garantir que a pessoa que está tentando acessar um sistema é quem ela realmente diz ser. Na maioria dos sistemas de informação e aplicações comerciais, a autenticidade dos usuários é garantida através de senhas alfanuméricas, que devem ser memorizadas pelos usuários e mantidas a salvo de outras pessoas. Apesar de ser a forma de autenticação mais difundida na atualidade, as senhas apresentam diversas vulnerabilidades do ponto de vista de segurança, pois se elas caírem em mãos erradas devido a descuidos ou ingenuidade por parte dos usuários, toda a segurança de um sistema é ameaçada. De fato, a maior parte das fraudes eletrônicas não ocorrem devido a falhas nos protocolos de segurança, mas sim devido a descuidos dos usuários que de alguma forma permitem que as suas senhas sejam descobertas por pessoas mal intencionadas.

Devido a estes problemas de segurança, técnicas de autenticação de usuários baseadas em características biométricas físicas, como as impressões digitais, o exame do fundo da retina e da palma das mãos vem sendo utilizadas como uma forma de garantir a autenticidade dos usuários [9]. Estas técnicas são muito mais seguras que as senhas, mas esbarram em alguns problemas operacionais que impedem seu uso em larga escala. O primeiro problema é o custo, pois sistemas de autenticação baseados em características biométricas físicas exigem um *hardware* especializado de custo elevado. O segundo problema é o grau de intrusão elevado apresentado por grande parte das técnicas de autenticação baseadas em características biométricas físicas, que causam desconfortos aos usuários [5]. Em relação as impressões digitais, seu uso possui uma conotação negativa devido ao fato delas estarem muito ligadas a delitos e investigações criminais [12]. As características biométricas físicas são muito utilizadas para o acesso a laboratórios e instalações de segurança, onde a segurança exigida justifica custos mais elevados e mesmo um maior grau de desconforto gerado pelas características mais intrusivas destes sistemas. Recentemente, até mesmo a aparentemente alta segurança dos sistemas de autenticação baseados em impressões digitais foi posta em xeque por uma técnica que se utilizou de dedos falsos de gelatina [36] para reproduzir as

impressões digitais. Esta técnica conseguiu enganar a maioria dos sistemas de autenticação de usuários baseados em impressões digitais.

Além das características biométricas físicas, também podem ser utilizadas para a autenticação de usuários características biométricas comportamentais, como gestos, expressões faciais e assinaturas manuscritas. Dentre estas características, a que mais se destaca é a autenticação de usuários baseada em assinaturas manuscritas, devido as diversas vantagens que ela apresenta em relação as outras técnicas. A primeira vantagem que as assinaturas apresentam é a segurança, pois ao contrário das senhas, mesmo que alguém conheça a assinatura de um usuário não é possível que ele consiga reproduzir esta assinatura de forma trivial. Outra vantagem é que os usuários estão acostumados a usar as assinaturas como uma forma de autenticação de transações financeiras, e assim sentem-se mais seguros em relação a sua utilização. O grau de intrusão apresentado pelos sistemas de autenticação de assinaturas também é baixo, e o *hardware* necessário para a autenticação de assinaturas tem um custo bastante acessível, não representando um alto investimento do ponto de vista financeiro.

Apesar de todas as vantagens apresentadas, a autenticação automatizada de assinaturas manuscritas não é uma tarefa trivial de se realizar, pois ao contrário das impressões digitais e do exame do fundo de retina, onde o padrão a ser reconhecido varia muito pouco, as assinaturas manuscritas apresentam uma grande variabilidade, até mesmo entre diferentes assinaturas de um mesmo usuário. Isto impede que técnicas mais simples de comparação visual possam ser utilizadas na autenticação de usuários através de assinaturas. As técnicas que apresentam melhores resultados na autenticação de assinaturas são as técnicas baseadas em Inteligência Artificial e no Aprendizado de Máquinas [14], como por exemplo as Árvores de Decisão [29], os Algoritmos Genéticos [31] e principalmente as Redes Neurais Artificiais [13].

Neste trabalho será estudada a autenticação de assinaturas, analisando alguns sistemas existentes, os diversos modelos existentes, as diversas técnicas possíveis de serem aplicadas a este problema, e em seguida será descrito o modelo proposto de um sistema de autenticação de assinaturas, denominado **Sistema NeuralSignX**. Também será descrito o protótipo do sistema que foi implementado com a finalidade de demonstrar as principais características do modelo e avaliar o seu funcionamento. Este trabalho está organizado da seguinte forma:

Capítulo 2: este capítulo apresenta os diversos conceitos relativos aos sistemas de autenticação de assinaturas, incluindo diversos modelos encontrados na literatura, apresentando também alguns sistemas existentes, e discutindo sobre as etapas que um sistema de autenticação de assinaturas deve possuir;

Capítulo 3: neste capítulo serão analisados alguns conceitos de Inteligência Artificial e Aprendizado de Máquinas, focando a atenção sobre as Redes Neurais Artificiais, e nos fatores que as tornam mais adequadas para a autenticação de assinaturas;

Capítulo 4: este capítulo descreve o sistema proposto, denominado de Sistema NeuralSignX. Nele serão apresentados os diversos módulos do sistema, o funcionamento de cada um e a forma como estes foram implementados no protótipo;

Capítulo 5: este capítulo descreve as diversas simulações realizadas com o objetivo de validar o sistema proposto, bem como os resultados obtidos em cada simulação. Através da apresentação dos resultados descritos neste capítulo, busca-se demonstrar a viabilidade e adequação do uso do Sistema NeuralSignX na solução do problema proposto, que é a autenticação de assinaturas.

2 Reconhecimento de assinaturas

Neste capítulo serão estudados os principais aspectos do reconhecimento de assinaturas, como os vários modelos de reconhecimento possíveis, as vantagens e desvantagens de cada modelo e as aplicações que podem se beneficiar do uso de sistemas de autenticação de assinaturas. Também serão apresentados alguns sistemas de reconhecimento de assinaturas existentes, e por último serão descritas as diversas etapas que um sistema de autenticação de assinaturas deve possuir.

2.1 Conceitos gerais

As assinaturas manuscritas são uma forma alternativa de se garantir a autenticidade dos usuários em sistemas computacionais de forma a minimizar os problemas de segurança que ocorrem devido ao uso de senhas. Elas apresentam diversas vantagens tanto em termos de segurança quanto em termos praticidade e confiança por parte dos usuários, mas a implementação de sistemas que façam o reconhecimento de assinaturas de forma automatizada é uma tarefa bastante complexa do ponto de vista computacional, principalmente devido a grande variabilidade que ocorre entre assinaturas de uma mesma pessoa. O reconhecimento de assinaturas é uma área que abrange diversas atividades, sendo as principais a identificação de assinaturas, a autenticação de assinaturas e a detecção de fraudes [1].

2.1.1 Identificação de assinaturas

Um sistema de identificação de assinaturas é um sistema que recebe uma assinatura e informa o nome do usuário que a desenhou, dentre os diversos usuários cadastrados em uma base de dados de assinaturas. Em sistemas deste tipo não há uma preocupação muito forte em garantir a autenticidade dos usuários, basta apenas que o sistema informe qual o usuário que tem a assinatura mais semelhante a que está sendo analisada. A utilidade de sistemas deste tipo é bastante limitada, pois na maioria dos casos o autor de uma assinatura já é conhecido de antemão, o que faz com que sistemas de identificação de assinaturas sejam pouco úteis na prática.

2.1.2 Autenticação de assinaturas

Sistemas de autenticação de assinaturas são sistemas que recebem uma assinatura e informam se esta assinatura pertence a um determinado usuário ou não. Nestes sistemas não há uma preocupação em identificar o autor da assinatura dentre os diversos usuários cadastrados em uma base de dados, pois o nome do usuário já é conhecido de antemão quando a autenticação de assinaturas é realizada.

2.1.3 Detecção de fraudes

Sistemas de detecção de fraudes são sistemas de autenticação de assinaturas especializados na detecção de fraudes, ou seja, em detectar usuários que estão querendo enganar o sistema. As fraudes podem ser de três tipos [1]:

- Fraudes Aleatórias: são fraudes realizadas por pessoas que desconhecem a forma original de uma assinatura, ou mesmo que conheçam a forma, não possuem muita

habilidade em reproduzi-la. Fraudes deste tipo são muito utilizadas em talões de cheques roubados, por exemplo;

- Fraudes traçadas: são fraudes realizadas por pessoas que possuem o desenho de uma assinatura original em papel e realizam a falsificação traçando com a caneta sobre esta assinatura, de forma que a assinatura resultante seja bastante próxima a original;
- Fraudes especializadas: são fraudes realizadas por usuários que conhecem a forma original de uma assinatura e treinam até que consigam reproduzi-la de forma satisfatória. Quanto mais hábil for o falsificador, mais difícil será a identificação deste tipo de fraude. Também existem tipos de assinaturas que são mais fáceis de serem falsificadas do que outras, como por exemplo as assinaturas em que a pessoa simplesmente escreve o próprio nome sem nenhum estilo especial na caligrafia.

As fraudes aleatórias são muito simples de serem detectadas, mas as fraudes traçadas e as especializadas necessitam de mecanismos especializados de detecção, que busquem indícios que possam levar a conclusão de que a assinatura foi falsificada.

Neste trabalho será enfocada a autenticação de assinaturas, sem uma maior preocupação com a detecção de fraudes ou com a identificação de assinaturas. Em trabalhos futuros estes itens poderão ser explorados com maior profundidade.

2.2 Formas de autenticação:

A autenticação de assinaturas pode ser realizada de duas formas no que se refere a obtenção dos dados, que são as formas *off-line* e *on-line*.

2.2.1 Off-line

Na autenticação *off-line* de assinaturas, o usuário faz a assinatura originalmente em papel. Esta assinatura é posteriormente digitalizada e submetida ao sistema que faz a autenticação a partir da imagem resultante do processo de digitalização. O equipamento de *hardware* necessário para este tipo de autenticação pode ser um *scanner* ou outro equipamento similar.

2.2.2 On-line

Na autenticação *on-line* de assinaturas, o usuário faz a assinatura sobre um dispositivo de *hardware* especial, conhecido como *tablet*, e este dispositivo envia para o computador os dados relativos a assinatura diretamente no formato digital, sem a necessidade de papel. Um *tablet* é um equipamento composto de uma superfície sensível e uma caneta especial, com a qual o usuário pode desenhar as suas assinaturas. A figura 2.1 mostra um exemplo de *tablet* que pode ser utilizado na autenticação *on-line* de assinaturas.



Figura 2.1 - Exemplo de um *tablet* [9]

Além do *tablet*, outros equipamentos de *hardware* similares podem ser utilizados, como por exemplo os computadores do tipo *Handheld*, que possuem uma tela de cristal líquido sensível e uma caneta especial, com a qual os usuários podem desenhar as assinaturas.

Em relação formas de obtenção dos dados, ambas apresentam vantagens e desvantagens, como será visto a seguir. A principal vantagem dos sistemas *off-line* é que eles podem ser aplicados na autenticação de assinaturas de documentos impressos e de cheques bancários, o que não pode ser realizado através do reconhecimento *on-line*. Já os sistemas *on-line*, por não necessitarem de papel, são muito mais práticos e mais seguros de serem utilizados em transações eletrônicas em tempo real. Outras vantagens que os sistemas de reconhecimento *on-line* apresentam em relação ao reconhecimento *off-line* são:

- Maior riqueza de informações: um sistema *on-line* de assinaturas disponibiliza não apenas a imagem final da assinatura, mas também diversas informações como a trajetória que a caneta realizou ao longo do tempo, a velocidade, a pressão da caneta e o tempo de duração da assinatura, e estas informações adicionais são muito úteis principalmente para a detecção de fraudes;
- Segurança: uma assinatura em papel pode ser facilmente utilizada para a realização de fraudes traçadas ou especializadas, e também pode ser digitalizada a partir de um documento antigo, o que não garante a autenticidade de uma transação específica. Já uma assinatura desenhada sobre um *tablet* pode inclusive ser mantida em sigilo, o que torna o sistema ainda mais seguro;
- Pureza da imagem: um sistema de reconhecimento *off-line* necessita obter a partir de uma imagem digitalizada os pontos que pertencem a assinatura e os pontos que pertencem ao fundo do papel, o que exige técnicas especiais de tratamento de imagens, e devido a variações de tonalidade da tinta da caneta ou impurezas provenientes do processo de digitalização, a imagem resultante pode ficar um pouco distorcida. Já no reconhecimento *on-line* não é necessário nenhuma técnica de tratamento de imagens para a obtenção dos pontos que pertencem a assinatura, pois estes já são informados diretamente pelo *tablet*.
- Baixo custo do equipamento: um *tablet* é um dispositivo de *hardware* que está se disseminando rapidamente e possui um baixo custo de aquisição, que é inferior ao de um *scanner* de mesa de média resolução.

Neste trabalho optou-se por utilizar o reconhecimento *on-line* de assinaturas devido as diversas vantagens que ele apresenta em relação ao reconhecimento *off-line*, e também por ser um método mais adequado e prático de ser utilizado em transações eletrônicas. No próximo capítulo serão apresentados diversos tipos sistemas que poderiam se beneficiar do uso de assinaturas como uma forma de autenticação de usuários.

2.3 Aplicações possíveis

Os primeiros sistemas de autenticação de assinaturas pesquisados e desenvolvidos utilizavam a metodologia *off-line*, pois tinham como foco principal a autenticação de assinaturas em cheques bancários, que até bem pouco tempo atrás representava a principal forma de se realizar transações financeiras. Mas com o advento de novas tecnologias, como por exemplo os cartões de crédito, os caixas eletrônicos, e mais recentemente os serviços de *home-banking* e de transferência eletrônica de fundos, os cheques perderam um pouco da sua importância, e estão sendo cada vez menos

utilizados em transações financeiras. Nas novas tecnologias, a principal forma de autenticação dos usuários não são mais as assinaturas, mas sim senhas numéricas ou alfanuméricas que os usuários necessitam decorar e manter a salvo de outras pessoas, sob pena de terem as suas economias furtadas por usuários mal intencionados.

Para se garantir uma maior segurança nas transações eletrônicas, sistemas de autenticação *on-line* de assinaturas poderiam ser utilizados, de forma que bastaria aos usuários assinarem em um *tablet* ou dispositivo de *hardware* similar para que suas transações fossem efetivadas, sem a necessidade do uso de senhas. Nas próximas seções serão descritos vários sistemas que poderiam se beneficiar da autenticação automatizada de assinaturas.

2.3.1 Caixas eletrônicos

O advento dos caixas eletrônicos representou um avanço significativo ao sistema financeiro, pois os bancos puderam reduzir seus custos e os clientes ganharam uma maior comodidade, podendo efetuar saques em qualquer horário do dia ou da noite, inclusive em finais de semana. Mas junto com os benefícios vieram os problemas de segurança e os golpes financeiros, principalmente contra usuários que escolhem senhas fáceis de serem descobertas ou que acabam revelando as senhas por descuido e até mesmo por ingenuidade. Um mecanismo de autenticação *on-line* de assinaturas poderia ser utilizado neste caso, bastando para as instituições financeiras adaptarem aos caixas eletrônicos dispositivos de *hardware* similares a um *tablet* e utilizarem sistemas de autenticação de assinaturas para validar as transações financeiras.

2.3.2 Cartões de crédito [5]

Os cartões de créditos são utilizados nos países desenvolvidos há várias décadas, e no Brasil seu uso está cada vez mais disseminado. Mas problemas de segurança que surgiram principalmente com o advento da *Internet* e do comércio eletrônico fizeram com que muitas pessoas perdessem a confiança neste tipo tecnologia, o que faz com que muitas delas se recusem a utilizar os cartões de crédito em transações *on-line*. Um sistema de autenticação de assinaturas aumentaria a segurança dos cartões de crédito, fazendo com que novos usuários viessem a aderir a esta tecnologia devido a elevação do nível de confiança que eles sentiriam pela tecnologia. De fato, a maioria das pessoas considera o uso de assinaturas uma forma extremamente segura de validar transações financeiras, o que afastaria o temor que muitos tem de serem lesados em golpes aplicados por *hackers* e outros criminosos digitais.

2.3.3 Serviços de *home-banking*

Da mesma forma que com os cartões de crédito, as assinaturas poderiam ser utilizadas para validar transferências eletrônicas de fundos, bastando que o usuário tivesse um *tablet* instalado em seu computador para desenhar as assinaturas. A disseminação do uso de computadores do tipo *Handheld*, que possuem um tela sensível ao toque de uma caneta especial, poderia fazer com que muitas pessoas acessassem os dados de suas contas bancárias e fizessem transferências monetárias diretamente de seus *Handhelds*, bastando para isto assinar sobre a tela sensível, sem a necessidade do uso de senhas ou de outro mecanismo similar.

2.3.4 Sistemas de gestão

Sistemas de gestão empresarial também poderiam se beneficiar do uso de sistemas de autenticação de assinaturas, principalmente para o acesso a dados altamente confidenciais. Atualmente ocorrem muitos casos de empresas que tem seus sistemas

invadidos por *hackers* que descobriram as senhas de algum usuário e com isto acabaram tendo acesso a informações confidenciais da empresa.

2.3.5 Acesso a instalações de segurança

Em muitos laboratórios de alta tecnologia, o acesso às instalações é controlado através do uso de um cartão magnético em conjunto com uma senha numérica para cada usuário. Se as senhas fossem substituídas por assinaturas *on-line*, a segurança destas instalações seria reforçada, permitindo inclusive que se realizassem auditorias mais apuradas quando a segurança fosse violada.

A partir destes exemplos pode-se concluir que a autenticação *on-line* de assinaturas apresenta diversas vantagens em relação as outras formas de autenticação de usuários, dentre as quais podemos destacar:

- A maioria das pessoas já está habituada e sente confiança em utilizar as assinaturas como uma forma de garantir a autenticidade de transações financeiras, e isto pode fazer com que elas tenham menos resistência em aderir as novas tecnologias como o comércio eletrônico;
- Com o uso de assinaturas as pessoas não precisam decorar uma seqüência complexa de letras e números, o que faz com que muitos problemas de segurança que ocorrem atualmente sejam evitados. De fato, grande parte das fraudes eletrônicas ocorrem porque os usuários acabam anotando as senhas em algum lugar com medo de esquecer ou acabam utilizando senhas simples e óbvias, que são muito fáceis de serem descobertas por *hackers*;
- Assinaturas são muito mais seguras que outras técnicas de autenticação pelo fato serem muito difíceis de serem forjadas. Um usuário que quiser se passar por outro usuário precisará conhecer a assinatura e treinar bastante até conseguir realizar uma falsificação de forma satisfatória. Se o sistema de autenticação utilizado disponibilizar informações temporais, como a trajetória e a velocidade da caneta, fica ainda mais difícil de se forjar uma assinatura, pois além de parecer com a assinatura original, a trajetória que a caneta percorre deve ser a mesma e a velocidade também deve ser muito próxima da velocidade original, o que dificulta a falsificação, pois um falsário geralmente precisa escrever mais devagar para conseguir reproduzir uma assinatura;
- Ao contrário das senhas, as assinaturas não podem ser transmitida a outras pessoas de forma trivial, o que reduz o número de golpes que ocorrem devido a ingenuidade de usuários que confiam suas senhas a outras pessoas.

Apesar de todas estas vantagens, as assinaturas são muito pouco utilizadas como mecanismo de autenticação de transações eletrônicas, devido a dificuldade de se implementar sistemas que façam a autenticação de assinaturas de forma satisfatória. Na próxima seção serão apresentados os principais trabalhos desenvolvidos na área de reconhecimento de assinaturas, analisando um pouco da metodologia utilizada e os resultados obtidos por cada um deles.

2.4 Sistemas de Reconhecimento de Assinaturas

Os sistemas de reconhecimento de assinaturas fazem parte da área de reconhecimento automatizado de caracteres e textos manuscritos [1], mostrada na figura 2.2. A área de reconhecimento de manuscritos derivou da área de reconhecimento de caracteres impressos. Nesta área, OSÓRIO [16] desenvolveu em sua dissertação de mestrado de

1991 um estudo sobre o reconhecimento visual de caracteres e propôs uma aplicação que utilizava Redes Neurais do tipo Adaline para o reconhecimento de caracteres tipografados. Os resultados obtidos foram bastante significativos, atingindo taxas de reconhecimento de 90 a 100%.



Figura 2.2 - Subdivisão dos tipos de reconhecimento de escrita

Na área de reconhecimento de textos manuscritos, XIAOLIN e PLAMONDON [17] estudaram o reconhecimento *on-line* de dígitos numéricos manuscritos baseado no uso de modelos de Markov (*hidden Markov models* – HMM), e obtiveram taxas de reconhecimento na ordem de 89 a 92%. Também na área de reconhecimento de textos manuscritos, SUEN ET. AL. [7] discutiram a viabilidade da implementação de sistemas de reconhecimento de datas manuscritas em cheques bancários, propondo um modelo que obteve taxas de 83% de acerto em seus experimentos. Outro trabalho importante na área de reconhecimento de textos manuscritos é o de WIENECKE ET AL. [20], que propôs um sistema de reconhecimento *on-line* de textos manuscritos, que utilizava para a obtenção dos dados de entrada um *tablet* em conjunto com uma câmera de vídeo que filmava a mão do usuário no momento da escrita. As taxas de acerto obtidas por este sistema foram em torno de 75%.

Na área de identificação de assinaturas, HAN ET AL. [2], estudou a identificação *off-line* de assinaturas através da extração de atributos visuais, como o número de laços fechados e de cruzamentos, e a utilização de tabelas *hash* para o processo de classificação. Os resultados obtidos foram da ordem de 90%.

Na área de autenticação de assinaturas, os primeiros trabalhos desenvolvidos utilizavam apenas a forma *off-line* para a obtenção dos dados. ABAS [1], em sua dissertação de mestrado, propôs um sistema que se baseava no uso de Redes Neurais do tipo *Backpropagation* e analisava uma assinatura em uma matriz de 160 x 160 pontos, onde cada ponto representava uma entrada na Rede Neural. Este tipo de análise é conhecido como comparação visual, e necessita que ajustes visuais de posição e escala sejam realizados anteriormente para que os resultados obtidos sejam satisfatórios. O sistema foi testado com uma base de dados de 480 assinaturas, e obteve taxas de 92% de acerto quando submetido apenas a assinaturas autênticas, mas quando foram utilizadas falsificações, o sistema obteve uma taxa de até 20% de aceites indevidos quando submetido a falsificações especializadas e de 10 a 40% quando submetido a falsificações traçadas. Um aceite indevido é quando uma assinatura que não pertence a determinado usuário é classificada como se fosse dele, e uma rejeição indevida é quando uma assinatura que pertence ao usuário é classificada como se não fosse dele. Em sistemas de autenticação *on-line* de assinaturas, é mais importante que a taxa de aceites indevidos seja baixa, pois é preferível pedir ao usuário que assine novamente quando uma assinatura verdadeira for rejeitada indevidamente do que aceitar uma assinatura falsificada como se fosse verdadeira.

Outros trabalhos na área de autenticação *off-line* de assinaturas que podem ser destacados são os de DROUHARD ET AL. [3], que utilizaram funções direcionais probabilísticas de densidade para a extração de atributos e Redes Neurais do tipo *Backpropagation* para a classificação, obtendo taxas de generalização na ordem 95% para uma base de dados de 800 assinaturas de 20 usuários diferentes (40 assinaturas por usuário). Também na área de autenticação *off-line*, HUANG e YAN [4] utilizaram a extração de atributos geométricos das assinaturas e Redes Neurais do tipo *Backpropagation* para a classificação, obtendo taxas de 90% de classificações corretas. BALTZAKIS e PAPAMARKOS [10] utilizaram um classificador de dois estágios baseado em Redes Neurais, onde o primeiro estágio era composto de três classificadores Neurais, e o segundo estágio recebia as saídas do primeiro estágio e emitia o resultado final da classificação. Somando todos os estágios o sistema trabalhava com 160 entradas na Rede Neural. Os resultados obtidos com este sistema foram em torno de 80,81% de generalização com uma base de dados de 500 assinaturas.

Pode ser notado que a boa parte dos trabalhos realizados na área de autenticação de assinaturas são baseados no uso de Redes Neurais. Dentre os trabalhos de autenticação *off-line* de assinaturas que não utilizam Redes Neurais, podemos destacar HERBST e COETZER [6], que utilizou algoritmos de programação dinâmica e transformações de Radon para o processo de classificação e obteve taxas de erro de 10% quando submetido apenas a assinaturas originais e 23% quando submetido a falsificações. Outro trabalho importante nesta área foi o de HUANG e YAN [11], que utilizou um método de extração da forma estrutural da assinatura e comparação estatística para o processo de autenticação *off-line* de assinaturas, obtendo taxas de 6,3% de rejeições indevidas e 8,2% de aceites indevidos.

No reconhecimento *on-line* de assinaturas, GUPTA e MCCABE [5] fizeram um extenso estudo sobre a autenticação de assinaturas utilizando as mais variadas técnicas e os mais variados métodos de extração de atributos. Na autenticação *on-line* de assinaturas usando modelos estatísticos de classificação, temos trabalhos como o de JAIN ET AL. [12], que utilizou a extração da estrutura básica da assinatura e a aplicação de funções de dissimilaridade para o processo de autenticação, obtendo resultados da ordem de 3,3% de rejeições indevidas e 2,7% de aceites indevidos. Também na área de classificação baseada em métodos estatísticos, PACUT e CZAJKA [21] trabalharam na autenticação *on-line* de assinaturas utilizando funções matemáticas para a classificação, e obtiveram resultados da ordem de 95,8% de acerto. SAKAMOTO ET AL. [22] utilizaram atributos dinâmicos como a posição da caneta, a pressão e a inclinação no processo de autenticação de assinaturas. Em seus experimentos eles utilizaram uma base de dados de 861 assinaturas autênticas, 1921 assinaturas falsificadas e 205 assinaturas geradas automaticamente a partir das originais. Os resultados obtidos foram de 97% de acerto. É importante ressaltar que a maioria das assinaturas da base de dados utilizavam caracteres japoneses do alfabeto Kanji, o que pode fazer com que o sistema não apresente os mesmos resultados quando submetido apenas a assinaturas ocidentais.

Também na área de autenticação *on-line* de assinaturas, mas utilizando métodos de Inteligência Artificial e Aprendizado de Máquinas, temos o trabalho de WESSELS e OMLIN [8], que desenvolveu um sistema híbrido baseado em modelos de Markov e mapas de Kohonen, obtendo taxas de acerto de 72% quando utilizado para a detecção de fraudes. Outro sistema nesta linha é o proposto por DARIUSZ ET AL. [23], que utiliza *wavelets* (funções ortogonais capazes de cortar os dados em diferentes frequências componentes) e Redes Neurais do tipo *Backpropagation* para o desenvolvimento de um sistema *on-line* de autenticação de assinaturas. As taxas de erro obtidas em suas

simulações foram próximas a 1%, com uma base de dados de 922 assinaturas originais, sendo a maioria delas escritas utilizando o alfabeto Kanji.

Pode ser concluído através deste estudo que os primeiros sistemas desenvolvidos utilizavam o reconhecimento *off-line* de assinaturas, mas a tendência atual é o crescimento de trabalhos utilizando o reconhecimento *on-line*, principalmente devido ao aumento do uso de transações eletrônicas a nível mundial. Também pode ser concluído que grande parte dos sistemas apresentados utilizava Redes Neurais para o processo de classificação, e a extração de atributos temporais, geométricos, visuais e dinâmicos é muito utilizada como forma de pré processar as assinaturas antes da etapa de classificação. Por esses motivos, o trabalho atual tem enfoque na área de autenticação *on-line* de assinaturas, utilizando a extração de atributos como uma forma de pré processamento e Redes Neurais para o processo de classificação. Na próxima seção serão apresentadas as diversas etapas que um sistema de autenticação de assinaturas deve possuir.

2.5 Etapas da autenticação de assinaturas

O processo de autenticação de assinaturas envolve diversas etapas, que podem variar de acordo com o modelo a ser implementado, mas basicamente todo sistema de autenticação de assinaturas necessita de uma etapa de obtenção e armazenamento das assinaturas, uma etapa de pré-processamento e uma etapa de processamento, como mostra a figura 2.3.

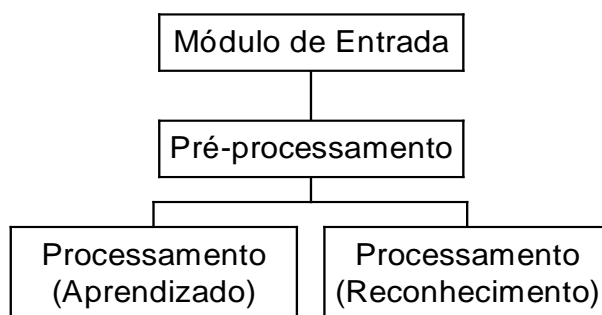


Figura 2.3 - Etapas de um sistema de autenticação de assinaturas

2.5.1 Obtenção e armazenamento das assinaturas

Esta etapa compreende a leitura dos dados provenientes do dispositivo de *hardware* utilizado para a coleta de assinaturas e o armazenamento destes dados de forma estruturada em um arquivo ou tabela de banco de dados, para que possam ser acessados posteriormente.

2.5.2 Pré processamento

Antes de um sistema de autenticação de assinaturas poder realizar a classificação das assinaturas recebidas entre verdadeiras e falsas, são necessárias algumas atividades para preparar os dados antes de serem submetidos a etapa de classificação. As atividades mais importantes a serem realizadas são o ajuste de posição, o ajuste de escala e a extração dos atributos. O ajuste de posição é necessário para eliminar as variações que ocorrem no posicionamento das assinaturas de um mesmo usuário, e o ajuste de escala serve para minimizar as variações de tamanho que podem ocorrer entre diversas assinaturas de uma mesma pessoa. Mas a tarefa mais importante a ser realizada na etapa de pré-processamento é a extração de atributos, que serve para obter a partir das

assinaturas informações que sirvam para diferenciar as assinaturas de um usuário das assinaturas dos demais usuários. Exemplos de atributos que podem ser extraídos são a velocidade de deslocamento da caneta, o número de laços fechados, quantas vezes a caneta foi levantada da superfície de escrita, entre muitos outros. A extração de atributos se faz necessária porque a comparação visual entre as assinaturas coletadas não é suficiente para garantir a autenticidade das mesmas, devido a grande variabilidade que ocorre entre assinaturas de uma mesma pessoa. Assim, torna-se necessária a obtenção de atributos que tenham a menor variação possível entre assinaturas de um mesmo usuário e variações maiores entre assinaturas de diferentes usuários.

2.5.3 Processamento

Esta etapa realiza a classificação das assinaturas, ou seja, ela determina se uma determinada assinatura pertence a um determinado usuário ou não. Em sistemas de autenticação de assinaturas baseados em técnicas de Aprendizado de Máquinas, a etapa de processamento tem duas fases distintas, uma fase de aprendizado, onde uma base de dados é submetida ao sistema para que ele aprenda a classificar corretamente as assinaturas, e uma fase de reconhecimento, onde uma assinatura é submetida ao sistema e ele responde dizendo se ela pertence ao usuário informado ou não. A etapa de processamento pode utilizar diversas técnicas de classificação, que vão desde análises matemáticas e estatísticas até sistemas de Inteligência Artificial e Aprendizado de Máquinas. As técnicas que costumam apresentar melhores resultados na classificação de assinaturas são as técnicas baseadas em Aprendizado de Máquinas, principalmente as Redes Neurais.

No decorrer deste capítulo foi constatado que a autenticação de assinaturas é uma tarefa complexa de reconhecimento de padrões que merece ser mais estudada e aperfeiçoada. O reconhecimento de padrões exige técnicas computacionais avançadas, como por exemplo as Redes Neurais, para que sejam obtidos resultados satisfatórios. No próximo capítulo serão estudadas as redes Neurais, destacando as propriedades e características que as tornam mais adequadas ao reconhecimento de padrões e a autenticação de assinaturas.

3 Redes Neurais Artificiais

Neste capítulo serão estudadas as Redes Neurais Artificiais (RNA) e como elas podem ser aplicadas no reconhecimento de padrões e na autenticação de assinaturas. Inicialmente será descrito o reconhecimento de padrões, em seguida serão descritas algumas técnicas de Inteligência Artificial e Aprendizado de Máquinas, e depois serão apresentadas as Redes Neurais Artificiais, detalhando o seu funcionamento, as principais características e alguns modelos importantes, como o *Perceptron*, o *Backpropagation* e o *Cascade-Correlation*. Em seguida serão analisadas as características das Redes Neurais que as tornam úteis para o processo de autenticação de assinaturas, e ao final será descrito o funcionamento do simulador de Redes Neurais Neusim, que é o simulador escolhido para ser utilizado neste trabalho.

3.1 Reconhecimento de Padrões

Segundo HALL [25], o reconhecimento de padrões consiste em perceber ou identificar um elemento a partir de uma experiência passada ou conhecimento, ou por algum detalhe deste elemento. O elemento a ser reconhecido é denominado de padrão, o qual é definido como sendo uma amostra representativa de uma classe ou tipo. O objetivo do reconhecimento de padrões é o de extrair, detectar e identificar elementos presentes em uma cena, para que se possa realizar atividades como a classificação de objetos, controle de qualidade, compactação de dados e uma série de outras aplicações científicas e comerciais [16].

O reconhecimento de padrões abrange uma gama muito extensa de aplicações e objetivos, estando dividido em duas principais áreas de acordo com o tipo de objetivo a ser alcançado. A primeira área é a obtenção de descrições, e a segunda área é a obtenção de classificações a partir dos padrões analisados.

- Obtenção de descrições: nesta categoria se encontram os procedimentos que visam a detecção de algum tipo de elemento em uma cena, não havendo uma preocupação em classificar estes elementos em categorias. A obtenção de descrições pode ser utilizada para a detecção de vetores em uma imagem, para a segmentação e para a detecção de elementos;
- Obtenção de classificações: esta forma de reconhecimento consiste da atribuição de um dado padrão a um dos conjuntos de classes pré-definidas por algum método [26]. Este reconhecimento é obtido através de um processo de abstração, onde descarta-se os detalhes não essenciais e destaca-se as propriedades relevantes, que são consideradas as características definidoras da classe [16]. Esta abstração leva a identificação de uma classe de elementos que possuam um conjunto de propriedades essenciais comuns a seus componentes.

O reconhecimento de padrões é uma tarefa extremamente complexa do ponto de vista computacional, e exige que sejam utilizadas técnicas computacionais sofisticadas para que se obtenham bons resultados. As técnicas que geralmente apresentam os melhores resultados no reconhecimento de padrões são as técnicas baseadas em Inteligência Artificial e Aprendizado de Máquinas.

3.2 Inteligência Artificial e Aprendizado de Máquinas

A Inteligência Artificial é uma área de estudos da computação que se interessa pelo estudo e criação de sistemas que possam exibir um comportamento inteligente e realizar tarefas complexas com um nível de competência que é equivalente ou superior ao de um especialista humano [27]. As primeiras ferramentas de I. A. desenvolvidas adquiriam os conhecimentos que eram explicitados pelos especialistas de uma determinada área, o que era similar a programar um sistema computacional para resolver um problema. Posteriormente, mecanismos de aquisição automática de conhecimentos foram acrescentados a estas ferramentas, de onde surgiram a linguagem de programação PROLOG e os sistemas especialistas de 2º geração. Estes mecanismos de aquisição automática de conhecimentos são conhecidos como técnicas de Aprendizado de Máquinas (*Machine Learning*) [14]. As técnicas de Aprendizado de Máquinas mais importantes são:

- Aprendizado por analogia ou por instâncias. Exemplo: Sistemas baseados em casos - CBR (*Case Based Reasoning*) [14],[28];
- Aprendizado por Indução. Exemplos: Árvores de Decisão - ID3, C4.5, CN2 (IDT – *Induction of Decision Trees*) [29], e ILP - *Inductive Logic Programming* (Prolog) [30];
- Aprendizado por evolução/seleção. Exemplo: Algoritmos Genéticos - GA e GP (*Genetic Algorithms / Genetic Programming*) [14],[31];
- Outros tipos de aprendizado: por reforço (*Reinforcement Learning*), não supervisionado, Bayesiano e por explicações (*Explanation Based*) [14],[30].

Grande parte destas ferramentas de Aprendizado de Máquinas possuem fortes limitações, pois usualmente assumem que os conhecimentos a serem adquiridos e as informações disponíveis estão corretos e devem estar completos em relação ao problema, o que dificilmente ocorre, e são orientadas para a manipulação de informações qualitativas, onde informações quantitativas representam um problema difícil de ser tratado [32].

Para se resolver estas limitações, muitos pesquisadores começaram a desenvolver ferramentas e sistemas inteligentes baseados na forma como o cérebro humano funciona, o que deu origem aos estudos que levaram ao desenvolvimento das Redes Neurais Artificiais [13].

3.3 Redes Neurais Artificiais

As Redes Neurais Artificiais surgiram como uma tentativa de reproduzir o funcionamento do cérebro humano, e assim se desenvolver máquinas capazes de realizar muitas funções que antes só eram possíveis de serem realizadas através da intervenção humana. As Redes Neurais possuem a capacidade de aprendizado a partir de exemplos e a capacidade de generalização, de forma que a partir dos exemplos analisados elas conseguem extrair as regras gerais que descrevem um problema e assim podem aplicar esta regra para a solução de novos exemplos não analisados anteriormente [16]. Outras vantagens que as Redes Neurais apresentam em relação as demais técnicas de Inteligência Artificial são o fato de serem tolerantes a dados incorretos ou incompletos e também de poderem lidar tanto com informações quantitativas quanto com informações qualitativas.

As Redes Neurais foram originalmente desenvolvidas baseadas nos estudos realizados sobre a forma como o conhecimento é armazenado no cérebro humano e a forma como ocorre o aprendizado. Nestes estudos foi constatado que nos seres humanos o conhecimento é armazenado nas ligações que os neurônios realizam uns com os outros, chamadas de sinapse, e a medida que o aprendizado ocorre mais ligações vão se formando e se fortalecendo entre estes neurônios. Outra característica percebida é que as ligações existentes entre os neurônios tendem a se fortalecer quando estes neurônios costumam ser ativados em conjunto. [32]

Para que seja possível compreender o funcionamento das Redes Neurais Artificiais, é necessário que se compreenda um pouco do funcionamento do cérebro humano. O núcleo do sistema nervoso é composto por milhões de células que são responsáveis pela geração, recepção e transmissão dos impulsos elétricos, conhecidas como neurônios. A figura 3.1 mostra o desenho de um neurônio biológico humano.

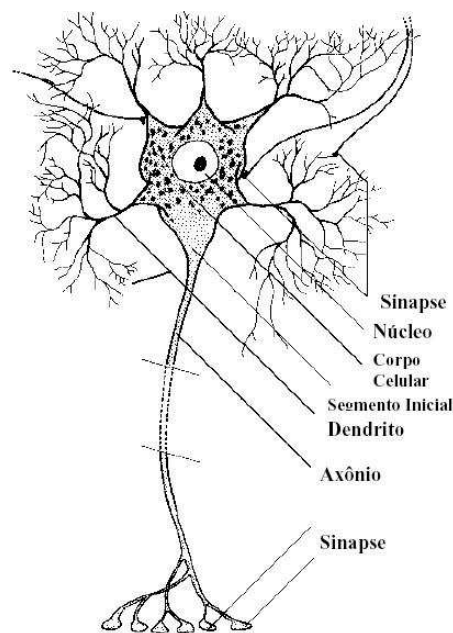


Figura 3.1 - Neurônio biológico humano [32]

Um neurônio biológico é formado por três partes principais, que são uma árvore dendrítica, um corpo celular e um prolongamento fino e longo, chamado de axônio. Os dendritos são as entradas pelas quais o neurônio recebe os impulsos elétricos provenientes de outros neurônios, e o axônio é o meio por onde os sinais elétricos saem do neurônio em direção aos outros neurônios. A ligação existente entre dois neurônios, conhecida como sinapse, é que irá influenciar sobre a transmissão do sinal recebido, podendo atuar de forma a atenuar ou amplificar os impulsos elétricos recebidos [16].

Baseado no funcionamento do neurônio biológico, foi desenvolvido o neurônio formal, que é uma simplificação matemática que tenta reproduzir as principais características dos neurônios humanos referentes a forma em que se processa a aquisição de conhecimentos. A figura 3.2 mostra um neurônio formal utilizado nas Redes Neurais Artificiais, mais conhecido como neurônio artificial.

Assim como um neurônio biológico, o neurônio artificial é constituído de entradas, que recebem os sinais provenientes do exterior, uma função de ativação, que realiza a combinação dos valores obtidos nas entradas, e saídas, que transmitem os sinais recebidos e processados pelo neurônio para o exterior. Associado a cada entrada de um

neurônio artificial existem pesos, que são valores numéricos que representam a força das conexões existentes entre os neurônios. Se um peso tiver um valor elevado, a sua respectiva entrada será amplificada, e se ele tiver um valor próximo de 0, a sua respectiva entrada terá seus valores atenuados [38].

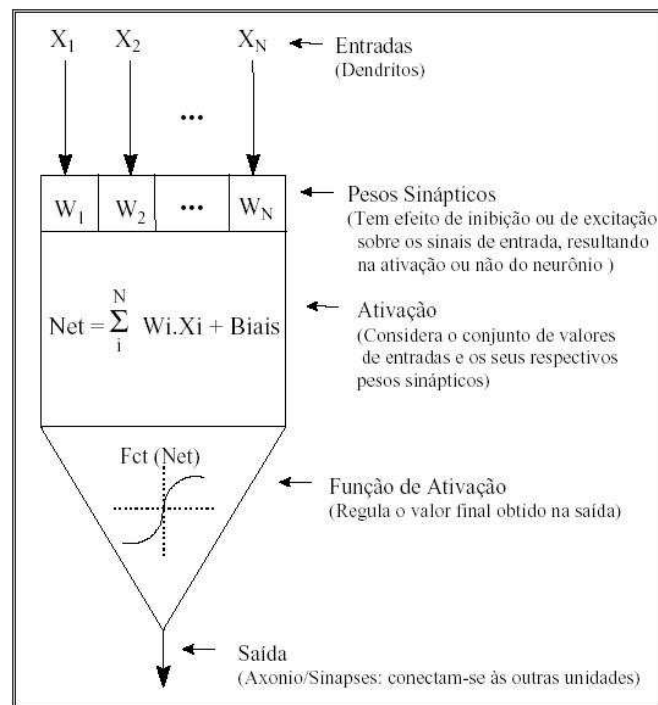


Figura 3.2 - Esquema de um neurônio artificial [32]

A ativação de um neurônio realiza o somatório de todas as entradas multiplicadas por seus respectivos pesos, mais um valor especial, chamado de bias, que é uma entrada que sempre recebe o valor 1 e possui um peso a ela associado que pode ser ajustado, assim como os pesos de todas as demais entradas. O resultado obtido no somatório é enviado para uma função de transferência, que serve para normalizar o valor de saída do neurônio. A função de transferência mais utilizada é a Sigmoide, cujo gráfico é mostrado na figura 3.3. A Sigmoide faz com que a saída de um neurônio seja normalizada entre 0 e 1, e devido a sua curvatura especial ela é usada no processo de aprendizado para gerar uma certa estabilidade no sistema depois que o valor da saída do neurônio se encontra próximo a um dos extremos. O valor obtido na função de transferência é então transmitido para a saída do neurônio.

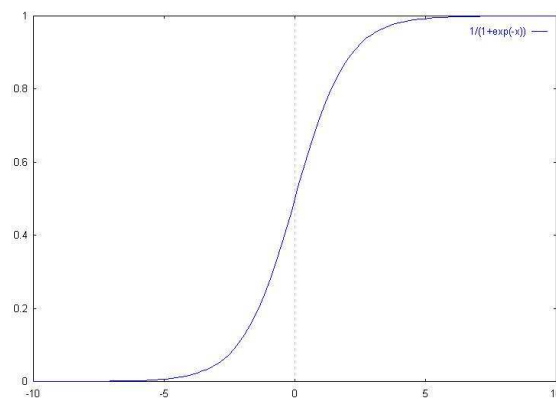


Figura 3.3 - Gráfico da função de transferência Sigmoide

Em uma Rede Neural, o conhecimento não é armazenado na forma de tabelas ou listas, como ocorre na maioria dos sistemas de informação, mas sim na forma de pesos sinápticos, ou melhor, na topologia e no valor dos pesos sinápticos dos neurônios que compõem uma Rede Neural. Inicialmente todos os pesos de uma Rede Neural são inicializados de forma aleatória, e a medida que o aprendizado se processa os pesos vão sendo ajustados de forma a fazer com que a Rede Neural responda na saída com os valores desejados para determinadas entradas. Devido a esta forma de armazenamento das informações, fica difícil de se extrair os conhecimentos de uma Rede Neural, ou seja, se sabe que ela aprendeu determinado problema e que ela consegue responder a ele de forma satisfatória, mas não é possível de se obter de forma direta quais foram as regras que a Rede Neural utilizou para tomar uma determinada decisão.

O aprendizado em uma Rede Neural se processa através de exemplos, o que significa que para realizar o treinamento de uma Rede Neural é necessário que se tenha uma base de dados com diversos exemplos para o problema a ser analisado. Quanto maior for a base de dados, melhores serão os resultados obtidos no treinamento em termos de generalização, mas não basta que uma base de dados seja extensa, ela precisa ser representativa. O aprendizado em uma Rede Neural pode ser do tipo supervisionado ou não supervisionado. No aprendizado supervisionado, são fornecidas as saídas desejadas para cada um dos exemplos da base de dados, que são os valores que se deseja que a Rede Neural responda quando encontrar exemplos do mesmo tipo. No aprendizado não supervisionado, não existe a necessidade de informar o valor desejado para as saídas, e a classificação ocorre por similaridade entre os exemplos de entrada, não havendo um controle muito grande sobre as classes de saída que serão obtidas. Neste trabalho será utilizado o aprendizado supervisionado, pois ele é mais adequado para o tipo de problema que está sendo abordado.

3.3.1 *Perceptron*

Um dos primeiros modelos de Redes Neurais desenvolvidos foi o *Perceptron* [33], que serviu de base para o desenvolvimento de algoritmos mais poderosos, como por exemplo o *Backpropagation* [15]. O *Perceptron* trabalha da seguinte forma: quando um neurônio artificial é ativado, ele recebe os valores obtidos nas entradas, realiza a soma ponderada, envia o resultado para a função de transferência, e assim se obtém o valor de saída. O valor de saída obtido é então comparado com a saída desejada, e o erro na saída é calculado diminuindo o valor da saída obtida pelo valor da saída desejada. Este erro é utilizado para o cálculo do ajuste dos pesos, dado pela fórmula:

$$\text{Peso_Novo}(i) = \text{Peso_Antigo}(i) + \frac{\beta * \text{Erro}(i) * \text{Entrada}(i)}{|\text{Entrada}(i)|}$$

onde *Entrada* é o valor recebido na entrada, *Erro* é o erro calculado na saída e β é o passo, que é um valor entre 0 e 1 que determina a velocidade do processo de aprendizado. Quando uma Rede Neural está sendo treinada, existe uma curva de erro a ela associada, como por exemplo a curva mostrada na figura 3.4. Esta figura representa que para certos valores de pesos o erro é menor e para outros é maior, e conforme se altera os valores dos pesos para cima ou para baixo o erro vai sendo alterado de acordo com a curva de erro do problema que está sendo analisado. Cada problema tem a sua curva de erro própria, sendo que a curva da figura 3.4 é apenas um exemplo ilustrativo.

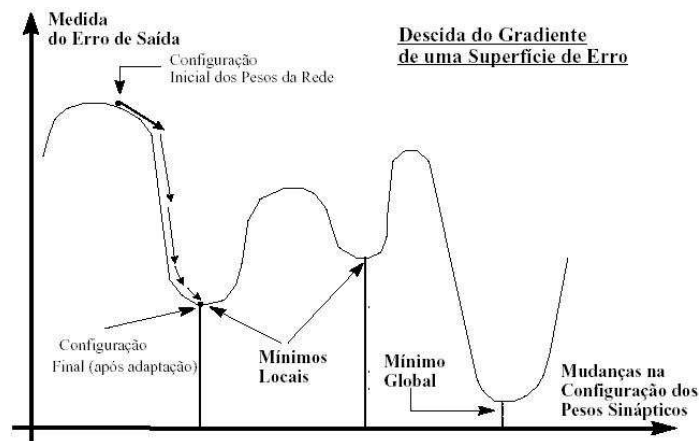


Figura 3.4 - Descida do gradiente de uma superfície de erro

Na figura 3.4 pode-se notar que existem espécies de “vales” na curva onde o erro é mais baixo que nos pontos vizinhos. Como os pesos são inicializados de forma aleatória, o erro inicial se situa em algum ponto da curva. A medida que os pesos vão sendo ajustados, vai se descendo na curva de erro, até que se atinja um ponto mínimo, de onde qualquer mudança que seja feita para baixo ou para cima fará com que o erro seja aumentado. Em uma curva de erro existe um mínimo global, que é o ponto onde ocorre o menor erro possível, e vários mínimos locais, que são mostradas na figura 3.4. O mínimo global é muito difícil de ser atingido, e mesmo que já se esteja nele não será possível de saber, pois é difícil conhecer o formato de toda a curva de erro do problema que está sendo analisado.

Um simulador que implemente uma Rede Neural a base de *Perceptrons* possui diversos parâmetros, sendo que os dois principais são o passo e o *momentum*. O passo é a velocidade com que se avança na curva de erro. Se for utilizado um passo muito pequeno, o aprendizado será muito lento, e se for utilizado um passo muito grande, pode-se acabar pulando os mínimos locais e o global e assim pode não ser possível atingir bons resultados no aprendizado. O *momentum* é um parâmetro que tem a função de evitar que o aprendizado pare em pequenas saliências da curva de erro, que podem impedir que se atinja o mínimo global ou algum mínimo local mais significativo. O *momentum* funciona como uma espécie de velocidade de descida, que impede que o aprendizado pare em pequenas saliências quando a taxa de erro estiver caindo rapidamente. Se o valor do *momentum* for elevado, pode-se acabar pulando os mínimos globais e assim não ser possível atingir um valor de erro satisfatório. Assim como o passo, não existe um valor ideal para o *momentum*, tudo depende da curva de erro e do problema que está sendo analisado.

Para que se compreenda melhor a forma como se processa o aprendizado em uma Rede Neural a base de *Perceptrons*, será analisado o exemplo da função AND, que responde com 1 quando as duas entradas forem 1 e com 0 para as demais combinações de entradas. A tabela de valores da função AND é mostrada na tabela 3.1.

Entrada 1	Entrada 2	Saída desejada
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 3.1 - Valores de entrada e saída para a função AND

Na figura 3.5 temos o esquema de um neurônio artificial que pode ser utilizado para o aprendizado da função AND. Este neurônio possui duas entradas e uma saída.

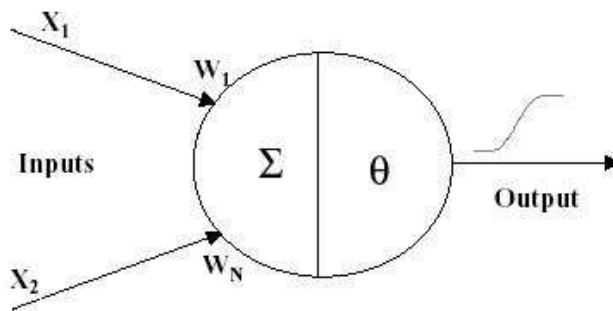


Figura 3.5 - Esquema de um Neurônio artificial com duas entradas e uma saída

A função de ativação de uma Rede Neural com duas entradas e uma saída é dada pela fórmula $F(x) = ax + by + c$, onde x e y são as entradas da Rede Neural, e a , b e c são os valores dos pesos das duas entradas e do Bias. Como pode ser notado, esta função nada mais é do que a equação da reta, o que significa que os pesos de um neurônio artificial formam uma reta que intersecciona o plano dividindo ele em regiões, como mostra a figura 3.6. Nesta figura são representados graficamente os valores da tabela 3.1 e a reta formada com o valor dos pesos dos neurônios depois de concluído o processo de aprendizado. Em Redes Neurais com três entradas, ao invés de uma reta temos um plano dividindo as classes em um cubo, e com quatro entradas ou mais, temos hiper-planos multidimensionais que não são possíveis de serem representados graficamente.

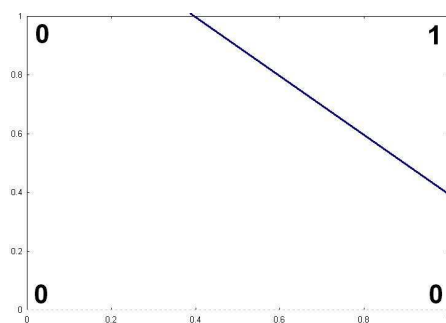


Figura 3.6 - Reta que divide as classes de saída da função AND

Note que a Rede Neural não precisa responder com 0 ou 1 exatamente, mas sim com valores próximos a 0 ou a 1. Um dos parâmetros que pode ser configurado em um simulador de Redes Neurais é o erro máximo, que indica a partir de qual valor se considera as saídas pertencentes a uma determinada classe. Se for utilizado um erro máximo de 0,4, por exemplo, qualquer valor de saída abaixo de 0,4 será considerado como sendo 0, e qualquer valor de saída acima de 0,6 será considerado como sendo 1. Na tabela 3.2 é mostrado um exemplo de ativação de uma Rede Neural para as entradas da função AND. Nas colunas *Entrada 1* e *Entrada 2* são mostradas as entradas de cada exemplo, na coluna *Peso 1* temos o peso associado a *Entrada 1* após o aprendizado, na coluna *Peso 2* temos o peso associado a *Entrada 2*, e na coluna *Peso Bias* temos o peso do Bias. Na coluna *Saída Obtida* temos os valores com os quais a Rede Neural respondeu para cada um dos exemplos após o processo de aprendizado. Pode-se notar pela tabela 3.2 que os pesos realmente formam uma equação da reta, e que as saídas obtidas estão corretas para um erro máximo de 0,4.

Entrada 1	Entrada 2	Peso 1	Peso 2	Peso Bias	Saída obtida
0	0	2,081044	3,072967	-4,107791	0,016178
0	1	2,081044	3,072967	-4,107791	0,262150
1	0	2,081044	3,072967	-4,107791	0,116423
1	1	2,081044	3,072967	-4,107791	0,740048

Tabela 3.2 - Valores das entradas, dos pesos e das saídas obtidos após o aprendizado da função AND

A função AND é um problema divisível linearmente, o que significa que com uma única reta é possível dividir as classes de saída. Mas existem funções que não são divisíveis linearmente, como por exemplo a função XOR, onde são necessárias duas ou mais retas para dividir as classes de saída, como mostra a figura 3.7.

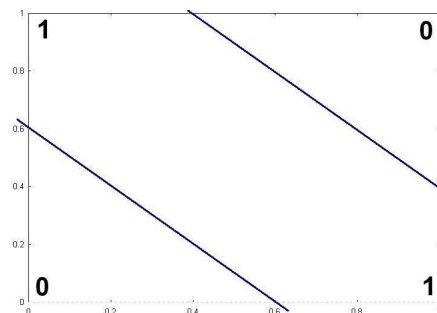


Figura 3.7 - Retas que dividem as classes de saída da função XOR

Uma Rede Neural formada por um único neurônio não é capaz de classificar corretamente a função XOR, pois cada neurônio é capaz de traçar uma única reta no gráfico. Muitos anos se levaram até que fosse descoberta uma forma de se combinar diversos *Perceptrons* para que fosse formada uma Rede Neural capaz de resolver problemas não divisíveis linearmente, e a solução definitiva para este problema só foi obtida em 1986, com as redes *Multi Layer Perceptron* (MLP) que utilizam o algoritmo *Backpropagation* [15].

3.3.2 Backpropagation

Um dos modelos de Redes Neurais mais utilizados na atualidade é o *Multi Layer Perceptron* (MLP) com *Backpropagation*, também chamado simplesmente de *Backpropagation*. O *Backpropagation* foi proposto em 1986 por RUMELHART [15] como uma forma de se combinar diversos *Perceptrons* em camadas para se resolver problemas não divisíveis linearmente, como por exemplo a função XOR. A forma como são estruturadas as diversas camadas do MLP é mostrada na figura 3.8.

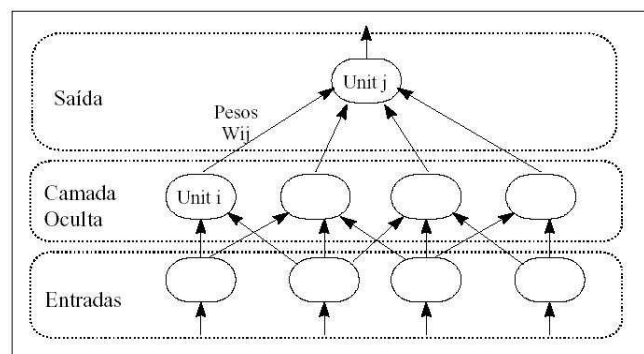


Figura 3.8 - Esquema de uma Rede Neural do tipo MLP [32]

A camada de entrada não possui pesos associados nem função de ativação, ela é apenas um *buffer* que recebe os valores externos e os repassa para a camada seguinte. Depois da camada de entrada, existem uma ou mais camadas ocultas, que possuem um número variável de neurônios que depende do problema que está sendo analisando. Por último temos a camada de saída, que recebe os valores da(s) camada(s) oculta(s), realiza a ativação e emite os resultados de saída da Rede Neural. O algoritmo do *Backpropagation* trabalha da seguinte forma: a camada de entrada recebe os valores externos e os repassa para os neurônios da camada seguinte, que é a primeira camada oculta. Estes neurônios fazem a ativação, da mesma forma que descrito anteriormente para o *Perceptron*, e o resultado é enviado para os neurônios da camada seguinte. Cada camada recebe como entrada os valores obtidos na saída dos neurônios da camada anterior. A camada de saída serve para combinar os valores obtidos nos neurônios das camadas anteriores e assim emitir o valor de saída da Rede Neural.

No *Backpropagation*, o ajuste dos pesos ocorre do fim para o início, sucessivamente em camadas. Para o ajuste dos pesos da camada de saída, calcula-se o erro comparando-se a saída obtida com a saída desejada, e se utiliza este erro para ajustar os pesos desta camada. Para o ajuste dos pesos da camada seguinte, é necessário calcular o nível de participação de cada neurônio na saída da Rede Neural. Este cálculo é realizado multiplicando-se o erro obtido na camada anterior (analisando-se no sentido da saída para a entrada) com o valor do peso que interliga os neurônios destas camadas, e assim se obtém um erro estimado, que é utilizado para o ajuste dos pesos desta camada. Pelo fato do erro ser estimado e não absoluto, Redes Neurais com mais de uma camada oculta costumam apresentar mais dificuldades no aprendizado.

O número de neurônios na camada de entrada de uma Rede Neural depende do número de variáveis envolvidas no problema a ser abordado. No caso da função AND, duas entradas são necessárias, uma para cada variável. O número de neurônios na camada de saída depende do tipo problema a ser abordado, pois para problemas que exigem uma resposta do tipo sim ou não, uma única saída é suficiente, mas se houver a necessidade de se classificar os exemplos em diversas categorias, podem ser necessárias várias saídas, uma para cada categoria de valores possíveis. Na camada oculta pode ser utilizada qualquer quantidade de neurônios, independente do número de entradas e de saídas utilizadas. Problemas linearmente separáveis não necessitam de neurônios na camada oculta, mas problemas de classificação ou de aproximação de funções mais complexos podem exigir vários neurônios na camada oculta. Para a determinação do número de neurônios da camada oculta, não existe uma regra fixa, pois tudo depende do problema que está sendo analisado, mas o que se costuma fazer é começar com poucos neurônios e ir aumentando o número até que seja possível realizar o aprendizado, o que exige que sejam feitas diversas simulações. O ajuste dos pesos em uma Rede Neural ocorre ao longo de muitas épocas, onde em cada época são submetidos todos os exemplos a Rede Neural, e assim os pesos vão sendo lentamente ajustados até que se chegue aos valores ideais para um determinado problema.

Outro fator importante em uma Rede Neural é o grau de generalização. Quando uma Rede Neural Artificial é treinada, os pesos vão sendo ajustados lentamente até que ela responda de forma adequada aos exemplos presentes na base de dados. Mas simplesmente responder de forma correta aos exemplos que estão sendo analisados não é o suficiente, é necessário que a Rede Neural consiga responder de forma correta a outros exemplos que sigam o mesmo padrão. Por esta razão, quando se realiza o aprendizado, costuma-se dividir a base de dados de exemplos em dois conjuntos de dados distintos, um conjunto de dados para o aprendizado, com a qual a Rede Neural é

ativada e tem os seus pesos ajustados, e um conjunto de dados de teste, com a qual a Rede Neural é ativada mas não sofre ajuste nos pesos. A base de dados de teste serve para medir como a Rede Neural se comporta com exemplos para os quais ela não foi treinada ao longo do processo de aprendizado, o que torna possível medir o grau de generalização da Rede Neural. A medida que as épocas vão passando, o erro em ambas as bases de dados vai sendo reduzido, como mostra a figura 3.9. Para a base de dados de aprendizado o erro se reduz indefinidamente, já que é a partir dela que os pesos são ajustados. Mas para a base com os dados de teste, chega um ponto em que o erro volta a subir, como pode ser visualizado na figura 3.9. Isto se deve ao fato de que a partir deste ponto a Rede Neural está começando a *decorar* os exemplos da base de aprendizado, e os novos conhecimentos adquiridos não representam mais uma regra geral, mas sim regras específicas para cada exemplo. O ideal é parar o treinamento no ponto ótimo de generalização, que é o instante imediatamente anterior ao ponto onde o erro começa novamente a subir na base de dados de teste.

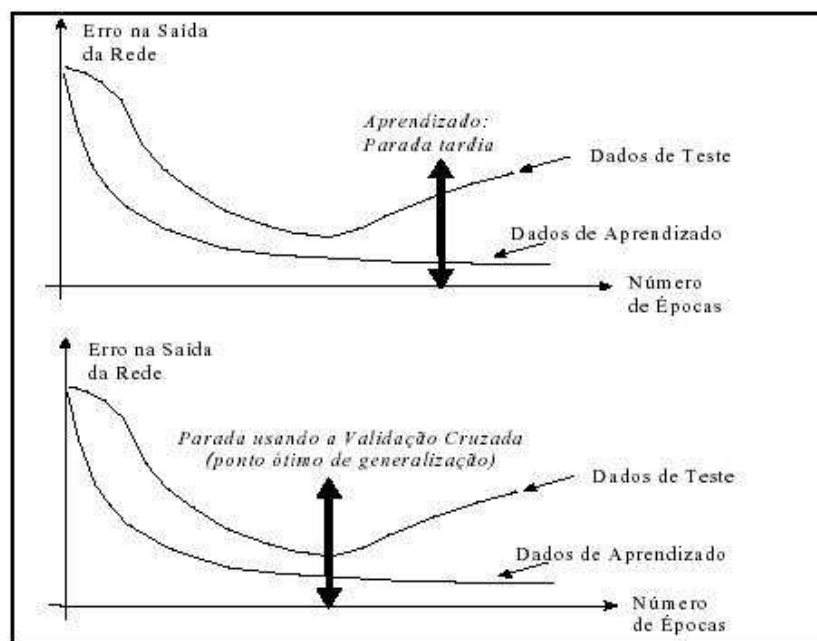


Figura 3.9 - Curvas de erro no aprendizado e na generalização [32]

3.3.3 Problemas e limitações do *Backpropagation*

O Algoritmo do *Backpropagation* possui diversos problemas e limitações, que foram levantadas por FAHLMAN [18]. As principais deficiências do algoritmo são:

- Definição da arquitetura: no *Backpropagation*, é preciso definir manualmente o número de neurônios da Rede Neural, o número de camadas e as interconexões entre os neurônios destas camadas, o que exige que sejam feitas várias simulações até que se consiga chegar a uma arquitetura ideal para a solução de determinado problema;
- Parâmetros em excesso: além da definição da arquitetura da rede, outros parâmetros, como o passo e o *momentum*, precisam ser configurados manualmente para cada problema a ser analisado, o que torna o uso das Redes Neurais extremamente complexo e exige que sejam feitas várias simulações até que se chegue aos valores ideais que tornem possível o aprendizado;
- Dependência da inicialização dos pesos: como os pesos são iniciados de forma aleatória, nunca se sabe exatamente em que ponto da curva de erro o aprendizado

irá iniciar, o que faz com que cada simulação realizada traga resultados diferentes, algumas vezes não sendo possível se chegar a valores satisfatórios;

- Minimização do erro lenta e incerta: são necessárias muitas épocas de aprendizado para que o aprendizado ocorra, e nunca se sabe com antecedência se o aprendizado irá ocorrer de forma satisfatória com a configuração e a topologia utilizada;
- Plasticidade e elasticidade: depois que uma Rede Neural convergiu para uma solução, é muito difícil adicionar novos exemplos e continuar o aprendizado do ponto onde parou, e ao se tentar fazer isto, corre-se o risco de fazer com que a Rede Neural esqueça todos os conhecimentos adquiridos anteriormente (esquecimento catastrófico).

Com a finalidade de tentar resolver a maioria destes problemas, FAHLMAN [19] propôs um novo modelo de Redes Neurais, chamado de *Cascade-Correlation*, que corrige muitas das deficiências presentes no *Backpropagation*.

3.3.4 *Cascade-Correlation*

O *Cascade-Correlation* é um modelo otimizado de MLP (*Multi Layer Perceptron*), desenvolvido por FAHLMAN [19] com a finalidade de solucionar a maioria dos problemas levantados anteriormente no algoritmo do *Backpropagation*. As principais vantagens que o *Cascade-Correlation* apresenta em relação aos demais modelos de Redes Neurais são:

- Definição da arquitetura: no *Cascade-Correlation*, não é necessário determinar o número de neurônios da camada oculta nem as interligações entre estes neurônios, pois o próprio algoritmo do *Cascade-Correlation* se encarrega de determinar a melhor topologia de rede possível para solucionar um determinado problema;
- Menos parâmetros: no *Cascade-Correlation*, o passo não precisa ser ajustado, pois o algoritmo altera o valor do passo durante o aprendizado com uma técnica chamada *Quickprop* [18] de forma a acelerar ao máximo o aprendizado. O *momentum* e outros parâmetros também não precisam ser ajustados no *Cascade-Correlation*.
- Continuidade do aprendizado: quando surgirem novos exemplos, pode-se treinar uma Rede Neural para aprender estes novos casos a partir de uma rede já treinada sem perder os conhecimentos adquiridos anteriormente;
- Velocidade de aprendizado: o *Cascade-Correlation* converge muito rapidamente para um ponto mínimo da curva de erro, e pelo fato de não serem necessárias várias simulações anteriores para ajustar a arquitetura da rede e os parâmetros, o tempo necessário para o aprendizado de um problema é reduzido drasticamente;

Todas estas vantagens fazem com que o *Cascade-Correlation* seja muito simples de ser utilizado em comparação a outros modelos de Redes Neurais, além de ser extremamente eficaz em termos de aprendizado. Uma das poucas limitações que o *Cascade-Correlation* apresenta é que ele só pode ser utilizado para problemas de classificação, e não para aproximação de funções. Uma vez que o *Cascade-Correlation* é muito adequado para resolver problemas de classificação, isto permite o seu uso em sistemas de autenticação de assinaturas.

A topologia do *Cascade-Correlation* é um pouco diferente da topologia do *Backpropagation* tradicional, como pode ser visto na figura 3.10. Os neurônios da camada oculta não são posicionados lado a lado, mas sim em cascata, com a saída de um neurônio alimentando a entrada do outro.

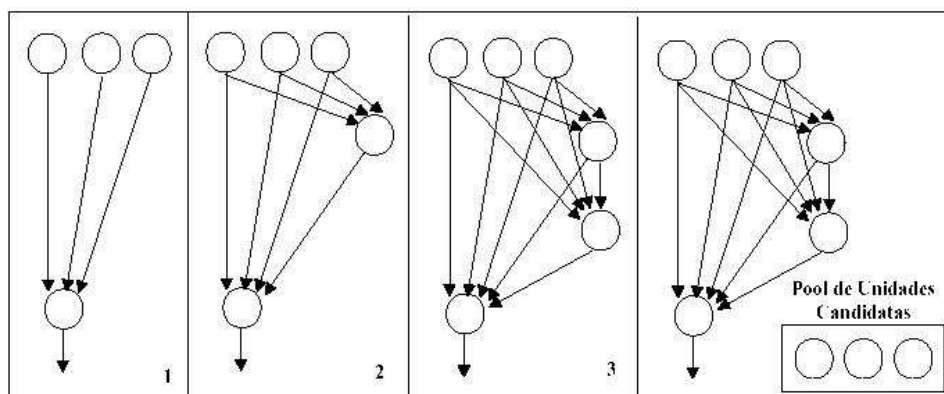


Figura 3.10 - Topologia incremental do *Cascade-Correlation*

O algoritmo de aprendizado utilizado pelo *Cascade-Correlation* funciona de uma forma bastante diferente do algoritmo utilizado pelo *Backpropagation*. A Rede Neural inicia sem nenhum neurônio na camada oculta, como mostra o quadro 1 da figura 3.10. A medida que as épocas vão passando, os pesos do(s) neurônio(s) de saída vão sendo ajustado de forma a minimizar ao máximo o erro. Quando o aprendizado estagnar e não for mais possível descer na curva de erro, um conjunto de unidades candidatas é alocado e treinado a parte, até se encontrar a unidade candidata que obtiver a maior correlação com o erro residual do neurônio de saída. A unidade selecionada é então inserida na Rede Neural, como mostra o quadro 2 da figura 3.10, e tem os seus pesos “congelados” de forma que não sejam mais ajustados. Depois disto, o neurônio de saída volta a ser treinado, e quando o aprendizado novamente estagnar, outro conjunto de unidades candidatas é alocado e o processo se repete, como mostra o quadro 3 da figura 3.10. O aprendizado é interrompido de acordo com o critério de parada selecionado, que pode ser quando se atingir 100% de aprendizado ou quando se exceder um número máximo de épocas.

3.4 Redes Neurais aplicadas a autenticação de assinaturas

A autenticação de assinaturas faz parte do reconhecimento de padrões, onde o padrão a ser reconhecido é a assinatura de determinado usuário, e as classes nas quais este padrão pode ser atribuído são as classes **Pertence** e **Não_Pertence**, relativas a determinado usuário. Para a autenticação de assinaturas, é necessário que se identifiquem nos padrões das assinaturas quais os elementos que podem diferenciá-las em relação as assinaturas dos demais usuários, e também é necessário que se abstraiam as características que não são relevantes para o processo de classificação.

Diversas características presentes nas Redes Neurais Artificiais as tornam muito eficazes na autenticação de assinaturas, sendo que as mais importantes são:

- Aprendizado a partir de exemplos: basta submeter a base de dados de assinaturas a uma Rede Neural que a rede irá aprender quais as características que melhor identificam as assinaturas de determinado usuário, sem que seja necessário explicitar as regras de classificação manualmente;
- Generalização: as assinaturas não precisam ser exatamente iguais para serem reconhecidas como pertencendo a um determinado usuário, pois o sistema é capaz de aprender as regras gerais que definem as assinaturas e realizar a classificação a partir destas regras;

- Possibilidade de trabalhar com informações incorretas e/ou incompletas: mesmo que o usuário tenha assinado de forma diferente em uma ou mais assinaturas da base de dados, isto não afetará de forma significativa o desempenho do sistema;
- Segurança dos dados: como a Rede Neural só precisa do arquivo de pesos para fazer a classificação, a base de dados original pode ser descartada ou mantida a salvo em outro local após do treinamento, o que torna o sistema mais seguro em relação ao ataque por parte de *hackers*.

Por todos estes motivos é que foi escolhido o uso de Redes Neurais para a autenticação de assinaturas neste trabalho, e o modelo de Redes Neurais escolhido foi o *Cascade-Correlation*, devido as diversas vantagens que ele apresenta em relação aos demais modelos de Redes Neurais.

3.5 Problema de classes fechadas

As funções AND e XOR, vistas anteriormente, são problemas em que as classes são abertas, de forma que apenas uma e duas retas, respectivamente, servem para a divisão das classes de saída. Já a autenticação de assinaturas é um problema de classes fechadas, onde não basta que se classifique uma assinatura na classe que ela mais se aproxime, é necessário que se tenha o máximo de certeza possível de que uma assinatura realmente pertença a um determinado usuário. A figura 3.11 mostra um exemplo de um problema com duas entradas (x e y) com três classes de saída, A , B e C .

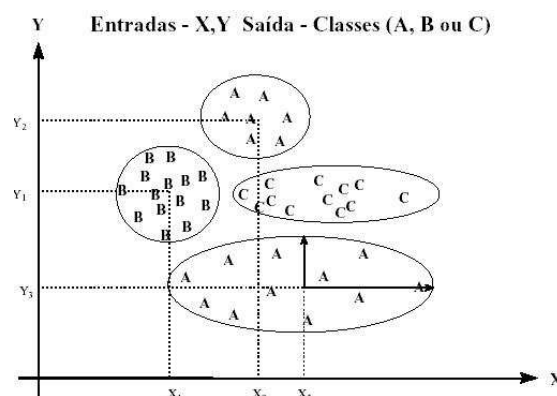


Figura 3.11 - Problema de classificação envolvendo classes fechadas

Podemos considerar cada classe da figura representando as assinaturas de um determinado usuário, assim, com três retas é possível classificar o problema, como mostra a figura 3.12.

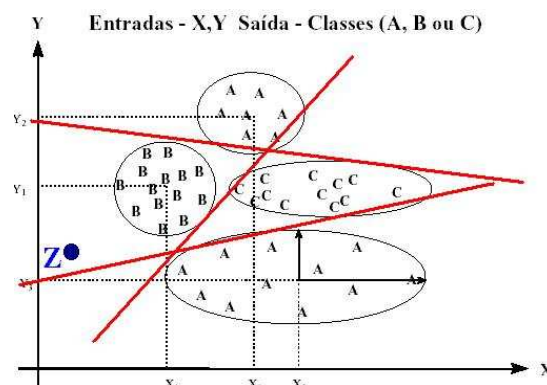


Figura 3.12 - Retas de divisão das classes A, B e C

A primeira vista pode parecer que o problema foi solucionado com apenas três neurônios, mas se o sistema recebesse como entrada uma assinatura desconhecida, como por exemplo a assinatura Z destacada na figura 3.12, a assinatura seria classificada como se pertencesse ao usuário B, mas é muito provável que esta assinatura não pertença a ele. Problemas assim irão ocorrer toda vez que o sistema se deparar com assinaturas para as quais ele não foi treinado, e dependendo da posição traçada pelas retas, o sistema pode dizer que uma assinatura pertence a determinado usuário mesmo que ela não se pareça nem um pouco com a assinatura original, e é impossível de se armazenar em uma base de dados todas as assinaturas possíveis de serem desenhadas em um *tablet*.

Problemas de classes fechadas exigem que a classificação delimite a área em que as assinaturas são consideradas válidas, como mostra a figura 3.13. Esta figura representa uma Rede Neural especializada em autenticar as assinaturas do usuário B, onde qualquer entrada que não se enquadre no padrão é considerada como se não pertencesse ao usuário.

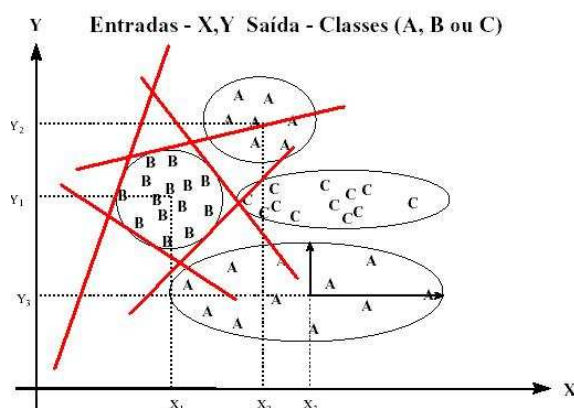


Figura 3.13 - Separação da classe fechada B em relação as demais classes

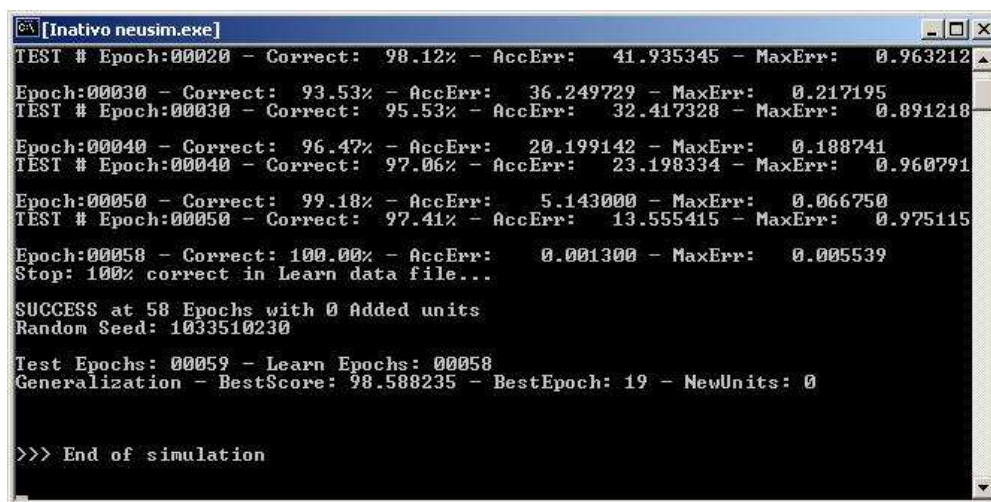
Mas o problema está em como fazer com que a rede delimite o padrão desta forma. Uma alternativa seria cobrir todo o espaço amostral com exemplos fictícios (exceto nas regiões onde existem padrões de assinaturas reais), que serviriam para indicar quais as regiões do espaço amostral que não pertencem a nenhum usuário. O problema desta abordagem é que com n entradas o número de exemplos fictícios necessários poderia tender ao infinito, principalmente quando as entradas são valores numéricos e contínuos.

A partir do estudo realizado sobre as classes fechadas, foi proposta aqui uma nova técnica para minimizar estes problemas, que denominamos de Nuvem de Pontos. Nesta técnica se escolhe a quantidade de assinaturas fictícias a serem utilizadas, e o sistema gera exemplos com valores aleatórios para cada entrada da Rede Neural. Os valores gerados para cada uma das entradas são distribuídos linearmente dentro de intervalos específicos, que variam conforme o atributo. Com esta técnica, é possível cobrir de modo esparsos o espaço amostral, o que ajuda a delimitar a área do hiper-plano na qual as assinaturas são classificadas como verdadeiras. Outra alternativa que pode ser utilizada é inserir assinaturas feitas com a caneta de forma aleatória, como riscos e desenhos livres, para delimitar as regiões do hiper-plano em que se concentram os exemplos possíveis de serem gerados a partir de um *tablet*, pois a Nuvem de Pontos pode gerar entradas com combinações de atributos impossíveis de serem obtidas em assinaturas reais. Estas duas técnicas utilizadas em conjunto costumam apresentar melhores resultados no processo de aprendizado.

3.6 Simulador Neusim

Para a utilização das Redes Neurais Artificiais, é necessário que se utilize um simulador que implemente o modelo de Rede Neural que foi escolhido. O simulador de Redes Neurais escolhido para ser utilizado neste trabalho foi o Neusim, desenvolvido por OSORIO [24]. O Neusim foi escolhido por ser um simulador ser simples de ser utilizado, e também por implementar o *Cascade-Correlation*, que é o modelo de Redes Neurais escolhido para ser utilizado neste trabalho.

A versão do Neusim utilizada no desenvolvimento do protótipo é uma aplicação de linha de comando desenvolvida para o sistema operacional MS-DOS, que suporta no máximo 1000 entradas, 29.999 exemplos na base de aprendizado e 29.999 exemplos na base de teste de generalização. Na figura 3.14 é mostrado o resultado de uma execução do Neusim na linha de comando do sistema operacional.



```
[Inativo neusim.exe]
TEST # Epoch:00020 - Correct: 98.12% - AccErr: 41.935345 - MaxErr: 0.963212
Epoch:00030 - Correct: 93.53% - AccErr: 36.249729 - MaxErr: 0.217195
TEST # Epoch:00030 - Correct: 95.53% - AccErr: 32.417328 - MaxErr: 0.891218
Epoch:00040 - Correct: 96.47% - AccErr: 20.199142 - MaxErr: 0.188741
TEST # Epoch:00040 - Correct: 97.06% - AccErr: 23.198334 - MaxErr: 0.960791
Epoch:00050 - Correct: 99.18% - AccErr: 5.143000 - MaxErr: 0.066750
TEST # Epoch:00050 - Correct: 97.41% - AccErr: 13.555415 - MaxErr: 0.975115
Epoch:00058 - Correct: 100.00% - AccErr: 0.001300 - MaxErr: 0.005539
Stop: 100% correct in Learn data file...
SUCCESS at 58 Epochs with 0 Added units
Random Seed: 1033510230
Test Epochs: 00059 - Learn Epochs: 00058
Generalization - BestScore: 98.588235 - BestEpoch: 19 - NewUnits: 0

>>> End of simulation
```

Figura 3.14 - Execução do simulador de Redes Neurais Neusim

O Neusim possui diversos arquivos de texto que servem para a entrada dos dados de exemplo, saída dos resultados das simulações e configuração dos parâmetros de execução. Os arquivos utilizados pelo Neusim podem ter qualquer nome de arquivo válido para o MS-DOS, mas todos eles devem ter o mesmo nome, mudando somente a extensão. Para que o Neusim seja executado, basta digitar na linha de comando **Neusim.exe arquivo**, onde **arquivo** é o nome utilizado nos arquivos, sem a extensão. Os arquivos mais importantes do Neusim são:

- Arquivo de configuração (CFG): permite ao usuário do Neusim configurar diversos parâmetros de simulação da Rede Neural;
- Arquivo de aprendizado (LRN): contém os dados utilizados para o aprendizado da Rede Neural;
- Arquivo de generalização (TST): contém os dados utilizados para o teste de generalização da Rede Neural;
- Arquivo de saída (OUT): contém o valor de ativação da Rede Neural para cada exemplo da base de dados de generalização. Este arquivo é gerado automaticamente pelo Neusim após o teste final de generalização;
- Arquivo de topologia (TOP): contém a topologia da Rede Neural que foi definida após aprendizado;

Arquivo de pesos (WTS): contém o valor dos pesos sinápticos da Rede Neural que foi definida após o processo de aprendizado.

As letras entre parênteses são as extensões de cada um dos arquivos. No anexo B são mostrados exemplos de alguns destes arquivos.

Em relação ao arquivo de configuração do Neusim, os parâmetros mais importantes que estão presentes neste arquivo são:

- Task: tarefa a ser realizada. Os valores possíveis são combinações das letras L (aprendizado), G (teste da base de generalização a cada época), WS (salva os pesos no final), P (carrega os pesos a partir do arquivo de pesos), T (teste da base de generalização no final da simulação) e O (gera o arquivo de saída após o teste final de generalização). Na fase de aprendizado costuma-se utilizar os valores LGWSOT, para que o simulador realize o aprendizado, faça os testes de generalização e salve os pesos no final da simulação. Na fase de classificação costuma-se utilizar as letras POT, para carregar os pesos salvos anteriormente, ativar a Rede Neural e gravar no arquivo de saída o valor de ativação da Rede Neural;
- MaxEpochs: número máximo de épocas de simulação;
- NInputs: número de neurônios na camada de entrada;
- NOutputs: número de neurônios na camada de saída;
- Learning: 0 para *Backpropagation* e 1 para *Cascade-Correlation*;
- Cascor: Número de unidades candidatas utilizadas pelo *Cascade-Correlation*;
- Epsilon: Velocidade do aprendizado (passo, *Learning rate*);
- MaxErr: Erro máximo (*Score Thershold*);
- StopCrit: condição de parada: 1 para terminar a simulação quanto atingir 100% de aprendizado e 2 para terminar a simulação quanto atingir 100% de generalização.

Informações adicionais sobre o Neusim podem ser obtidas em [24]. No próximo capítulo será estudado o modelo de sistema de autenticação de assinaturas proposto, denominado Sistema NeuralSignX.

4 Sistema NeuralSignX

Este capítulo apresenta a proposta de um sistema *on-line* de autenticação de assinaturas baseado em Redes Neurais, denominado de Sistema NeuralSignX. O sistema é composto de três módulos principais, que são o módulo de entrada, o módulo de pré-processamento e o módulo de processamento. A figura 4.1 mostra um diagrama de blocos dos diversos módulos do sistema. Nas próximas seções será descrito o funcionamento de cada um dos módulos, e ao final será apresentado o protótipo do Sistema NeuralSignX, que foi implementado com a finalidade de demonstrar as principais características do sistema proposto.

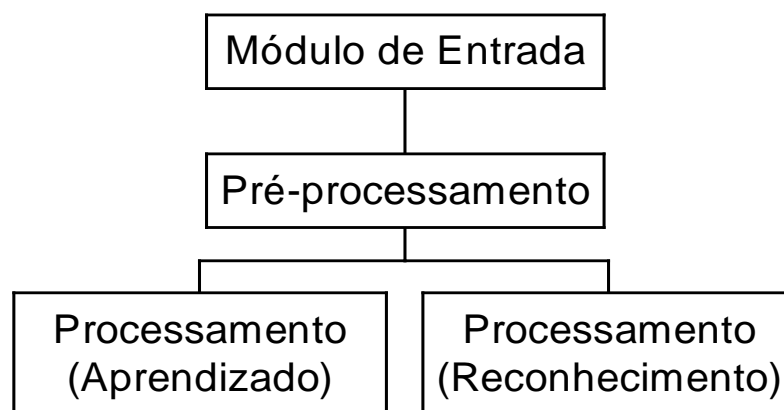


Figura 4.1 - Módulos do Sistema NeuralSignX

4.1 Módulo de entrada

O módulo de entrada é responsável pela leitura e armazenamento dos dados provenientes do dispositivo de *hardware* utilizado para a coleta de assinaturas. Este dispositivo é o PenTablet SuperPen WP4030, fabricado pela UC-Logic¹. Este dispositivo é composto de duas partes, que são uma caneta especial e uma superfície sensível sobre a qual a caneta se desloca quando uma assinatura é desenhada. A área sensível da superfície do *tablet* tem dimensões de 120 x 76 milímetros, e é capaz de captar a posição da caneta quando esta está a uma distância de no máximo 12,7 milímetros da superfície do *tablet*. As informações que o *tablet* envia para o computador são as coordenadas da caneta em relação a área sensível, se a caneta está tocando ou não a superfície do *tablet* e a pressão exercida sobre a caneta. O funcionamento de um *tablet* é similar ao funcionamento de um *mouse*, que envia as coordenadas em que a caneta se encontra a cada n milissegundos, de forma que a aplicação que faz a coleta das assinaturas precisa coletar e armazenar estas informações em tempo real para poder montar posteriormente a imagem da assinatura.

No Sistema NeuralSignX, as rotinas que realizam a leitura dos dados provenientes do *tablet* funcionam da seguinte forma: quando uma assinatura é iniciada, as informações enviadas pelo *tablet* vão sendo armazenadas em um vetor com a seguinte estrutura:

¹ UC-Logic SuperPen – <http://www.superpen.com/>


```

Ponto = Registro

    Coordenada_X:      Inteiro
    Coordenada_Y:      Inteiro
    PressaoMaiorQueZero: Booleano
    HoraRecebimento:   DateTime

Fim

```

As coordenadas x e y são as coordenadas de cada ponto amostrado, o item *PressaoMaiorQueZero* indica se a caneta está ou não tocando o *tablet* e o item *HoraRecebimento* indica o instante de tempo em o ponto foi coletado, com precisão de milissegundos. O instante de tempo é utilizado posteriormente para o cálculo de atributos como o tempo de duração da assinatura e a velocidade de deslocamento da caneta. Quando a assinatura é encerrada, os pontos que foram armazenados no vetor são salvos em um arquivo com o seguinte formato:

```

LOGIN=MILTON HEINEN
127 466 1 21:42:23:821
129 455 1 21:42:23:831
131 442 0 21:42:23:841
137 409 0 21:42:23:861
138 391 1 21:42:23:871

```

A linha iniciada por LOGIN é o cabeçalho da assinatura, que indica que uma nova assinatura está iniciando e pertence ao usuário Milton Heinen. As linhas subsequentes informam as coordenadas x e y de cada ponto amostrado pelo *tablet*, se a caneta estava ou não tocando o *tablet* (1 ou 0) naquele momento e o instante de tempo em que cada ponto foi coletado pelo *tablet* no formato hh:nn:ss:zzz. (horas, minutos, segundos e milissegundos). Este arquivo texto armazena as coordenadas dos pontos de todas as assinaturas coletadas pelo *tablet*, formando assim uma base de dados de assinaturas. No Anexo B é mostrada uma parte de um arquivo de assinaturas do Sistema NeuralSignX.

4.2 Pré-processamento

O módulo de pré-processamento do Sistema NeuralSignX é responsável por transformar os dados provenientes do módulo de entrada em informações que possam ser utilizadas no processo de autenticação de assinaturas. As principais atividades realizadas pelo módulo de pré-processamento são a montagem da assinatura, o ajuste de posição, o ajuste de escala e a extração de atributos. A figura 4.2 mostra um diagrama de blocos das diversas atividades realizadas pelo módulo de pré-processamento do Sistema NeuralSignX.

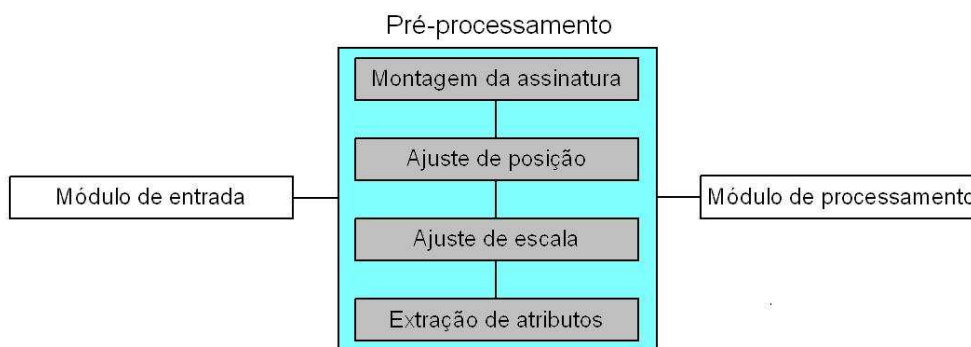


Figura 4.2 - Atividades do módulo de pré-processamento

4.2.1 Montagem da assinatura

Em sistemas de autenticação de assinaturas do tipo *off-line*, a assinatura é obtida diretamente do dispositivo digitalizador na forma de uma imagem, mas em sistemas *on-line*, o que o *tablet* disponibiliza é apenas uma lista seqüencial dos pontos que foram coletados durante a assinatura. Portanto, a primeira tarefa que deve ser realizada pelo módulo de pré-processamento é a montagem da assinatura a partir dos pontos coletados pelo *tablet*. O primeiro passo para a montagem da assinatura é desenhar os pontos coletados em um *bitmap*, conforme mostra a figura 4.3.

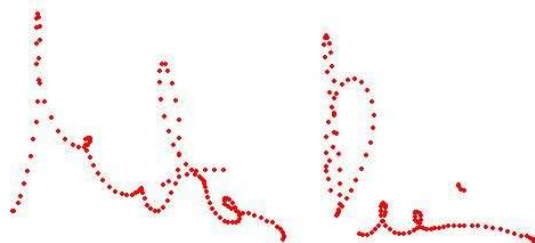


Figura 4.3 - Pontos de uma assinatura coletados pelo *tablet*

Como se pode notar através da figura 4.3, o resultado obtido não é exatamente uma imagem da assinatura, mas sim uma coleção de pontos que percorrem a trajetória da assinatura, alguns mais afastados e outros mais próximos entre si. Isto ocorre devido ao fato de o *tablet* não coletar todos os pontos de uma assinatura, mas apenas os pontos amostrados em intervalos regulares de tempo. O segundo passo necessário para se obter a imagem da assinatura é ligar os pontos originais através de retas, conforme mostra a figura 4.4, onde os pontos coletados estão destacados para uma melhor visualização do processo.

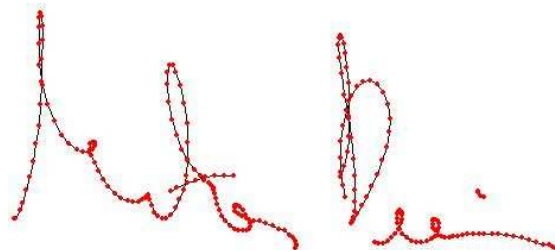


Figura 4.4 - Pontos coletados sendo interligados através de retas

A figura 4.5 mostra a mesma assinatura sem destacar os pontos originais. Pode-se notar que o desenho da assinatura é bastante realístico, não apresentando grandes distorções na curvatura. Sobre a imagem da assinatura obtida na figura 4.5, podemos aplicar todas as técnicas de extração de atributos utilizadas no reconhecimento *off-line*, com a vantagem de não ser necessário que se façam tratamentos na imagem para eliminar as impurezas e nem separar o fundo da imagem do desenho da assinatura.

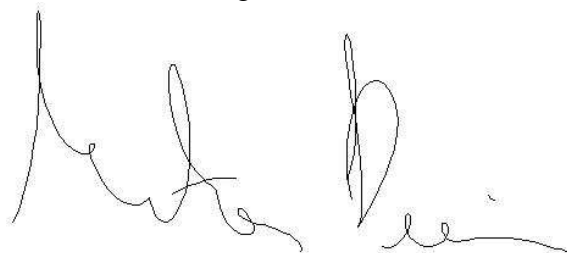


Figura 4.5 - Forma final de uma assinatura depois da montagem

A obtenção da imagem da assinatura é importante para o processo de autenticação de assinaturas, mas o maior diferencial do reconhecimento *on-line* em relação ao *off-line* está na lista de pontos coletados pelo *tablet*, pois a partir destes pontos, que estão ordenados conforme foram coletados, é possível que sejam obtidas informações temporais e dinâmicas a respeito da assinatura que não poderiam ser obtidas através da imagem.

4.2.2 Ajuste de posição

O ajuste de posição é uma alteração que se faz nas assinaturas de forma que elas fiquem todas na mesma posição na tela, o que facilita o processo de autenticação. O ajuste de posição pode ser feito de duas formas, pelo canto superior esquerdo ou pelo centro de massa da assinatura. O ajuste pelo canto superior esquerdo consiste em deslocar a assinatura de forma que as suas extremidades superior e esquerda coincidam com as extremidades superior e esquerda da tela. A figura 4.6 mostra como se realiza o ajuste de posição pelo canto superior esquerdo.

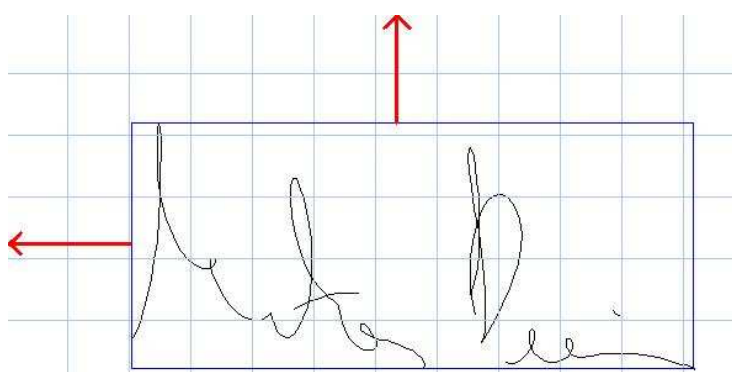


Figura 4.6 - Ajuste de posição pelo canto superior esquerdo

O ajuste de posição pelo centro de massa da assinatura altera a posição da assinatura de forma que o centro de massa da assinatura coincida com o centro da imagem. O centro de massa da assinatura é o local que corresponde a média das coordenadas de todos os pontos coletados da assinatura. Na figura 4.7 vemos o centro de massa de uma assinatura representado por um círculo.

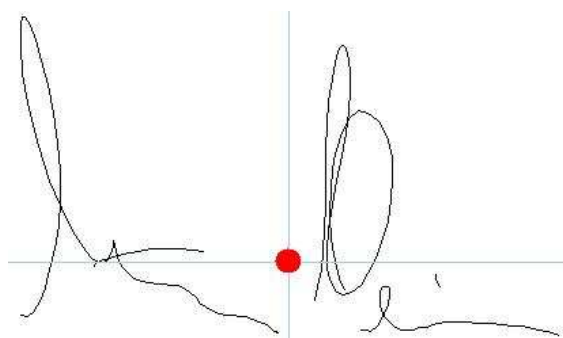


Figura 4.7 - Centro de massa de uma assinatura

A figura 4.8 mostra como se realiza o ajuste de posição pelo centro de massa. A seta próxima ao centro da figura mostra o deslocamento que a assinatura deve realizar até que o seu centro de massa coincida com o centro da tela. Este deslocamento é realizado alterando-se o valor das coordenadas de cada um dos pontos da assinatura. Na figura 4.9 vemos a assinatura depois que o ajuste de posição pelo centro de massa da assinatura foi realizado, com o centro de massa coincidindo com o centro da tela.

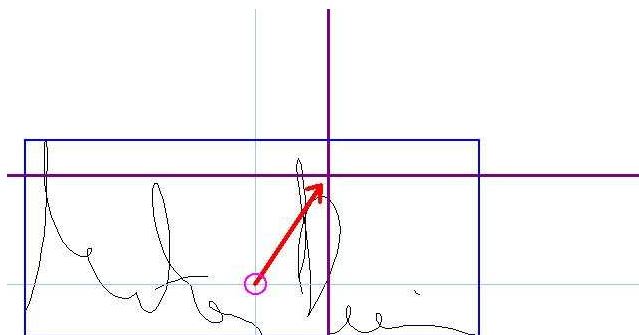


Figura 4.8 - Ajuste de posição pelo centro de massa de uma assinatura

O ajuste de posição pelo centro de massa da assinatura é mais complexo de ser implementado, mas apresenta melhores resultados devido ao fato dele não ser sensível a variações nos extremos da assinatura, que no ajuste pelo canto superior esquerdo poderiam fazer com que toda a assinatura ficasse deslocada.

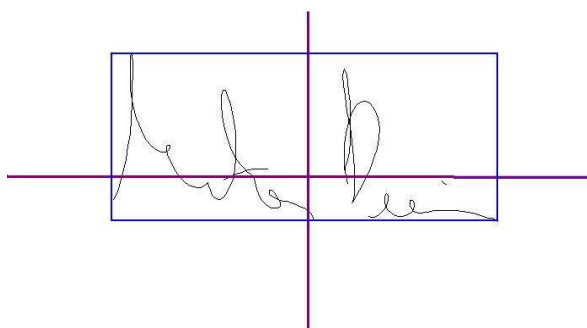


Figura 4.9 - Assinatura após o ajuste de posição pelo centro de massa

4.2.3 Ajuste de escala

O ajuste de escala visa minimizar as variações de tamanho que podem ocorrer entre assinaturas de uma mesma pessoa. Ele é realizado através da alteração das coordenadas dos pontos coletados pelo *tablet* de forma que a assinatura sofra um incremento no tamanho, e este incremento é cessado quando um dos extremos da assinatura coincidir com um dos extremos da tela. A figura 4.10 demonstra este processo. O retângulo menor da figura corresponde as dimensões originais da assinatura, e o retângulo maior corresponde as dimensões que a assinatura terá depois que o ajuste for realizado. O fator de incremento deve ser o mesmo em todas as direções para que a assinatura não fique distorcida.

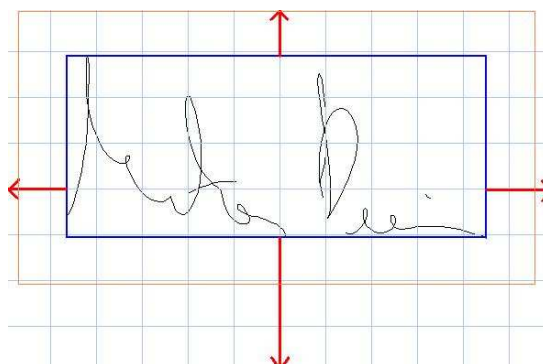


Figura 4.10 - Ajuste de escala (assinaturas centralizadas)

Se uma assinatura teve a posição ajustada anteriormente pelo canto superior esquerdo, o ajuste de escala deve expandir a assinatura apenas para a direita e para baixo, como mostra a figura 4.11. O processo de expansão cessa quando um dos extremos da assinatura coincidir com um dos extremos da tela.

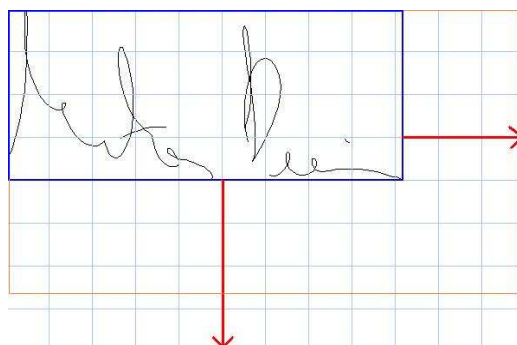


Figura 4.11 - Ajuste de escala (assinaturas ajustadas pelo canto superior esquerdo)

4.2.4 Extração de atributos

A etapa mais importante do módulo de pré-processamento do Sistema NeuralSignX é a extração de atributos, que consiste em obter a partir das assinaturas informações que permitam realizar posteriormente a classificação das assinaturas no módulo de processamento. Uma série de atributos foram levantados, estudados e implementados, e os que obtiveram os melhores resultados são:

4.2.4.1 Tempo de duração da assinatura

É o tempo que o usuário utilizou para a realização da assinatura. O cálculo é realizado diminuindo-se o instante final da assinatura pelo instante inicial, obtendo-se assim o tempo de duração da assinatura em termos de milissegundos. Este atributo é importante principalmente para a detecção de fraudes, pois assinaturas falsificadas costumam levar muito mais tempo para serem desenhadas que as assinaturas originais [5].

4.2.4.2 Número de vezes em que a caneta foi levantada

Este atributo conta quantas vezes o usuário afastou a caneta do *tablet* durante a assinatura, excluindo-se os extremos. O usuário costuma levantar a caneta para separar nome e sobrenome ou para colocar acentos. Este atributo, em conjunto com outros, é muito útil diferenciar assinaturas de diferentes usuários, pois ele praticamente não varia entre assinaturas de um mesmo usuário.

4.2.4.3 Velocidade média da assinatura

É a velocidade média de deslocamento da caneta sobre o *tablet*, calculada dividindo-se a distância total que a caneta percorreu durante o traçado pelo tempo de duração da assinatura. Este atributo é muito útil em processos de detecção de fraudes, pois assinaturas falsificadas costumam ser desenhadas muito mais lentamente que as assinaturas originais [5]. Este atributo deve ser extraído antes de se realizar o ajuste de escala da assinatura, pois este ajuste altera o comprimento da assinatura e assim afeta o cálculo da velocidade.

4.2.4.4 Velocidade máxima da assinatura

Este atributo é similar a velocidade média, mas com a diferença que ao invés de se considerar a assinatura como um todo, o conjunto dos pontos amostrados é dividido em

diversos subconjuntos e é calculada a velocidade média para cada subconjunto. A velocidade máxima será a maior velocidade encontrada nestes subconjuntos. Este atributo é útil para diferenciar assinaturas que possuam características especiais como um pequeno trecho muito mais rápido que o restante da assinatura, e em sistemas de autenticação de assinaturas, quanto mais informações estiverem disponíveis, melhores serão os resultados. O tamanho do subconjunto de pontos pode ser configurado nos parâmetros do sistema. Na assinatura da figura 4.3, vista no início do capítulo, podemos observar áreas de maior velocidade da assinatura, que são as regiões onde os pontos amostrados pelo *tablet* estão mais distantes entre si.

4.2.4.5 Amostragem seqüencial da assinatura

Em assinaturas simples, onde o usuário apenas escreve seu nome, a trajetória que a caneta percorre é fácil de ser identificada quando a assinatura é inspecionada visualmente. Mas em assinaturas mais complexas, como a da figura 4.12, é mais difícil determinar com precisão qual a trajetória realizada pela caneta apenas com a inspeção visual. Em sistemas de autenticação de assinaturas, as informações relativas a trajetória da assinatura são muito úteis para o processo classificação e de detecção de fraudes, pois se um falsário fizer uma assinatura muito semelhante a original, mas com uma trajetória diferente, a falsificação poderá ser detectada facilmente. A trajetória da assinatura é mais importante para a detecção de falsificações traçadas, pois neste tipo de falsificação a trajetória da assinatura costuma ser muito diferente da trajetória original, embora a forma da assinaturas seja bastante semelhante.



Figura 4.12 - Exemplo de assinatura com uma trajetória não usual

Existem muitas formas de se obter informações relativas a trajetória da assinatura. Uma das formas mais simples é analisar as coordenadas do ponto inicial e do ponto final da assinatura e comparar com o início e o término das assinaturas dos outros usuários. Se os pontos estiverem relativamente próximos, a assinatura é considerada autêntica, caso contrário a assinatura é considerada falsa. Devido a variabilidade existente entre assinaturas de um mesmo usuário, as coordenadas do ponto inicial e final da assinatura nunca estarão no mesmo local, mas tendem a se localizar em regiões relativamente próximas dentro da imagem, o que faz que um mecanismo de classificação que possua generalização, como as Redes Neurais, possa utilizar este tipo de informação para diferenciar as assinaturas de diferentes usuários. Na figura 4.13 é mostrada a assinatura da figura 4.12 com os pontos de início e término destacados para uma melhor visualização.

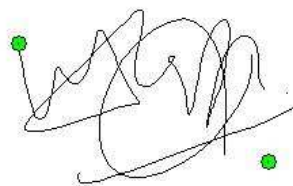


Figura 4.13 - Assinatura com os pontos inicial e final destacados

Para minimizar as variações que ocorrem entre as assinaturas de um mesmo usuário, as coordenadas dos pontos da assinatura podem ser normalizadas entre 0 e 1, onde o ponto mais a esquerda terá a coordenada x com valor 0, o ponto mais a direita terá a coordenada x com valor 1, e os pontos intermediários terão valores entre 0 e 1. Para o eixo y se procede da mesma forma, mas em relação aos extremos superior e inferior da assinatura. Nas configurações do protótipo é possível escolher se as coordenadas vão ser informadas com os valores absolutos ou com os valores normalizados.

Com apenas dois pontos amostrados é possível determinar qual a região em que uma assinatura deve começar e qual a região em que uma assinatura deve terminar para ser considerada autêntica, mas a trajetória do restante da assinatura não é analisada. Para que seja analisada toda a trajetória da assinatura, seria necessário utilizar todos os pontos que foram coletados a partir do *tablet*, mas isto seria impraticável devido ao grande número de entradas que seria necessárias na Rede Neural. Uma solução que pode ser utilizada é amostrar os pontos coletados originalmente em intervalos regulares em relação ao tempo, de forma que se saiba em que região a caneta deve se encontrar quando tiver passado 10% do tempo total da assinatura, 20%, 30%, e assim por diante. A figura 4.14 mostra uma assinatura que foi amostrada em oito pontos equidistantes em relação ao tempo de duração da assinatura. A cor dos pontos destacados serve para uma melhor visualização da trajetória, onde o ponto inicial é o ponto destacado mais claro e o ponto final é o ponto destacado mais escuro. O número de pontos amostrados que serão utilizados neste atributo pode ser configurado nos parâmetros do Sistema NeuralSignX. Para cada ponto amostrado, são necessárias duas entradas na Rede Neural, uma para o valor da coordenada x e outra para o valor da coordenada y .

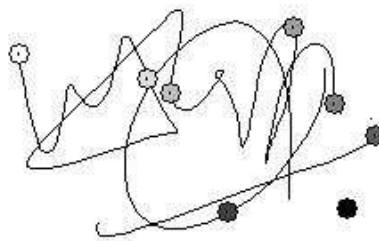


Figura 4.14 - Assinatura com 8 pontos da trajetória amostrados

4.2.4.6 Simetria da assinatura

Cada assinatura possui uma forma particular de se distribuir geograficamente em relação ao centro de massa, algumas sendo mais simétricas e outras sendo distribuídas mais para um lado do que para outro. Na figura 4.15 é mostrada uma assinatura que é relativamente simétrica no sentido vertical, mas no sentido horizontal ela se distribui mais para a direita.

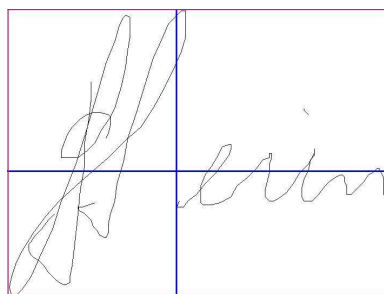


Figura 4.15 - Assinatura com maior simetria vertical

Esta assinatura difere da assinatura mostrada na figura 4.16, que é bastante simétrica no sentido horizontal mas em relação ao sentido vertical ela se distribui muito mais na parte superior do que na parte inferior.

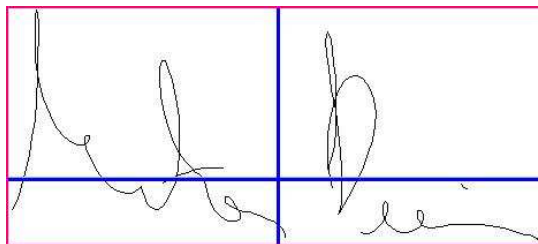


Figura 4.16 - Assinatura com maior simetria horizontal

A forma como uma assinatura se distribui espacialmente em relação ao centro de massa é um atributo que pode ser utilizado no processo de autenticação de assinaturas, e é calculado da seguinte forma: mede-se a distância do centro de massa da assinatura em relação aos extremos superior, esquerdo, inferior e direito da assinatura. Os valores obtidos são normalizados entre 0 e 1 para ficarem independentes das dimensões da assinatura. Este atributo é mais importante para a identificação das assinaturas devido ao fato dele ter relação direta com o formato de uma assinatura, mas também acrescenta importantes informações no processo de autenticação de assinaturas. Esta técnica foi utilizada por OSORIO [16] para a identificação de caracteres impressos.

4.2.4.7 Densidade da assinatura com informações de grade

Este atributo acrescenta importantes informações relacionadas com a forma de uma assinatura. Ele é calculado da seguinte forma: divide-se a imagem da assinatura em diversas células de um determinado tamanho, como se fosse um tabuleiro de xadrez. Para cada célula, calcula-se o número de pontos da assinatura que se encontram dentro desta célula. Os valores encontrados dentro de cada célula são normalizados entre 0 e 1, sendo 0 para a célula que tiver o menor número de pontos e 1 para a célula que tiver o maior número de pontos. Na figura 4.17 é demonstrado o uso desta técnica. Para uma melhor visualização dos resultados, os valores obtidos foram desenhados em tons de cinza no fundo das células, onde um tom claro representa uma baixa densidade de pontos e um tom escuro uma alta densidade de pontos.

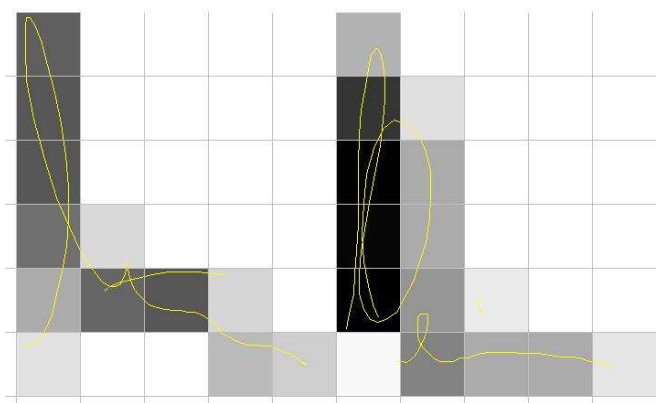


Figura 4.17 - Densidade de pontos com informações de grade

Esta técnica foi adaptada dos trabalhos de BALTZAKIS e PAPAMARKOS [10] no reconhecimento *off-line* de assinaturas. A altura e a largura das células podem ser especificadas nos parâmetros do sistema. O tamanho das células é um importante fator a ser considerado, pois se as células forem muito grandes não será possível diferenciar

todas as assinaturas, e se as células forem muito pequenas o sistema terá dificuldades para generalizar, pois como cada célula representa uma entrada na Rede Neural, o número de entradas necessário seria muito grande.

4.2.4.8 Interseções de linhas verticais em relação a assinatura

A partir da assinatura desenhada em um *bitmap* são traçadas linhas verticais imaginárias cortando a assinatura em intervalos fixos em relação ao tamanho total da imagem. Para cada linha imaginária, são contadas quantas vezes esta linha intersecciona a assinatura. Esta técnica foi utilizada por OSORIO [16] para o reconhecimento visual de caracteres impressos. A distância entre as linhas pode ser configurada nos parâmetros do sistema.

4.2.4.9 Interseções de linhas horizontais em relação a assinatura

Similar a interseção de linhas verticais, só que utilizando linhas horizontais. Na figura 4.18 vemos as linhas imaginárias de interseção verticais e horizontais em relação a assinatura. Estes dois atributos costumam ser utilizados em conjunto.

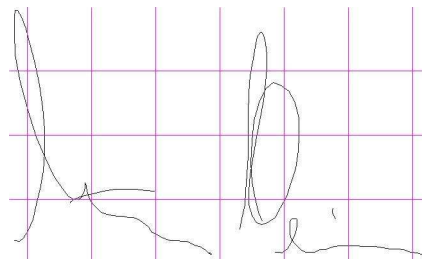


Figura 4.18 - Linhas verticais e horizontais que interseccionam uma assinatura

4.2.4.10 Número de vetores apontando para cada um dos pontos geográficos

Uma assinatura é uma figura bastante complexa, com diversas curvaturas em sua trajetória. Para uma análise dos atributos geométricos de uma assinatura, um artifício que pode ser utilizado é a extração da forma estrutural da assinatura, que é uma simplificação da forma original. Na figura 4.19 é mostrada a forma estrutural da assinatura da figura 4.5, vista anteriormente neste capítulo.

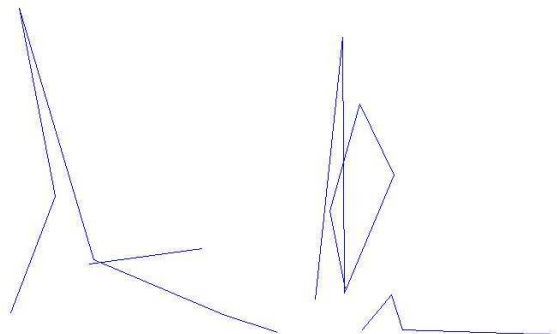


Figura 4.19 - Forma estrutural de uma assinatura

Para se obter a forma estrutural de uma assinatura, são calculados os pontos chave da assinatura, que são o início e o término de cada segmento e os pontos onde houve troca de sentido do traçado em relação aos eixos x ou y . Uma troca de sentido do traçado é quando o valor de uma coordenada estava em crescimento e começou a decrescer, ou vice-versa. Depois de calculados os pontos chave, estes são interligados através de retas, formando a estrutura básica da assinatura. No cálculo dos pontos onde houve trocas de sentido do traçado, uma distância mínima entre cada ponto chave é estipulada para

evitar que pequenas ondulações na trajetória da assinatura tornem a forma estrutural muito detalhada. Quanto maior for esta distância, mais simplificada será a forma estrutural da assinatura.

As retas que compõe a forma estrutural de uma assinatura são uma coleção de vetores, que possuem comprimento, orientação e sentido. Um atributo que pode ser utilizado na autenticação de assinaturas é o número de vetores que apontam para cada ponto geográfico da assinatura. Se fizermos uma analogia com os pontos cardeais, podemos considerar o topo da assinatura como sendo o norte, o lado esquerdo como sendo o oeste, a parte de baixo o sul e a direita o leste. Na figura 4.20 vemos os pontos geográficos traçados em relação a uma assinatura. Com esta analogia é possível determinar para cada vetor qual o ponto geográfico que este aponta com maior intensidade, e depois podem ser contabilizados quantos vetores apontam para cada um dos pontos geográficos. Este atributo é muito útil na autenticação de assinaturas, pois está diretamente relacionado a forma da assinatura, sofrendo poucas alterações entre assinaturas de um mesmo usuário e sendo independente da posição e da escala da assinatura.

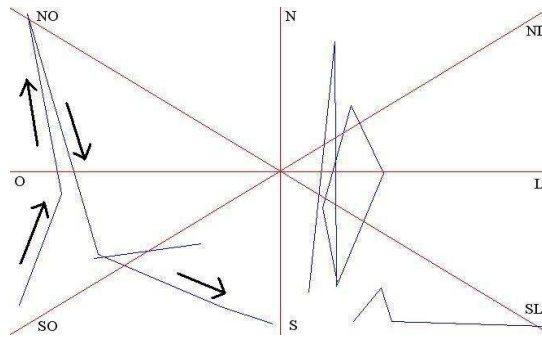


Figura 4.20 - Analogia de uma assinatura com os pontos cardeais

4.2.4.11 Soma do comprimento dos vetores para cada ponto geográfico

Similar ao atributo anterior, mas ao invés de se calcular o número de vetores que apontam para cada ponto geográfico, calculamos a soma do comprimento dos vetores que apontam para cada ponto geográfico. Os valores obtidos são normalizados entre 0 e 1 para que fiquem independentes da escala da assinatura. O diferencial introduzido por este atributo é a noção de grandeza, que indica para qual sentido a assinatura se desloca com mais intensidade, tendo assim uma relação direta com a trajetória da assinatura.

Estes são os atributos mais importantes implementados no protótipo do Sistema NeuralSignX. Abaixo são descritos vários atributos que foram levantados, estudados e implementados, mas nos testes preliminares foi constatado que as informações que eles forneciam já estavam contidas em outros atributos, ou não eram relevantes para o processo de classificação.

4.2.4.12 Número de pontos coletados pelo tablet

Este atributo fornece informações de grandeza temporal, pois quanto mais tempo um usuário demora para assinar, mais pontos serão coletados. Este atributo foi excluído porque em computadores mais lentos o número de pontos coletados em um mesmo intervalo de tempo é menor do que em computadores mais rápidos, o que torna o sistema dependente do *hardware*. Este atributo foi substituído pelo tempo de duração da assinatura, que consegue fornecer as mesmas informações temporais sem ser dependente do *hardware*.

4.2.4.13 Comprimento total da assinatura

É a distância total que a caneta percorre durante o traçado da assinatura. Para se obter o comprimento, calcula-se o somatório da distância de cada ponto amostrado com seu sucessor. Este atributo foi excluído porque é dependente da escala da assinatura, e se torna redundante quando os atributos tempo de duração da assinatura e velocidade média já estão sendo utilizados, uma vez que o comprimento da assinatura é possível de ser obtido através destes dois atributos.

4.2.4.14 Número de trocas de sentido da caneta nos eixos x e y

Este atributo conta quantas vezes a caneta trocou de direção em relação aos eixos x e y. Uma troca de direção é quando o valor da coordenada x estava em crescimento e começou a decrescer ou vice-versa. Um limite mínimo é determinado para evitar que pequenas ondulações no sentido do traçado sejam reconhecidas como uma troca de sentido. Este atributo foi descartado porque as informações obtidas a partir dele já estavam contidas no atributo Número de vetores apontando para os pontos geográficos.

4.2.4.15 Densidade de pontos da assinatura por quadrantes

Se forem traçadas duas linhas imaginárias, uma horizontal e uma vertical, passando pelo centro de massa de uma assinatura, esta assinatura é dividida em quatro regiões, conforme ilustrado na figura 4.21. Um atributo que foi levantado e implementado é a densidade de pontos da assinatura em cada um dos quadrantes, que difere da densidade de pontos da assinatura com informações de grade apenas na forma de divisão as regiões. Este atributo foi descartado porque trazia quase as mesmas informações que a densidade de pontos com informações de grade, e não era tão eficiente quanto este último.

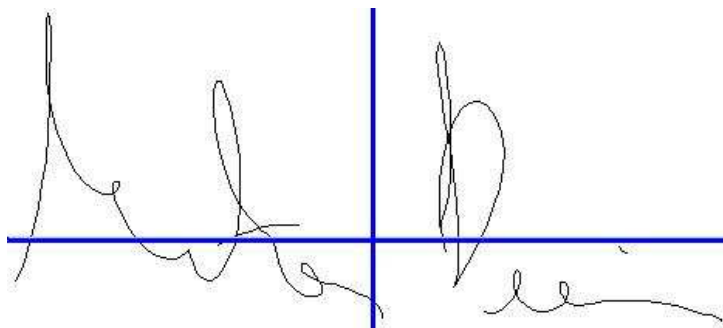


Figura 4.21 - Divisão de uma assinatura a partir do centro de massa

Abaixo são listados alguns atributos que foram levantados a partir da bibliografia e que podem vir a ser implementados em trabalhos futuros:

4.2.4.16 Número de cruzamentos

Este atributo conta quantos pontos de interseção existem na trajetória da assinatura, ou seja, conta quantas vezes a linha que descreve a assinatura corta ela mesma. Este atributo foi utilizado por BALTZAKIS e PAPAMARKOS em [10].

4.2.4.17 Número de laços fechados da assinatura

Este atributo informa a quantidade de laços fechados que uma assinatura apresenta. Este atributo foi utilizado nos trabalhos de BALTZAKIS e PAPAMARKOS [10] e é calculado a partir do número de cruzamentos e do número de extremidades da assinatura.

4.2.4.18 Pressão média da caneta

Este atributo, que necessita de um *tablet* com sensor de pressão para ser implementado, informa a pressão média que o usuário exerceu sobre a caneta durante a trajetória da assinatura.

4.2.4.19 Pressão máxima da caneta

Similar ao atributo anterior, mas considerando a pressão máxima ao invés da pressão média.

4.2.4.20 Número de interseções da assinatura em relação a linhas diagonais

Similar ao número de interseções verticais e horizontais em relação a assinatura, mas considerando linhas que cortam a assinatura diagonalmente. Este atributo foi utilizado em por OSORIO [16].

4.2.4.21 Trajetória da assinatura em relação a grade

Divide-se a assinatura em diversas células de tamanho fixo, formando uma grade com a mostrada anteriormente na figura 4.18. Depois disso se verifica qual a ordem em que cada uma das células foi visitada durante a trajetória da assinatura.

O Sistema NeuralSignX permite que sejam escolhidos quais os atributos que serão utilizados em cada simulação, de forma que seja possível determinar quais são os atributos mais relevantes para o processo de classificação. A princípio quanto mais atributos forem utilizados, melhores serão os resultados, desde que cada um dos atributos escolhidos seja capaz de fornecer informações relevantes que não estejam contidas nos demais atributos. Depois de ser realizada a extração dos atributos, o sistema grava o valor obtido nos atributos de cada uma das assinaturas em arquivos de texto que são enviados para o módulo de processamento, para que este realize a classificação das assinaturas.

4.3 Processamento

No Sistema NeuralSignX, o módulo de processamento é implementado através do uso de Redes Neurais do tipo *Cascade-Correlation* [19], devido as diversas vantagens que estas apresentam em relação aos demais modelos de Redes Neurais Artificiais, descritas em [18],[19],[34],[35]. Pelo fato de ser baseado em técnicas de aprendizado, o módulo de processamento possui duas etapas distintas. Na primeira etapa é realizado o aprendizado da Rede Neural, onde toda a base de dados de assinaturas é submetida ao simulador Neusim, que realiza o treinamento de forma que o sistema aprenda como autenticar as assinaturas de determinado usuário. Na segunda etapa, que ocorre depois que o processo de aprendizado foi concluído, é realizada o reconhecimento das assinaturas, onde uma assinatura é submetida ao sistema, e baseado no conhecimento que a Rede Neural adquiriu na etapa de aprendizado, é realizada a classificação que informa se a assinatura submetida pertence ou não ao usuário. Cabe salientar que o módulo de processamento não recebe as assinaturas de forma direta, mas sim o valor dos atributos extraídos no módulo de pré-processamento.

4.4 Protótipo do Sistema NeuralSignX

O protótipo do Sistema NeuralSignX é um conjunto de aplicações que implementam as principais funcionalidades do sistema, de forma a ser possível realizar experimentos e avaliar o desempenho do modelo proposto. As principais aplicações que compõe o protótipo do Sistema NeuralSignX são WinTablet, que implementa o módulo de

entrada, Feature, que implementa o módulo de pré-processamento, e FinalCut, que implementa o módulo de processamento na fase de reconhecimento.

4.4.1 Módulo de entrada

A aplicação que implementa o módulo de entrada do Sistema NeuralSignX chama-se WinTablet. Quando ela é iniciada, a caixa de diálogo mostrada na figura 4.22 é exibida para que seja informado o nome do usuário que irá fornecer as assinaturas.



Figura 4.22 - Caixa de mensagem para informar o nome do usuário

Depois que o usuário digitar o seu nome e pressionar o botão "OK", é carregada a tela principal do WinTablet, mostrada na figura 4.23. Na parte superior da tela temos um painel com quatro botões que servem para comandar a aplicação. O primeiro botão, da esquerda para a direita, inicia uma nova assinatura, o segundo botão salva a assinatura corrente, o terceiro botão cancela a assinatura corrente e o quarto botão encerra a aplicação. Entre os botões é exibido o nome do usuário e quantas assinaturas já foram coletadas desde que a aplicação foi iniciada. Abaixo do painel superior existe uma área retangular que é responsável por captar as coordenadas do *tablet* e exibir a assinatura depois que ela foi concluída.

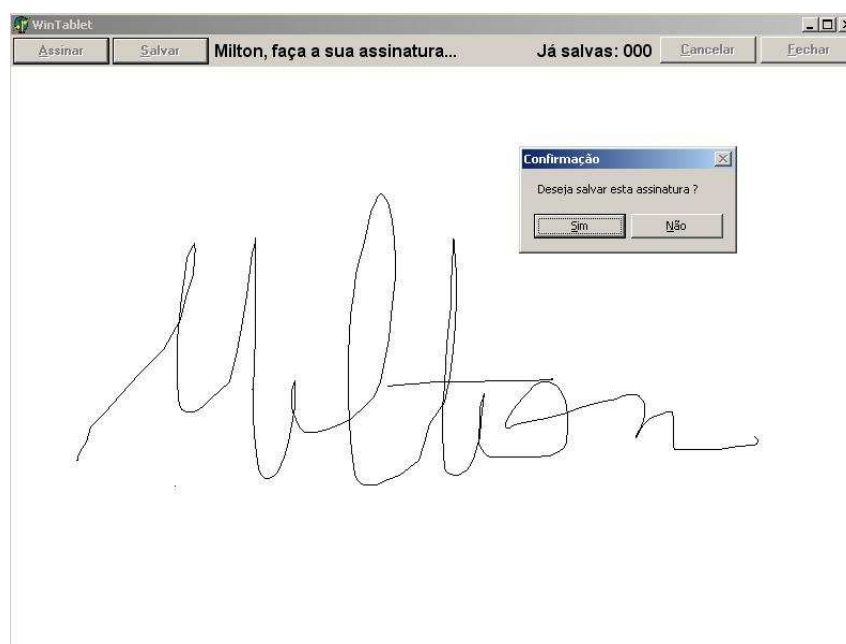


Figura 4.23 - Módulo de entrada do Sistema NeuralSignX

O processo de coleta de assinaturas funciona da seguinte forma: depois que o usuário digitou o seu nome e iniciou a aplicação, ele pressiona o botão "Assinar" para iniciar uma nova assinatura. A medida que a assinatura vai sendo desenhada, o cursor do sistema operacional vai se movimentando de forma a percorrer a trajetória da assinatura. Isto ocorre porque o *tablet* utilizado tem a mesma funcionalidade de um *mouse*, e a assinatura é capturada pelo protótipo amostrando-se as coordenadas do cursor do

sistema operacional. Quando a assinatura for concluída, o usuário pressiona o botão “Salvar”, que faz com que a assinatura seja mostrada na tela e uma caixa de diálogo, mostrada na figura 4.23, pergunta ao usuário se ele deseja salvar a assinatura. Se o usuário pressionar o botão “Sim”, a assinatura é salva na base de dados de assinaturas. O usuário também pode pressionar o botão “Cancelar” durante a assinatura se cometer algum erro no traçado, e assim pode reiniciar a assinatura.

4.4.2 Módulo de pré-processamento

A aplicação do protótipo que implementa o módulo de pré-processamento do Sistema NeuralSignX chama-se Feature. Esta aplicação lê o arquivo de assinaturas gerado pelo módulo de entrada, realiza o pré-processamento e gera os arquivos de saída com o valor dos atributos que serão utilizados pelo módulo de processamento. A tela principal do módulo de pré-processamento é mostrada na figura 4.24.

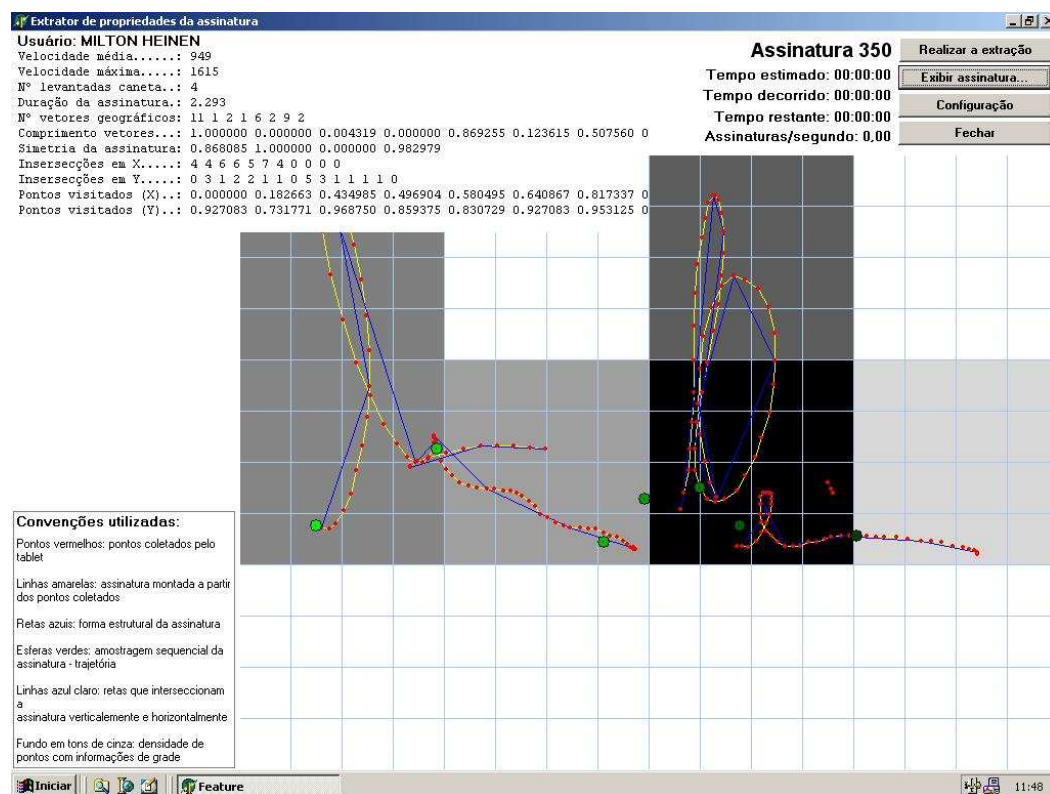


Figura 4.24 - Tela principal do módulo de pré-processamento

No canto superior esquerdo da tela é exibido o nome do usuário que desenhou a assinatura que está sendo analisada, e abaixo do nome são exibidos os valores dos atributos obtidos a partir da assinatura. No canto superior direito estão os botões utilizados para comandar a aplicação, e a esquerda dos botões são mostradas diversas informações relacionadas com o andamento do processo de extração, como o número de assinaturas que já foram analisadas, o número total de assinaturas, o tempo estimado de duração do processo e a velocidade do processo de extração. No centro da tela é exibida a imagem da assinatura que está sendo analisada, e sobre ela são desenhadas informações relativas aos seus atributos, como a forma estrutural e a densidade de pontos. No canto inferior direito existe uma legenda que descreve o significado de cada um dos elementos visuais presentes na imagem.

O funcionamento da aplicação é bastante simples. O primeiro botão no canto superior direito, com o rótulo “Realizar a Extração”, dispara o processo de extração dos atributos

a partir da base de dados de assinaturas e gera os arquivos de saída para o módulo de processamento. As assinaturas vão sendo mostradas na tela a medida que o processo de extração avança, para que seja possível verificar visualmente o andamento da extração dos atributos, e também para que seja possível realizar uma conferência visual dos valores obtidos. O segundo botão, com o rótulo “Exibir assinatura...”, permite que o usuário selecione uma assinatura pelo número sequencial dela, que é exibida na tela junto com os valores dos atributos apenas para conferência visual, sem que sejam gerados arquivos na saída. O quarto botão, com o rótulo “Fechar”, cancela a extração dos atributos e encerra a aplicação, e o terceiro botão, com o rótulo “Configuração”, chama a tela de configuração do protótipo, que é mostrada na figura 4.25.

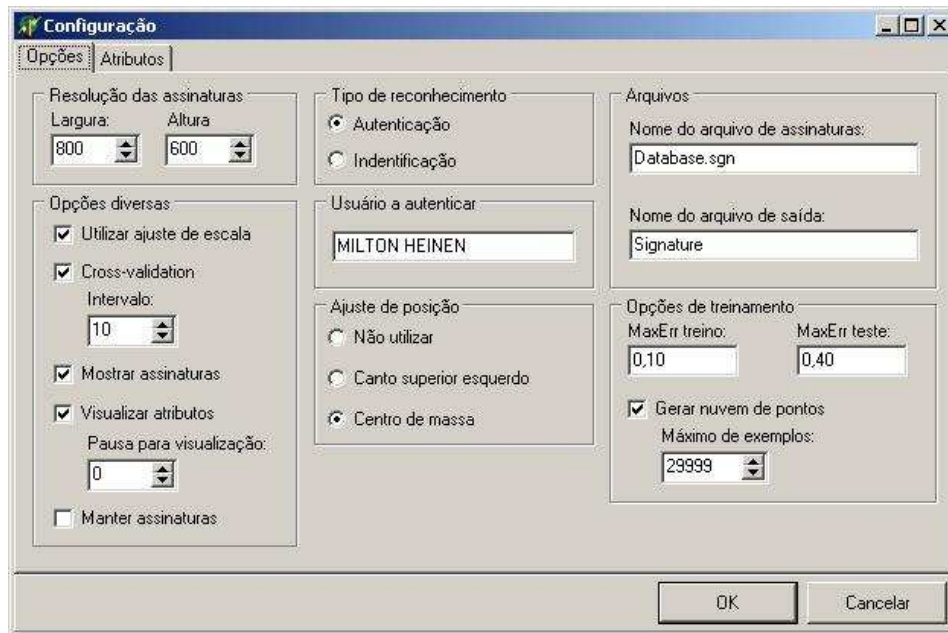


Figura 4.25 - Tela de configuração do protótipo

A tela de configuração é utilizada não apenas pelo módulo de pré-processamento, mas por todas as aplicações do protótipo. A primeira página da tela de configuração, mostrada na figura 4.25, serve para configurar os seguintes parâmetros:

- Resolução das assinaturas: é a resolução com a qual as assinaturas serão salvas e analisadas. Este atributo é utilizado pelo módulo de entrada e pelo módulo de pré-processamento e não deve ser alterado depois que já foram coletadas assinaturas. O valor padrão é 800 x 600;
- Utilizar ajuste de escala: se estiver ativado o ajuste de escala será realizado antes da extração dos atributos;
- Cross-validation: se for utilizado, a base de dados de assinaturas será dividida entre assinaturas utilizadas no aprendizado e assinaturas utilizadas para o teste de generalização, quando forem gerados os arquivos de saída;
- Intervalo: é forma como as assinaturas serão divididas entre aprendizado e generalização. Se for escolhido 10, por exemplo, as primeiras dez assinaturas serão utilizadas para o aprendizado, as dez seguintes para o teste de generalização, e assim por diante. A proporção entre as assinaturas é de 50% para cada conjunto. O valor padrão para este atributo é 1.

- Mostrar assinaturas: se estiver ativado, as assinaturas vão sendo exibidas na tela a medida que o sistema extrai os atributos. É possível desativar a visualização das assinaturas para melhorar a performance do sistema;
- Visualizar atributos: se estiver ativado, informações visuais relacionadas com os valores dos atributos vão sendo exibidas na tela a medida que o sistema extrai os atributos. Pode ser desativado para melhorar a performance em termos de velocidade de extração;
- Pausa na visualização: permite que seja determinado um tempo de espera entre a análise das assinaturas para que seja possível visualizar os atributos na tela. Em computadores mais velozes a extração dos atributos é tão rápida que não permite que o usuário visualize o valor dos atributos;
- Manter assinaturas: se estiver ativado, o sistema desenha a próxima assinatura na tela sem apagar a anterior, a não ser que ela pertença a um outro usuário. Isto é útil para verificar de forma visual as variações que ocorrem entre assinaturas de um mesmo usuário;
- Tipo de reconhecimento: permite que o protótipo seja configurado para trabalhar com identificação de usuários ou com autenticação de usuários. A identificação de usuários não é o objetivo deste trabalho, mas foi introduzida no protótipo apenas para uma melhor avaliação de alguns atributos. A diferença está no número de saídas da Rede Neural, de forma que na identificação de usuários são geradas múltiplas saídas, uma para cada usuário da base dados de assinaturas, e na autenticação de assinaturas são geradas apenas duas saídas, a primeira indicando o nível de aceitação de uma assinatura e a segunda o nível de rejeição de uma assinatura, como será visto mais adiante na seção 4.4.3;
- Usuário a autenticar: é o usuário para o qual se deseja que o sistema aprenda a reconhecer as assinaturas. O aprendizado deve ser realizado para cada usuário que se deseje autenticar;
- Ajuste de posição: define se será utilizado ajuste de posição e se ele será pelo canto superior esquerdo ou pelo centro de massa da assinatura;
- Nome de arquivo de assinaturas: este parâmetro é utilizado pelo módulo de entrada e pelo módulo de pré-processamento, e define o nome do arquivo no qual serão salvas as assinaturas coletadas pelo módulo de entrada. O nome padrão é Database.sgn;
- Nome do arquivo de saída: é o nome sem a extensão que será utilizado pelos arquivos de saída do módulo de pré-processamento. O nome padrão é Signature;
- MaxErr treino: este parâmetro é utilizado pelo módulo de processamento e informa qual o valor do erro máximo utilizado na fase de aprendizado pelo simulador Neusim;
- MaxErr teste: este parâmetro é utilizado pelo módulo de processamento e informa qual o valor do erro máximo utilizado na fase de reconhecimento pelo simulador Neusim;
- Gerar Nuvem de Pontos: se estiver ativado, o sistema gera saídas com valores aleatórios para os atributos. A Nuvem de Pontos é uma técnica utilizada neste trabalho para lidar com classes fechadas, como foi visto na seção 3.5.

- Máximo de exemplos: é o número máximo de exemplos que poderão ser gerados pela Nuvem de Pontos. O limite máximo do Neusim é 29.999.

A segunda página da tela de configuração do protótipo é mostrada na figura 4.26, e serve para definir quais os atributos que serão utilizados nas simulações, além das configurações específicas utilizadas por alguns atributos.

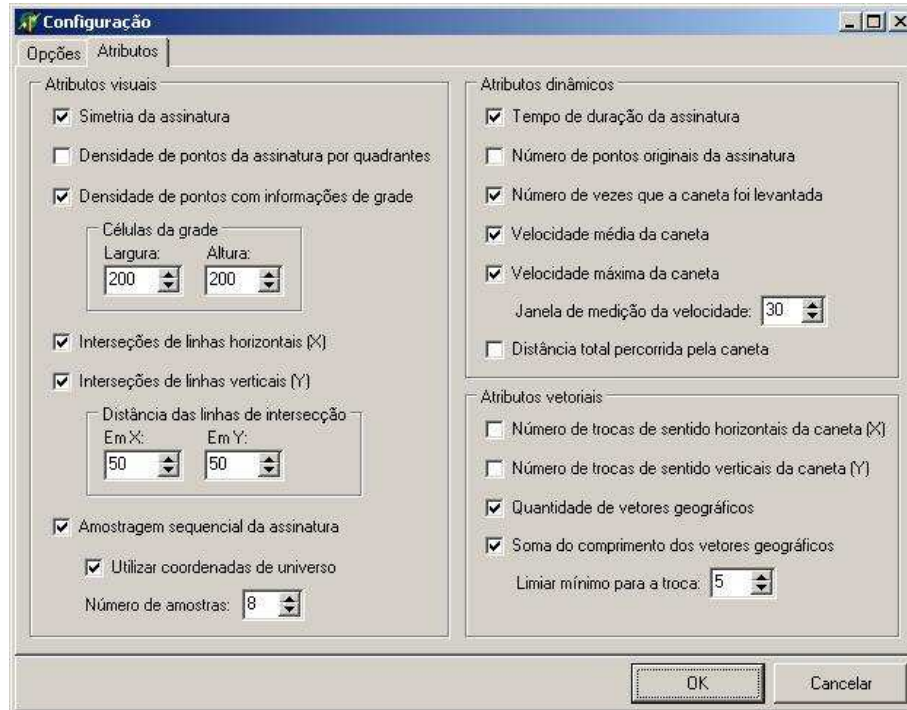


Figura 4.26 - Tela de escolha dos atributos serão utilizados nas simulações

As caixas de seleção servem para que sejam escolhidos quais os atributos que serão utilizados nas simulações. Os ajustes que podem ser realizados sobre os diversos atributos são:

- Tamanho das células da grade: define a altura e a largura utilizada pelas células do atributo densidade da assinatura com informações de grade. O valor padrão é 200 x 200;
- Distância das linhas de interseção: distância entre as linhas verticais e horizontais utilizadas no atributo interseções de linhas verticais e horizontais em relação a assinatura;
- Utilizar coordenadas de universo: se estiver ativado, os pontos da amostragem sequencial da assinatura são enviados para o módulo de processamento com valores normalizados entre 0 e 1;
- Número de amostras sequenciais: define quantos pontos serão amostrados pelo atributo amostragem sequencial da assinatura;
- Janela de medição da velocidade: define o tamanho das sub-regiões utilizadas no cálculo da velocidade máxima da assinatura. O valor padrão é 30;
- Limiar mínimo para a troca: utilizado no cálculo dos pontos onde houveram trocas de sentido do traçado. Define a distância mínima entre cada ponto para impedir que pequenas ondulações sejam reconhecidas como uma troca de sentido. O valor padrão é 5.

Para que seja utilizada a aplicação Feature, que implementa o módulo de pré-processamento do Sistema NeuralSignX, é necessário que a base de dados de assinaturas esteja no mesmo diretório da aplicação e também é necessário que o nome do arquivo da base de dados de assinaturas esteja configurado corretamente nos parâmetros do Sistema NeuralSignX.

Para executar a aplicação Feature, inicia-se ela, e em seguida pressiona-se o botão configurar caso seja necessário fazer alguns ajustes nos parâmetros. Depois os ajustes que foram realizados, pressiona-se o botão “OK” para voltar a tela principal da aplicação. Na tela principal, pressiona-se o botão “Realizar a extração”, que irá extrair os atributos das assinaturas contidas na base e gerar os arquivos de saída utilizados pelo módulo de processamento. Quando a extração estiver concluída, basta encerrar a aplicação e passar para a etapa de processamento do Sistema NeuralSignX.

4.4.3 Processamento

O módulo de processamento é implementado através do uso do simulador de Redes Neurais Neusim, desenvolvido por OSORIO [24]. Na etapa de pré-processamento, vista anteriormente, são gerados dois arquivos de saída, um com a extensão LRN e outro com a extensão TST. Estes arquivos são escritos no formato aceito pelo Neusim, e contém os valores dos atributos das assinaturas utilizadas para o aprendizado e para o teste de generalização. A estruturação dos arquivos ocorre da seguinte forma: as entradas correspondem aos valores dos atributos extraídos das assinaturas, podendo cada atributo representar uma ou mais entradas; As saídas informam para qual o usuário se deseja realizar o aprendizado, sendo que para este usuário a saída desejada é “1 0”, e para os demais a saída desejada é “0 1”. O motivo de se utilizar duas saídas é minimizar os problemas decorrentes do reconhecimento de classes fechadas e assim reforçar a segurança do sistema.

Ao gerar os arquivos de saída, o módulo de pré-processamento automaticamente altera os principais parâmetros do Neusim, como o número de entradas, o número de saídas, a tarefa e o erro máximo, de forma que o usuário não precise alterar manualmente estes parâmetros no arquivo de configuração do Neusim. O número de entradas é definido conforme o número de atributos escolhidos e o número de saídas depende do tipo de reconhecimento (2 saídas para a autenticação de usuários e uma saída por usuário para a identificação de assinaturas). A tarefa é definida conforme a fase em que se encontra o módulo de processamento. Se estiver na fase de aprendizado, a tarefa será LGWSOT, e se estiver na fase de autenticação, a tarefa será POT. O erro máximo também é alterado conforme a fase em que se encontra o módulo de processamento, de forma que o usuário pode cadastrar nos parâmetros do Sistema NeuralSignX os valores que serão utilizados na fase de aprendizado e na fase de reconhecimento do módulo de processamento.

A execução do módulo de processamento na fase de aprendizado é bastante simples, basta digitar na linha de comando Neusim Signature (onde Signature é o nome dos arquivos utilizados pelo Neusim sem as extensões). O Neusim iniciará, carregará os arquivos necessários, realizará o aprendizado e criará os arquivos TOP, com a topologia da Rede Neural, WTS, com o valor dos pesos sinápticos, e o arquivo OUT, com os valores de ativação da Rede Neural para cada um dos exemplos da base de teste de generalização. O arquivo OUT é útil para que seja possível verificar visualmente quais assinaturas o sistema não conseguiu classificar corretamente. Se o aprendizado não for satisfatório, pode ser repetida a simulação alterando alguns parâmetros ou atributos utilizados (se forem alterados os parâmetros do NeuralSignX, é necessário extrair os

atributos novamente). Quando o aprendizado estiver concluído, basta encerrar o Neusim e passar para a próxima fase, que é a fase de reconhecimento.

4.4.4 Fase de reconhecimento

Para a fase de reconhecimento do NeuralSignX, foi desenvolvida uma aplicação especial chamada FinalCut, que integra todos os módulos do sistema desde a entrada até a emissão dos resultados. Esta aplicação utiliza os arquivos da topologia (TOP) e de pesos (WTS) gerados em tempo de aprendizado e realiza a classificação das assinaturas em tempo real, informando se as assinaturas pertencem ao usuário ou não. Na figura 4.28 é mostrada a tela principal da aplicação FinalCut.

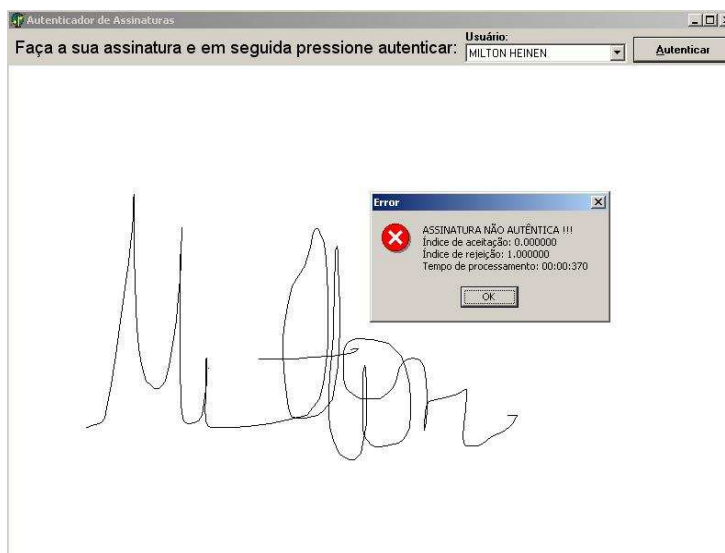


Figura 4.27 - Tela principal do programa que realiza a autenticação das assinaturas

O funcionamento da aplicação FinalCut ocorre da seguinte forma: o usuário inicia a aplicação, desenha a assinatura e pressiona o botão “Autenticar”. O sistema lê os dados provenientes do *tablet*, realiza o pré-processamento, chama o Neusim para realizar a classificação e emite os resultados, tudo de forma automática sem a intervenção do usuário. Ao lado do botão autenticar é mostrado o nome do usuário que está sendo autenticado no momento. É importante ressaltar que para trocar o usuário corrente, é preciso repetir a etapa de aprendizado ou carregar o arquivo de pesos salvo anteriormente para o novo usuário. Os resultados emitidos após a autenticação são:

- Parecer: Assinatura autêntica ou não autêntica;
- Índice de aceitação: Valor obtido na primeira saída da Rede Neural;
- Índice de rejeição: Valor obtido na segunda saída da Rede Neural;
- Tempo de processamento: Tempo que demorou do instante que o usuário pressionou o botão “autenticar” até ser emitido o resultado da autenticação.

Para validar o modelo proposto, foram necessárias diversas simulações com o Sistema NeuralSignX, utilizando uma base de dados com assinaturas de diversas pessoas. No próximo capítulo serão descritas as diversas simulações realizadas com o protótipo do Sistema NeuralSignX.

5 Resultados obtidos

Neste capítulo serão descritos os testes realizados para a validação do Sistema NeuralSignX, bem como os resultados obtidos nestes testes. Inicialmente será descrita a forma como a base de dados de assinaturas foi obtida, em seguida serão discutidos os testes realizados para a configuração dos parâmetros do protótipo, e os experimentos realizados para a escolha dos atributos mais relevantes. Por último serão descritas as diversas simulações realizadas com a finalidade de validar o sistema proposto, bem como os resultados obtidos nestas simulações.

5.1 Base de dados de assinaturas

A base de dados de assinaturas utilizada nas simulações foi coletada ao longo de quatro meses, e é composta por 2550 assinaturas. Cada usuário contribuiu com aproximadamente 30 assinaturas, mas alguns usuários contribuíram com mais assinaturas, para que fosse possível realizar uma avaliação do desempenho do sistema utilizando diferentes quantidades de assinaturas por usuário. A composição da base de dados de assinaturas ficou da seguinte forma:

- 1350 assinaturas autênticas, coletadas de 46 usuários diferentes;
- 950 assinaturas pictográficas desenhadas por 12 pessoas diferentes, que reproduzem elementos gráficos como desenhos, letras isoladas, palavras, linhas, curvas e outros elementos visuais;
- 250 falsificações realizadas com as assinaturas de oito usuários diferentes, divididas em 130 falsificações traçadas e 120 falsificações especializadas. As falsificações foram realizadas por seis pessoas diferentes com o consentimento prévio dos autores originais das assinaturas.

As assinaturas autênticas foram coletadas com a finalidade de verificar o funcionamento do sistema quando submetido a assinaturas reais de diversos usuários diferentes, e as assinaturas pictográficas foram coletadas com a finalidade de verificar como o sistema se comporta quando submetido a assinaturas fora do padrão usual, que tanto podem ser assinaturas utilizadas no dia a dia por algum usuário como podem ser tentativas de invasão do sistema através de fraudes aleatórias. Tecnicamente falando, uma assinatura pode ser qualquer padrão desenhado a mão livre por um usuário, sem a necessidade de ter qualquer relação com seu nome.

As assinaturas falsificadas foram coletadas com a finalidade de verificar a robustez do sistema quando submetido a assinaturas muito próximas do padrão original analisado. As falsificações traçadas foram desenhadas percorrendo-se a trajetória de diversas assinaturas desenhadas em papel, onde para cada modelo de assinatura a ser falsificado existiam três amostras de assinaturas desenhadas no papel. Cinco pessoas diferentes percorreram a trajetória de cada uma das assinaturas, resultando num valor médio de 15 falsificações traçadas por usuário. Com esta técnica foi possível obter uma maior variabilidade nas falsificações, que evita que o sistema simplesmente decore o padrão de falsificação de uma determinada pessoa. As falsificações especializadas também foram desenhadas de forma semelhante, com diversas pessoas falsificando cada modelo

de assinatura, mas ao invés de simplesmente percorrer a trajetória da assinatura sobre o papel, cada pessoa praticou até que fosse possível desenhar a assinatura da forma mais aceitável possível. A qualidade de cada falsificação variou de acordo com o talento da pessoa que desenhou e com o formato da assinatura, pois assinaturas diferentes requerem graus diferentes de habilidade para serem reproduzidas a contento.

5.2 Configuração dos parâmetros do sistema

O Sistema NeuralSignX possui diversos parâmetros, e para determinar qual a configuração ideal para cada um deles foram necessários vários testes até que se chegasse aos valores ideais. O critério utilizado para determinar qual a melhor configuração do sistema foi o grau de generalização obtido na saída do módulo de processamento, e cada parâmetro foi avaliado de forma independente, para que a configuração de um não influísse nos demais. Por questões de espaço não serão descritos aqui todos os testes realizados, mas os valores ideais obtidos são mostrados na tabela 5.1. Alguns parâmetros, como a resolução das assinaturas, foram determinados de forma empírica baseado em critérios técnicos, sem que fosse necessário a realização de simulações.

Parâmetro	Valor
Resolução das assinaturas	800 x 600
Intervalo do <i>Cross-validation</i>	1
Utilizar ajuste de escala	Sim
Ajuste de posição utilizado	Pelo centro de massa da assinatura
MaxErr treino	0,1
MaxErr teste	0,4
Utilizar Nuvem de Pontos	Sim (pode não ser utilizado em algumas simulações)
Número máximo de exemplos	29.999 (limite máximo do simulador Neusim)

Tabela 5.1 - Valores obtidos para os parâmetros genéricos do Sistema NeuralSignX

Além dos parâmetros genéricos, o Sistema NeuralSignX possui diversos parâmetros que são específicos de alguns atributos. Para a configuração destes parâmetros, foram necessários testes com os atributos de forma isolada, para que os resultados de um atributo não influíssem sobre os demais. Na tabela 5.2 são mostrados os valores obtidos na configuração dos parâmetros específicos de alguns atributos. Os valores são válidos para assinaturas coletadas com uma resolução de 800 x 600.

Atributo	Parâmetro	Valor
Densidade de pontos com informações de grade	Tamanho das células da grade	100 x 100
Interseções de linhas verticais e horizontais	Distância entre as linhas de interseção	25 x 25
Amostragem seqüencial da assinatura	Utilizar coordenadas de universo	Sim
Amostragem seqüencial da assinatura	Número de amostras utilizadas	8
Velocidade máxima da assinatura	Janela de medição da velocidade	30
Número de vetores apontando para os pontos geográficos	Limite mínimo para a troca de sentido	5

Tabela 5.2 - Valores obtidos para os parâmetros específicos de alguns atributos do Sistema NeuralSignX

5.3 Determinação dos atributos a serem utilizados

No capítulo 3 foram descritos os diversos atributos implementados no Sistema NeuralSignX. Alguns atributos foram retirados da bibliografia, outros foram adaptados de outros sistemas para a autenticação de assinaturas, e alguns foram desenvolvidos especialmente para o Sistema NeuralSignX, de forma que não era possível saber previamente quais atributos trariam bons resultados ao processo de autenticação de assinaturas. Além disto, alguns atributos poderiam ser redundantes, ou seja, trazer informações que já estavam contidas nos demais atributos, sendo portanto desnecessários.

Em um sistema de autenticação de assinaturas, o essencial é que se possa obter o máximo de informações possíveis sobre as assinaturas, pois quanto mais informações estiverem disponíveis, melhor será o processo de classificação e mais difícil será para um usuário enganar o sistema. Se um sistema analisar só a velocidade da assinatura, por exemplo, em pouco tempo os usuários descobrirão isto e conseguirão facilmente enganar o sistema com assinaturas que nem se parecem com a original. Se além da velocidade, o sistema analisar o número de trocas de sentido da caneta, a falsificação já se torna mais difícil, e quanto mais informações forem utilizadas, mais robusto o sistema se torna, fazendo com que os usuários não consigam enganar o sistema de forma trivial.

Assim se conclui que o ideal é utilizar o maior número de atributos possível, desde que nenhum atributo seja redundante em relação aos demais. Nesta seção serão utilizadas várias técnicas para verificar a importância de cada um dos atributos. A primeira técnica utilizada é a Análise de Componentes Principais [37], em seguida será utilizada a matriz de correlação, depois serão realizadas simulações com o Sistema NeuralSignX para verificar a utilidade de cada um dos atributos, e por último serão utilizadas as Árvores de Decisão [29] para verificar quais os atributos são mais importantes para a autenticação de determinadas assinaturas.

5.3.1 Análise de Componentes Principais (ACP)

A Análise de Componentes Principais (ACP) [37] é uma técnica estatística que utiliza a correlação entre os diversos campos de uma base de dados para calcular o quanto cada campo contribui para a variabilidade total de um conjunto de exemplos analisados. No caso do Sistema NeuralSignX, cada campo corresponde uma entrada da Rede Neural. O funcionamento desta técnica ocorre da seguinte forma: as entradas submetidas a análise são decompostas em diversos fatores, onde cada fator pode conter diversas entradas em graus diferentes de participação. Para cada fator é calculado o valor próprio (*eigenvalue*), que é o quanto este fator possui de informações próprias, que não estão contidas nos outros fatores. As entradas mais relevantes para o sistema são as que estiverem presentes nos fatores que obtiverem os maiores *eigenvalues*. Dentro de cada fator também é possível verificar o quanto cada entrada contribuiu para a formação do fator, quanto mais alto for o valor, mais relevante é esta entrada.

Para a realização da Análise de Componentes Principais, foi procurada a assessoria estatística do Centro 6 da Unisinos, com o prof. Renato Carlson, que auxiliou na realização da análise e na interpretação dos resultados obtidos. Quando se utiliza todos os atributos do Sistema NeuralSignX, com as configurações ideais, são necessárias 150 entradas na Rede Neural. Para facilitar a análise, foi reduzida a precisão de alguns atributos (aumentando o tamanho das células do atributo densidade de pontos com informações de grade, por exemplo), de forma que foram necessárias apenas 86 entradas

na Rede Neural. O *software* utilizado para a Análise de Componentes Principais foi o SPSS², e os dados de entrada para este *software* são os valores dos atributos para cada uma das assinaturas da base de dados. O *software* decompôs as 86 entradas em 86 fatores e os ordenou de forma decrescente pelo *eigenvalue*, de forma que os 17 primeiros fatores foram responsáveis por 74,207% da variância total, como mostra a tabela 5.3. Por questões de espaço foram mostrados apenas os 20 primeiros fatores, que são os mais importantes.

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	19,223	22,353	22,353	19,223	22,353	22,353
2	7,904	9,191	31,544	7,904	9,191	31,544
3	6,432	7,479	39,023	6,432	7,479	39,023
4	4,948	5,754	44,777	4,948	5,754	44,777
5	3,815	4,436	49,213	3,815	4,436	49,213
6	2,941	3,419	52,632	2,941	3,419	52,632
7	2,609	3,034	55,666	2,609	3,034	55,666
8	2,355	2,738	58,404	2,355	2,738	58,404
9	2,051	2,385	60,789	2,051	2,385	60,789
10	1,922	2,234	63,023	1,922	2,234	63,023
11	1,709	1,988	65,011	1,709	1,988	65,011
12	1,671	1,943	66,953	1,671	1,943	66,953
13	1,479	1,720	68,674	1,479	1,720	68,674
14	1,351	1,570	70,244	1,351	1,570	70,244
15	1,205	1,401	71,645	1,205	1,401	71,645
16	1,171	1,362	73,007	1,171	1,362	73,007
17	1,032	1,200	74,207	1,032	1,200	74,207
18	,996	1,158	75,365			
19	,941	1,094	76,459			
20	,847	,985	77,444			

Tabela 5.3 - Eigenvalues obtidos para os vinte primeiros fatores da Análise de Componentes Principais

O primeiro aspecto importante que se nota a partir da tabela 5.3 é que a variação total está bastante espalhada por todos os fatores da análise, de forma que não se pode facilmente isolar os fatores mais importantes e descartar os demais. Os fatores abaixo do 20º, por exemplo, correspondem isoladamente a menos de 1% da variância total, mas em conjunto correspondem a 22,556%, que é uma variância muito significativa para ser descartada em um sistema de autenticação de assinaturas. O fator que possui o maior *eigenvalue* é o fator 1, responsável por 22,353% de toda a variância do conjunto de exemplos. A tabela 5.4 mostra a composição das entradas no primeiro fator. Por questões de espaço não foram mostradas as contribuições de todas as entradas, mas apenas das que contribuíam com mais de 0,6. A faixa de valores para as contribuições pode variar de 0 a 1, e quanto mais elevado for o valor, mais a entrada participa na formação do fator. Pela análise da tabela 5.4 nota-se que a maior parte dos atributos do sistema tem grande participação na formação deste fator, com valores de participação acima de 0,6. Para os demais fatores a participação também foi bastante distribuída, com grande parte das entradas tendo uma participação maior que 0,5 na formação dos fatores. Segundo o professor Renato Carlson, pelos resultados obtidos se chega a conclusão que a Análise de Componentes Principais pode não ser uma boa técnica para a redução do número de entradas do problema em questão.

² <http://www.spss.com>

Entradas	Contribuição
Velocidade_media:	0,771
Velocidade_maxima:	0,635
Num_pontos_coletados:	-0,744
Duracao_assinatura:	-0,668
Num_trocas_sentido_X:	-0,777
Num_trocas_sentido_Y:	-0,805
Num_vetores_sentido_Norte:	-0,627
Num_vetores_sentido_Sul:	-0,729
Num_vetores_sentido_Sudeste:	-0,606
Num_vetores_sentido_Nordeste:	-0,722
Simetria_Left:	-0,677
Simetria_Botton:	0,645
Simetria_Right:	-0,616
Interseccao_X_2:	0,630
Interseccao_X_6:	-0,725
Interseccao_X_9:	0,715
Interseccao_X_10:	0,689
Densidade_celula_1_2:	0,794
Densidade_celula_1_3:	0,719
Densidade_celula_2_1:	-0,744
Densidade_celula_2_4:	-0,71
Densidade_celula_3_2:	0,791
Densidade_celula_3_3:	0,652

Tabela 5.4 - Entradas que mais contribuíram na formação do primeiro fator da Análise de Componentes Principais

A partir dos resultados obtidos pela Análise de Componentes Principais, se chega a conclusão que a maioria dos atributos utilizados pelo Sistema NeuralSignX são importantes para a classificação das assinaturas, o que faz com que a variância total fique distribuída entre as diversas entradas da Rede Neural. Mas um aspecto que pode ter distorcido os resultados obtidos é a grande variabilidade presente entre as assinaturas de um mesmo usuário, que faz com que a variância total do conjunto de exemplos seja muito grande.

Pela Análise de Componentes Principais se constatou que a redução do número de atributos teria grandes chances de provocar uma perda de informações, e não é isto que se deseja em um sistema de autenticação de assinaturas. Se um atributo servir para identificar uma única assinatura, mesmo assim ele deve ser mantido no sistema, pois com bases de dados maiores, este atributo poderá servir para identificar centenas de usuários. É preferível manter um atributo desnecessário do que descartar um atributo que possa ser necessário para identificar uma determinada assinatura.

Através da Análise de Componentes Principais não foi possível se descobrir quais os atributos não são relevantes para o sistema. Uma outra técnica sugerida por Renato Carlson para a verificação da importância de cada um dos atributos é o uso de uma matriz de correlação, que será vista a seguir.

5.3.2 Matriz de correlação

Uma matriz de correlação é uma tabela na qual são informadas as correlações existentes entre cada uma das variáveis de um conjunto de dados, servindo assim para auxiliar na redução do número de variáveis do problema. Uma variável é candidata a exclusão

quando tiver uma alta correlação com alguma outra variável do problema. No Sistema NeuralSignX, a matriz de correlação pode auxiliar na detecção de atributos que estejam contidos em outros atributos, e portanto não acrescentam informações ao sistema. A partir dos valores dos atributos, onde foi considerada toda a base de dados de assinaturas, foi calculada uma matriz de correlação, e a partir da análise dos resultados obtidos se chegou as seguintes conclusões:

- A velocidade média e a velocidade máxima possuem entre si uma correlação de 85,4%. Entretanto, de acordo com os testes realizados com o Sistema NeuralSignX, ao se remover um destes dois atributos, o índice de generalização do sistema é reduzido, de forma que uma correlação de 85% ainda se requer a presença de ambos os atributos;
- O número de trocas de sentido horizontal possui uma alta correlação com o número de trocas de sentido vertical da assinatura (97,3%), o que pode fazer com que um dos atributos seja desnecessário;
- O número de pontos coletados pelo *tablet* possui uma alta correlação com o tempo de duração da assinatura (96,8%), o que torna possível que se remova um destes atributos;
- O número de trocas de sentido da caneta, tanto vertical quanto horizontal, possui uma alta correlação com o tempo de duração da assinatura (91,3%). Assim, um destes atributos pode ser desnecessário;
- Para as demais entradas, a correlação foi menor que 80%, o que justificaria a permanência dos demais atributos.

Cabe salientar que os valores obtidos na matriz de correlação se referem as assinaturas presentes na base de dados, mas nada impede que para outras assinaturas as correlações obtidas sejam diferentes. Por exemplo, o número de trocas de sentido da caneta apresentou uma alta correlação com o tempo de duração da assinatura, mas pelo conhecimento que se tem a respeito dos atributos, sabe-se que eles possuem uma certa independência. Portanto, fica difícil determinar se as correlações obtidas ocorrem para todos os tipos de assinaturas ou se apenas representam características da maioria das assinaturas presentes na base de dados.

Para os atributos que são responsáveis por mais de uma entrada na Rede Neural, como por exemplo a densidade de pontos com informações de grade, a análise da matriz de correlação não trouxe bons resultados, pois para cada uma das entradas as correlações obtidas foram muito baixas, devido aos atributos estarem distribuídos entre diversas entradas da Rede Neural. Para os atributos que podem ser calculados a partir de outros atributos, como por exemplo a distância percorrida, a matriz de correlação não auxilia na detecção da importância destes atributos. Para verificar a importância dos atributos que não puderam ser analisados através da matriz de correlação, utilizou-se próprio Sistema NeuralSignX para verificar a importância dos atributos, através de um conjunto de simulações que serão descritas a seguir.

5.3.3 Utilização do Sistema NeuralSignX para a seleção de atributos

Na configuração dos parâmetros do sistema, foram realizadas diversas simulações onde os resultados obtidos no aprendizado serviram para medir a performance do sistema, e assim ajustou-se cada parâmetro de forma a atingir os melhores resultados possíveis. Como o sistema usa o *Cascade-Correlation* para a classificação, e este utiliza a correlação no processo de aprendizado, informações como o número de neurônios da

camada oculta, o número de épocas de aprendizado, a taxa de aprendizado e a taxa de generalização podem ser utilizadas para avaliar o desempenho do sistema com ou sem algum atributo específico. Para isto, se verifica quais os resultados obtidos no aprendizado com a presença de todos os atributos, e depois se retira um atributo de cada vez e se comparam os resultados para verificar se houveram alterações na qualidade do aprendizado. Se a ausência de um determinado atributo não causar nenhuma influência significativa sobre os resultados, significa que este atributo não é necessário. Cabe salientar que os resultados obtidos se referem apenas ao conjunto de assinaturas analisado, e não a todo o universo possível de assinaturas.

Para se verificar a importância dos atributos utilizando o Sistema NeuralSignX, é necessário que se compare o desempenho do sistema na autenticação das assinaturas de todos os usuários da base de dados, pois assinaturas de diferentes usuários podem requerer diferentes atributos para serem identificadas. Para evitar que seja necessário realizar testes separados para as assinaturas de cada um dos usuários presentes na base de dados, foi incluída uma nova funcionalidade no Sistema NeuralSignX, que não estava prevista inicialmente. Esta funcionalidade é a identificação de usuários.

5.3.3.1 Identificação de usuários

Para a autenticação de assinaturas, são necessárias uma ou duas saídas na Rede Neural, que servem para indicar se uma assinatura pertence ou não a um determinado usuário. Se quisermos diferenciar as assinaturas de todos os usuários da base de dados entre si, são necessárias várias saídas na Rede Neural, uma para cada usuário. Se houverem cinco usuários cadastrados em um sistema, por exemplo, as saídas desejadas para identificar as assinaturas de cada usuário seriam similares aos valores mostrados na tabela 5.5.

N.º usuário	Nome do usuário	Saída desejada
1	Milton	1 0 0 0 0
2	Marilei	0 1 0 0 0
3	Jalton	0 0 1 0 0
4	Melânia	0 0 0 1 0
5	Arno	0 0 0 0 1

Tabela 5.5 - Exemplos de saídas desejadas para um sistema de identificação de usuários com cinco usuários cadastrados

Cada uma das saídas é responsável pela identificação de um usuário diferente, e para avaliar a resposta que o sistema fornece, basta considerar a saída com o valor mais elevado, descartando as demais. Esta técnica chama-se *winner takes all*. Um sistema de identificação de usuários implementado desta forma é bastante limitado quanto ao número máximo de usuários, e a topologia da Rede Neural precisa ser alterada toda vez que se altera o número de usuários cadastrados no sistema. O objetivo aqui não é propor um sistema de identificação de usuários, mas sim implementar uma técnica que permita determinar quais os atributos são relevantes para o sistema com o menor número de testes possíveis.

5.3.3.2 Atributos similares

Existem duas formas possíveis de se eliminar atributos com a identificação de usuários. A primeira forma é medindo o desempenho do sistema com todos os atributos e depois sem cada um deles, retirados um a um. A segunda forma é identificar os atributos que possuem uma certa similaridade, e verificar se o desempenho deles em conjunto é

melhor do que de forma isolada. Pelo conhecimento que se tem a respeito dos atributos e pelas experimentações práticas, se chega a conclusão que os atributos que possuem uma maior similaridade entre si são:

- Densidade de pontos por quadrantes em relação a densidade de pontos com informações de grade;
- Número de trocas de sentido verticais e horizontais em relação ao número de vetores apontando para cada um dos pontos geográficos;
- Número de vetores apontando para cada um dos pontos geográficos em relação a soma do comprimento dos vetores geográficos;
- Velocidade máxima da caneta em relação a velocidade média da caneta;
- Amostragem sequencial da assinatura em relação a soma do comprimento dos vetores geográficos;
- Densidade de pontos com informações de grade em relação as interseções verticais e horizontais da assinatura;
- Tempo de duração da assinatura em relação ao número de pontos coletados pelo *tablet*;
- Simetria da assinatura em relação a soma do comprimento dos vetores geográficos;
- Velocidade média da caneta, tempo de duração da assinatura e comprimento total da assinatura.

Na escolha dos atributos a serem utilizados pelo sistema, ambas as técnicas foram utilizadas, e assim se chegou a conclusão que os seguintes atributos não fornecem informações relevantes ao processo de classificação:

- Simetria da assinatura;
- Densidade de pontos da assinatura por quadrantes;
- Número de pontos coletados pelo *tablet*;
- Distância total percorrida pela caneta;
- Número de trocas de sentido da caneta verticais e horizontais.

Ao final dos testes, permaneceram no sistema os seguintes atributos, responsáveis por 138 entradas na Rede Neural:

- Densidade de pontos com informações de grade;
- Interseções de linhas verticais e horizontais em relação a assinatura;
- Amostragem sequencial da assinatura;
- Tempo de duração da assinatura;
- Número de vezes que a caneta foi levantada;
- Velocidade média da caneta;
- Velocidade máxima da caneta;
- Número de vetores apontando para cada um dos pontos geográficos;
- Soma do comprimento dos vetores geográficos.

Através da matriz de correlação e dos testes práticos foi possível descobrir quais os atributos são mais importantes para a base de dados como um todo, mas não é possível saber quais as regras que levaram determinado atributo a ser considerado não importante. Também não é possível saber quais os atributos que são mais importantes para a identificação de uma determinada assinatura. Para responder a estas questões, se optou por utilizar as Árvores de Decisão [29], para identificar quais os atributos são mais importantes para determinadas assinaturas, e assim verificar o quanto cada atributo participa na solução do problema. As Árvores de Decisão também podem ser úteis para verificar se os atributos selecionados anteriormente realmente cumprem o seu papel.

5.3.4 Árvores de Decisão

Uma das principais desvantagens das Redes Neurais é que elas não permitem que se visualize de forma trivial o conhecimento adquirido no processo de aprendizado. Quando se deseja saber quais as regras que levaram a tomada de uma certa decisão, outras técnicas de Aprendizado de Máquinas são mais adequadas, como por exemplo as Árvores de Decisão [29]. As Árvores de Decisão são um conjunto de regras *booleanas* do tipo *se/então/senão*, que formam uma árvore onde os nodos mais elevados correspondem as entradas mais relevantes na classificação dos dados de entrada. No Sistema NeuralSignX, foi utilizado o algoritmo C4.5, proposto por QUINLAN [29].

Para que fosse possível verificar a importância dos atributos na autenticação das assinaturas de diversos usuários, foram geradas Árvores de Decisão para alguns usuários da base de dados. As informações enviadas para as Árvores de Decisão são as mesmas enviadas para a Rede Neural, sendo necessário apenas alterar o formato dos arquivos para que sejam aceitos pelo *software* C4.5. Na figura 5.1 é mostrada a árvore de decisão para a autenticação das assinaturas do usuário Arno Heinen.

```

Decision Tree:
Interseccao_Y_3 <= 0 :
| Soma_comprimento_vetores_Norte > 30 : FALSIFICATION (225.0/1.0)
| Soma_comprimento_vetores_Norte <= 30 :
| | Ponto_amostrado_Y_1 <= 84 : FALSIFICATION (22.0/1.0)
| | Ponto_amostrado_Y_1 > 84 : ARNO_JOSE_HEINEN (14.0)
Interseccao_Y_3 > 0 :
| Interseccao_Y_2 > 0 : FALSIFICATION (2153.0)
| Interseccao_Y_2 <= 0 :
| | Ponto_amostrado_Y_5 <= 15 :
| | | Velocidade_media <= 235 : ARNO_JOSE_HEINEN (2.0)
| | | Velocidade_media > 235 : FALSIFICATION (5.0)
| | Ponto_amostrado_Y_5 > 15 :
| | | Interseccao_Y_4 <= 3 : FALSIFICATION (116.0)
| | | Interseccao_Y_4 > 3 :
| | | | Soma_comprimento_vetores_Noroeste <= 0 : ARNO_JOSE_HEINEN (2.0)
| | | | Soma_comprimento_vetores_Noroeste > 0 : FALSIFICATION (11.0)

```

Figura 5.1 - Árvore de decisão para as assinaturas do usuário Arno Heinen

Pela análise da árvore pode-se perceber que os atributos mais importantes para a autenticação das assinaturas deste usuário são:

- Interseção de linhas verticais e horizontais;
- Soma do comprimento dos vetores geográficos;
- Amostragem sequencial da assinatura;
- Velocidade média da assinatura.

Para a identificação deste usuário, bastaria a utilização destes quatro atributos. Mas para um sistema ser genérico, ele precisa ser capaz de reconhecer as assinaturas de qualquer

usuário, portanto serão feitos mais testes com outros usuários. Para o usuário Jorge Felipe, a árvore de decisão obtida é mostrada na figura 5.2. Utilizando apenas os atributos velocidade média da caneta e o número de vezes em que a caneta foi levantada é possível classificar de forma correta 91% dos exemplos.

```
Decision Tree:

Numero_pen_ups <= 17 : FALSIFICATION (1259.0)
Numero_pen_ups > 17 :
|   Velocidade_media <= 376 : FALSIFICATION (61.0)
|   Velocidade_media > 376 : JORGE_FELIPE (30.0)
```

Figura 5.2 - Árvore de decisão para as assinaturas do usuário Jorge Felipe

Na figura 5.3 é mostrada a árvore de decisão para o usuário Paulo César Flores. Para as assinaturas deste usuário, os atributos mais importantes são a densidade de pontos com informações de grade, a soma do comprimento dos vetores geográficos e as linhas de interseção verticais e horizontais.

```
Decision Tree:

Densidade_celula_2_7 > 69 : PAULO_CESAR_FLORES (28.0/1.0)
Densidade_celula_2_7 <= 69 :
|   Densidade_celula_2_7 <= 47 :
|   |   Densidade_celula_2_7 <= 21 : FALSIFICATION (1224.0/1.0)
|   |   Densidade_celula_2_7 > 21 :
|   |   |   Interseccao_Y_3 <= 0 : PAULO_CESAR_FLORES (4.0)
|   |   |   Interseccao_Y_3 > 0 : FALSIFICATION (68.0)
|   |   Densidade_celula_2_7 > 47 :
|   |   |   Soma_comprimento_vetores_Nordeste <= 59 : PAULO_CESAR_FLORES (8.0)
|   |   |   Soma_comprimento_vetores_Nordeste > 59 : FALSIFICATION (18.0)
```

Figura 5.3 - Árvore de decisão para as assinaturas do usuário Paulo César Flores

Através da análise das Árvores de Decisão das figuras 5.1, 5.2 e 5.3, pode-se perceber que para usuários diferentes existem atributos diferentes que melhor os identificam. Através de várias simulações com diferentes usuários se chegou a conclusão que os nove atributos selecionados anteriormente são úteis no processo de classificação e todos eles servem para a identificação das assinaturas de pelo menos um usuário.

As Árvores de Decisão foram introduzidas junto ao Sistema NeuralSignX não como uma forma de classificação, mas para auxiliar na escolha dos atributos mais relevantes e também para que fosse possível visualizar quais os atributos que mais se destacam em cada assinatura, e assim entender quais as regras o sistema estaria utilizando para a classificação de determinada assinatura. A compreensão de quais fatores levaram a tomada de uma decisão é importante quando se deseja saber porque uma determinada assinatura não foi corretamente classificada.

Através da Análise de Componentes Principais, da matriz de correlação, das Árvores de Decisão e de simulações utilizando o Sistema NeuralSignX foi possível avaliar a importância de cada um dos atributos do sistema, e assim se chegou ao menor número de atributos possível sem que houvessem perdas significativas de informação. Também foi possível verificar através das simulações que os atributos selecionados são bastante eficientes e servem para identificar as assinaturas da maioria dos usuários presentes na base de dados. Na próxima seção serão estudados os testes realizados com o objetivo de validar a performance do modelo proposto.

5.4 Validação do modelo proposto

Nesta seção serão descritos os testes realizados com o objetivo de validar o Sistema NeuralSignX. Inicialmente serão descritos os testes realizados para a identificação de usuários, em seguida serão descritos os testes realizados para a autenticação de assinaturas e por último serão descritos os testes realizados com o objetivo de verificar a robustez do sistema quando submetido a assinaturas falsificadas.

5.4.1 Identificação de usuários

Os primeiros testes realizados com o Sistema NeuralSignX foram realizados com o sistema em modo de identificação de usuários, para descobrir quais os atributos que seriam utilizados nas simulações. Pelo fato da identificação de usuários ter sido amplamente utilizada na configuração do Sistema NeuralSignX, aqui serão descritos os resultados obtidos com o Sistema NeuralSignX na identificação de usuários.

O primeiro teste foi realizado com uma base de dados de 750 assinaturas de 20 usuários diferentes. Esta base de dados foi dividida simetricamente em 375 assinaturas para o aprendizado e 375 assinaturas para o teste de generalização. O número de assinaturas de cada usuário é o mesmo em ambas as bases de dados, ou seja, se um usuário contribuiu com 20 assinaturas, existem 10 assinaturas deste usuário na base de aprendizado e 10 assinaturas na base de teste de generalização. Em todos os testes realizados com o Sistema NeuralSignX a base de dados foi dividida simetricamente, de forma que os exemplos de ambas as bases de dados fossem representativos. Como os resultados obtidos em uma Rede Neural são diferentes a cada execução do simulador, em cada simulação foram realizados cinco testes distintos. A técnica *winner takes all*, descrita anteriormente, foi utilizada para verificar as saídas obtidas pelo sistema, e assim calcular o número de classificações corretas e incorretas. Na tabela 5.6 são mostrados os resultados obtidos pelo Sistema NeuralSignX na identificação das assinaturas de 20 usuários diferentes.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Média	DP
Aprendizado	100%	100%	100%	100%	100%	100%	0
Generalização	98,4%	98,13%	98,4%	98,4%	98,4%	98,35%	0,12
Neurônios	0	0	0	0	0	0	0
Melhor época	63	68	69	75	70	69	4,3
N.º Acertos	369	368	369	369	369	368,8	0,45
N.º Erros	6	7	6	6	6	6,2	0,45
% Erros	1,6%	1,87%	1,6%	1,6%	1,6%	1,65%	0,12

Tabela 5.6 - Resultados obtidos com a identificação de 20 usuários diferentes

Cada uma das colunas da tabela representa um teste diferente, e as duas últimas colunas representam a média e o desvio padrão (DP) obtido nas cinco simulações. As linhas da tabela representam as seguintes informações:

- Aprendizado: percentual de acertos obtido no aprendizado;
- Generalização: percentual de acertos obtido na base de teste de generalização com a técnica *winner takes all*;
- Neurônios: número de neurônios adicionados na camada oculta;
- Melhor época: época em que ocorreu a melhor generalização (*best epoch*);

- N.º Acertos: número de exemplos da base de teste de generalização corretamente classificados;
- N.º Erros: número de exemplos da base de teste de generalização incorretamente classificados;
- % Erros: percentual de erros em relação ao número total de exemplos presentes na base de teste de generalização.

Pelos valores obtidos e apresentados na tabela 5.6 se percebe que o problema é linearmente separável, pois não necessita de neurônios na camada oculta. O número de épocas necessário para o aprendizado ficou em torno de 69, o que indica que a Rede Neural rapidamente convergiu para a solução do problema. O percentual de erros obtido na base de teste de generalização foi em torno de 1,65%, e na base de dados de aprendizado todos os exemplos foram classificados corretamente (100% de aprendizado).

O segundo teste realizado utilizou todas as assinaturas reais da base de dados, que totalizavam 1350 assinaturas de 46 usuários diferentes, divididas simetricamente em 675 assinaturas para o aprendizado e 675 assinaturas para o teste de generalização. Os resultados obtidos são mostrados na tabela 5.7.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Média	DP
Aprendizado	100%	100%	100%	100%	100%	100%	0
Generalização	96%	96,15%	97,63%	96,74%	97,19%	96,74%	0,35
Neurônios	0	0	0	0	0	0	0
Melhor época	84	90	81	86	84	85	3,32
N.º acertos	648	649	659	653	656	653	4,64
N.º erros	27	26	16	22	19	22	4,64
% Erros	4%	3,85%	2,37%	3,26%	2,81%	3,26%	0,69

Tabela 5.7 - Resultados obtidos na identificação de 46 usuários diferentes

O maior número de exemplos e de classes de saída presentes nas bases de dados fez com que fossem necessárias mais épocas para realizar o aprendizado, e o percentual de erros subiu de 1,65% para 3,26%. O percentual de aprendizado continuou sendo de 100%, e não foram necessários neurônios na camada oculta.

Um sistema de identificação de usuários procura diferenciar as assinaturas de todos os usuários entre si, de forma que ele precisa descobrir as regras gerais que servem para diferenciar as assinaturas dos diversos usuários presentes na base de dados. Já um sistema de autenticação de assinaturas precisa identificar as assinaturas de um único usuário, o que faz com que os resultados obtidos sejam bem melhores, pois a Rede Neural pode se especializar na identificação das assinaturas deste usuário. Na próxima subseção serão descritos os resultados obtidos com o Sistema NeuralSignX para a autenticação de assinaturas, que é o objetivo principal do Sistema NeuralSignX.

5.4.2 Autenticação de assinaturas

Do ponto de vista de implementação, a diferença entre um sistema de identificação de usuários e um sistema de autenticação de assinaturas está no número de saídas e na forma como elas são estruturadas. No Sistema NeuralSignX, por questões de segurança se optou por utilizar duas saídas para a autenticação de assinaturas, uma representando o índice de aceitação e outra o índice de rejeição de determinada assinatura. Para que uma assinatura seja considerada autêntica, ela precisa ter na primeira saída um valor superior

a $(1 - MaxErr)$, e na segunda saída um valor inferior a $MaxErr$, como é mostrado na figura 5.4 para um $MaxErr$ de 0,4. As áreas escuras no gráfico da figura representam os valores da saída necessários para uma assinatura seja considerada autêntica. Se ambas as saídas responderem com valores nas áreas escuras do gráfico, a assinatura é considerada autêntica, caso contrário ela é considerada falsa.

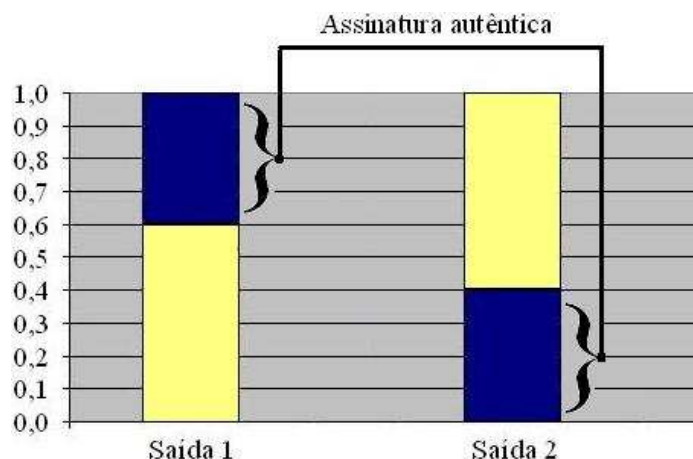


Figura 5.4 - Faixa de valores de saída para uma assinatura autêntica

O treinamento de uma Rede Neural para a autenticação de assinaturas é realizado para identificar as assinaturas de um usuário de cada vez, portanto é necessário que seja realizado o aprendizado para cada usuário de forma independente. Para medir o desempenho do Sistema NeuralSignX na autenticação de assinaturas, foram selecionados cinco usuários em cada simulação, e para cada usuário foram realizados cinco testes. Ao final foram calculadas a média e o desvio padrão dos valores obtidos nos cinco testes de cada usuário. Para uma melhor compreensão dos resultados obtidos, inicialmente serão descritos os resultados obtidos para o primeiro usuário de forma detalhada, e em seguida serão mostradas as médias dos valores obtidos para os demais usuários.

5.4.2.1 Primeira simulação – Somente assinaturas autênticas

Na primeira simulação realizada com a autenticação de assinaturas, foi utilizada uma base de dados com 1350 assinaturas de 46 usuários diferentes, dividida simetricamente em 675 assinaturas para o aprendizado e 675 assinaturas para a generalização. Os resultados obtidos na autenticação das assinaturas do usuário Milton Heinen são mostrados na tabela 5.8. Das 675 assinaturas em cada base de dados, 70 pertenciam ao usuário Milton Heinen e 605 pertenciam aos demais usuários.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Média	DP
Aprendizado	100%	100%	100%	100%	100%	100%	0
Generalização	100%	99,85%	100%	99,85%	100%	99,94%	0,08
Neurônios	0	0	0	0	0	0	0
Melhor época	74	68	72	69	75	71,6	3,05
N.º AI	0	0	0	0	0	0	0
N.º RI	0	1	0	1	0	0,4	0,55
% AI	0%	0%	0%	0%	0%	0%	0
% RI	0%	1,43%	0%	1,43%	0%	0,57%	0,78

Tabela 5.8 - Resultados obtidos na autenticação das assinaturas do usuário Milton Heinen

Cada uma das colunas da tabela 5.8 representa um teste diferente, e as duas últimas colunas representam a média e o desvio padrão (DP) das cinco simulações. As linhas da tabela contém as seguintes informações:

- Aprendizado: percentual de acertos obtido no aprendizado;
- Generalização: percentual de acertos obtido no teste de generalização;
- Neurônios: número de neurônios da camada oculta;
- Melhor época: época em que ocorreu a melhor generalização (*best epoch*);
- N.º AI: número de aceites indevidos obtidos no teste de generalização;
- N.º RI: número de rejeições indevidas obtidas no teste de generalização;
- % AI: percentual de aceites indevidos em relação ao número de assinaturas da base de dados de generalização que não pertencem ao usuário;
- % RI: percentual de rejeições indevidas em relação ao número de assinaturas do usuário presentes na base de dados de generalização;

Os aceites indevidos (AI) são assinaturas que não pertencem ao usuário, mas que foram classificadas como se fossem dele, e as rejeições indevidas (RI) são assinaturas que pertencem ao usuário, mas que foram classificadas como se não fossem dele.

Analisando os resultados obtidos na tabela 5.8, percebe-se que a taxa de aprendizado foi de 100%, não foram necessários neurônios na camada oculta, e o número de épocas necessário para o aprendizado ficou em torno de 72. Não houveram casos de aceites indevidos nestas simulações, e o número médio de rejeições indevidas ficou em torno de 0,4, que corresponde em média a 0,57% do número de assinaturas do usuário presentes na base de teste de generalização. A figura 5.5 mostra uma assinatura do usuário Milton Heinen.

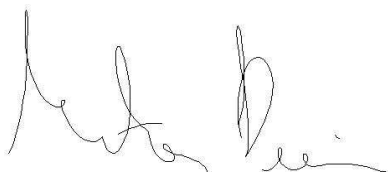


Figura 5.5 - Exemplo de uma assinatura do usuário Milton Heinen

Os resultados obtidos para os demais usuários foram bastante similares. Em todos os testes o aprendizado foi de 100%, não foram necessários neurônios na camada oculta e o número de épocas para o aprendizado ficou entre 60 e 140. Os valores que mais variaram de usuário para usuário foram o número de aceites indevidos e o número de rejeições indevidas. Na tabela 5.9 são mostradas as médias dos valores obtidos na autenticação das assinaturas de cinco usuários.

Usuário	Média AI	Média RI	DP AI	DP RI	QTDE
Milton Heinen	0	0,4 (0,57%)	0	0,55	70
Arno Heinen	0	2 (20%)	0	0	10
Marilei da Costa	1 (0,15%)	0	0	0	25
Wagner Cambuzzi	0	0	0	0	15
Débora Fidelis	0	1,2 (8%)	0	0,84	15
Média Geral	0,2	0,72	0	0,32	32,4

Tabela 5.9 - Resultados obtidos na autenticação de assinaturas de cinco usuários

Cada linha da tabela 5.9 representa a média dos valores obtidos nos cinco testes realizados para cada um dos usuários. A última linha da tabela representa a média geral obtida para estes cinco usuários. As colunas da tabela representam as seguintes informações:

- Média AI: média do número de aceites indevidos obtido no arquivo de saída do Neusim (OUT), que representa o teste final dos dados do arquivo de teste de generalização. Os valores entre parênteses mostram o percentual de aceites indevidos em relação ao número de assinaturas da base de teste de generalização que não pertencem ao usuário;
- Média RI: média do número de rejeições indevidas obtido no arquivo de saída do Neusim (OUT). Os valores entre parênteses mostram o percentual de rejeições indevidas em relação ao número de assinaturas da base de teste de generalização que pertencem ao usuário (QTDE);
- DP AI: desvio padrão do número de aceites indevidos;
- DP RI: desvio padrão do número de rejeições indevidas;
- QTDE: quantidade de assinaturas do usuário presentes na base de teste de generalização.

Analisando a tabela 5.9 é possível notar que apenas para as assinaturas da Marilei da Costa ocorreram casos de aceites indevidos, e para os demais usuários não houve nenhum caso. Na figura 5.6 é mostrada uma assinatura da Marilei da Costa.

A handwritten signature in black ink that reads "Marilei H. da Costa". The script is cursive and fluid.

Figura 5.6 - Exemplo de uma assinatura da Marilei da Costa

Na figura 5.7 é mostrada a assinatura que foi confundida com as assinaturas da Marilei da Costa em todas as cinco simulações realizadas, que na verdade pertence a Cláudia do Couto. O índice de aceitação obtido para esta assinatura (primeira saída) ficou em torno de 0,99 e o índice de rejeição obtido para esta assinatura (segunda saída) ficou em torno de 0,2. Embora visualmente estas assinaturas não sejam muito semelhantes, pode-se perceber que elas apresentam diversos pontos em comum, que podem fazer com que vários atributos respondam da mesma forma para ambas as assinaturas.

A handwritten signature in black ink that reads "Cláudia do Couto". The script is cursive and fluid, similar in style to the one in Figure 5.6.

Figura 5.7 - Assinatura indevidamente aceita como sendo da Marilei da Costa

Em relação as rejeições indevidas, os valores da tabela 5.9 ficaram dentro do aceitável, com exceção dos resultados obtidos para as assinaturas do Arno Heinen. Para este usuário o percentual foi elevado (20%), e isto se deve principalmente ao pequeno número de amostras coletado para este usuário, que é de apenas 10 assinaturas em cada base de dados. Quanto menor for o número de assinaturas coletado, mais rígido se torna o sistema, fazendo com que apenas as assinaturas que forem muito próximas das presentes na base dados de aprendizado sejam reconhecidas como autênticas. Pelos testes realizados se percebe que 30 assinaturas por usuário (15 para o aprendizado e 15 para a generalização) são suficientes para que se atinja bons resultados.

5.4.2.2 Segunda simulação – assinaturas autênticas e pictográficas

Nesta simulação, buscou-se verificar como o sistema se comporta quando estiverem presentes na base de dados assinaturas reais e assinaturas pictográficas. No total foram utilizadas 2300 assinaturas, 1350 pertencendo a 46 usuários reais e 950 representando mais de 100 pictogramas diferentes. A base de dados foi dividida de forma simétrica, com 1150 assinaturas para o aprendizado e 1150 assinaturas para o teste de generalização. Na tabela 5.10 são mostrados os resultados obtidos para o usuário Carlos Schneider. Das 1150 assinaturas presentes em cada base de dados, 15 pertencem ao usuário Carlos Schneider, 660 pertencem aos demais usuários e 475 são assinaturas pictográficas.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Média	DP
Aprendizado	100%	100%	100%	100%	100%	100%	0
Generalização	99,83%	100%	99,91%	100%	100%	99,95%	0,0008
Neurônios	0	0	0	0	0	0	0
Melhor época	57	65	59	61	64	61,2	3,35
N.º AI	0	0	0	0	0	0	0
N.º RI	2	0	1	0	0	0,6	0,89
% AI	0%	0%	0%	0%	0%	0%	0
% RI	13,33%	0%	6,67%	0%	0%	4%	5,96

Tabela 5.10 - Resultados obtidos para as assinaturas de Carlos Schneider

Como na primeira simulação, o aprendizado foi de 100%, e não houveram neurônios adicionados na camada oculta. Foram necessárias cerca de 61 épocas para se realizar o aprendizado no ponto ótimo de generalização. A assinatura do usuário Carlos Schneider é mostrada na figura 5.8. Em todos os testes realizados com o Sistema NeuralSignX após a configuração do sistema e escolha dos atributos, o aprendizado foi de 100%, e em nenhum teste o Neusim precisou adicionar neurônios na camada oculta.



Figura 5.8 - Exemplo de uma assinatura do usuário Carlos Schneider

Para os demais usuários, os resultados obtidos foram bastante similares, só havendo diferenças no número de aceites indevidos e no número de rejeições indevidas. Na tabela 5.11 são mostradas as médias dos resultados obtidos para as assinaturas de cinco usuários diferentes.

Usuário	Média AI	Média RI	DP AI	DP RI	QTDE
Carlos Schneider	0	0,6 (4%)	0	0,89	15
Melânia Heinen	0	0,8 (2%)	0	0,45	40
Renato Wagner	0	0	0	0	15
Marlete Wagner	1,2 (0,11%)	1,2 (8%)	0,84	0,45	15
Lori Huther	1,4 (0,12%)	2 (13,33%)	1,34	0	15
Média Geral	0,52	0,92	0,44	0,36	20

Tabela 5.11 - Resultados obtidos na autenticação de assinaturas de cinco usuários (assinaturas reais e pictográficas)

As rejeições indevidas ficaram dentro da média, com valores geralmente abaixo de 10%. As assinaturas da Lori Huther foram as que tiveram o maior nível de rejeição. A figura 5.9 é mostra uma assinatura da Lori Huther. Analisando várias assinaturas da Lori, se percebe que a variabilidade delas é muito grande, principalmente na parte final da assinatura. Esta variabilidade acima do normal colabora para que o índice de rejeições indevidas seja mais elevado.

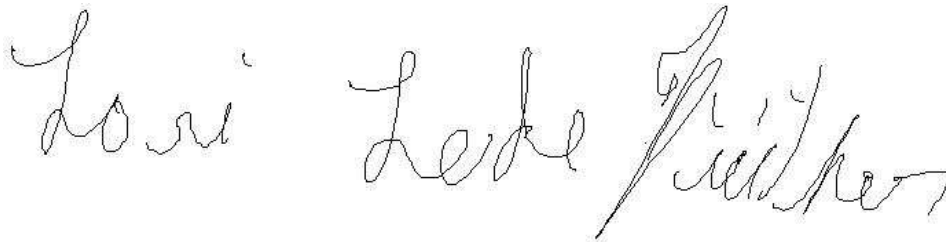


Figura 5.9 - Exemplo de uma assinatura da Lori Huther

Em relação aos aceites indevidos, ocorreram em média 1,2 aceites indevidos com as assinaturas da Marlete Wagner e 1,4 aceites indevidos com as assinaturas da Lori Huther. Na figura 5.10 é mostrada uma das assinaturas que foram confundidas com as assinaturas da Lori Huther, que na verdade pertence ao usuário Miguel de Oliveira.

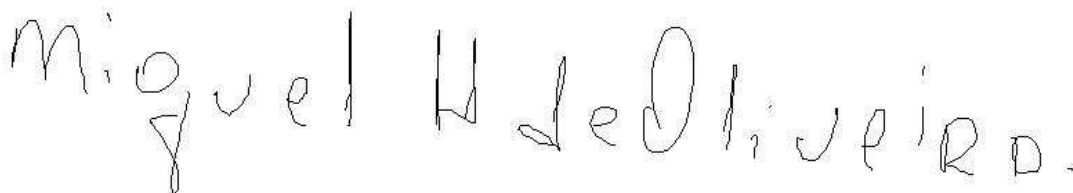


Figura 5.10 - Assinatura indevidamente aceita como sendo da Lori Huther

Observando as duas assinaturas, se percebe que visualmente elas não se parecem, portanto o sistema não deveria ter se confundido. A hipótese que se levanta é que embora não haja semelhança visual, em termos de atributos estas assinaturas podem ser parecidas. Para que se possa verificar esta hipótese, serão utilizadas Árvores de Decisão para descobrir quais os atributos que melhor representam cada uma das assinaturas. Na figura 5.11 é mostrada a árvore de decisão obtida para as assinaturas da Lori Huther.

```

Decision Tree:
Duracao_assinatura <= 1033 : FALSIFICACION (1027.0)
Duracao_assinatura > 1033 :
|  Soma_comprimento_vetores_Nordeste <= 97 :
|  |  Velocidade_media <= 321 : FALSIFICACION (102.0)
|  |  Velocidade_media > 321 :
|  |  |  Num_vetores_sentido_Oeste <= 3 : LORI_LEDE_HUTHER (2.0)
|  |  |  Num_vetores_sentido_Oeste > 3 : FALSIFICACION (4.0)
|  |  Soma_comprimento_vetores_Nordeste > 97 :
|  |  |  Velocidade_maxima <= 685 : FALSIFICACION (2.0)
|  |  |  Velocidade_maxima > 685 : LORI_LEDE_HUTHER (13.0)

```

Figura 5.11 - Árvore de decisão gerada para as assinaturas da Lori Huther

Além das Árvores de Decisão, o *software* C4.5 também pode extrair regras de produção, que são as regras que melhor caracterizam o comportamento dos exemplos da base de

dados. Na figura 5.12 são mostradas as regras de produção para as assinaturas da Lori Huther. Se percebe que utilizando apenas a velocidade máxima e a duração da assinatura é possível classificar corretamente 89.9% dos casos.

```
Rule 3:
  Velocidade_maxima > 685
  Duracao_assinatura > 1033
  -> class LORI_LEDE_HUTHER [89.9%]

Default class: FALSIFICATION
```

Figura 5.12 - Regras de produção geradas para as assinaturas da Lori Huther

Para as assinaturas do usuário Miguel de Oliveira, as regras de produção são mostradas na figura 5.13. Com apenas um atributo (número de vezes que a caneta foi levantada) é possível identificar 90,6% das assinaturas do Miguel.

```
Rule 1:
  Numero_pen_ups > 20
  -> class MIGUEL_DE_OLIVEIRA [90.6%]

Default class: FALSIFICATION
```

Figura 5.13 - Regras de produção geradas para as assinaturas de Miguel de Oliveira

Através da análise destas regras se percebe o porque do sistema estar se confundindo na classificação. Quando uma assinatura possui para um determinado atributo valores muito diferentes dos ocorridos para os demais exemplos, a Rede Neural rapidamente entende que basta este atributo para identificar as assinaturas deste usuário, podendo desconsiderar as demais entradas. Ou seja, o aprendizado ocorre tão depressa que para as demais entradas os pesos ainda não chegam a ser corretamente ajustados. No caso das assinaturas da Lori, toda vez que uma outra assinatura durar mais de 10 segundos e ter uma velocidade máxima superior a 685, esta assinatura poderá ser confundida com as assinaturas da Lori. Quanto maior e mais diversificada for a base de dados utilizada no aprendizado, menores serão os índices de aceites indevidos, mas por mais que se aumente o tamanho de uma base de dados, nunca será possível cobrir todas as possibilidades em termos de atributos.

Uma assinatura possui diversos atributos que lhe são característicos, e a combinação de certos valores para cada atributo é que faz com que se tenha certeza de que uma assinatura é verdadeira. Ou seja, a classificação não pode ser realizada com base em apenas um atributo, mas sim na combinação de vários atributos que juntos possam garantir a autenticidade. No caso das assinaturas do Miguel de Oliveira, se houvessem na base de dados de aprendizado outros casos em que o número de vezes que a caneta foi levantada fosse superior a 20, o Rede Neural necessariamente teria de analisar outros atributos, caso contrário a taxa de aprendizado não chegaria a 100%.

Estes problemas ocorrem porque a autenticação de assinaturas é um problema de classes fechadas, e portanto deve ser tratado como tal. Após diversas tentativas, se chegou a proposta de uma nova técnica capaz de minimizar este problema, que foi chamada de Nuvem de Pontos.

5.4.2.3 Terceira simulação – utilização da Nuvem de Pontos

A Nuvem de Pontos é uma técnica desenvolvida durante a fase de implementação do Sistema NeuralSignX para reduzir o número de aceites indevidos na classificação das assinaturas. Nesta técnica, além das assinaturas presentes na base de dados, são inseridos nos arquivos de aprendizado e de teste de generalização do Neusim diversos exemplos com valores aleatórios para cada uma das entradas, dentro das faixas possíveis para cada atributo. A saída desejada para estes exemplos é sempre “0 1”, indicando que o exemplo não pertence ao usuário. A maioria dos exemplos gerados pela Nuvem de Pontos possuem combinações entre os atributos impossíveis de existir em assinaturas reais, o que faz com que o sistema dificilmente os trate como tal. Mas a presença de valores distribuídos uniformemente para cada atributo exigirá que a Rede Neural analise vários atributos para poder autenticar uma assinatura. Quanto maior for o número de exemplos gerados pela Nuvem de Pontos, melhores serão os resultados. Nas simulações foram utilizados 29.999 exemplos para aprendizado e 29.999 para generalização, que é o limite máximo da versão do Neusim utilizada. O ideal seria utilizar muito mais exemplos, e isto poderá vir a ser realizado em trabalhos futuros, através do uso de outros simuladores de Redes Neurais.

Para se comprovar a eficácia da Nuvem de Pontos, foi gerada uma árvore de decisão para as assinaturas da Lori Huther a partir de uma base de dados com 2300 assinaturas reais e 97.700 exemplos gerados aleatoriamente. Os resultados são mostrados na figura 5.14. Se percebe claramente que a árvore de decisão é muito mais complexa que a da figura 5.11, o que comprova que o C4.5 precisa de muito mais informações para realizar a classificação. Da mesma forma o Neusim também precisará ajustar os pesos de muito mais entradas para conseguir realizar a classificação de forma correta.

```
Decision Tree:
Soma_comprimento_vetores_Nordeste <= 99 :
|  Densidade_celula_4_6 <= 99 :
|  |  Densidade_celula_4_7 <= 99 : FALSIFICATION (97699.0)
|  |  Densidade_celula_4_7 > 99 :
|  |  |  Ponto_amostrado_X_7 <= 95 : FALSIFICATION (611.0)
|  |  |  Ponto_amostrado_X_7 > 95 :
|  |  |  |  Interseccao_Y_1 <= 0 : LORI_LEDE_HUTHER (2.0)
|  |  |  |  Interseccao_Y_1 > 0 : FALSIFICATION (25.0)
|  |  Densidade_celula_4_6 > 99 :
|  |  |  Interseccao_Y_20 > 0 : FALSIFICATION (605.0)
|  |  |  Interseccao_Y_20 <= 0 :
|  |  |  |  Interseccao_X_7 <= 0 : LORI_LEDE_HUTHER (3.0)
|  |  |  |  Interseccao_X_7 > 0 : FALSIFICATION (29.0)
Soma_comprimento_vetores_Nordeste > 99 :
|  Interseccao_Y_1 > 0 : FALSIFICATION (957.0/2.0)
|  Interseccao_Y_1 <= 0 :
|  |  Ponto_amostrado_X_6 <= 76 : FALSIFICATION (44.0/1.0)
|  |  Ponto_amostrado_X_6 > 76 :
|  |  |  Soma_comprimento_vetores_Leste <= 35 : LORI_LEDE_HUTHER (22.0)
|  |  |  Soma_comprimento_vetores_Leste > 35 : FALSIFICATION (3.0)
```

Figura 5.14 - Árvore de decisão criada utilizando a Nuvem de Pontos

Depois de serem verificadas as alterações causadas pela Nuvem de Pontos nas Árvore de Decisão, é necessário que sejam realizadas simulações para comprovar a eficácia da Nuvem de Pontos na autenticação de assinaturas. Na tabela 5.12 são mostrados os resultados obtidos com a Nuvem de Pontos para as assinaturas da Tatiane Boll. Das 29.999 assinaturas presentes em cada base de dados, 10 pertenciam a Tatiane Boll e 29989 não pertenciam a ela.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Média	DP
Aprendizado	100%	100%	100%	100%	100%	100%	0
Generalização	99,99%	99,99%	99,98%	99,99%	99,98%	99,99%	0,005
Neurônios	0	0	0	0	0	0	0
Melhor época	63	90	72	111	66	80,4	17,94
N.º AI	0	0	1	0	1	0,4	0,49
N.º RI	3	3	3	3	3	3	0
% AI	0%	0%	0,003	0%	0,003	0,001%	0,0016
% RI	30%	30%	30%	30%	30%	30%	0

Tabela 5.12 - Resultados obtidos na autenticação de assinaturas da Tatiane Boll

O que mais chama a atenção nos resultados obtidos é o elevado índice de rejeições indevidas. Na figura 5.15 é mostrada uma das assinaturas da Tatiane Boll que foram aceitas, e na figura 5.16 uma das assinaturas que foram rejeitadas. Pode-se notar que as assinaturas são muito semelhantes, e o sistema deveria ter condições de aceitar a assinatura da figura 5.16 como se fosse verdadeira.

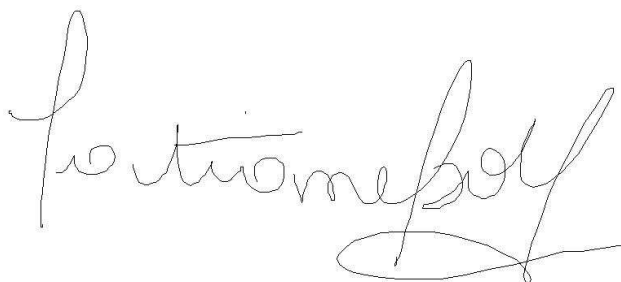


Figura 5.15 - Exemplo de uma assinatura da Tatiane Boll corretamente aceita

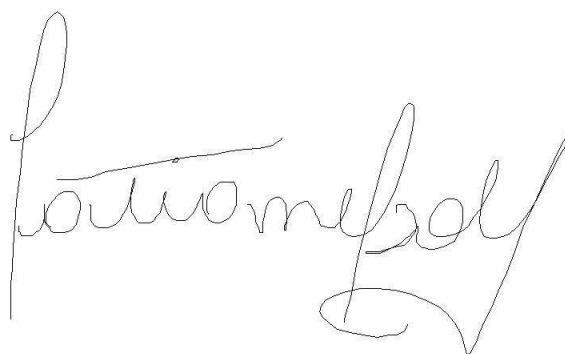


Figura 5.16 - Exemplo de uma assinatura da Tatiane Boll indevidamente rejeitada

O motivo que levou o sistema a rejeitar a assinatura da figura 5.16 é o pequeno número de assinaturas da Tatiane que foram coletadas (apenas 20), o que fez com que o sistema ficasse muito rígido e rejeitasse algumas assinaturas autênticas.

Na tabela 5.13 são mostradas as médias dos resultados obtidos para as assinaturas de cinco usuários diferentes. Se optou por realizar novamente as assinaturas da Marlete Wagner e da Lori Huther, para que fosse possível verificar se a Nuvem de Pontos reduziu os índices de aceites indevidos.

Usuário	Média AI	Média RI	DP AI	DP RI	QTDE
Miguel de Oliveira	0	0,8 (5,33%)	0	0,4	15
Jalton Heinen	0,6 (0,002%)	0	0,49	0	25
Alessandra Huther	0,2 (0,0007%)	0,6 (2,4%)	0,4	0,49	25
Marlete Wagner	2,4 (0,008%)	1,2 (8%)	0,5	0,4	15
Lori Huther	3,6 (0,012%)	1,8 (12%)	1,5	1,17	15
Média Geral	1,36	0,88	0,58	0,49	19

Tabela 5.13 - Resultados obtidos na autenticação de assinaturas de cinco usuários utilizando a Nuvem de Pontos

Em relação aos valores obtidos na tabela 5.11, houve um aumento do número de aceites indevidos, mas como o número total de exemplos aumentou bastante, o percentual de aceites indevidos em relação ao número total de exemplos diminuiu consideravelmente. Em relação ao percentual de rejeições indevidas, não se notou alterações significativas em relação aos resultados obtidos na tabela 5.11. Para que seja possível fazer uma comparação mais realista dos resultados obtidos com e sem a Nuvem de Pontos, na tabela 5.14 são mostrados os resultados obtidos utilizando-se a Nuvem de Pontos apenas no arquivo de aprendizado, e não no arquivo de generalização. Comparando-se os resultados da tabela 5.14 com os obtidos da tabela 5.11, pode-se perceber que houve uma ligeira redução no número de aceites indevidos, e os índices de rejeições indevidas não sofreram alterações significativas.

Para que os resultados ficassem ainda melhores, seria necessário que fossem gerados mais exemplos com a Nuvem de Pontos, mas devido as limitações da versão do Neusim utilizada, isto não foi possível de ser realizado. Mas pelos resultados obtidos com 29.999 exemplos, se percebe que a técnica da Nuvem de Pontos é capaz de reduzir o número de aceites indevidos, sem alterar de forma significativa o número de rejeições indevidas.

Usuário	Média AI	Média RI	DP AI	DP RI	QTDE
Miguel de Oliveira	0	0,6 (4%)	0	0,49	15
Jalton Heinen	0,6 (0,05%)	0	0,49	0	25
Alessandra Huther	0,2 (0,018%)	0	0,4	0	25
Marlete Wagner	0,8 (0,07%)	0,8 (5,3%)	0,75	0,75	15
Lori Huther	1,2 (0,1%)	2 (13,3%)	1,17	0	15
Média Geral	0,56	0,68	0,58	0,25	19

Tabela 5.14 - Resultados obtidos utilizando a Nuvem de Pontos apenas no arquivo de aprendizado

Pelas simulações realizadas, chega-se a conclusão de que o sistema obteve resultados bastante satisfatórios na autenticação de assinaturas, com taxas de aceites indevidos bastante reduzidas e taxas de rejeições indevidas dentro de valores considerados aceitáveis. Mas ainda resta ser estudada a hipótese de uma pessoa tentar enganar o sistema, falsificando as assinaturas de um determinado usuário. Na próxima seção serão apresentados os resultados obtidos pelo um Sistema NeuralSignX quando assinaturas falsificadas estão presentes nas bases de dados de aprendizado e de generalização.

5.4.3 Utilização de assinaturas falsificadas

O objetivo do Sistema NeuralSignX não é a detecção de fraudes, mas foram realizadas algumas simulações com assinaturas falsificadas para medir o comportamento do

sistema em situações extremas. Existem três tipos de fraudes, as aleatórias, as traçadas e as especializadas. Nas seções anteriores já foi testado o desempenho do sistema quando submetido a fraudes aleatórias (com as assinaturas pictográficas e a Nuvem de Pontos), agora serão realizados testes com falsificações traçadas e especializadas. Na tabela 5.15 são mostrados os resultados obtidos para a autenticação das assinaturas do usuário Milton Heinen. Das 29.999 assinaturas base de teste de generalização, 70 pertenciam ao usuário Milton Heinen, 15 eram falsificações traçadas, 17 eram falsificações especializadas, 697 pertenciam aos demais usuários, 476 eram assinaturas pictográficas e 28724 foram geradas aleatoriamente através da Nuvem de Pontos.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Média	DP
Aprendizado	100%	100%	100%	100%	100%	100%	0
Generalização	99,98%	99,99%	99,98%	99,98%	99,99%	99,98%	0,0049
Neurônios	0	0	0	0	0	0	0
Melhor época	111	145	126	92	148	124,40	21,04
N.º AI	0	0	0	0	0	0	0
N.º RI	4	2	3	4	2	3	0,89
% AI	0%	0%	0%	0%	0%	0%	0
% RI	5,71%	2,86%	4,29%	5,71%	2,86%	4,29%	1,28

Tabela 5.15 - Resultados obtidos na autenticação de assinaturas do usuário Milton Heinen (com fraudes envolvidas)

Se percebe que para este usuário o número de falsas rejeições subiu em relação aos valores obtidos na tabela 5.8, mas em compensação não houveram casos de falsas aceitações nas cinco simulações realizadas. Na figura 5.17 são mostradas três assinaturas presentes na base de dados. A primeira é uma assinatura original que pertence ao usuário Milton Heinen, a segunda é uma falsificação traçada e a terceira é uma falsificação especializada.

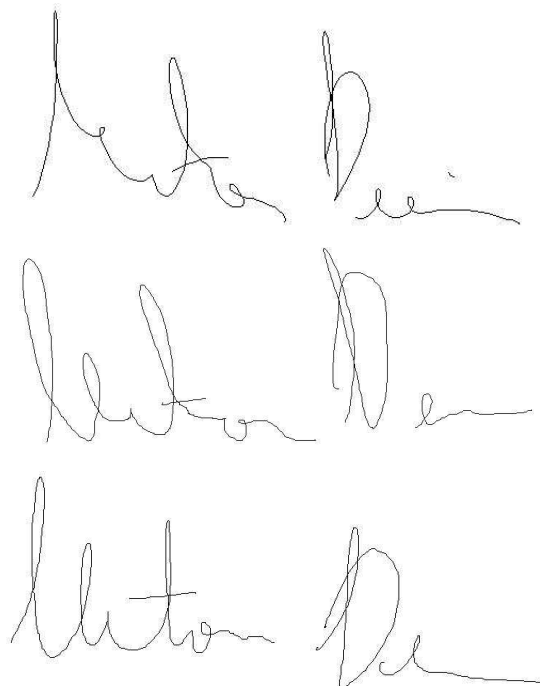


Figura 5.17 - Assinatura original, falsificação traçada e falsificação especializada das assinaturas do usuário Milton Heinen

A tabela 5.16 mostra os resultados obtidos para cinco dos oito usuários para os quais foram coletadas assinaturas falsificadas. Pode-se perceber que a média dos resultados obtidos na tabela 5.16 são similares aos obtidos na tabela 5.13, o que demonstra o potencial do Sistema NeuralSignX em evitar que assinaturas falsificadas sejam reconhecidas como verdadeiras.

Usuário	Média AI	Média RI	DP AI	DP RI	QTDE
Milton Heinen	0	3 (4,29%)	0	0,89	70
Melânia Heinen	0,6 (0,002%)	1 (2,5%)	0,49	0	40
Jalton Heinen	1,8 (0,006%)	0,2 (1,33%)	1,47	0,4	15
Rosana Colling	1,6 (0,005%)	1 (6,67%)	0,49	0	15
Alessandra Huther	0,2 (0,0007%)	0,6 (2,4%)	0,4	0,8	15
Média Geral	0,84	1,16	0,57	0,42	31

Tabela 5.16 - Resultados obtidos na autenticação de assinaturas de cinco usuário (com fraudes envolvidas)

Para que se possa ter uma noção de quanto as falsificações são similares, na figura 5.18 são mostradas algumas assinaturas da base de dados. A primeira é uma assinatura original do usuário Jalton Heinen, a segunda é uma falsificação traçada e a terceira é uma falsificação especializada. Pode-se perceber que visualmente as assinaturas são bastante próximas.

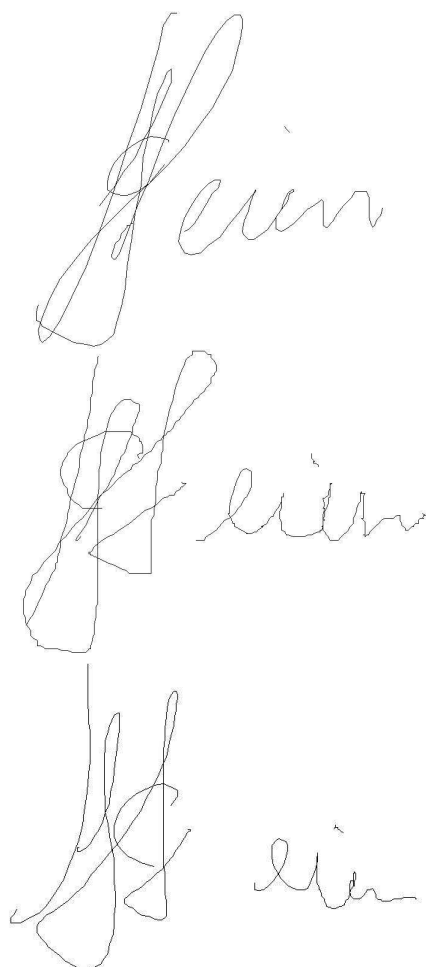


Figura 5.18 - Assinatura original, falsificação traçada e falsificação especializada das assinaturas do usuário Jalton Heinen

Na figura 5.19 são mostradas mais três assinaturas, desta vez da Alessandra Huther. A primeira é uma assinatura original, a segunda é uma falsificação traçada e a terceira uma falsificação especializada.

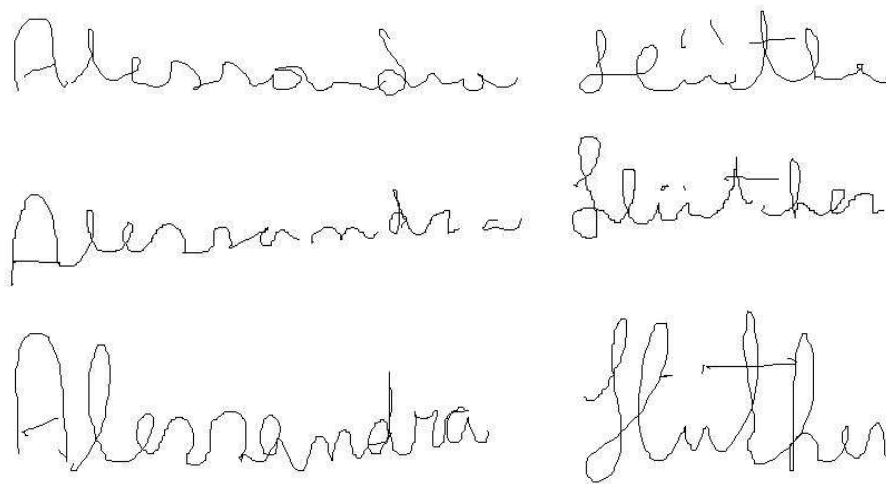
The image displays three handwritten signatures of 'Alessandra Huther' arranged vertically. The top signature is the original, showing fluid, connected cursive strokes. The middle signature is a traced forgery, where the lines are less fluid and more segmented, appearing as if drawn over a guide. The bottom signature is a specialized forgery, which mimics the original's style more closely but still shows some differences in stroke thickness and connection points.

Figura 5.19 - Assinatura original, falsificação traçada e falsificação especializada das assinaturas da Alessandra Huther

A presença de assinaturas mais parecidas com as que estavam sendo autenticadas fez com que o número médio de épocas necessário para se atingir o ponto ótimo de generalização fosse aumentado, e isto também forçou o sistema a utilizar atributos que antes não eram necessários para a classificação.

Através das diversas simulações descritas neste capítulo, foi possível verificar que o Sistema NeuralSignX tem um grande potencial não apenas como um sistema de autenticação de assinaturas, mas também como um sistema de identificação de usuários e de detecção de fraudes. A taxa de aprendizado em todas as simulações realizadas foi de 100%, e a taxa de generalização ficou acima de 99% para a autenticação de assinaturas e acima de 95% para a identificação de usuários. O percentual de aceites indevidos ficou abaixo de 0,05%, que é um resultado excelente, e o percentual de rejeições indevidas ficou em torno de 5%, que é um valor bastante razoável. Em trabalhos futuros o sistema poderá ser expandido, com a adição de novos atributos e com o aumento do número de exemplos gerados pela Nuvem de Pontos, o que poderá tornar os resultados obtidos ainda melhores.

6 Conclusão

Este trabalho teve por objetivo o estudo da autenticação *on-line* de assinaturas através da utilização de Redes Neurais Artificiais. Para atingir tal objetivo foram estudados diversos sistemas de autenticação de assinaturas existentes, diversas formas de implementação e as etapas que um sistema de autenticação de assinaturas deve possuir. Também foram estudados e descritos diversos tópicos referentes a Inteligência Artificial e ao Aprendizado de Máquinas, em especial as Redes Neurais Artificiais. Por fim, foi criada uma proposta de solução e implementação de um sistema de autenticação *on-line* de assinaturas, denominado Sistema NeuralSignX. Para validar esta proposta de implementação foram desenvolvidas diversas ferramentas que implementam as diversas operações e etapas estudadas neste trabalho, criando assim um protótipo completo e operacional do Sistema NeuralSignX.

Os resultados obtidos nas simulações realizadas com o protótipo comprovam que o Sistema NeuralSignX não é apenas viável, mas também representa uma ótima solução para a autenticação *on-line* de usuários em sistemas de informação. A proposta inicial era que o sistema conseguisse realizar a autenticação de assinaturas manuscritas, sem prever situações onde um usuário estivesse querendo enganar o sistema. Entretanto, o sistema foi capaz não de somente autenticar as assinaturas, mas também conseguir realizar a identificação dos usuários a partir destas assinaturas, e também, foi capaz de realizar a detecção de fraudes, não permitindo que assinaturas falsificadas fossem aceitas como verdadeiras. Os resultados obtidos pelo sistema foram de 100% de aprendizado em todas as simulações, e no teste de generalização os resultados obtidos foram de 96,74% na identificação de usuários, 99,99% na autenticação de assinaturas e 99,98% na detecção de fraudes. A taxa de rejeições indevidas ficou em torno de 5%, e a taxa de aceites indevidos foi menor que 0,05% em todas as simulações realizadas.

Pelos resultados obtidos se comprovam as diversas vantagens da autenticação *on-line* de assinaturas em relação a autenticação *off-line*, demonstrada principalmente nos resultados práticos obtidos que ultrapassam a média dos resultados de todos os sistemas de autenticação *off-line* estudados na bibliografia. Também se comprova a partir dos resultados obtidos que as Redes Neurais Artificiais são muito adequadas ao processo de autenticação de assinaturas, e que os atributos selecionados são muito úteis para a classificação das assinaturas, pois permitiram que a Rede Neural conseguisse não apenas obter taxas elevadas de aprendizado, como também de generalização, mantendo baixos índices de aceites indevidos, o que é fundamental em um sistema de autenticação de assinaturas.

Cabe salientar que são encontradas no mercado outras ferramentas similares ao Sistema NeuralSignX, que permitam a autenticação de assinaturas com o mesmo grau de segurança apresentado pelo sistema proposto, o que torna o Sistema NeuralSignX uma ferramenta com grandes potenciais do ponto de vista comercial.

Em trabalhos futuros novos atributos poderão vir a ser incorporados ao sistema, permitindo que os resultados obtidos fiquem ainda melhores, principalmente na detecção de fraudes. O sistema poderá também ser melhorado de forma a exigir menos assinaturas por usuário na etapa de aprendizado, através de uma aplicação que poderia

gerar automaticamente variações a partir das assinaturas originais de forma a aumentar a base de dados com assinaturas artificiais. Este aumento da base de dados de assinaturas poderá inclusive reduzir o número de rejeições indevidas, que é inversamente proporcional ao número de exemplos de assinaturas do usuário presentes na base de dados. Outra técnica que pode ser utilizada para um aumento da base de dados de assinaturas é o incremento progressivo, no qual cada nova assinatura reconhecida como autêntica é inserida na base de dados, e de tempos em tempos o sistema é novamente treinado com as novas assinaturas que foram incorporadas. Esta técnica permite inclusive que as variações que ocorrem nas assinaturas com o passar dos anos não tornem a base de dados obsoleta.

Além dos aspectos relacionados ao aprendizado, outras técnicas poderiam vir a ser incorporadas ao sistema de forma a facilitar o seu uso comercial, como por exemplo a criptografia, que impediria que pessoas estranhas tivessem acesso aos arquivos confidenciais do sistema, como os arquivos de pesos e a base de dados de assinaturas. Outras técnicas tradicionais, como limitar o número de tentativas de se conectar a um sistema, poderia impedir que os usuários que estejam querendo falsificar uma assinatura tenham acesso ao sistema.

Os sistemas de autenticação de assinaturas provaram ser uma alternativa viável de autenticação de usuários, e em breve o seu uso comercial poderá ser difundido, aliado principalmente a redução dos custos e a popularização dos equipamentos de *hardware* necessários para a utilização do sistema.

Anexo A Exemplos de assinaturas coletadas



Figura A.1 - Arno José Heinen



Figura A.8 - Marlete Wagner



Figura A.2 - Débora Fidelis



Figura A.9 - Melânia Heinen



Figura A.3 - Germano Schreiber

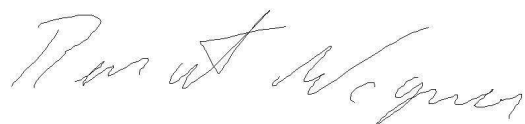


Figura A.10 - Renato Wagner



Figura A.4 - Isabel de Oliveira



Figura A.5 - Jorge Felipe

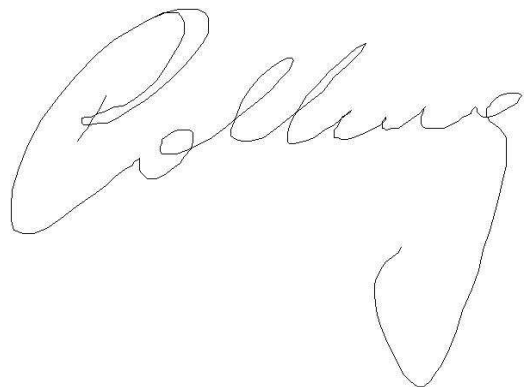


Figura A.11 - Rosana Colling

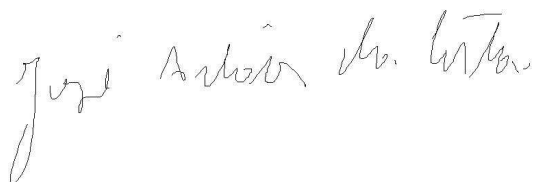


Figura A.6 - José da Costa



Figura A.7 - Juciane Soares

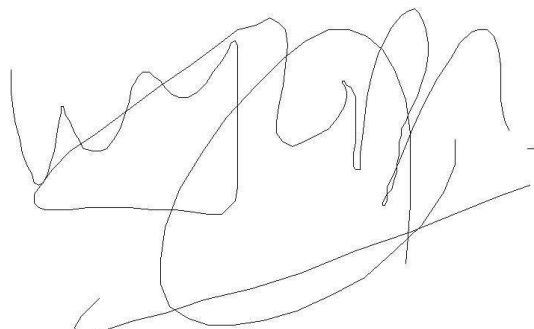


Figura A.12 - Wagner Cambruzzi

Anexo B Exemplos de arquivos do Sistema NeuralSignX

B.1 Arquivo de assinaturas do Sistema NeuralSignX

```
LOGIN=MILTON HEINEN
391 434 1 00:35:45:795
391 432 1 00:35:45:815
391 432 1 00:35:45:815
391 432 0 00:35:45:815
273 375 0 00:35:45:925
266 372 0 00:35:45:935
266 372 1 00:35:45:935
262 371 1 00:35:45:935
259 369 1 00:35:45:945
278 368 1 00:35:45:985
287 368 1 00:35:45:995
295 370 1 00:35:46:005
304 373 1 00:35:46:015
304 373 1 00:35:46:015
304 373 0 00:35:46:015
309 375 0 00:35:46:025
322 384 0 00:35:46:045
409 403 0 00:35:46:226
409 403 1 00:35:46:226
412 393 1 00:35:46:226
415 379 1 00:35:46:236
```

```
LOGIN=MELANIA HEINEN
135 417 1 20:32:56:028
134 419 1 20:32:56:038
134 420 1 20:32:56:048
134 422 1 20:32:56:058
134 424 1 20:32:56:068
133 425 1 20:32:56:078
133 425 1 20:32:56:078
133 425 0 20:32:56:078
133 423 0 20:32:56:138
152 431 0 20:32:56:789
152 431 0 20:32:56:799
152 431 1 20:32:56:799
```

B.2 Arquivo de configuração do Sistema NeuralSignX

```
%% ARQUIVO DE CONFIGURAÇÃO DO SISTEMA NEURAL SIGN X
%% LINHAS COMEÇADAS COM %% SÃO COMENTÁRIOS

WIDTH_IMAGE=800
HEIGHT_IMAGE=600
TYPE_RECOGNITION=0
NAME_AUTENTICATE_USER=MILTON HEINEN
TYPE_POSITION_AJUST=2
USE_SCALE_AJUST=S
USE_CROSS_VALIDATION=S
```

```

CROSS_VALIDATION_INTERVAL=1
SHOW_SIGNATURE=S
SHOW_VISUAL_FEEDBACK=S
VISUALIZATION_DELAY=0
KEEP_SIGNATURES_OLD=N
INPUT_FILE=Database.sgn
OUTPUT_FILE=Signature
MAX_ERR_LEARNING=0.10
MAX_ERR_TESTING=0.40
FEATURE_SIMETRY=N
FEATURE_POINT_DENSITY=N
FEATURE_GRID_POINT_DENSITY=S
WIDTH_CELLS=100
HEIGHT_CELLS=100
FEATURE_INTERSECTION_X=S
FEATURE_INTERSECTION_Y=S
INTERSECTION_DISTANCE_X=25
INTERSECTION_DISTANCE_Y=25
FEATURE_SEQUENTIAL_EXAMPLE_POINTS=S
USE_UNIVERSE_COORDENATES=S
NUMBER_SEQUENTIAL_EXAMPLE_POINTS=8
FEATURE_SIGNATURE_DURACTION=S
FEATURE_NUMBER_COLETED_POINTS=N
FEATURE_NUMBER_PEN_UPS=S
FEATURE_AVERAGE_PEN_VELOCITY=S
FEATURE_MAXIMUM_PEN_VELOCITY=S
MAX_VELOCITY_WINDOW=30
FEATURE_TOTAL_PERCORRED_DISTANCE=N
FEATURE_NUMBER_DIRECTION_CHANGES_X=N
FEATURE_NUMBER_DIRECTION_CHANGES_Y=N
FEATURE_NUMBER_GEOGRAPHICAL_VECTORS=S
FEATURE_LENGTH_GEOGRAPHICAL_VECTORS=S
CHANGE_DIRECTION_THERESHOLD=5
GENERATE_RANDOM_EXAMPLES=S
MAX_GENERATE_EXAMPLES=29999

```

B.3 Arquivo de configuração do Neusim (CFG)

```

%%
%% NEUSIM Configuration File
%%
Task          LGWSOT          % Learning tasks
MaxEpochs    1500
NInputs       138
NOutputs      2              % Start with no hidden units
Learning      1              % Learning method:
Cascor        16              % Cascade-Correlation (16 candidate units)
%
% CasCor parameters
%
CCOutPat      32
CCOutECh      0.01
CCHidPat      32
CCHidECh      0.03
EpLearn       1              % CasCor use quick-prop (epoch oriented)
%
% Learning parameters
%
RndRange      0.001
Epsilon       0.000001
Momentum      0.0

```



```

MaxErr      0.40
StopCrit    2           % Stop when 100% correct (testing data)
StopErr     0
SPOffset    0.1
%
% Report
%
RepLevel    128
ErrLevel    0
RepFreq     50
UserFreq    10
UserRep     1
%%
%% End of CFG
%%

```

B.4 Arquivo de Aprendizado do Neusim (LRN)

```

6 2 20 0
575 1038 2 3.115 26 29 0 1
568 1029 2 3.175 27 28 0 1
553 980 2 3.285 27 30 0 1
558 1021 2 3.215 27 30 0 1
504 934 2 3.324 28 27 0 1
557 1062 4 4.427 35 33 1 0
580 1050 4 4.166 31 36 1 0
547 1047 4 4.616 34 33 1 0
563 1070 4 4.236 37 33 1 0
570 1077 4 4.206 31 35 1 0
565 1101 6 4.236 50 45 0 1
345 616 0 3.960 42 49 0 1
459 566 0 0.600 3 5 0 1
371 569 0 0.920 4 9 0 1
458 663 0 0.840 2 8 0 1
354 589 0 1.000 3 9 0 1
251 451 0 1.260 3 12 0 1
260 529 1 4.760 50 50 0 1
312 514 1 4.720 47 45 0 1
292 472 1 4.100 44 52 0 1

```

B.5 Arquivo de saída do Neusim (OUT)

```

0.008618 0.991680
0.010437 0.989771
0.010318 0.989861
0.008538 0.991738
0.011331 0.988804
0.009066 0.991198
0.011087 0.989099
0.010961 0.989125
0.009546 0.990669
0.009894 0.990255
0.009964 0.990209
0.007333 0.992935
0.007656 0.992614
0.010565 0.989599
0.009157 0.991110

```

Bibliografia

- [1] ABAS, Rasha. **Backpropagation Networks prototype for off-line signature verification**. Minor thesis, RMIT, Department of Computer Science, Melbourne, March 1994.
- [2] HAN, K., SETHI, I. K. **Handwritten signature retrieval and identification**. Pattern Recognition, Letters 17, pp-83-90. Elsevier Press, 1996.
- [3] DROUHARD, J. P., SABOURIN, R., GODBOUT, M. **A neural network approach to off-line signature verification using directional PDF**. Pattern Recognition, V29, N3. pp-415-424. Elsevier Science Ltd., 1996.
- [4] HUANG, Kai; YAN, Hong. **Off-line signature verification based on geometric feature extraction and neural network classification**. Pattern Recognition, V30, N1. pp-9-17. Elsevier Press, 1997.
- [5] GUPTA, Gopal; MCCABE, Alan. **A review of dynamic handwritten signature verification**. Technical Report - James Cook University, Townsville, Australia, 1997.
- [6] HERBST, Ben; COETZER, Hanno. **On an offline signature verification system**. Proceedings of the 9th Annual South African Workshop on Pattern Recognition. Pp39-43. 1998.
- [7] SUEN, Ching Y., XU, Qizhi; LAM, Louisa. **Automatic recognition of handwritten data on cheques - Factor Fiction?** Pattern Recognition Letters, V20 (1999). Pp.1287-1295. Elsevier Press. 1999.
- [8] WESSELS, T., OMLIN, C. W. **A hybrid system for signature verification**. South African Telecommunication Networks and Application Conference. SATNAC Conference 1999.
- [9] RIHA, Zdenek; MATYAS, Vaclav. **Biometric authentication systems**. FI MU Report Series, RS-2000-08, November 2000.
- [10] BALTZAKIS, H., PAPAMARKOS, N. **A new signature verification technique based on a two stage neural network classifier**. Engineering applications of Artificial intelligence 14(2001), pp.95-103, September 2000.
- [11] HUANG, Kai; YAN, Hong. **Off-line signature verification using structural feature correspondence**. Pattern Recognition (In Press, Uncorrected Proof). Elsevier Press. September 2001.
- [12] JAIN, A. K., GRIESS, F. D., CONNELL, S. D. **On-line signature verification**. Pattern Recognition (In Press, Uncorrected Proof). Elsevier Press. Jan. 2002.
- [13] BRAGA, Antônio; LUDERMIR, Teresa; CARVALHO, André. **Redes Neurais Artificiais: Teoria e Aplicações**. Editora LTC. Rio de Janeiro. 2000.
- [14] MITCHELL, Tom. **Machine Learning**. WCB / McGrall-Hill – Computer Science Series. Boston, MA. 1997.

- [15] RUMELHART, D., HINTON, G., WILLIAMS, R. **Learning Internal Representations by Error Propagation**. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition - Vol. 1. Cambridge: MIT Press, 1986.
- [16] OSORIO, Fernando Santos. **Um estudo sobre reconhecimento visual de caracteres através de Redes Neurais**. UFRGS, CPGCC. Dissertação de mestrado, 1991.
- [17] LI, Xiaolin; PLAMONDON, Réjean. **Model-Based Online Handwritten Digit Recognition**. Proceedings of the 14th International Conference on Pattern Recognition, Brisbane (Australia), pp. 1134-1136, August 1998.
- [18] FAHLMAN, Scott E. **An empirical study of learning speed in back-propagation networks**. Carnegie Mellon University – CMU. Computer Science Technical Report CMU-CS-88-162. September 1988.
- [19] FAHLMAN, Scott E., LEBIERE, C. **The Cascade-Correlation learning architecture**. Carnegie Mellon University – CMU. Computer Science Technical Report CMU-CS-90-100. February 1990.
- [20] WIENECKE, Marcus; FINK, Gernot A., SAGERER, Gerhard. **A Handwriting Recognition System Based on Visual Input**. 2nd International Workshop on Compute Vision Systems. IEEE, 2001.
- [21] PACUT, Andrzej; CZAJKA, Adam. **Recognition of Human Signatures**. Proceedings IJCNN'01 International Joint Conference on NN Vol.2 pp1560-1564. NEURAL NETWORKS 2001.
- [22] SAKAMOTO, D., MORITA, H., OHISHI, T., KOMIYA, Y., MATSUMOTO, T. **On-line Signature Verification Algorithm Incorporating Pen Position, Pen Pressure and Pen Inclination Trajectories**. Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing 2001. pp993-996 vol. 2.
- [23] Z., DARIUSZ; Lejtman; GEORGE, Susan E. **On-line handwritten signature verification using wavelets and back-propagation neural networks**. Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing 2001. pp992-993 vol. 2.
- [24] OSORIO, Fernando Santos. **INSS: A hybrid system for constructive machine learning**. Neurocomputing 28 (1999) 191-205. Elsevier Press 1999.
- [25] HALL, Ernest L. **Computer image processing and Recognition**. New York: Academic Press, 1979. 584p.
- [26] KOVALEWSKY, V. A. **Image pattern recognition**. New York: Springer-Verlag, 1980, 241p.
- [27] NIKOLOPOULOS, Chris. **Expert Systems – Introduction to First and Second Generation and Hybrid Knowledge Based Systems**. Marcel Dekker Inc. Press. 1997.
- [28] KOLODNER, J. **Case-Based Reasoning**. Morgan Kaufmann Series in Representation and Reasoning, San Mateo, CA. 1993.
- [29] QUINLAN, J. R. **C4.5- Programs for Machine Learning**. Morgan Kauffman Publishers. San Mateo, CA. 1993.

- [30] NILSSON, N. J. **Artificial Intelligence: A New Synthesis**. Morgan Kaufmann Publishers. 1998.
- [31] GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley Publishing. 1989.
- [32] OSORIO, Fernando S., BITTENCOURT, João R. **Sistemas Inteligentes baseados em Redes Neurais Artificiais aplicados ao Processamento de Imagens**. I WORKSHOP DE INTELIGÊNCIA ARTIFICIAL UNISC – Universidade de Santa Cruz do Sul Departamento de Informática- Junho 2000.
- [33] ROSENBLATT, R. **Principles of Neurodynamics**. Spartan Books. New York. 1959.
- [34] SCHIFFMANN, W., JOOST, M., WERNER, R. **Comparison of Optimized Backpropagation Algorithms**. Proceedings of the European Symposium on Artificial Neural Networks, ESANN'93, Brussels, p.97-104, 1993.
- [35] SCHIFFMANN, W., JOOST, M., WERNER, R. **Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons**. Technical Report, University of Koblenz, Deutschland. September 1995.
- [36] MATSUMOTO, T., MATSUMOTO, H., YAMADA, K., HOSHIRO, S. **Impact of artificial “Gummy” Fingers on Fingerprint Systems**. Proceedings of SPIE Vol. # 4677, Optical Security and Counterfeit Deterrence Techniques IV.
- [37] CATTELL, Raymond B. **Factor Analysis**. New York: Harper Brothers, 1952.
- [38] HARTMANN, Fernando R. **Redes Neurais, Conceitos Básicos e Análise**. Universidade do Vale do Rio dos Sinos. Trabalho de Conclusão. 90p, 1996.