

Reconhecimento de Faces Utilizando Redes Neurais Autoassociativas

Trabalho de Conclusão de Curso

Engenharia da Computação

Sérgio Guerra Prazeres
Orientador: Bruno José Torres Fernandes

SÉRGIO GUERRA PRAZERES

**Reconhecimento de Faces Utilizando
Redes Neurais Autoassociativas**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do título
de Bacharel em Engenharia da Computação
pela Escola Politécnica de Pernambuco –
Universidade de Pernambuco.

Recife, Dezembro de 2010.

À minha família.

Resumo

Com o aumento do uso de computadores para acesso a informações gerais e muitas vezes também restritas é evidente a necessidade de aumento da segurança através de meios de autenticação rápidos, práticos e seguros. Devido a isso, o sistema de reconhecimento de pessoas vem ganhando destaque dentro e fora da academia. Apesar de serem desenvolvidos também sistemas de reconhecimento por outros tipos de biometria, existe um grande interesse de empresas e universidade no campo do reconhecimento facial, pois este é um método que não exige contato físico. Este trabalho visa analisar o desempenho da utilização de redes neurais para o problema de autenticação, comparando os resultados de duas abordagens. Uma usando redes neurais MLP e outra usando MLP Autoassociativa tomando por base o resultado do PCA. No final discutem-se as vantagens da utilização de cada uma delas.

Abstract

With the increased use of computers to access general information and also often restricted is evident the need for increased security through authentication means fast, practical and safe. Because of this recognition system has been gaining attention from people inside and outside academia. Despite being developed recognition systems also for other types of biometrics, there is great interest from companies and universities in the field of face recognition, because this is a method that requires no physical contact. This study aims to examine the performance of neural networks to the problem of authentication, comparing the results of two approaches. One uses a MLP neural networks and the other uses MLP Autoassociative building on the outcome of the PCA. In the end we discuss the advantages of each.

Sumário

Resumo	i
Abstract	ii
Índice de Figuras	v
Índice de Tabelas	vi
Tabela de Símbolos e Siglas	vii
Capítulo 1 Introdução	8
1.1 Motivação e Problema	8
1.2 Objetivos	9
1.3 Metodologia	10
1.4 Organização do Trabalho	10
Capítulo 2 Fundamentação Teórica	12
2.1 Análise de Componentes Principais (PCA)	12
2.1.1 Matemática do PCA	13
2.2 Classificadores de uma única classe	14
2.3 Redes Neurais Artificiais	14
2.3.1 Neurônio biológico	15
2.3.2 Neurônio artificial	16
2.3.3 Perceptron	17
2.3.4 MLP ou Perceptron de múltiplas camadas	18
2.3.5 MLP autoassociativas	21
Capítulo 3 Reconhecimento de Faces	24
3.1 Introdução	24
3.1.1 Tarefas na identificação de uma face	24
3.2 Sistema Proposto	25
Capítulo 4 Experimentos	27
4.1 Redes Neurais usadas para Classificação	28

4.1.1	MLP	28
4.1.2	MLP Autoassociativa	29
4.2	PCA	30
4.3	Comparativo entre as técnicas utilizadas nos experimentos	31
Capítulo 5 Conclusão e Trabalhos Futuros		32
Bibliografia		33
Apêndice A Fundamentação Matemática		36
Apêndice B Telas do Sistema Proposto		38

Índice de Figuras

Figura 1.	Estrutura do neurônio [12]	15
Figura 2.	Comunicação entre neurônios [12].....	16
Figura 3.	Modelo matemático do neurônio artificial, imagem adaptada de [13].....	16
Figura 4.	Analogia dos neurônios biológico e artificial [14].....	17
Figura 5.	Classes linearmente separáveis e não linearmente separáveis [15].....	18
Figura 6.	Exemplo de uma rede neural MLP [17]	20
Figura 7.	MLP Autoassociativa [19]	22
Figura 8.	Rede Neural Autoassociativa, imagem adaptada de [18].....	22
Figura 9.	Etapas do reconhecimento automático de faces, imagem adaptada de [4]	24
Figura 10.	Painel de configuração de estrutura da rede neural	26
Figura 11.	Vetorização da matriz de duas dimensão de uma imagem, imagem adaptada de [20].....	26
Figura 12.	Amostras de imagens do banco ORL [21].....	27
Figura 13.	Tela de configuração de estrutura da Rede Neural	38
Figura 14.	Tela de configuração de propriedades do treinamento da Rede Neural	38
Figura 15.	Tela de parâmetros de teste do PCA	39
Figura 16.	Tela do gerador de imagens aleatórias para base de imagens ORL....	39

Índice de Tabelas

Tabela 1.	Tipos básicos de autenticação	8
Tabela 2.	Resultados obtidos durante os dez experimentos com a MLP, mostrado a média dos resultados.....	28
Tabela 3.	Resultados obtidos durante os dez experimentos com a MLP Autoassociativa, mostrado a média dos resultados.....	29
Tabela 4.	Resultados obtidos durante os dez experimentos com o PCA, mostrado a média dos resultados.....	30
Tabela 5.	Comparativo dos resultados obtidos para redes neurais para classificação das imagens da base de dados ORL	31

Tabela de Símbolos e Siglas

RNA – Redes Neurais Artificiais

MLP – Multi Layer Perceptron

PCA – Principal Component Analysis

MLPAA - Multi Layer Perceptron Autoassociative

Capítulo 1

Introdução

1.1 Motivação e Problema

Com o passar do tempo, os computadores vêm se tornando cada vez mais indispensáveis, tanto no dia a dia de pessoas como no de corporações. Mais importante que os computadores são as informações armazenadas e a possibilidade de acessá-las de várias localidades do mundo.

Essas informações podem ser de acesso público ou de acesso privado. No caso de informações pessoais e corporativas, onde o seu conteúdo deve ser restrito a algumas pessoas, deve-se utilizar um método de autenticação para a autorização. A autenticação é o processo de verificação da identidade de uma pessoa [1]. Existem três tipos básicos de autenticação [2]:

Tabela 1. Tipos básicos de autenticação

Tipo de Solução	Resumo	Exemplos
Baseadas no Conhecimento	O que se sabe	Senhas
Baseadas na Propriedade	O que se tem	Tokens, cartões, chips
Baseadas em Características	O que se é	Biometria

As autenticações baseadas no conhecimento exigem memorizações de senhas por parte do usuário. Por sua vez, estas senhas devem ser as mais complexas possíveis para dificultar a ação de fraudadores, o que aumenta ainda mais a dificuldade de memorização.

Como forma alternativa ao uso de senhas surgem as autenticações baseadas na propriedade, evitando a necessidade de memorização de senhas, contudo esses artefatos (Tokens, cartões ou chips) podem ser roubados ou fraudados.

É aí que a biometria ganha importância. Por ser um método de autenticação baseado em características corporais próprias e únicas para identificação do usuário. Eliminando a necessidade de memorização, como é o caso das senhas e o

perigo de perder, como é o caso dos cartões. Ainda, este método de autenticação exige a presença física do usuário, evitando a fraude por substituição de pessoas.

A biometria é o estudo estatístico de características físicas ou comportamentais das pessoas, usado como forma de identificá-las tanto para a identificação criminal como para o controle de acesso [3]. Os principais mecanismos Biométricos existentes comercialmente são [2]:

- Reconhecimento de Impressão Digital
- Reconhecimento de Íris
- Reconhecimento de Voz
- Reconhecimento de Veias das Mãos
- Reconhecimento de Face
- Reconhecimento de Retina
- Reconhecimento de Geométrico das Mãos
- Reconhecimento de Assinatura

Existem também alguns sistemas não usados comercialmente em fase de teste como:

- Reconhecimento por Odor
- Reconhecimento por Geometria da Orelha
- Reconhecimento por DNA
- Reconhecimento por Ondas Cerebrais

Um dos mecanismos biométricos mais utilizados é o Reconhecimento Facial. Apesar de não ser um dos mais precisos, devido a grande variação de poses, iluminação e expressões faciais ele tem algumas vantagens. A principal delas é o fato de ser um sistema de autenticação não intrusivo, ou seja, o usuário não tem a necessidade de tocar ou interagir com nenhum dispositivo biométrico durante um período de tempo significativo, possibilitando assim a identificação de um indivíduo mesmo sem consentimento do mesmo [4].

1.2 Objetivos

Este Trabalho de Conclusão de Curso tem como principal objetivo modelar e implementar dois ambientes para reconhecimento facial. O primeiro utilizando Redes Neurais Artificiais Autoassociativas e o segundo usando RNA Multi Layer Perceptron

como One-Class-Classifiers. Ambos com o treinamento baseados no conjunto de imagens ORL disponibilizado em [5].

A meta principal é construir uma prova de conceito a partir da comparação dos resultados obtidos do teste da RNA Autoassociativa com os resultados obtidos do teste da RNA MLP, identificando as vantagens e desvantagens na utilização da RNA Autoassociativa para autenticação direta do usuário. Além disso, utilizar a RNA Autoassociativa como extrator de característica e comparar os resultados com as características extraídas pelo algoritmo de Análise de Componentes Principais, avaliando a precisão das características extraídas usando um algoritmo KNN para autenticação do usuário.

1.3 Metodologia

Inicialmente, será construída uma rede neural com parâmetros de configuração que possibilitem o treinamento e o teste com a base de dados de imagens de faces ORL, na modalidade Multi Layer Perceptron e MLP Autoassociativa.

Em seguida, será confrontada a performance do uso da MLP Autoassociativa, da MLP e do PCA como classificadores de uma única classe. Ambas aplicadas para reconhecimento da face em questão.

Finalmente, as conclusões obtidas serão analisadas com os resultados encontrados nas comparações propostas acima.

1.4 Organização do Trabalho

No primeiro capítulo encontra-se o texto introdutório, organizado em subseções como motivação e problema, objetivos, metodologia e uma visão geral deste trabalho de conclusão de curso (TCC).

O segundo capítulo contém os fundamentos teóricos necessários para o entendimento do sistema proposto, apresentando conceitos de redes neurais MLP, Autoassociativas, PCA e Classificadores de Uma Única Classe.

No terceiro capítulo será apresentado os principais problemas do reconhecimento de face, o sistema proposto para manipulação das imagens e das redes neurais.

O quarto capítulo é destinado aos resultados alcançados nos experimentos das RNAs Autoassociativa comparados com os resultados da RNAs MLP e do PCA como classificadores diretos.

Por fim, no ultimo capítulo são apresentadas as conclusões referentes aos resultados obtidos no capítulo anterior, as dificuldades encontradas durante o desenvolvimento do estudo e possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Análise de Componentes Principais (PCA)

Análise de Componentes Principais ou PCA é uma ferramenta bastante utilizada em reconhecimento estatístico de padrões para seleção de características. Envolve um procedimento matemático que tem como objetivo reduzir uma grande dimensionalidade do espaço de dados (variáveis observadas) para uma menor dimensionalidade intrínseca do espaço de características (variáveis independentes), que são necessárias para descrever os dados resumidamente. Este espaço de característica é chamado de componentes principais, que estão relacionados com as variáveis originais por uma transformação ortogonal. Esta transformação é definida de tal forma que o primeiro componente principal deve capturar tanto da variabilidade quanto possível. A segunda dimensão é ortogonal à primeira e, sujeita a esta restrição, captura tanto da variabilidade restante quanto possível, e assim por diante. Dependendo do campo de aplicação, o PCA também é conhecido por transformada de Karhunen-Loève (KLT), transformada de Hotelling ou Decomposição Ortogona Própria (POD) [6].

O PCA pode ser utilizados para remoção de redundância, extração de características, compressão de dados, etc. Por ser uma técnica clássica no domínio linear, aplicações com modelos lineares são apropriadas como o processamento de imagens, processamento de sinais, teoria de sistemas e controle, etc [7].

A idéia principal do uso do PCA para reconhecimento de face é expressar um grande vetor de uma dimensão de pixel construído a partir da matriz de duas dimensões da imagem da face, em um vetor reduzido de espaço de características de uma dimensão. Essa transformação de duas dimensões da matriz da imagem para uma dimensão, chamada de vetor coluna da imagem é demonstrado na figura 11.

2.1.1 Matemática do PCA

De acordo com Gilliano [4], o PCA é descrito como uma transformada de um conjunto de vetores de entrada com o mesmo tamanho K formados a partir do vetor de n dimensões $\vec{x} = [x_1, x_2, \dots, x_n]^T$ em um vetor y de acordo com a equação

$$\vec{y} = A(\vec{x} - m_x) \quad (1)$$

Lembrando que cada linha do vetor x consiste em K valores pertencentes a uma entrada. O vetor m_x na equação (1) é o vetor de valores médios de todas as variáveis de entrada definidas pela relação

$$m_x = E\{x\} = \frac{1}{K} \sum_{k=1}^K x_k \quad (2)$$

A matriz A da equação (1) é determinada pela matriz de covariância C_x . As linhas da matriz A são formadas pelos autovetores e de C_x organizados em ordem decrescente de acordo com os autovalores correspondentes. O apêndice A possui uma descrição detalhada do que são matrizes de covariância, autovalores e autovetores. A avaliação da matriz C_x é possível de acordo com a relação

$$C_x = E\{(x - m_x)(x - m_x)^T\} = \frac{1}{K} \sum_{k=1}^K x_k x_k^T - m_x m_x^T \quad (3)$$

Como a dimensão do vetor \vec{x} de entradas variáveis é de n dimensões, o que implica que a dimensão de C_x é de $n \times n$. Os elementos $C_x(i, i)$ situados na diagonal principal são as variâncias

$$C_x(i, i) = E\{(x_i - m_i)^2\} \quad (4)$$

de \vec{x} e os outros valores $C_x(i, j)$ determinam a covariância entre as variáveis de entrada x_i, x_j .

$$C_x(i, j) = E\{(x_i - m_i)(x_j - m_j)\} \quad (5)$$

As linhas de A na equação (1) são ortonormais, então a inversão do PCA é possível de acordo com a relação

$$\vec{x} = A^T \vec{y} + m_x \quad (6)$$

O núcleo do PCA definido pela equação (1) possui algumas outras propriedades interessantes, que podem ser usadas em processamento de imagens e de sinais para alcançar outros objetivos.

2.2 Classificadores de uma única classe

Os classificadores de uma única classe são descritores capazes de aprender domínios restritos em um espaço multi-dimensional de padrões utilizando apenas um conjunto positivo dos exemplos. O uso desses descritores é possível quando se tem disponível apenas uma das classes de padrões, ou quando as outras classes são grandes demais ou difíceis de caracterizar [8].

Apenas os padrões dentro do domínio do classificador, previamente treinado, são reconhecidos, de forma que todos os outros objetos não pertencentes a este domínio são rejeitados. Isso é, esses descritores são capazes de distinguir se o objeto faz parte da classe alvo ou de uma classe desconhecida, a qual não fez parte do aprendizado [8].

Alguns exemplos da utilização desses descritores são nas áreas de monitoramento de máquinas, detecção de novidades, recuperação de dados de imagens e identificação pessoal [9].

2.3 Redes Neurais Artificiais

O cérebro humano tem aproximadamente 10 bilhões de neurônios que se interconectam formando uma rede neural ou rede neuronal. Esta rede neural tem a capacidade de processar milhões de informações e realizar em torno de 60 trilhões de ligações sinápticas [10].

O cérebro é um sistema de processamento maciçamente paralelo, que embora seja constituído por neurônios individualmente lentos, pode processar rapidamente milhões de informações ao mesmo tempo [10].

Redes Neurais Artificiais ou simplesmente Redes Neurais, são sistemas computacionais que procuram reproduzir, através de programas de computadores ou circuitos eletrônicos, de maneira simplificada, o funcionamento de uma rede neural biológica do cérebro humano. Uma RNA é constituída por um conjunto de unidades de processamento denominadas de neurônios artificiais. Essas unidades emulam, através de modelos matemáticos, o processamento de informações dos neurônios biológicos por meio de aprendizagem e aproximação. A interconexão desses elementos possibilita o processamento de informações em paralelo.

Uma Rede Neural MLP é constituída por elementos de processamento interconectados dispostos em camadas, onde os neurônios da camada escondida são responsáveis pela não linearidade desta rede e os pesos sinápticos, que interconectam os neurônios, são responsáveis pela memória da mesma. Este modelo de rede neural apresenta algumas características análogas às observadas no cérebro humano. Como por exemplo o conhecimento, que é adquirido pela rede neural através de um processo de aprendizagem. Quando este aprendizado é bem sucedido proporciona à rede uma capacidade de generalização baseado no conhecimento adquirido, ou seja, a rede neural consegue produzir saídas adequadas para as entradas que não foram vistas durante o processo de treinamento [11].

2.3.1 Neurônio biológico

O neurônio biológico é formado por um corpo celular de onde saem finos prolongamentos mais curtos e juntos do corpo celular chamados de dendritos e um prolongamento mais comprido chamado de axônio, como mostra a figura 1. No axônio encontram-se terminais denominados individualmente de telodendro e coletivamente de árvore de terminal [12].

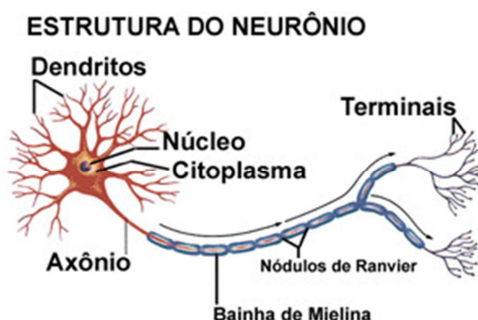


Figura 1. Estrutura do neurônio [12]

A passagem do impulso nervoso entre dois neurônios ocorre na região da sinapse, geralmente formada pela união do axônio de um neurônio com os dendritos de outro. Essa transmissão sucede do axônio de um neurônio para os dendritos do neurônio seguinte, nunca no sentido contrário [10].



Figura 2. Comunicação entre neurônios [12]

Para a compreensão do funcionamento do neurônio artificial é essencial o entendimento da Lei do Tudo ou Nada a qual rege o funcionamento do neurônio biológico. Esta lei tem relação com a intensidade do estímulo ou limiar excitatório, a partir do qual o neurônio dispara ou não o impulso nervoso. Ou seja, se a intensidade do estímulo for inferior ao limiar excitatório, não ocorrerá o impulso nervoso. Caso a intensidade do estímulo passe esse limiar, o potencial de ação do neurônio será sempre o mesmo, qualquer que seja a intensidade do estímulo [10].

2.3.2 Neurônio artificial

Segundo Valença [10], o neurônio artificial é um modelo análogo de uma célula nervosa, definido como a menor unidade de uma rede neural. O comportamento do neurônio artificial utilizado para caracterizar o neurônio biológico é o primeiro passo para compreender as redes neurais. A primeira proposta de uma representação matemática foi feita por McCulloch e Pitts em 1943.

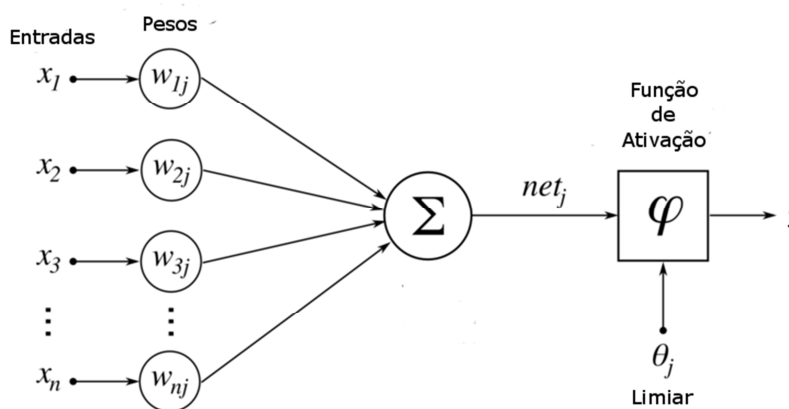


Figura 3. Modelo matemático do neurônio artificial, imagem adaptada de [13]

Esse modelo representa o neurônio biológico utilizando uma regra de propagação e uma função de ativação. A emissão de sinal deste neurônio é baseada em um limiar em analogia com a lei do tudo ou nada.

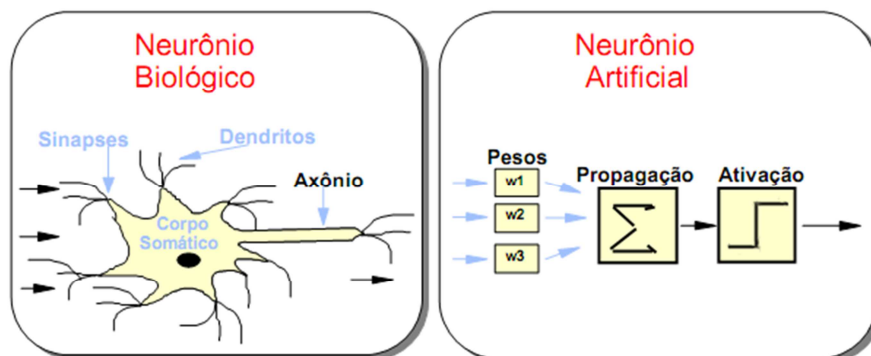


Figura 4. Analogia dos neurônios biológico e artificial [14]

Na figura 3, $x_1, x_2, x_3, \dots, x_N$ são as entradas do neurônio denominadas de i e a saída de chamada de j . A entrada líquida net_j é dada por:

$$net_j = \sum_{i=1}^N w_{ij}x_i - \theta \quad (7)$$

sendo w_{ij} os pesos sinápticos e θ o limiar.

Onde:

$$\varphi = f(net_j) = \begin{cases} 1, \forall net_j \geq 0 \\ 0, \forall net_j < 0 \end{cases} \quad (8)$$

A função de ativação φ na figura 3 é a função degrau.

2.3.3 Perceptron

O Perceptron é o modelo de rede neural mais simples. Formado por duas camadas, a camada de entrada e a camada de saída. Em sua arquitetura são apresentadas uma ou várias unidades de processamento na camada de entrada e um ou vários neurônios artificiais na camada de saída. Os neurônios de entrada se conectam através de pesos sinápticos com os neurônio de saída [10].

O perceptron tem capacidade de aprender, ou seja, dependendo dos exemplos apresentados à rede, os pesos sinápticos são reajustados através de uma regra de aprendizagem, por exemplo, a regra de aprendizagem de Hebb que diz:

“Se um neurônio recebe um estímulo de outro neurônio e ambos são altamente ativos, o peso entre estes neurônios deve ser fortalecido, caso contrário enfraquecido” [10].

A rede perceptron tem como limitação o fato de só resolver problemas de classificação de conjuntos linearmente separáveis.



Figura 5. Classes linearmente separáveis e não linearmente separáveis [15]

O treinamento do Perceptron é do tipo supervisionado, uma vez que, o valor desejado na saída é apresentado à rede juntamente com as variáveis de entrada durante o treinamento. O cálculo para reajuste dos pesos é simples, porque todos os valores da rede são conhecidos. O ajuste Δw_{ij} a ser realizado é calculado pela expressão:

$$\Delta w_{ij} = \eta x_i (d_j - y_j) \quad (9)$$

onde, η é a taxa de aprendizagem; x_i é o valor apresentado a rede; d_j é o valor desejado na saída e y_j é o valor calculado na saída. Inicialmente os valores dos pesos são atribuídos aleatoriamente, em seguida a resposta da rede é calculada e comparada com o valor desejado para o reajuste ou não dos pesos sinápticos [10].

Mais detalhes sobre as redes neurais Perceptron são encontrados em [16].

2.3.4 MLP ou Perceptron de múltiplas camadas

As redes MLP são uma generalização das redes Perceptron Simples adicionando uma ou mais camadas intermediária. A MLP resolve o problema da restrição do Perceptron, ou seja, uma MLP com uma única camada escondida e uma função de ativação sigmóide pode aproximar qualquer função contínua arbitrária [10].

As redes MLPs são formadas por no mínimo três camadas com pelo menos um neurônio em cada camada. A camada de entrada, onde cada neurônio representa uma variável de entrada do problema proposto. Pelo menos uma camada intermediária, onde esta camada é responsável pela não linearidade da rede, permitindo que a rede resolva problemas reais. Por fim, uma camada de saída onde seus neurônios são as respostas da rede, ou seja, as variáveis que se quer prever ou classificar [16].

O modelo de cada uma das unidades de processamento da rede neural possui uma função de ativação não linear. Uma função muito utilizada em MLPs é a função sigmoide, como por exemplo, a logística ou a tangente hiperbólica. A definição da função de ativação é o que ditará o funcionamento da rede. Outra característica da MLP é que ela possui um alto grau de conectividade entre seus elementos processadores. De forma que todos os neurônios de uma camada devem estar ligados a todos os neurônios das camadas adjacentes [11], como mostra a figura 6.

O processamento de cada camada é influenciado pelo processamento das unidades da camada anterior, onde cada camada deve desempenhar o seu papel. As camadas de entradas são as receptoras de estímulos. A primeira camada oculta tem cada unidade definindo uma reta no espaço de decisão, refletindo as características dos padrões apresentados. A segunda camada oculta combina as retas definidas na camada anterior, formando regiões convexas onde o número de lados é definido pelo número de unidades da camada anterior conectados a cada unidade de processamento desta camada. A camada de saída combina as regiões da camada anterior, definindo o espaço de saída da rede [11].

As camadas intermediárias de uma rede neural são como detectores de características. Essas características são representadas internamente através dos pesos sinápticos.

Estudos sobre o número de camadas intermediárias, necessárias para implementar uma rede neural, indicam que uma camada oculta é suficiente para aproximar qualquer função contínua. Enquanto que duas camadas ocultas são suficientes para aproximar qualquer função matemática [11].

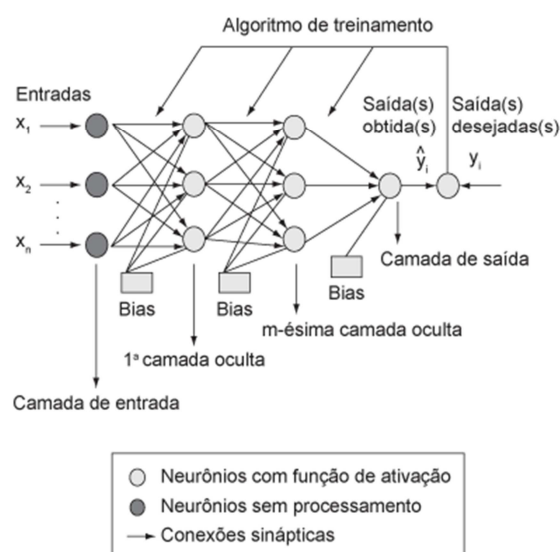


Figura 6. Exemplo de uma rede neural MLP [17]

As redes MLPs ficaram populares depois do aparecimento do algoritmo de aprendizagem conhecido por backpropagation, que teve origem no trabalho desenvolvido por Paul Werbos em 1974. Este algoritmo consiste em uma generalização da técnica do gradiente descendente [16].

O treinamento das MLPs, assim como o do perceptron, são do tipo supervisionado. Porém o treinamento não é tão simples como o do perceptron, devido a inclusão de pelo menos uma camada oculta. A complexidade do treinamento da MLP está associada ao fato do não conhecimento do erro da camada intermediária, o que é uma necessidade para o reajuste dos pesos [16].

Para reajustar os pesos sinápticos da rede MLP, o backpropagation faz uma propagação recursiva do erro. Isso é feito em dois passos. O primeiro passo é a propagação do sinal da camada de entrada para a camada de saída, calculando o sinal de saída e o erro. No segundo passo, o erro é propagado recursivamente da camada de saída para a camada de entrada, onde o ajuste nos pesos deve ser proporcional ao sentido contrário do gradiente da função erro. Para isso é adicionado um termo chamado de sensibilidade ao cálculo do valor Δw_{ij}^m a ser ajustado nos pesos de cada camada. Esse ajuste é calculado pela expressão

$$\Delta w_{ij}^m = \alpha \delta_i^m y_j^{m-1} \quad (10)$$

onde, α é a taxa de aprendizagem, δ_i^m é a sensibilidade calculada para a camada m e y_j^{m-1} é o resultado calculado no neurônio j da camada anterior $m - 1$ [16].

Quando se utiliza como função objetivo o Erro Médio Quadrático a sensibilidade da camada de saída é facilmente calculada a partir da expressão

$$\delta_i^{m-1} = f^{m-1'}(net_i^{m-1}) \cdot e_i(n) \quad (11)$$

onde δ_i^{m-1} é a sensibilidade para os neurônios da camada de saída, $f^{m-1'}(net_i^{m-1})$ é a derivada da função de ativação dos neurônios da camada de saída e $e_i(n)$ é o erro de um neurônio da camada de saída " i " qualquer para um exemplo " n " qualquer apresentado a rede.

Quando j é um neurônio escondido, o valor da sensibilidade não é conhecido, pois não se conhece o erro da camada intermediária. Como apenas a sensibilidade da camada de saída é conhecida, então essa informação é usada para calcular as sensibilidades das demais camadas, através da retro-propagação da sensibilidade da camada de saída. A sensibilidade da camada escondida é calculada através da expressão

$$\delta_j^{m-1} = f^{m-1'}(net_j^{m-1}) \sum_{i=1}^N w_{ij}^m \cdot \delta_i^m \quad (12)$$

onde, " i " corresponde aos neurônios da camada que recebe o sinal, " j " os neurônios que emitem o sinal e δ_i^m a sensibilidade da camada de saída. Desta forma, calcula-se os valores das sensibilidades da última para a primeira camada, permitindo que os pesos sejam reajustados de forma a minimizar o erro entre a saída calculada e a saída desejada pela rede [16].

2.3.5 MLP autoassociativas

As MLP Autoassociativas são redes neurais multicamadas que tem o mesmo número de neurônios nas camadas de entrada e saída, e um número menor de neurônios na camada escondida como mostra a figura 7. No treinamento desta rede, o vetor de entrada é usado como a saída desejada [18].

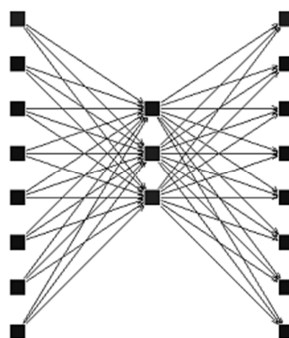


Figura 7. MLP Autoassociativa [19]

Isso, acarreta na organização de uma rede de codificação (compressão) entre a camada de entrada e a camada escondida, e uma rede de decodificação entre a camada escondida e a camada de saída como mostra a figura 8. A rede neural autoassociativa aplicada a imagem, provê uma redução de dimensionalidade na camada escondida transformando a informação contida na mesma em um espaço de característica, como no PCA, permitindo o uso desta rede para extração de característica na área de reconhecimento facial [18].

Outra abordagem da MLP Autoassociativa também é possível, utilizando a rede para verificação de padrões. O critério de verificação é baseado na distância euclidiana entre a entrada e a saída, através da verificação de um limiar, chamado de T . Ou seja, um padrão x é aceito se

$$|f(x) - x| \leq T \quad (13)$$

onde $f(x)$ é a saída da rede, x é a entrada e T é o limiar. A idéia é que desta forma apenas os padrões das classes usadas no treinamento serão susceptíveis a reproduzir uma saída com uma tal aproximação que a distância euclidiana seja menor que o limiar [19].

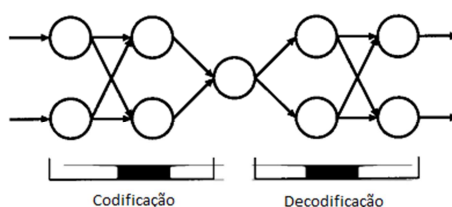


Figura 8. Rede Neural Autoassociativa, imagem adaptada de [18]

O limiar pode ser escolhido como sendo o maior limiar para as entradas durante a validação, ou seja,

$$t_{max} = \text{Max}(x_1, x_2, \dots, x_n) \quad (14)$$

onde x_i são os padrões usados para a validação [19].

As redes neurais MLPs autoassociativas são mais indicadas para reconhecimento de padrões que as redes MLP que funcionam como classificadores de 1 para N. O motivo disso é o fato das MLPs feedforward realizarem a discriminação das classes por superfícies de separação no espaço de padrões os quais são gerados pelo algoritmo de aprendizagem com o objetivo de discriminar os exemplos dados. As superfícies de separação resultantes são geradas de forma a particionar os dados de entrada, mas tais superfícies não são necessariamente fechadas, ou seja, elas não cobrem os exemplos capturando suas distribuições de probabilidade.

As MLPs Autoassociativas resolve as inconveniências advindas das superfícies de separação abertas. Nos autoassociadores como o critério de rejeição estão relacionados à maneira com que a entrada é aproximada pela saída, o que proporciona que os mesmos produzam superfícies de separação fechadas cobrindo os dados de treinamento.

Capítulo 3

Reconhecimento de Faces

3.1 Introdução

Um sistema de reconhecimento facial utiliza atributos exclusivos dos seres humanos. Esses sistemas procuram identificar características faciais únicas de cada pessoa, que possam ser medidos e utilizados como referências para o reconhecimento de quem os porta. Como por exemplo a distância entre os olhos, sobrancelhas, etc [20].

3.1.1 Tarefas na identificação de uma face

O reconhecimento de face é uma particularização do reconhecimento de padrões, onde tem o objetivo de classificar informações baseados em conhecimento a priori ou em informações estatísticas dos padrões [4].

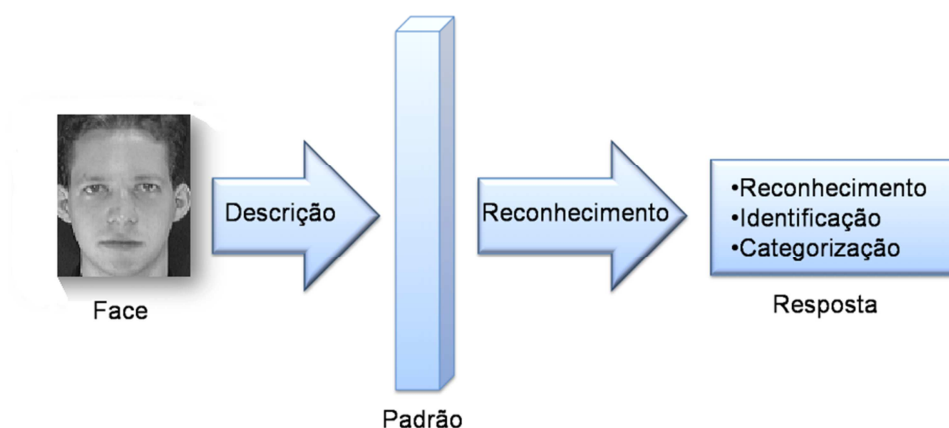


Figura 9. Etapas do reconhecimento automático de faces, imagem adaptada de [4]

É importante destacar algumas tarefas do reconhecimento de faces como a classificação, conhecido-desconhecido, verificação e reconhecimento completo.

A classificação consiste na identificação de uma face x considerando que essa pessoa faça parte do conjunto Ω , onde Ω é o conjunto de todas as pessoas existentes.

O conhecido-desconhecido tem o objetivo de decidir se a face é ou não um membro de Ω , verificando se x pode ser classificado como um padrão de alguma classe de Ω .

Na verificação procura-se confirmar a identidade de uma pessoa, parecido com o conhecido-desconhecido, porém com apenas uma pessoa conhecida dentro da base de treinamento.

O reconhecimento completo visa determinar a identidade de uma pessoa através da face pertencente a uma classe de Ω .

3.2 Sistema Proposto

Para os experimentos feitos neste TCC foi desenvolvido um sistema composto por uma MLP de parâmetros configuráveis através de uma interface gráfica. Na aba “Estrutura da Rede Neural”, mostrado na figura 10, é possível configurar os parâmetros relativos a estrutura da rede como: o número de neurônios da camada de entrada, da camada escondida e da camada de saída; “Tipo da Rede Neural” é possível optar por dois tipos: MLP ou MLP Autoassociativa; “Tipo de dados do arquivo de entrada” é definido o tipo de dados contido no arquivo de entrada, devendo ser “Dados” para um arquivo contendo dados inseridos diretamente no arquivo de entrada ou “Imagem” para um arquivo contendo a URL dos arquivos de imagens. Neste ultimo, se selecionado imagem, será habilitado um botão, onde é possível passar como parâmetro um exemplo de uma das imagens contidas no arquivo de entrada.

Esse botão definirá a quantidade de neurônios de entrada baseado no tamanho da imagem de exemplo, os campos de arquivo de entrada e de saída, onde será selecionado ou escrito o caminho dos arquivos, e a seleção de “Função de Ativação” contendo os tipos de função que será usado nos neurônios da camada de saída, podendo ser selecionados os tipos: “Sigmoide Logística”, “Degral” ou “Linear”. No Apêndice B são mostradas todas as telas do sistema proposto.

Figura 10. Painel de configuração de estrutura da rede neural

As imagens usadas são da base de dados ORL disponíveis em [5], pré-processadas para o tamanho de 23 colunas por 28 linhas e convertidas para escala de cinza, onde os valores dos pixels variam entre 0 e 255. Como a entrada da rede neural é de uma dimensão, se fez necessário a vetorização da imagem, ou seja, concatenar todas as 23 colunas em uma como mostra a figura 11. Os valores dos pixels foram normalizados entre 0,15 e 0,85.

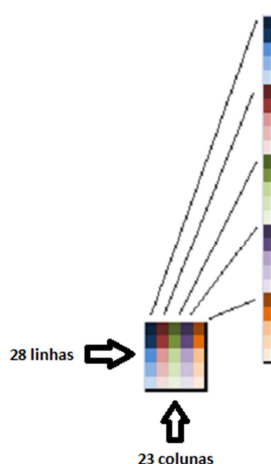


Figura 11. Vetorização da matriz de duas dimensão de uma imagem, imagem adaptada de [20]

Capítulo 4

Experimentos

Para os experimentos realizados no trabalho foram utilizadas quatrocentas imagens da base de dados ORL [5]. Essa base de dados é composta por dez imagens de pessoas num total de quarenta pessoas. Os experimentos foram realizados para todas as quarenta pessoas, sendo a seleção das imagens de treino, validação e teste de forma aleatória. Por exemplo, para uma configuração de três imagens de treino, duas de validação e cinco de testes foram escolhidos três imagens aleatórias entre as dez de uma mesma pessoa, mais duas imagens aleatórias entre as sete restantes e as demais imagens dessa pessoa foram usadas para o teste.



Figura 12. Amostras de imagens do banco ORL [21]

Os testes das redes neurais foram executados variando a quantidade de imagens de treino, validação e teste. Essa quantidade será indicada como 1-2-7 para uma imagem de treino, duas de validação e sete para teste por pessoa. Os ensaios para todas as redes foram realizados variando a relação entre treino-validação-teste em 1-2-7, 3-2-5, 5-2-3 e 7-2-1, isso quer dizer que sempre as redes foram validadas com oitenta imagens das quatrocentas imagens disponíveis. Anotados os números de acertos, erros e a taxa de acerto para cada experimento, onde a taxa de acerto é o numero de acertos da rede dividido pelo numero total de imagens testadas. Foram realizadas dez iterações com cada relação treino-validação-teste para cada tipo de rede usada, e obtida a média aritmética destas dez iterações.

4.1 Redes Neurais usadas para Classificação

4.1.1 MLP

Para as redes neurais MLPs foi treinada uma rede neural para todas as quarenta pessoas de forma que, quando usada uma imagem de treino por pessoa, a MLP foi treinada com quarenta imagens.

O maior número de acertos foi encontrado quando a rede foi treinada com 280 imagens e testada com 40, usando a relação 7-2-1 por pessoa. A MLP acertou 100% das 40 imagens de teste apresentadas a mesma em uma das dez iterações. A média aritmética da taxa de acertos das dez iterações foi de 0,9625, o que corresponde a 96,25% de acertos.

A pior taxa de acerto, a partir de agora chamada de T.A., foi encontrada quando a rede foi treinada com 40 imagens e testada com 280 das 400 imagens. Com a relação 1-2-7, a T.A. foi de 0,56. Ou seja, a rede chegou a errar 123 vezes durante os 280 testes.

Na tabela abaixo é mostrada a média das dez iterações para cada configuração da rede.

Tabela 2. Resultados obtidos durante os dez experimentos com a MLP, mostrado a média dos resultados

Relações	Média T.A.
1-2-7	0,6201
3-2-5	0,8810
5-2-3	0,9291
7-2-1	0,9625

4.1.2 MLP Autoassociativa

As redes autoassociativas foram avaliadas treinando uma rede MLP autoassociativa para cada pessoa. Desta forma, foram usadas quarenta redes, cada uma treinada com a quantidade de imagens dependendo da relação utilizada. Por exemplo, para a relação 1-2-7 foi empregada uma imagem para treinamento, duas para validação e sete imagens de teste de uma pessoa para cada rede neural referente àquela pessoa.

A melhor taxa de acerto foi de também de 1,0, ou seja, 100% de acerto para a relação 7-2-1. A média aritmética das dez iterações foi de 0,9550 referente a 95,50% de acerto.

A pior T.A. foi de 0,5920 para a relação 1-2-7, usando apenas uma imagem para treino por pessoa. A média obtida para essa configuração de treinamento foi de 0,6847.

Na tabela abaixo é mostrada a média das dez iterações para cada configuração de treinamento da rede.

Tabela 3. Resultados obtidos durante os dez experimentos com a MLP Autoassociativa, mostrado a média dos resultados

Relações	Médias T.A.
1-2-7	0,6847
3-2-5	0,8620
5-2-3	0,9191
7-2-1	0,9550

4.2 PCA

Para os experimentos do PCA foi usado o algoritmo Eigenfaces em Java. Como não se usa imagens para validação no algoritmo do PCA, os testes foram realizados somando as quantidades das imagens de treino e validação das relações usadas para o treino das Redes Neurais, onde estas imagens foram transformadas nos vetores de características (base de treino).

Para cada pessoa testada a imagem é transformada pelo PCA também no vetor de característica e comparado através da menor Distância Euclidiana encontrada na base de treino.

O número de características utilizadas para o cálculo da Distância Euclidiana foi fixado em 10.

Para o PCA, também foram repetidos os experimentos dez vezes, onde o melhor resultado da T.A. foi igual a 1,0, ou seja, 100% de reconhecimento para as dez imagens testadas e o pior resultado T.A. foi de 0,5551.

Tabela 4. Resultados obtidos durante os dez experimentos com o PCA, mostrando a média dos resultados

Relações	Relações	Médias T.A.
-	1-0-9	0,5801
1-2-7	3-0-7	0,8285
3-2-5	5-0-5	0,8750
5-2-3	7-0-3	0,9412
7-2-1	9-0-1	0,9750

4.3 Comparativo entre as técnicas utilizadas nos experimentos

Tabela 5. Comparativo dos resultados obtidos para redes neurais para classificação das imagens da base de dados ORL

Relação	MLP	MLPAA	PCA
1-2-7	0,6201	0,6847	0,5801
3-2-5	0,8810	0,8620	0,8285
5-2-3	0,9291	0,9191	0,8750
7-2-1	0,9625	0,9550	0,9412
9-0-1	-	-	0,9750

Após a visualização dos resultados obtidos através dos experimentos descritos no capítulo anterior, é possível observar a proximidade dos resultados entre a MLP e a MLP Autoassociativa para classificação. Essa diferença também foi influenciada pela aleatoriedade na escolha das imagens para o treinamento e testes. Esta semelhança implica em um desempenho próximo de ambos os tipos de redes para utilização em um problema prático.

Uma grande vantagem na utilização da MLP Autoassociativa deve ser mencionada. Como é treinada uma rede diferente para cada pessoa a ser reconhecida, ao necessitar que mais uma pessoa seja adicionada ao problema, não é necessário um retreinamento das redes das outras classes. Para inclusão de uma nova pessoa, adiciona-se uma nova rede autoassociativa e treina a mesma apenas com padrões da nova classe adicionada.

Ao contrário da rede Autoassociativa, as redes MLP necessitam ser treinadas novamente com todos os padrões cada vez que se deseja adicionar uma nova classe de padrões. Porque uma única rede é treinada para todos os padrões.

Capítulo 5

Conclusão e Trabalhos Futuros

Este trabalho de conclusão propôs a comparação de duas técnicas para o problema de reconhecimento de face. A primeira delas foi a rede neural MLP simples e a segunda a rede neural MLP Autoassociativa. O PCA foi treinado com uma única configuração, para ter um parâmetro comparativo, pois é uma técnica bastante utilizada no campo de reconhecimento de padrões.

Como os resultados dos experimentos ficaram muito próximos entre as duas redes, conclui-se que ambas podem ser utilizadas mantendo o desempenho da aplicação. Contudo, é importante ressaltar uma vantagem da rede autoassociativa, onde ao adicionar um novo indivíduo ao problema, não é necessário refazer o trabalho de treinamento para as outras classes.

Como trabalho futuro pretende-se expandir os testes das redes com outras bases de treinamento, com diferenças mais complexas e maior variedade de relações de treinamento, validação e teste, efetuando os mesmos para o PCA.

Bibliografia

[1] AUTENTICAÇÃO. **Wikipédia**, 2010. Disponível em: <<http://pt.wikipedia.org/wiki/Autentica%C3%A7%C3%A3o>>. Acesso em: 01 out. 2010.

[2] GRUPO DE TELEINFORMATICA E AUTOMAÇÃO. **UFRJ**, 2006. Disponível em: <http://www.gta.ufrj.br/grad/06_2/miguel/REDES2_Miguel_Biometria.htm>. Acesso em: 01 out. 2010.

[3] BIOMETRIA. **Wikipedia**, 2010. Disponível em: <<http://pt.wikipedia.org/wiki/Biometria>>. Acesso em: 27 set. 2010.

[4] MENEZES, G. G. S. D. Reconhecimento de Face Usando Análise de Componentes Principais e Morfologia Matemática, Recife, nov. 2009.

[5] AT & T, L. The Database of Faces. Disponível em: <<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>>.

[6] PRINCIPAL Component Analysis. **Wikipedia**. Disponível em: <http://en.wikipedia.org/wiki/Principal_component_analysis>. Acesso em: 20 out. 2010.

[7] KIM, K. **Face Recognition using Principle Component Analysis**. University of Maryland, College Park. MD 20742, USA.

[8] FENG, A.; CHEN, B.; LIU, X. **Learning the Boundary of One-Class-Classififer Globally and Locally**. Computer Science & Technology College - Nanjing University of Aeronautics & Astronaltics. Nanjin, P.R. China.

[9] EL-YANIV, R.; NISENSEN, M. **Optimal Single-Class Classification Strategies**. Departament of Computer Science Technion - Israel Institute of Technology. Technion, Israel 32000.

[10] VALENÇA, M. **Aplicando redes neurais: um guia completo**. Olinda: Livro Rápido, 2005.

[11] GATTASS, M. Redes Neurais. **Tecgraf, PUC - Rio**. Disponível em: <<http://www.tecgraf.puc-rio.br/~mgattass/RedeNeural/redeneural.html>>. Acesso em: 10 nov. 2010.

[12] BALLONE, G. Neurônios e Neurotransmissores. **Psiquweb**, 2008. Disponível em: <<http://www.psiqweb.med.br/site/DefaultLimpo.aspx?area=NO/LerNoticia&idNoticia=290>>. Acesso em: 02 nov. 2010.

[13] ARTIFICIAL Neuron Model. **wikimedia**, 2005. Disponível em: <<http://commons.wikimedia.org/wiki/File:ArtificialNeuronModel.png>>. Acesso em: 09 out. 2010.

[14] VELLASCO, M. M. B. R. **Redes Neurais**. ICA: Núcleo de Pesquisa em Inteligência, PUC - Rio. Rio de Janeiro. 2009.

[15] MENDES, Q.; OLIVEIRA, M. F. D. S. Tutorial de redes neurais, aplicações em bioinformática. **Laboratório Nacional de Computação Científica**. Disponível em: <http://www.lncc.br/~labinfo/tutorialRN/frm3_modeloPerceptron.htm>. Acesso em: 06 nov. 2010.

[16] VALENÇA, M. J. S. **Fundamentos das Redes Neurais**: exemplos em Java. 2ª. ed. Olinda: Livro Rápido, 2009.

[17] COELHO, D. S.; SANTOS, A. P.; COSTA, C. A. D. Podemos prever a taxa de câmbio brasileira? Evidência empírica utilizando inteligência computacional e modelos econométricos. **Scientific Electronic Library Online**, 2008. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2008000300016#f1>. Acesso em: 10 nov. 2010.

[18] KIMURA, F. et al. **Handwritten Numeral Recognition using Autoassociative Neural Networks**. Faculty of Engineering, Mie University. Tsu, Japan. 1998.

[19] CAVALCANTI, D. C.; PEREIRA, C. S.; CARVALHO, E. C. B. **Auto-Associative Neural Network and Eigenbands Fusion for Frontal Face Verification**. Center of Informatics, Federal University of Pernambuco. Recife.

[20] FREITAS, E. D. G. D.; ALEXANDRIA, A. R. D. **Implementação de um Sistema de Reconhecimento Facial Utilizando Redes Neurais Artificiais.** Instituição Federal de Educação Ciências e Tecnologia do Ceará. Iguatu, CE.

[21] ROLIN, A. L. **UM SISTEMA DE IDENTIFICAÇÃO AUTOMÁTICA DE FACES PARA AMBIENTES VIRTUAIS DE APRENDIZAGEM.** Universidade Federal da Paraíba. João Pessoa. 2009.

Apêndice A

Fundamentação Matemática

A Matriz de Covariância

Em um conjunto de dados com mais de duas dimensões, pode-se calcular mais de uma covariância. Por exemplo, a partir de um conjunto de dados de três dimensões (x, y, z) é possível calcular $cov(x, y)$, $cov(x, z)$ e $cov(y, z)$, sendo $cov(x, y)$ a covariância entre x e y . Para um conjunto de dados de n dimensões é possível calcular N valores de covariâncias diferentes, onde N é calculado por:

$$N = \frac{n!}{(n-2)! \cdot 2} \quad (15)$$

Usualmente todos os valores de covariâncias possíveis são calculados entre todas as n dimensões e inseridos em uma matriz, formando assim a matriz de covariância, definida por:

$$C^{n \times n} = (C_i, C_j) = cov(Dim_i, Dim_j) \quad (16)$$

onde $C^{n \times n}$ é uma matriz n linhas e n colunas, e Dim_x é a x -ésima dimensão, e cada entrada da matriz é o resultado do cálculo covariância entre as dimensões separadas.

A seguir é mostrada uma matriz de covariância para um conjunto de dados imaginários de três dimensões, usando as dimensões usuais x, y, z :

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{pmatrix} \quad (17)$$

A diagonal principal desta matriz é formada pelos valores das covariâncias entre uma das dimensões e ela mesma, onde esse valor é a variância para essa dimensão [4].

Autovetores e autovalores

Em uma transformação especial da forma $T: V \rightarrow W$

$$T(v) = \lambda v \quad (18)$$

onde, λ é o autovalor (escalar) e v é o autovetor se $v \geq 0$. Como toda transformação linear pode ser escrita pela multiplicação de uma matriz por um vetor, então:

$$T(v) = Av \quad (19)$$

Igualando as equações (18) e (19) obtem-se o sistema homogêneo:

$$(A - \lambda I)v = 0 \quad (20)$$

onde, A é $n \times n$, $v = 0$ é sempre solução trivial. Os vetores $v \geq 0$ para os quais existe um λ que resolve a equação (20) são chamados de autovetores da matriz A e os valores de λ , que juntamente com v resolvem a equação são chamados de autovalores da matriz A associados aos respectivos autovetores.

Para que a equação (20) tenha solução além da trivial é necessário que o determinante da matriz dos coeficientes seja zero, ou seja:

$$\det(A - \lambda I) = 0 \quad (21)$$

o que resulta em um polinômio de grau n em λ , conhecido como polinômio característico. As raízes do polinômio característico são os autovalores da matriz A . Para se encontrar os autovetores basta substituir o valor do autovalor na equação original e encontrar o autovetor. O autovalor será, então, associado ao autovetor encontrado.

O autovetor encontrado forma uma base para o espaço de solução da equação (20), dado o respectivo autovalor. Logo, qualquer múltiplo do autovetor também é um autovetor [4].

Apêndice B

Telas do Sistema Proposto

Redes Neurais

Figura 13. Tela de configuração de estrutura da Rede Neural



Estrutura da Rede Neural | Propriedades do Treinamento | Saídas do Treinamento | Teste ORL | Gerador Exemplos ORL | PCA ORL

Números de Neurônios:

Neurônios de Entrada: 644

Camadas Escondidas: 1

Neurônios Escondidos: 50

Neurônios de Saída: 40

Tipo da Rede Neural: MLP

Tipo de dados do arquivo de entrada: Dados

Arquivo de Entrada: entradasImagens.csv [Selecio...]

Arquivo de Saída: saida.txt [Selecio...]


Função de Ativação: SigmoideLogistica

Exemplos:

☐ Do Arquivo

☒ Do Gerador ORL

Figura 14. Tela de configuração de propriedades do treinamento da Rede Neural



Estrutura da Rede Neural | **Propriedades do Treinamento** | Saídas do Treinamento | Teste ORL | Gerador Exemplos ORL | PCA ORL

Dados de Treinamento:

Taxa de Aprendizagem: 0.2

Momento: 0.4

Num. Ciclos de Validação: 100

Num Ciclos Máximo: 2000

Exemplos Treinamento(%): 50

Exemplos Validação(%): 25

Exemplos Teste(%): 25

Número de Redes Neurais:

☒ Única

☐ Multiplas

Número de Redes: 40

Inicializar Rede(s) Neuraui(s) **Treinar Rede** **Limpar Rede Nural**

PCA

Figura 15. Tela de parâmetros de teste do PCA

The screenshot shows a software interface with a tabbed menu at the top. The 'PCA ORL' tab is selected. The interface contains the following elements:

- Directorio:** A text field containing the path 'D:/Meus Documentos/NetBeansProjects/MlpAutoassociativa/imagensPCA'.
- Arquivo de Entrada:** A section with a text field 'URL da imagem a ser testada!' and a 'Selecionar' button.
- Numero de Eingenfaces:** A text field containing the value '10'.
- Usar PCA:** A section with two radio buttons: 'Usando PCA' (which is selected) and 'Usando Redes Neurais'.
- Testar PCA:** A large button at the bottom left.

Gerador Imagens ORL

Figura 16. Tela do gerador de imagens aleatórias para base de imagens ORL

The screenshot shows a software interface with a tabbed menu at the top. The 'Gerador Exemplos ORL' tab is selected. The interface contains the following elements:

- Quantidade de Exemplos para 40 pessoas:** A section with three input fields: 'Treino:' with value '5', 'Validação:' with value '2', and 'Teste:' with value '3'.
- Resultados dos Testes:** A section with three output fields: 'Quantidade de Erros:' with value '7.0', 'Quantidade de Acertos:' with value '113.0', and 'Acertos / exemplo de teste:' with value '0.941666...'.
- String Base para URL de imagens:** A text field containing the path 'D:/Meus Documentos/NetBeansProjects/MlpAutoassociativa/att_faces_23x28/s#/@.jpg'.
- Instructions:** A line of text stating 'Onde # numero da pessoa e @ opção de imagem dessa pessoa.'
- Carregar EXEMPLOS:** A large button at the bottom.