



UNILASALLE



CENTRO UNIVERSITÁRIO LA SALLE
Curso de Bacharelado em Ciência da Computação

EVERTON JOHNNY DA SILVA

**UM ESTUDO DE ALGORITMOS DE APRENDIZAGEM
APLICADOS À MORFOLOGIA DA LÍNGUA PORTUGUESA**

CANOAS, 2008

EVERTON JOHNNY DA SILVA

**UM ESTUDO DE ALGORITMOS DE APRENDIZAGEM
APLICADOS À MORFOLOGIA DA LÍNGUA PORTUGUESA**

Trabalho de conclusão apresentado para a banca examinadora do curso de Ciência da Computação do Centro Universitário La Salle, com exigência parcial para a obtenção do grau de Bacharel em Ciência da Computação, sob orientação da Prof^a. Dr^a. Alessandra Dahmer e co-orientação do Prof. Me. Celso A. N. da Conceição.

CANOAS, 2008

TERMO DE APROVAÇÃO

EVERTON JOHNNY DA SILVA

UM ESTUDO DE ALGORITMOS DE APRENDIZAGEM APLICADOS À MORFOLOGIA DA LÍNGUA PORTUGUESA

Trabalho de conclusão aprovado como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Curso de Ciência da Computação do Centro Universitário La Salle - UNILASALLE, pela seguinte banca examinadora:

Prof^a. Me. Ana Margô Mantovani
Unilasalle

Prof^a. Me. Patrícia B. M. Vianna
Unilasalle

Canoas, 07 de julho de 2008.

DEDICATÓRIA

Dedico este trabalho de conclusão aos meus pais, que são minhas bases sólidas de apoio, e que nos momentos mais difíceis dessa longa jornada, sempre dedicaram o seu amor e incentivo. Em especial a minha noiva Fernanda, por ser minha fonte inspiradora e estar sempre firme ao meu lado. Aos grandes amigos que compartilham comigo esta grande lição chamada Vida. Aos familiares e colegas que compartilharam de muitos dias na execução de relatórios, artigos e seminários. Enfim a todas as pessoas que contribuíram e ajudaram para a conclusão dessa caminhada.

AGRADECIMENTOS

Agradeço a professora e orientadora Alessandra Dahmer pela dedicação e apoio imprescindível. Ao professor Celso pela ajuda e atenção. Ambos foram essenciais e muito me ajudaram na elaboração, desenvolvimento e conclusão deste trabalho. E claro não poderia deixar de agradecer ao professor Lincoln que me deu esta oportunidade de pesquisa e também quem exerceu, digamos assim, o 'calor' necessário para que este trabalho acontecesse.

Aos professores Javier García, Simão Toscani, Gaspare Bruno, Tasso Faria, Marcos Barreto, Débora Nice, Mozart Siqueira e Patrícia Kayser que compartilharam de seu tempo e conhecimento comigo durante estes anos de graduação, meu muito obrigado.

Também gostaria de registrar aqui os meus grandes amigos que também contribuíram nessa caminhada, Thiago Oliveira, Nelson Sonttag, Tonismar Bernardo, Sandro Souza, Diogo Elmur, Wanderson, Rafael Garbin e a todos que ofereceram seu apoio contribuindo em favor desse trabalho.

RESUMO

Há tempos a idéia de um modelo computacional que simule certas condições que acontecem em um cérebro humano, entre elas a capacidade de aprender, gera duas reações: a primeira, que é a negação desta possibilidade, que por trás esconde um gigantesco temor em relação ao uso incorreto deste conhecimento, e a segunda reação, onde muitos ficam admirados antevendo diversas aplicações benéficas no uso deste modelo. Esta monografia apresenta uma nova alternativa para esta capacidade inteligente. Desta forma, é importante avaliar o comportamento desse modelo computacional frente às características da morfologia da língua portuguesa e para tanto foi usado o caso da formação do plural das palavras. Esta monografia também visa contribuir de forma efetiva com os anseios da comunidade científica no que diz respeito a qual, ou quais, modelos computacionais, com ou sem supervisão humana, oferecem a melhor solução, tendo como objeto de conhecimento a língua portuguesa.

PALAVRAS-CHAVE: morfologia, plural, redes neurais, algoritmos de aprendizagem

ABSTRACT

There are times the idea of a computational model that simulate certain conditions that occur in a human brain, including the ability to learn, generates two reactions: first, which is the negation of this possibility, which hides behind a giant fear in relation to misuse of this knowledge, and the second reaction, where many are surprised seeing beneficial in various applications using this model. This paper presents a new alternative for this intelligent capability. Thus it is important to analyse the behaviour of that computer model to the characteristics of the front morphology of the portuguese language and was used for both the case of training the plural of words. The monograph also aims to contribute effectively to the wishes of the scientific community with regard to which, or what, computational models with or without human supervision, offer the best solution, with the aim of understanding the portuguese language.

KEYWORDS: morphology, plural, neural networks, the learning algorithms

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
RNA	Rede Neural Artificial
MLP	Multilayer Perceptron
RAD	Rapid Application Development
LMCPC	Léxico Multifuncional Computadorizado do Português Contemporâneo

LISTA DE FIGURAS

Figura 1 -	O nodo Z separa X de Y.....	16
Figura 2 -	Uma rede de Bayes simples.....	16
Figura 3 -	Modelo de McCulloch e Pitts (1943).....	18
Figura 4 -	Modelo de solução adotada.....	37
Figura 5 -	Grafo arquitetural da MLP.....	40
Figura 6 -	Casos de uso: funcionalidades providas para usuário Administrador.....	44
Figura 7 -	Casos de uso: funcionalidades providas para usuário Professor.....	44
Figura 8 -	Tela de configuração da rede neural.....	50
Figura 9 -	Tela do conjunto de entrada (singular x plural).....	51
Figura 10-	Tela de treinamento da rede MLP.....	52
Figura 11-	Tela de log do treinamento da rede MLP.....	53
Figura 12-	Tela de teste da rede MLP.....	54
Figura 13-	Relação entre Treinamento X Aprendizado.....	57
Figura 14-	Tela de simulação: apresentação de erro.....	59
Figura 15-	Tela de simulação: apresentação de acerto.....	59
Figura 16-	Log de aprendizado para quatro regras.....	60

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Relevância do trabalho	11
1.2 Objetivos do trabalho.....	12
1.3 Estrutura do trabalho	13
2 INTELIGÊNCIA ARTIFICIAL	14
2.1 Redes Bayesianas.....	15
2.2 Redes Neurais Artificiais	17
2.2.1 Algoritmos de aprendizagem.....	19
2.2.2 Redes Multilayer Perceptron	20
2.3 Considerações sobre o capítulo	22
3 MORFOLOGIA	24
3.1 Estrutura das Palavras.....	24
3.2 Processos de Formação das Palavras	24
3.3 Flexões dos Substantivos	26
3.3.1 Flexão dos substantivos quanto ao gênero.....	26
3.3.2 Flexão dos substantivos quanto ao número	27
3.4 Flexões dos Adjetivos.....	30
3.4.1 Flexão dos adjetivos quanto ao gênero.....	31
3.4.2 Flexão dos adjetivos quanto ao número.....	31
3.5 Considerações sobre o capítulo	31
4 MODELO PROPOSTO	32
4.1 Metodologia	32
4.1.1 Projeto de Redes Neurais	33
4.1.2 Coleta e Seleção de Dados.....	33
4.1.3 Configuração da Rede.....	34
4.1.4 Treinamento	35
4.1.5 Teste	36

4.1.6 Integração	36
4.2 Solução adotada.....	37
4.2.1 Análise Léxico-Morfológica.....	37
4.2.2 Estudo da Palavra	38
4.2.3 Arquitetura da RNA	39
4.3 Considerações sobre o capítulo	42
5 IMPLEMENTAÇÃO.....	43
5.1 Modelagem das funcionalidades	43
5.1.1 Atores	44
5.1.2 Casos de uso	45
5.2 Desenvolvimento do Protótipo	47
5.2.1 Ferramentas utilizadas	47
5.2.2 Implementação do banco de dados	47
5.2.3 Implementação do aplicativo	49
5.3 Considerações sobre o capítulo	55
6 ANÁLISE DOS RESULTADOS	56
6.1 Treinamento X Aprendizado.....	56
6.2 Simulação de uso do Plural RNA.....	57
6.3 Simulação para todas as regras do plural	60
6.4 Considerações sobre o capítulo	63
7 TRABALHOS RELACIONADOS	64
7.1 NETtalk.....	64
7.2 Parallel Distributed Processing	65
7.3 Distributed Representations, Simple Recurrent Network, and Grammatical Structure	66
8 CONCLUSÃO	68
8.1 Limitações.....	69
8.2 Trabalhos futuros.....	69
REFÊRENCIAS BIBLIOGRÁFICAS	70
APÊNDICE	73

1 INTRODUÇÃO

Um modelo computacional que simule o cérebro humano, que tenha a capacidade de aprender, é estudado já algum tempo. Dentre as muitas coisas para aprendizado, está o entendimento de linguagem natural, seja ele no reconhecimento de fala ou na leitura de textos. Muitas pesquisas nesta área revelam que ainda não é possível manter um domínio pleno sobre esta arte, mas está cada vez mais próxima a possibilidade de uma máquina mostrar consciência dos sentidos de uma linguagem natural.

O fator motivador para a investigação nesta área é decorrente da necessidade de sistemas/produtos que sejam flexíveis quanto à usabilidade, independentemente da experiência do usuário com o computador. Portanto, este poderia usufruir desta tecnologia de maneira tão simples quanto ao uso de um eletrodoméstico (reservado as devidas proporções de cada caso e sua aplicabilidade).

O objetivo deste estudo é apresentar os algoritmos de aprendizagem existentes e quais os comportamentos e desempenhos quando aplicados na morfologia da língua portuguesa.

A flexão do substantivo em número, o plural, não é um processo fácil para uma criança, mas a partir do momento que ela aprende as “regras de formação do plural” fica automático, tornando-se um processo inconsciente. A justificativa para este trabalho está em avaliar os algoritmos de aprendizagem com capacidade de identificar e/ou reconhecer estes padrões de formação dos plurais das palavras na língua portuguesa, aprendendo a aplicá-los nas diversas palavras do nosso idioma.

1.1 Relevância do trabalho

A relevância desta pesquisa está na oportunidade de desenvolver novos métodos que possibilitem utilizar de forma mais direta e clara os modelos gerados a

partir de Redes Neurais Artificiais. O desenvolvimento de procedimentos que sejam capazes de aplicar de maneira efetiva modelos elaborados com Redes Neurais Artificiais facilitaria, aos planejadores e tomadores de decisão, a condução de análises sem exigir o *software* onde foi treinada a rede e, por conseguinte dos arquivos da rede treinada.

Esses fatos alentam a pesquisa e o desenvolvimento de novos métodos que possam ser utilizados no estudo do conexionismo. Além disso, embora a aplicação do presente projeto seja específica para avaliação do uso na morfologia da língua portuguesa, alguns métodos a serem desenvolvidos poderão ser utilizados em outras áreas do conhecimento que empregam Inteligência Artificial.

1.2 Objetivos do trabalho

O objetivo geral deste trabalho foi estudar os modelos computacionais das redes neurais supervisionadas, não supervisionadas e redes bayesianas e sua adequação para serem aplicados à língua portuguesa, sendo objeto de teste formar o plural das palavras.

E os objetivos específicos deste trabalho foram:

- Avaliar o uso da técnica de aprendizagem supervisionada, na qual o estudo do algoritmo denominado “backpropagation”, aplicando algumas regras da morfologia da língua portuguesa, executando um treinamento da rede neural, antes de tentar resolver o caso;
- Avaliar as possibilidades de uso da técnica de aprendizagem não supervisionada, na qual o estudo de Mapas Auto Organizáveis, denominado “algoritmo de Kohonen”, onde a rede neural aprende sozinha;
- Avaliar as possibilidades de uso de Redes Bayesianas, que são redes de conhecimento, para através de probabilidades gerarem palavras pluralizadas;
- Escolher um dos modelos estudados e aplicá-lo para tentar gerar o plural das palavras;
- E avaliar o modelo escolhido através de um protótipo.

1.3 Estrutura do trabalho

A estrutura desta monografia dar-se-á através da apresentação dos algoritmos de aprendizagem e das regras gramaticais envolvidas na morfologia da língua portuguesa.

O segundo capítulo detalhará as Redes Bayesianas e Redes Neurais Artificiais, nas quais serão abordados seus algoritmos de aprendizagem, as características de cada modelo e aspectos como arquitetura e funcionamento.

No terceiro capítulo, serão apresentados os conceitos sobre a Morfologia da Língua Portuguesa, onde são apresentadas definições sobre as palavras como estrutura, os processos de formação e principalmente suas flexões. Também serão descritas diferentes resultados para uma mesma regra do plural, pois sabe-se que esta variação, ocorre entre os gramáticos da Língua Portuguesa.

Para o quarto capítulo será descrito o modelo proposto, onde são detalhados os pontos determinantes para a solução e definição dos dados a serem avaliados e que definem os resultados deste trabalho.

O quinto capítulo apresentará a descrição da implementação de um software que usa o modelo de solução proposto e todas as soluções adotadas para avaliar esse modelo.

O sexto capítulo apresentará a metodologia e os resultados do trabalho, onde estarão os gráficos elaborados a partir dos dados retirados das simulações, bem como a análise dos valores coletados.

No penúltimo capítulo, serão descritos os trabalhos relacionados a esta monografia, onde serão relatadas as limitações do modelo proposto, mas também destacará as potencialidades referentes a cada estudo.

Por fim, no sétimo capítulo, será descrita a conclusão desta monografia, onde serão apresentados os resultados do trabalho e sugestão de trabalhos futuros.

2 INTELIGÊNCIA ARTIFICIAL

Segundo o dicionário da língua portuguesa Aurélio, a palavra “inteligência” é definida como a faculdade de entender, pensar, raciocinar e interpretar, o que caracteriza a capacidade que permite ao ser humano escolher entre uma coisa ou outra. Também define a palavra “artificial” como algo produzido pelo homem e não por causas naturais.

A Inteligência Artificial (IA) é uma parte da ciência da computação voltada para o desenvolvimento de sistemas de computadores inteligentes, isto é, sistemas que exibem características, as quais são comparadas com a inteligência no comportamento humano – por exemplo: compreensão da linguagem, aprendizado, raciocínio, resolução de problemas, etc (FERNANDES, 2005). Seu objetivo principal está separado em dois: o teórico: a criação de teorias e modelos para a capacidade cognitiva; e prático: a implementação de sistemas computacionais baseados nos modelos teóricos (BITTENCOURT, 2006).

Fernandes (2005 apud SIMON, 1998) cita duas importantes abordagens para IA:

- Abordagem Cognitiva: chamada por alguns autores como “Simbolista”, esta abordagem dá destaque à forma como o ser humano raciocina, ou seja, aos processos cognitivos. Baseados em aspectos psicológicos e em processos algorítmicos procuram encontrar uma explicação para comportamentos inteligentes. Os pioneiros desta corrente foram John McCarthy, Marvin Minsky, Newell e Simon;
- Abordagem Conexionista: ou também denominada “Biológica”, esta abordagem dá ênfase no modelo de funcionamento do cérebro, dos neurônios e das conexões neurais. Os pioneiros desta abordagem foram McCulloch, Pitts, Helder, Rosenblatt e Widrow.

Este trabalho contempla uma abordagem conexionista. A seguir serão descritas duas áreas desta abordagem que foram estudadas.

2.1 Redes Bayesianas

As redes Bayesianas são tipos específicos de redes de conhecimento. A idéia principal é que, para descrever um modelo do mundo real, não é necessário usar uma enorme tabela de probabilidades conjuntas, na qual, são listadas as probabilidades de todas as combinações possíveis de eventos. A maioria dos eventos é condicionalmente independente da maioria dos outros, portanto suas interações não precisam ser consideradas. Em vez disso, usa-se uma representação mais local, que descreve agrupamentos de eventos que interagem (HRUSCHKA, 1997).

As redes Bayesianas foram desenvolvidas nos anos 70 com objetivo de modelar processamento distribuído na compreensão da leitura, onde as expectativas semânticas e evidências perceptivas deveriam ser combinadas para formar uma interpretação coerente. A habilidade para coordenar inferências bidirecionais preencheu uma lacuna na tecnologia de sistemas especialistas no início dos anos 80, e as redes Bayesianas têm emergido como um esquema de representação genérico para conhecimento incerto (PEARL, 1988).

A abordagem de redes Bayesianas é usada para modelar conhecimento porque oferece uma estrutura unificada e intuitiva para representar modelos de seus dados. O seu caráter unificado torna possível comparar diferentes hipóteses sobre os dados e a natureza intuitiva, do formalismo gráfico, torna Redes Bayesianas um dos melhores métodos analíticos disponível para tomada de decisão (KOEHLER, 2004). Um primeiro trabalho há sugerir a representação de conhecimento incerto baseado na estatística foi o do reverendo Bayes em 1763 (SCHREIBER, 2003).

Uma rede bayesiana é um Grafo Direcionado Acíclico (DAG) onde os nós representam as variáveis (de interesse) de um domínio e os arcos representam a dependência condicional ou informativa entre as variáveis. A força da dependência é representada por probabilidades condicionais que são associadas a cada cluster de nós pais-filhos na rede (PEARL, 1997). Para se representar que um fato X é independente de um outro fato Y, caso um terceiro fato Z seja conhecido, todos os caminhos do grafo que ligam X e Y devem passar por Z (Figura 1):

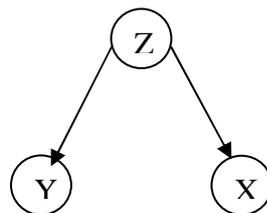


Figura 1 - o nodo Z separa X de Y

Assim, se a variável Z é conhecida, então X e Y são independentes para um dado Z. Isto possibilita a redução dos parâmetros numéricos das condicionadas que fazem parte da distribuição em questão.

O conhecimento *causa/efeito* de um domínio permite estruturar suposições de independência condicional.

Uma rede de Bayes é um grafo dirigido acíclico tal que:

1. Cada nodo representa uma variável aleatória
2. Um arco dirigido a partir do nodo J até o nodo F significa que J tem influência direta sobre F.
3. Para cada nodo existe uma tabela de probabilidade condicional do nodo, considerado os seus ancestrais imediatos.

Suponha que os grafos com que se esteja trabalhando sejam fracamente conectados. Isto significa que ao remover qualquer arco do grafo este se divide em dois. Por exemplo, suponha as seguintes proposições:

E: Enxaqueca

F: Fotofobia

D: Dor de cabeça

Além disso, suponhamos que E é uma causa primária e F e D são efeitos de E. Esta informação corresponde à informação estruturada do domínio. A rede de Bayes resultante é apresentada na Figura 2:

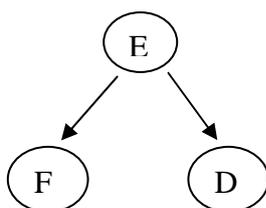


Figura 2 - Uma rede de Bayes simples

Para completar a rede de Bayes especifica-se os seguintes parâmetros:

$$\begin{aligned} P(E) &= 0.04 \\ P(D|E) &= 0.9 \\ P(D|\neg E) &= 0.02 \\ P(F|E) &= 0.7 \\ P(F|\neg E) &= 0.05 \end{aligned}$$

As probabilidades que faltam (por ex. $P(\neg D|E)$) se calculam a partir do axioma básico de probabilidade $P(\text{verdadeiro}) = 1 - P(\text{falso}) = 1$.

A rede de Bayes torna explícita a informação estrutural do domínio. Esta informação estrutural traduz-se em certas suposições de *independência condicional*. A suposição de independência condicional básica é que as evidências associadas a uma causa comum são independentes se a causa comum estabeleceu-se. No exemplo isto significa que:

$$P(D|E,F) = P(D|E)$$

Assim, $P(D|\neg E,F) = P(D|\neg E) = 0.02$

A partir da rede de Bayes inicial, é possível calcular as probabilidades *a priori*. Utilizando condicionamento:

$$\begin{aligned} P(D) &= P(D|E) * P(E) + P(D|\neg E) * P(\neg E) = 0.056 \\ P(F) &= P(F|E) * P(E) + P(F|\neg E) * P(\neg E) = 0.076 \end{aligned}$$

Uma rede de Bayes permite realizar inferências de diversos tipos. Por exemplo, se o objetivo é fazer um diagnóstico, o que interessa é avaliar a probabilidade de que uma variável aleatória seja verdadeira dada a evidência disponível. Pode-se querer avaliar a probabilidade de que ocorra enxaqueca dado que se observa fotofobia. Isto corresponde a calcular $P(E|F)$. Se fotofobia é evidência positiva para enxaqueca, então deveria verificar-se que a probabilidade condicional $P(E|F)$ seja maior que a probabilidade *a priori* $P(E)$.

2.2 Redes Neurais Artificiais

O cérebro humano é considerado o mais fascinante processador baseado em carbono existente, sendo composto por aproximadamente 100 bilhões neurônios. Todas as funções e movimentos do organismo estão relacionados ao funcionamento destas pequenas células. Os neurônios estão conectados uns aos outros através de

sinapses, e juntos formam uma grande rede, chamada Rede Neural. Esta grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informação (KOVACS, 2002).

Em 1943 McCulloch e Pitts criaram a representação e formalização matemática dos neurônios artificiais, o que fez surgir os primeiros modelos de redes neurais artificiais (Figura 3).

Segundo Medeiros (2003) a idéia básica é a partir do modelo biológico de um neurônio gerar o modelo matemático (artificial). Neste modelo, cada sinal positivo ou negativo que entra pelo sistema, é multiplicado por um número, ou peso, que indica a sua influência na saída. Caso a soma ponderada dos sinais exceda certo limite, é gerada uma resposta na saída. No modelo geral de neurônio (Figura 3) as entradas $W_i X_i$ são combinadas usando uma função f , para produzir um estado de ativação do neurônio (correspondente à frequência de descarga do neurônio biológico). As entradas chegam através dos dendritos e tem um peso atribuído pela sinapse. A função básica de um neurônio é somar as entradas e retornar uma saída, caso esse valor seja maior que o valor de soma limite “threshold” (BILOBROVEC, 2004).

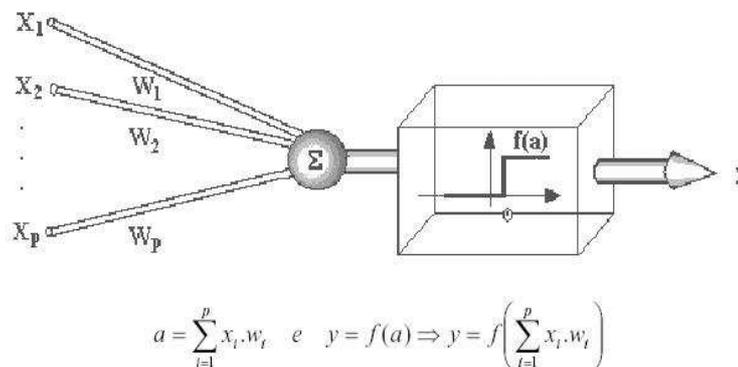


Figura 3 – Modelo de McCulloch e Pitts (1943)

Fonte: Bilobrovec, 2004.

Uma rede neural artificial consegue o comportamento inteligente porque tem a mesma característica de um neurônio biológico, o qual pode formar novas conexões com outros neurônios, o que acontece através dessas interações entre as unidades de processamento da rede (MEDEIROS, 2003).

2.2.1 Algoritmos de aprendizagem

Denomina-se algoritmo de aprendizado um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais que diferem entre si principalmente pelo modo como os pesos são modificados (FERNANDES, 2005).

O aprendizado pelo computador, chamado aprendizado conexionista, é em geral um processo gradual e iterado. O aprendizado é realizado utilizando-se um conjunto de dados de aprendizado disponível (base de exemplos). Cada iteração deste processo gradativo de adaptação dos pesos de uma rede neural é chamada de aprendizado (LUGER, 2004).

De todas as propriedades interessantes das Redes Neurais Artificiais, nenhuma “captura” tão bem uma característica humana como a habilidade de aprender.

Ao invés de especificar todos os detalhes de uma computação, tem-se a possibilidade de treinar uma rede para fazer esta computação. Isto significa que se pode tratar problemas onde regras apropriadas são muito difíceis de se conhecer a priori. O objetivo do treinamento de uma RNA é fazer com que a aplicação de um conjunto de entradas produza um conjunto de saídas desejado, ou no mínimo, um conjunto de saídas consistentes. Cada conjunto de entrada ou saída é chamado de vetor.

O treinamento é realizado pela aplicação seqüencial dos vetores de entradas (e em alguns casos também os de saída), enquanto os pesos da rede são ajustados de acordo com um procedimento de treinamento pré-determinado. Durante o treinamento, os pesos da rede gradualmente convergem para determinados valores, de maneira tal que a aplicação dos vetores de entrada produza as saídas necessárias. Os procedimentos de treinamento que levam as RNAs a aprender determinadas tarefas podem ser classificados em duas classes de treinamento:

- Supervisionado
- Não Supervisionado

O treinamento supervisionado necessita de um par de vetores composto do vetor de entrada e do vetor alvo que se deseja como saída. Juntos, estes vetores

são chamados de “par de treinamento” ou vetor de treinamento, sendo interessante ressaltar que geralmente a rede é treinada com vários vetores de treinamento.

O procedimento de treinamento funciona da seguinte forma: o vetor de entrada é aplicado. A saída da rede é calculada e comparada com o correspondente vetor alvo. O erro encontrado é então realimentado através da rede e os pesos são atualizados de acordo com um algoritmo determinado a fim de minimizar este erro. Este processo de treinamento é repetido até que o erro, para os vetores de treinamento, tenha alcançado níveis bem baixos.

O treinamento não supervisionado, por sua vez, não requer vetor alvo para as saídas, e obviamente, não faz comparações para determinar a resposta ideal. O conjunto de treinamento modifica os pesos da rede de forma a produzir saídas que sejam consistentes, isto é, que tanto a apresentação de um dos vetores de treinamento, como a apresentação de um vetor que é suficientemente similar, irão produzir o mesmo padrão nas saídas. O processo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores similares em classes. A aplicação de um vetor de uma determinada classe à entrada da rede irá produzir um vetor de saída específico, mas não existe maneira de se determinar, antes do treinamento, qual o padrão que será produzido na saída para um vetor de entrada de uma determinada classe. Desta forma, a saída de algumas RNAs deve ser transformada, em uma forma compreensível, após o processo de treinamento, o que é um simples problema de identificação das relações entrada/saída estabelecidas pela rede.

2.2.2 Redes Multilayer Perceptron

As arquiteturas do tipo perceptron de MLP (múltiplas camadas) constituem os modelos neurais artificiais mais utilizados e conhecidos atualmente. Tipicamente, esta arquitetura consiste de um conjunto de unidades sensoriais que formam uma camada de entrada, uma ou mais camadas intermediárias (ou escondidas) de unidades computacionais e uma camada de saída. Os sinais de entrada são propagados camada a camada pela rede em uma direção positiva, ou seja, da entrada para a saída. Esta arquitetura representa uma generalização do perceptron apresentado anteriormente.

As redes do tipo MLP têm sido utilizadas com sucesso para a solução de vários problemas envolvendo altos graus de não-linearidade. Seu treinamento é do tipo supervisionado e utiliza um algoritmo muito popular chamado retro-propagação do erro (error backpropagation). Este algoritmo é baseado numa regra de aprendizagem que “corrige” o erro durante o treinamento (HAYKIN, 1994).

Basicamente, o processo de retro-propagação do erro é constituído de duas fases: uma fase de propagação do sinal funcional (feedforward) e uma de retropropagação do erro (backpropagation). Na fase positiva, os vetores de dados são aplicados às unidades de entrada, e seu efeito se propaga pela rede, camada a camada.

Finalmente, um conjunto de saídas é produzido como resposta da rede. Durante a fase positiva, os pesos das conexões são mantidos fixos. Na retro-propagação do erro, por outro lado, os pesos são ajustados de acordo com uma regra de correção do erro.

Especificamente, a resposta da rede em um instante de tempo é subtraída da saída desejada (target) para produzir um sinal de erro. Este sinal de erro é propagado da saída para a entrada, camada a camada, originando o nome “retro-propagação do erro”. Os pesos são ajustados de forma que a “distância” entre a resposta da rede e a resposta desejada sejam reduzidas.

2.2.2.1 Algoritmo de Aprendizagem para uma rede MLP

Definida a arquitetura da rede temos a seguinte situação para o algoritmo backpropagation onde dado um conjunto de padrões de treinamento $P = \{(z_1, d_1), (z_2, d_2), \dots, (z_p, d_p)\}$, onde z_i é $(I \times 1)$, d_i é $(K \times 1)$ e $i = 1, 2, \dots, P$. Temos ainda a camada escondida, onde as saídas de seus neurônios são y e o é $(J \times 1)$ e o é $(K \times 1)$:

1. Inicialização da rede: Os parâmetros da rede são inicializados por quantidade de iterações (ciclos) e taxa de aprendizagem – η ;
2. Inicialização dos pesos: Os pesos W e V são inicializados randomicamente com valores pequenos (entre 0 e 1). O tamanho de W é $(K \times J)$ e V é $(J \times I)$;
3. Treinamento é inicializado: Um padrão de entrada p é apresentado à camada de entrada e as saídas das camadas da rede são computadas utilizando a função de ativação sigmóide:

$$y_j = f(v_j^t z), \text{ para } j = 1, 2, \dots, J$$

Onde v_j é a j -ésima linha de V e y é a saída dos neurônios das camadas escondidas.

$$o_k = f(v_k^t y), \text{ para } k = 1, 2, \dots, K$$

Onde v_k é a k -ésima linha de W e o é a saída da rede.

4. Erro é calculado: O erro da rede é calculado para cada neurônio da camada de saída.

$$E = 1/2 (d_k - o_k)^2 + E, \text{ para } k = 1, 2, \dots, K$$

Onde d_k é o valor esperado e o é a saída calculada para o neurônio k .

5. Retro propagação do erro: Sinal de erro é calculado para camada de saída $\delta_o(K \times 1)$ e para as camadas escondidas $\delta_y(J \times 1)$. Para camada de saída, temos:

$$\delta_{ok} = 1/2 (d_k - o_k)(1 - o_k^2), \text{ para } k = 1, 2, \dots, K$$

E para as camadas escondidas, temos:

$$\delta_{yj} = 1/2 (1 - y_j^2) \sum \delta_{ok} w_{kj}, \text{ para } j = 1, 2, \dots, J$$

Com os sinais de erro calculados devemos reajustar os pesos da rede.

6. Ajustes de pesos: O ajuste dos pesos é feito em dois passos. O primeiro passo é ajustar os pesos da camada de saída:

$$w_{kj} = w_{kj} + (n \delta_{ok} y_j), \text{ para } k = 1, 2, \dots, K \text{ e } j = 1, 2, \dots, J$$

Com os pesos da camada de saída calculados, reajustamos os pesos das camadas anteriores, da seguinte maneira:

$$v_{ji} = v_{ji} + (\delta_{yj} z_i), \text{ para } j = 1, 2, \dots, J \text{ e } i = 1, 2, \dots, I$$

7. Verifica conjunto de padrões: Se existir algum padrão, retorna ao passo 3. Caso contrário passa-se ao passo 8.
8. Finaliza treinamento: Se o erro $E < E_{max}$, finalize o treinamento. Caso contrário inicia-se um novo ciclo de treinamento retornando ao passo 3.

2.3 Considerações sobre o capítulo

Uma palavra que define bem redes Bayesianas é a “probabilidade”, e esta definição, mostra que deve ser executadas previsões de algum acontecimento, e por isso logo se descartou a hipótese do uso de redes Bayesianas, pois no contexto

deste trabalho a incerteza não faz parte do contexto morfológico das classes de palavras (substantivos, adjetivos, artigos, numeral, pronome, verbo, advérbio, etc.). A classificação de palavras quanto a sua classe na língua portuguesa é uma tarefa exata, onde não existem porcentagens de chance de ser um verbo ou substantivo, por exemplo (TUFANO, 1995).

Já o aprendizado não-supervisionado das redes neurais, os pesos da rede são modificados em função de critérios internos, tais como, por exemplo, a repetição de padrões de ativação em paralelo de vários neurônios. O comportamento resultante deste tipo de aprendizado é usualmente comparado com técnicas de análise de dados empregadas na estatística (OSÓRIO, 2000). Assim, por ser uma técnica que se utiliza de estatística, o que não acontece na flexão em número das palavras, não foi adotado este modelo na elaboração deste trabalho.

Finalmente, no aprendizado supervisionado o usuário “ensina” a rede neural. Sendo assim, a rede deve ser capaz de medir a diferença entre seu comportamento atual e o comportamento de referência, e então corrigir os pesos de maneira a reduzir o desvio de comportamento em relação aos exemplos de referência (OSÓRIO, 1991). Este modelo de aprendizado foi escolhido neste trabalho, pois existe uma grande discussão entre gramáticos da língua portuguesa sobre algumas regras do plural e em determinados momentos, como será vista no capítulo sobre a implementação, se faz necessário a intervenção de um especialista.

E no que diz respeito aos algoritmos de treinamento, existe uma grande variedade, tanto para o treinamento supervisionado, como para o não supervisionado. Entre esses, um dos mais difundidos com certeza é o algoritmo utilizado neste trabalho, conhecido por algoritmo backpropagation (retro-propagação).

No capítulo **Erro! Fonte de referência não encontrada.**, serão apresentadas os conceitos sobre morfologia e as regras gramaticais utilizadas para a geração do plural das palavras. O estudo desses conceitos foi fundamental para a realização deste trabalho.

3 MORFOLOGIA

O escopo deste trabalho está focado na morfologia. Conforme Tufano (1995) e Cegalla (1998), na língua portuguesa, morfologia é a parte da gramática que estuda a estrutura, a formação, a flexão e a classificação das palavras.

3.1 Estrutura das Palavras

As palavras são formadas de unidades ou elementos mórficos e/ou estruturais das palavras, são eles CEGALLA (1998):

- a) raiz, radical, tema: elementos básicos e significativos;
- b) afixos (prefixos, sufixos), desinência, vogal temática: elementos modificadores da significação dos primeiros;
- c) vogal de ligação, consoante de ligação: elementos de ligação ou eufônicos.

3.2 Processos de Formação das Palavras

As palavras, na língua portuguesa, formam-se principalmente através de dois processos: composição e derivação. Tendo ainda hibridismo e onomatopéia como processos de formação de palavras (CEGALLA, 1998).

Composição é o processo segundo o qual a palavra nova é constituída a partir de dois ou mais radicais. A composição pode ser:

- a) Por justaposição – Os radicais primitivos não apresentam alteração fonética na palavra composta. Exemplos:

papel-moeda, mãe-pátria, pontapé, couve-flor

b) Por aglutinação – Quanto ao ser constituída a nova palavra, os radicais primitivos sofrem fonética. Exemplos:

fidalgo (filho + de + algo)

embora (em + boa + hora)

aguardente (água + ardente)

Derivação é o processo segundo o qual se constituem palavras novas acrescentando-se afixo(s) à palavra primitiva. A derivação pode ser:

a) Prefixal – Acrescente-se prefixo à palavras primitiva. Exemplo:

dispor (dis + por)

benemérito (bene + mérito)

ingrata (in + grata)

infeliz (in + feliz)

b) Sufixal – Quando se acrescenta um sufixo ao radical: Exemplo:

barcaça (barc – aça)

batalha (bat + alha)

areal (are + al)

gostoso (gost + oso)

c) Prefixal e sufixal – Processo segundo o qual a palavra nova surge da adição de um prefixo e um sufixo ao radical. Exemplo:

indelicadamente = in + delicada + mente

Neste caso, se tirar um dos afixos a palavra restante existe na língua:

in – delicada ou delicada – mente.

d) Parassintética – Quando prefixo e sufixo são adicionados simultaneamente à palavra primitiva. Exemplo:

apedrejar = a + pedr + ejar

desalmado = des + alm + ado

Na derivação parassintética, não obtemos uma palavra do nosso idioma, caso tirarmos os afixos.

e) Regressiva – Segundo Ismal de Lima Coutinho (in Gramática Histórica): “Consiste a derivação regressiva na dedução de uma forma primitiva com base numa outra que se julga derivada”. A derivação regressiva pode ser:

1. Nominal – Exemplo: revoa (revoada), estranja (pop. Estrangeiro)
2. Verbal – Exemplo: o gasto (gastar), a compra (comprar), a reserva (reservar), o ganho (ganhar)

3.3 Flexões dos Substantivos

O substantivo é a palavra que usamos para designar seres, coisas, idéias (CEGALLA, 1998). Na língua portuguesa existem palavras variáveis e invariáveis. São variáveis: o substantivo, o adjetivo, o artigo, o pronome, o numeral e o verbo. Classificam-se entre as palavras invariáveis: o advérbio, a preposição, a conjunção e a interjeição.

Segundo Cegalla (1998) algumas observações são válidas quanto à classificação das palavras:

- a) A mesma palavra pode figurar em mais de uma classe:
O céu é azul. (azul, adjetivo).
O azul triste de seus olhos fascinava-me (azul, substantivo);
- b) Alguns advérbios admitem ‘as flexões de grau:
cedo, cedinho; agora, agorinha; muito, muitíssimo.
- c) Não poucos substantivos e pronomes e a maioria dos numerais são invariáveis:
pires, isto, alguém, dez, etc.

3.3.1 Flexão dos substantivos quanto ao gênero

Quanto ao gênero os substantivos flexionam-se em gênero masculino e gênero feminino, dividindo-se em biformes e uniformes. São biformes os substantivos que apresentam uma forma para cada gênero. E são uniformes os substantivos que apresentam a mesma forma no masculino e no feminino.

Segundo Tufano (1995) não se pode confundir o gênero com o sexo dos seres a que se referem os substantivos. Na análise gramatical, a palavra nuvem, por exemplo, é um substantivo feminino e a palavra sapato, um substantivo masculino, não havendo nesses casos, evidentemente, nenhuma alusão a características sexuais. A palavra criança é um substantivo feminino e tatu, um substantivo masculino, embora possam designar seres de ambos os sexos.

A flexão do substantivo quanto ao grau pode apresentar-se no grau aumentativo ou grau diminutivo. Exemplo: nariz – narigão – narizinho. A indicação de grau de um substantivo pode ser feita de duas maneiras:

a) Por meio de sufixos apropriados, como *inho*, *ão*, *ona* (processo sintético).

Exemplos: livrinho, paredão

b) Por meio de adjetivos (processo analítico).

Exemplos: livro pequeno, parede grande

Nem sempre, porém, o uso de sufixos aumentativos ou diminutivos está relacionado à idéia de tamanho. Muitas vezes eles dão ao substantivo um sentido afetivo, carinhoso ou, então, pejorativo, irônico.

Exemplos: Rodrigo é meu *amigão*!

Mas que *sujeitinho* covarde!

3.3.2 Flexão dos substantivos quanto ao número

A flexão do substantivo quanto ao número pode estar no singular ou no plural. Há, porém, alguns só são usados no singular ou no plural, como fé, caridade, parabéns, pêsames.

a) Plural dos substantivos simples:

1. Como regra geral, o plural é feito pelo acréscimo do *s* à forma do singular.

Exemplos: aluno – alunos

casa – casas

urubu – urubus

2. Os substantivos terminados em *al*, *el*, *ol* e *ul* fazem o plural trocando o *l* por *is*.

Exemplos: jornal – jornais

anzol – anzóis

pastel – pastéis

3. Os substantivos terminados em *il* fazem o plural de duas formas:

a. Quando são oxítonos, trocam o *l* por *s*. Exemplo: funil - funis

b. Quando são paroxítonos, trocam o *l* por *eis*. Exemplo: fóssil – fósseis

4. Os substantivos terminados em *r* e *z* fazem o plural pelo acréscimo de *es*

Exemplos: pastor - pastores
juiz – juízes
doutor – doutores

5. Os substantivos terminados em *n* fazem o plural pelo acréscimo de *es* ou *s* (a segunda forma é mais usada)

Exemplos: pólen – polens
hífen – hífens
abdômen – abdomens

6. Os substantivos monossílabos e os oxítonos terminados em *s* fazem o plural pelo acréscimo de *es*.

Exemplos: país – países
mês – meses
francês – franceses

Quando não são oxítonos, os substantivos terminados em *s* não mudam de forma no plural.

Exemplo: o tênis – os tênis

7. Os substantivos terminados em *x* não mudam de forma no plural.

Exemplo: o tórax – os tórax

8. Os substantivos terminados em *ão* podem formar o plural de três maneiras:

- a. Pelo acréscimo de *s*. Exemplo: irmão - irmãos
- b. Pela transformação do *ão* em *ães*. Exemplo: alemão - alemães
- c. Pela transformação do *ão* em *ões*. Exemplo: leão – leões

Os aumentativos fazem o plural pela transformação do *ão* em *ões* (TUFANO, 1995). Exemplo: casarão – casarões

Vários substantivos em *–ão* ainda não encontraram uma forma definitiva para o plural, se bem que a flexão *–ões*, por ser mais eufônica, se venha impondo vitoriosamente (CEGALL, 1998). Eis alguns exemplos:

aldeão	aldeões	aldeãos	
castelão	castelões	castelãos	
ermitão	ermitões	ermitãos	ermitães
faisão	faisões		faisães
hortelão	hortelões	hortelãos	
sultão	sultões	sultãos	sultães
verão	verões	verãos	
vilão	vilões	vilãos	

9. Quando o substantivo está no grau diminutivo e é formado com os sufixos – *zinho* ou –*zito*, perde o *s* do plural do substantivo primitivo.

Exemplos: animalzinho – animai(-s) + zinhos = animalzinhos
 cãozinho – cãe (-s) + zinhos = cãesinhos

b) Plural dos substantivos compostos:

1. Pluralizam-se os dois elementos, quando houver:

a) substantivo + substantivo:

abelha-mestra	abelhas-mestras
couve-flor	couves-flores

b) substantivo + adjetivo:

amor-perfeito	amores-perfeitos
capitão-mor	capitães-mores

c) adjetivo + substantivo:

boa-vida	boas-vidas
curta-metragem	curtas-metragens

d) numeral + substantivo:

terça-feira	terças-feiras
quinta-feira	quintas-feiras

2. Varia apenas o segundo elemento, quando houver:

a) elementos unidos sem hífen:

os pontapés, os girassóis, os vaivéns

b) verbo + substantivo:

os guarda-roupas, os guarda-louças, os beija-flores

c) elemento invariável + palavra variável:

as sempre-vidas, as ave-marias, os vice-reis

os alto-falantes, os auto-serviços, as auto-sugestões

d) palavras repetidas:

os quero-queros, os tico-ticos, os ruge-ruges

3. Varia apenas o primeiro elemento, quando houver:

a) substantivo + de + substantivo:

os pés-de-moleque, os pães-de-ló, as quedas-d'agua

os paus-d'arco, os sinais-da-cruz, os autos-de-fé

b) o segundo elemento limita ou determina o primeiro:

os pombos-correio, as canetas-tinteiro, os navios-escola

os peixes-boi, as frutas-pão, os cafés-concerto

4. Os dois elementos ficam invariáveis:

a) verbo + advérbio:

os bota-fora, os pisa-mansinho

b) verbo + substantivo plural:

o louva-a-deus, os louva-a-deus,

o diz-que-diz, os diz-que-diz,

o bem-te-vi, os bem-te-vis

o bem-me-quer, os bem-me-queres,

o João-ninguém, os Joões-ninguém

3.4 Flexões dos Adjetivos

Adjetivo é a palavra que modifica o substantivo atribuindo-lhe uma característica (TUFANO, 1995). Um mesmo substantivo pode ser modificado por mais de um adjetivo. Segundo Cegalla (1995) o adjetivo é uma palavra que modifica o substantivo, expressando condição, estado ou qualidade do ser.

Exemplo: Ele é um homem *elegante*.
Essa história é *engraçada*.

3.4.1 Flexão dos adjetivos quanto ao gênero

A flexão do adjetivo quanto ao gênero pode ser uniforme ou biforme. Uniforme quando possui a mesma forma no masculino e no feminino. Exemplo: menino gentil, menina gentil. Biforme quando possui uma forma para o masculino e outra para o feminino.

Exemplo: menino educado, menina educada.

3.4.2 Flexão dos adjetivos quanto ao número

A flexão do adjetivo quanto ao número segue as mesmas regras aplicadas ao substantivo. E nos adjetivos compostos, geralmente apenas o segundo elemento flexiona-se no plural.

Exemplos:

lugar <i>agradável</i>	lugares <i>agradáveis</i>
cão <i>feroz</i>	cães <i>ferozes</i>
mulher <i>gentil</i>	mulheres <i>gentis</i>
poema <i>herói-cômico</i>	poemas <i>herói-cômicos</i>

3.5 Considerações sobre o capítulo

Nesse capítulo foram apresentadas as regras gramáticas e questões referente à língua portuguesa. Além disso, também foram detalhadas todas as regras para flexão quanto ao gênero e número dos substantivos e adjetivos. Sendo mais importante para o presente trabalho a flexão quanto ao número (regras do plural).

No próximo capítulo, será descrito a metodologia de pesquisa adotada neste trabalho e o modelo de solução proposto.

4 MODELO PROPOSTO

A proposta desta monografia refere-se à análise de técnicas computacionais de inteligência artificial para resolver a questão da geração dos plurais e responder algumas perguntas em relação a capacidade de aprendizagem pelo computador. Baseado neste estudo foi elaborado um modelo, que foi avaliado através da implementação de um protótipo.

No capítulo 0, será demonstrada a metodologia utilizada para a análise, elaboração, coleta dos dados e parâmetros necessários, para as avaliações.

4.1 Metodologia

Este capítulo apresenta a metodologia empregada nessa monografia, sendo essa caracterizada como uma pesquisa de natureza aplicada, tendo uma abordagem quantitativa, com objetivos exploratórios e procedimentos comparativos.

Um método científico configura-se como “o conjunto das atividades sistemáticas e racionais, com maior segurança e economia, que permite alcançar o objetivo (conhecimentos válidos e verdadeiros), traçando o caminho a ser seguido, detectando erros e auxiliando as decisões do pesquisador” (MARCONI; LAKATOS, 2001). Com relação à natureza da pesquisa exploratória, ela gerou conhecimentos novos e úteis para o avanço da ciência e para aplicação prática, dirigidos à solução do problema de reconhecimento de linguagem natural.

Apos o estudo sobre as três alternativas propostas percebeu-se que neste trabalho, não se tem como executar previsões de algum acontecimento, e por isso logo se descartou a hipótese do uso de Redes Bayesianas, pois a incerteza não faz parte do contexto morfológico das classes de palavras (substantivos, adjetivos, artigos, numeral, pronome, verbo, advérbio, etc.) (MAGALHÃES, 1999). A

classificação de palavras quanto a sua classe na língua portuguesa é uma tarefa exata, onde não existem porcentagens de chance de ser um verbo ou substantivo.

A morfologia é a parte da gramática que estuda a forma e a estrutura das palavras (CEGALLA, 1998). A partir deste conceito percebe-se que se faz necessário um modelo computacional capaz de reconhecer padrões na formação das palavras e por este motivo à escolha deliberada por Redes Neurais Artificiais (RNA) supervisionadas. A seguir são apresentados os passos executados para montagem e definição de um modelo de RNA, conforme (LUDWIG JR; MONTGOMERY, 2007). Estes passos foram seguidos neste trabalho.

4.1.1 Projeto de Redes Neurais

Com uma boa arquitetura para a rede neural é possível evitar problemas inerentes à memorização (quando a rede neural decora um conjunto de dados), ao invés de aprender, gerando o problema de não ter resultados corretos quando a entrada é diferente do conjunto de dados de treinamento. Para garantir uma boa funcionalidade da rede neural foi necessária a execução de cinco fases distintas no projeto: coleta e seleção de dados; configuração da rede; treinamento; teste e integração. Estas fases são descritas a seguir.

4.1.2 Coleta e Seleção de Dados

A coleta criteriosa dos dados relativos ao problema é fundamental para o desempenho de uma rede neural. Esta tarefa requer uma análise cuidadosa dos dados com a finalidade de minimizar ambigüidades e erros. Além disso, os dados coletados devem cobrir amplamente o domínio do problema, incluindo as exceções e as condições limites.

Os dados coletados são separados em duas categorias: dados de treinamento, que serão utilizados para o treinamento da rede e dados de validação, que serão utilizados para verificar o desempenho da rede. O exemplo abaixo mostra a definição de uma das regras do plural:

“Em substantivos simples, acrescenta-se a desinência ‘s’ aos substantivos terminados em vogal, ditongo oral ou ditongo nasal ‘ãe’” (CEGALLA, 1998).

Tabela 1 – Exemplo de palavra no singular e no plural.

Singular	Plural
sarau	saraus
mãe	mães
casa	casas
pai	pais

Fonte: Cegalla, 1998

É recomendável a reordenação aleatória dos dados para prevenção de tendências associadas à ordem de apresentação dos dados (polarização). Deve-se notar que o número de sinapses de uma RNA é função do número de posições do vetor de entrada da rede. Sendo assim, um vetor de entrada com grande dimensão implica em um grande número de sinapses, que deverão ser ajustadas.

O número adequado de dados para o treinamento de uma RNA é função do número de sinapses desta rede. Assim, quanto mais variáveis livres (sinapses e bias) uma RNA tiver, mais restrições (exemplos de treinamento) são necessárias.

Existem inúmeras ferramentas matemáticas para a escolha das informações adequadas na composição do vetor de entrada. Dentre tais ferramentas podemos destacar: a análise do nível de entropia (LUDWIG et. al, 2005); a distância de Battacharia (LUDWIG et. al, 2004) e a análise dos componentes principais.

4.1.3 Configuração da Rede

Esta fase pode ser dividida em três etapas:

- a) Seleção do paradigma neural apropriado à aplicação, isto é, qual a configuração da rede neural a ser utilizada. No presente trabalho foram estudados os principais paradigmas (Perceptron Simples, Perceptron Múltiplas Camadas, Rede de Kohonen) e encontrou-se logo boa resposta com um Perceptron Simples, conforme será mostrado mais adiante;
- b) Determinação da topologia da rede, ou seja, o número de camadas e o número de neurônios ou nós em cada cama;
- c) Determinação do algoritmo de treinamento, taxa de aprendizagem e demais parâmetros de treinamento;
- d) Escolha do tipo de função de transferência.

Estas escolhas ainda são feitas de forma empírica, embora existam algumas heurísticas que conduzem a opções mais acertadas. Em geral o procedimento requer grande experiência dos projetistas, o que só é obtido com o esforço ao longo do tempo na aplicação a variadas aplicações. A solução, empírica, apresentada neste trabalho com certeza não é única possível, mas algum esforço já foi realizado para a encontrar a melhor solução.

4.1.4 Treinamento

Nesta fase, serão ajustados os pesos das conexões. A tarefa aqui é a determinação dos valores iniciais dos pesos sinápticos (inicialização da rede), qual o algoritmo de aprendizagem e qual o tempo de treinamento para o aprendizado da rede.

Os valores iniciais dos pesos da rede são, normalmente, números aleatórios uniformemente distribuídos, em um intervalo definido. Para esta fase, (NGUYEN; WIDROW, 1990) encontraram uma função que pode ser utilizada para determinar valores iniciais melhores que valores puramente aleatórios.

O algoritmo de aprendizagem é escolhido em função do tipo de rede utilizada e das características dos dados que serão ajustados. Entretanto, o processo é basicamente empírico.

Quanto ao tempo de treinamento, podem ser adotados alguns indicadores, dentre os quais um número máximo de ciclos, a taxa de erro médio por ciclo, ou ainda, a capacidade de generalização da rede. Pode ocorrer que um determinado instante do treinamento a generalização comece a degenerar, causando o problema de over-training, ou seja, a rede se especializa no conjunto de dados do treinamento e perde a capacidade de generalização, ou seja, memoriza os dados e não mais consegue definir uma saída correta para outros dados que estejam fora do padrão de entrada: a rede decora.

O ideal é que o treinamento venha a ser interrompido quando a rede apresentar uma boa capacidade de generalização e quando a taxa de erro for admissível. Assim, deve-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

4.1.5 Teste

Durante esta fase, o conjunto de validação é utilizado para determinar o desempenho da rede com dados que não foram apresentados à mesma. Esta fase é a que verificar se a rede não decorou os dados de entrada e valida a rede para a aplicação desejada.

Devem ser considerados ainda outros testes como a análise dos pesos sinápticos e níveis de bias, pois se existirem valores muito pequenos, as conexões associadas podem ser consideradas insignificantes e assim serem eliminadas (prunning). De modo inverso, valores muito maiores que os outros indicam a possibilidade de over-training da rede.

4.1.6 Integração

A integração é o processo final do projeto de uma rede neural, para sua aplicação ao objetivo desejado. Assim, com a rede treinada e validada, e possível à integração dela a um sistema. Este sistema deverá conter facilidades de utilização como interface conveniente e facilidades de aquisição de dados através de planilhas eletrônicas, interfaces com unidades de processamento de sinais, ou arquivos padronizados. Além disso, o sistema deve periodicamente monitorar seu desempenho e fazer a manutenção da rede quando for necessário ou indicar aos projetistas a necessidade de novas seções de treinamento.

Muitos dos projetos de redes neurais são realizados utilizando o software Matlab (HANSELMAN, 1999), que apresenta funções específicas para desenvolvimento de sua arquitetura, treinamento e validação. Mas qualquer compilador pode gerar programas que treinem e simulem redes neurais, como a linguagem C.

Este trabalho apresenta um modelo computacional de RNA e um software específico para testes e avaliações deste modelo. O objetivo principal deste software será obter o plural das palavras usando RNA, que reconhecerá o padrão na formação do plural das palavras, e terá a capacidade 'inteligente' de montar o plural de quaisquer palavras.

A seguir no capítulo 4.2, será descrito o modelo, a arquitetura da rede neural e os procedimentos adotados para execução da RNA.

4.2 Solução adotada

De acordo com o que foi discutido na introdução e apresentado como objetivo deste trabalho foi pesquisado e definido um modelo computacional, utilizando redes neurais artificiais, que torna possível ao computador o comportamento inteligente de gerar o plural das palavras.

A figura 4 demonstra, passo a passo, o modelo de solução adotado, para atingir a capacidade inteligente de pluralizar as palavras.

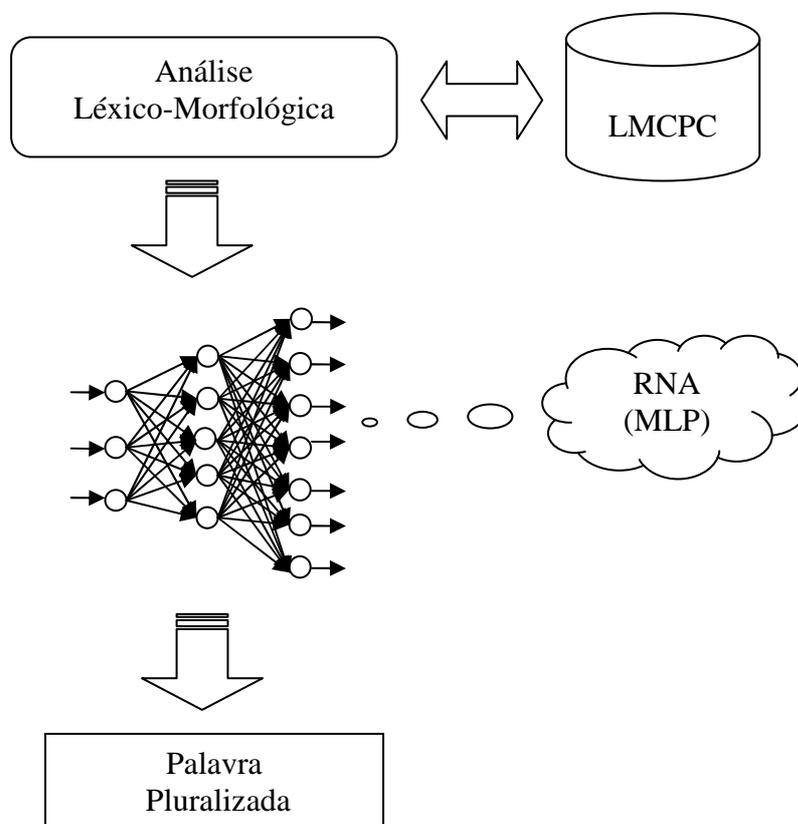


Figura 4 - Modelo de solução adotada

Fonte: Próprio, 2008.

4.2.1 Análise Léxico-Morfológica

A primeira etapa consiste em uma Análise Léxico-Morfológica: processo que classifica as palavras e que validará se a palavra é um substantivo ou adjetivo. Certas palavras podem ser tratadas de uma maneira que não corresponde muito à

maneira como elas estão sendo definidas normalmente, por exemplo, por meio do tipo “nome” se pode representar nomes e substantivos (MICHEL, 2000).

Para resolver esta etapa é utilizado o dicionário léxico com base no Léxico Multifuncional Computadorizado do Português Contemporâneo (LMCPC) (NASCIMENTO, 2000) que foi usado no trabalho de Junior (2004). O dicionário possui um grande número de palavras, cerca de 128.000, e suas possíveis classificações, possibilitando assim uma identificação bastante abrangente de palavras do idioma português. Por outro lado, este dicionário léxico não possui informações mais detalhadas da palavra, como por exemplo, gênero, número, grau, pessoa, o que não é um problema, pois, esta informação espera-se alcançar utilizando algoritmos de aprendizagem.

Este trabalho restringe-se as desinências nos nomes (adjetivos e substantivos) como o gênero (masculinos ou feminino) e o número (singular ou plural), pois, pode-se também trabalhar com as desinências nos verbos, a pessoa e o número, o tempo e o modo. Abaixo exemplos de desinências:

a) Nos nomes (adjetivos e substantivos):

menin	a	s
radical	desinência de gênero (feminino)	desinência de número (plural)

b) Nos verbos (a pessoa e o número, o tempo e o modo):

and	à	sse	mos
radical	vogal temática	desinência de tempo (imperfeito e de modo substantivo)	desinência de pessoa (1º) e de número (plural)

Após a análise léxico-morfológica da palavra para confirmar a desinências nos nomes (adjetivos e substantivos), esta será estudada (morfologia).

4.2.2 Estudo da Palavra

A mesma palavra do processo anterior será classificada quanto ao número de sílabas (monossílaba, dissílaba, trissílaba e polissílaba), quanto ao acento tônico ou quanto a acentuação gráfica, como denominam algumas bibliografias e quanto a

sílaba tônica onde é levado em conta a pronúncia da palavra, a questão fonética, a força de pronúncia da sílaba (oxítona, paroxítona e proparoxítona).

Para a classificação da palavra é necessário fazer a separação das sílabas, onde foi utilizado o algoritmo de Divisão Silábica de Texto Escrito e Falado (GOUVEIA, 2000) e como solução para o caso da fonética foi aplicado o algoritmo utilizado no projeto do Sintetizador de Voz de Qualidade LianeTTS (BORGES, 2007) em software livre desenvolvido na UFRJ.

Estas classificações são importantes, pois são informações fundamentais que definem de qual maneira pluralizar uma palavra. No processo de treinamento da RNA serão mostradas as regras dos plurais e o porque destas classificações. Também os algoritmos para classificação da palavra serão mais abordados no capítulo sobre a implementação.

4.2.3 Arquitetura da RNA

Como solução do problema de classificação de padrões abordado neste trabalho, foi escolhida uma rede neural do tipo multicamadas. A MLP foi escolhida devido à sua fácil implementação e também pelo fato de bons resultados alcançados na literatura utilizando essa abordagem (CONRADS; ROEHL, 1999), (YONG; LIM, 2001). Sua estrutura é composta por uma camada sensorial, que recebe três sinais de entrada (duas letras e a classificação da palavra), uma camada oculta, contendo cinco neurônios, média ponderada entre entradas e saídas, conforme (KOVACS, 2002) e uma camada de saída, constituída de seis neurônios conforme o grafo arquitetural apresentado na figura 4.2. Sua alimentação é do tipo feedforward, função de transferência sigmóide e o algoritmo de treinamento escolhido é o backpropagation.

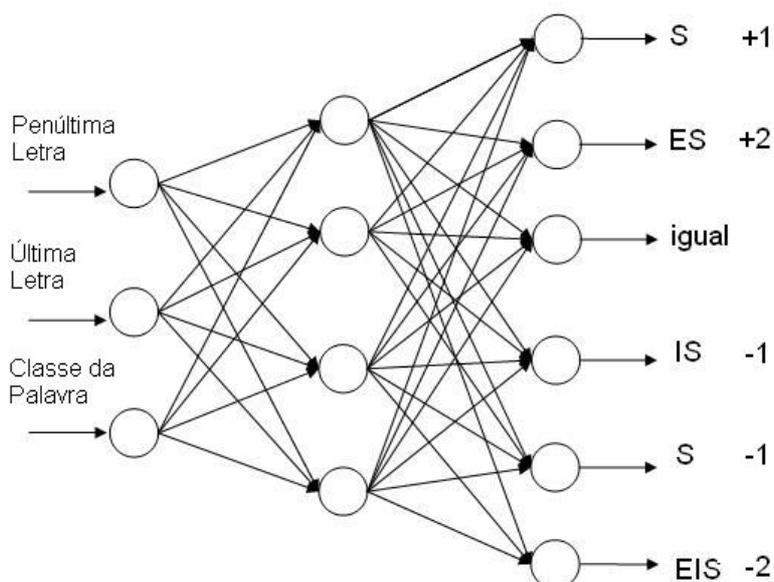


Figura 5 - Grafo arquitetural da MLP

Fonte: Próprio, 2008

A seguir cada uma das camadas do grafo arquitetural da MLP será descrita.

4.2.3.1 Camada de Entrada

A camada sensorial, camada de entrada, recebe três sinais de entrada que são as duas últimas letras da última sílaba da palavra e a classificação da palavra que são obtidas, respectivamente, através do algoritmo de Divisão Silábica de Texto Escrito e Falado (GOUVEIA, 2000) e do algoritmo para o Sintetizador de Voz de Qualidade LianeTTS (BORGES, 2007).

Como visto no referencial teórico, são as terminações das palavras que determinam se a palavra está no singular ou no plural. Desta maneira, o primeiro passo é codificar as informações em valores numéricos, por exemplo, para as regras (CEGALLA, 1998):

- a) Em substantivos simples, acrescenta-se a desinência 's' ao substantivo terminado em vogal, ditongo oral ou ditongo nasal 'ãe';
- b) Os substantivos terminados em 'al', 'el', 'ol' e 'ul' fazem o plural trocando o 'l' por 'is'.

A tabela 2 demonstra um exemplo de representação dos sinais de entrada convertidos de caracteres para valores numéricos.

Tabela 2 – Terminações codificadas em valores numéricos.

Terminação e Classificação da Palavra	=	Valores Numéricos de Entrada
s – a – oxítona	=	78 – 62 – 1
m – e – oxítona	=	74 – 67 – 1
a – l – paroxítona	=	62 – 73 – 2
e – l – paroxítona	=	67 – 73 – 2
o – l – proparoxitona	=	70 – 73 – 3

Fonte: Próprio, 2008.

4.2.3.2 Camada Intermediária ou Escondida

É o local onde realiza-se a maioria dos processamentos, através de conexões ponderadas. O número de elementos numa camada intermediária define a treinabilidade e a precisão da aproximação funcional. Para o trabalho foram utilizados 4 neurônios na camada escondida, nos quais são a média ponderada entre a camada de entrada e a camada de saída, pois o estudo de (KOVACS, 2002) mostra que isto aumenta em muito a capacidade de aprendizado da rede.

4.2.3.3 Camada de Saída

Nesta camada ocorre a apresentação do resultado final obtido após os cálculos efetuados nas camadas anteriores, ou seja, o neurônio que possuir a saída certa será ativado.

Como visto no referencial teórico somente é possível representar de seis formas o plural das palavras (CEGALLA, 1998), por isso, a quantidade de neurônios de saída é igual a seis. São exemplos de saídas as operações realizadas em: aluno(s), casa(s), funil(is), pastor(es). A tabela 3 apresenta um resumo das regras do plural e as saídas possíveis.

Tabela 3 – Resumo das regras do plural

Terminação	Saída
a, o, u	Adicionar o “S”
al, el, ol, ul	Reduzir uma letra e adicionar o “IS”

il – oxítonas	Reduzir uma letra e adicionar o “IS”
il – paroxítonas	Reduzir duas letras e adicionar o “EIS”
r, z	Reduzir duas letras e adicionar o “ES”
n	Adicionar o “S”
s – oxítonos	Adicionar o “ES”
s – não oxítonos	Palavra fica igual
x	Palavra fica igual
ão	Adicionar o “S”

Fonte: Próprio, 2008

Na tabela é mostrado as terminações que determinam as regras do plural (conforme visto no referencial, item 3.3.2) e que formam um conjunto de seis saídas diferentes representadas na arquitetura da rede neural proposta.

4.3 Considerações sobre o capítulo

Nesse capítulo foi apresentado o escopo do trabalho, suas premissas, restrições e limites. Além disso, também foi detalhada a metodologia utilizada para elaborar a rede neural artificial assim como aspectos referentes à modelagem dessas funcionalidades, demonstrando a interação do modelo.

No próximo capítulo serão abordadas questões referentes a implementação e ao projeto físico do protótipo do software que foi denominado PluralRNA e foi utilizado para avaliação do presente modelo.

5 IMPLEMENTAÇÃO

De acordo com o que foi discutido na introdução e apresentado como objetivo deste trabalho de pesquisa, foi desenvolvido um protótipo de software para avaliar o comportamento e a utilização de Redes Neurais Artificiais na morfologia da língua portuguesa.

O principal requisito do software foi permitir ao usuário a entrada de uma palavra no singular e a partir desta gerar a palavra no plural.

Neste capítulo então, serão expostos as principais soluções de implementação para cada problema encontrado no processo de desenvolvimento desse trabalho, assim como a arquitetura do ambiente proposto e a avaliação do trabalho.

5.1 Modelagem das funcionalidades

Para documentar a modelagem proposta para o trabalho, foram utilizados diagramas especificados pela notação UML (OMG, 2006) que foram confeccionados com o auxílio da ferramenta JUDE na versão Community 5.2.1 (JUDE, 2006).

A partir de uma análise funcional do principal requisito deste protótipo de software verificou-se a necessidade de dois níveis de usuário do software conforme os diagramas demonstrados abaixo.

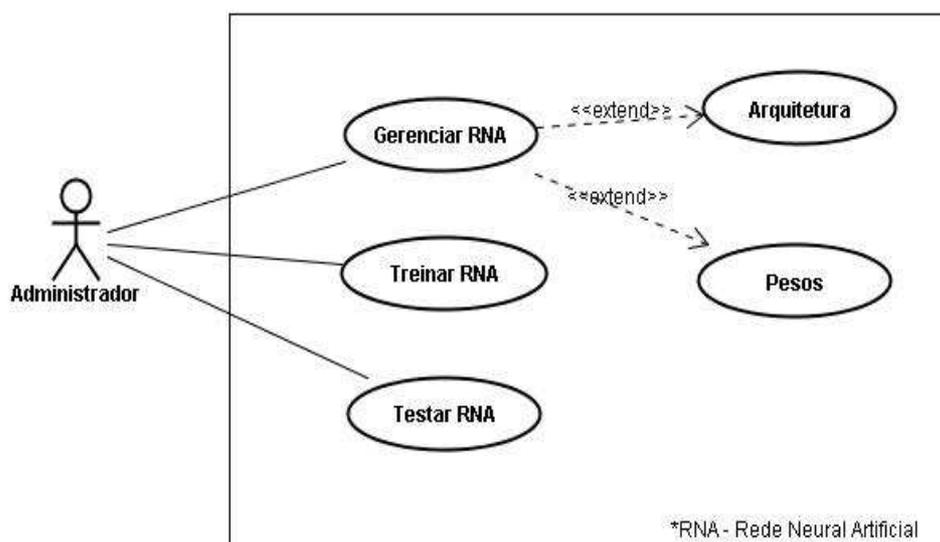


Figura 6 – Casos de uso: funcionalidades providas para usuário Administrador
Fonte: Próprio, 2008

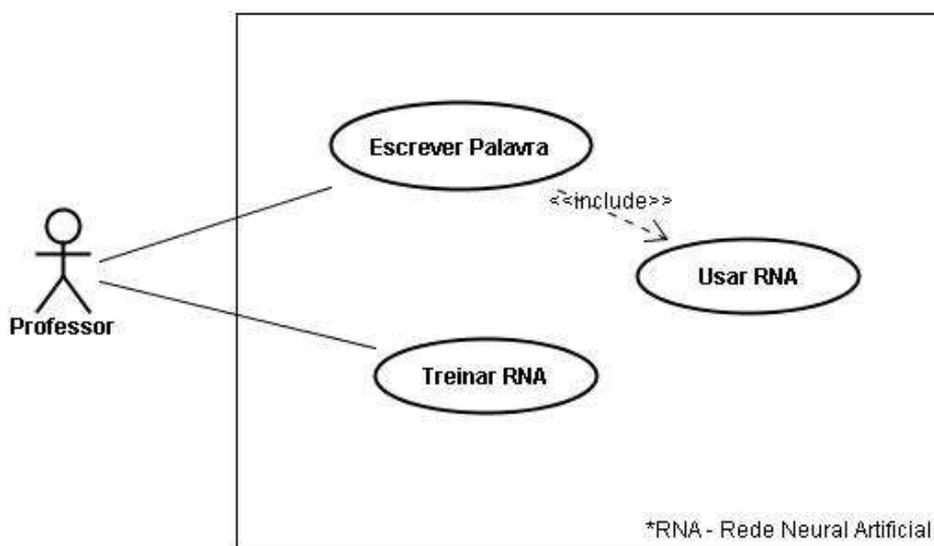


Figura 7 – Casos de uso: funcionalidades providas para usuário Professor
Fonte: Próprio, 2008

Para o melhor entendimento do diagrama, as próximas seções descreverão os componentes e as suas funcionalidades.

5.1.1 Atores

Os atores são as pessoas ou entidades computacionais envolvidas na execução das funcionalidades do sistema. Cada um deles será descrito a seguir, assim como qual o seu papel dentro da aplicação:

- **Administrador:** Representa o usuário com a responsabilidade de configurar parâmetros como número de épocas, neurônios da camada de entrada, neurônios da camada oculta, neurônios da camada de saída, taxa de aprendizagem e taxa de inércia. Todos estes parâmetros estão diretamente ligados a arquitetura que foi indicada por este trabalho, este o porquê de existir este nível de usuário no sistema;
- **Professor:** É a representação do usuário com a tarefa principal de utilizar a aplicação e avaliar o funcionamento e as respostas obtidas da RNA que foi implementada. No caso de uma resposta inválida, este usuário tem a opção de efetuar um treinamento e mostrar para a RNA qual a resposta certa.

-

5.1.2 Casos de uso

Os casos de uso demonstram como ocorre à interação entre os atores e o sistema, e eles serão descritos a seguir:

5.1.2.1 Gerenciar a RNA

Esse caso de uso engloba as operações realizadas pelo administrador para gerenciar a arquitetura da rede neural. Dividindo-se em duas operações:

- **Configurar arquitetura da RNA:** O administrador pode configurar parâmetros como número de épocas, neurônios da camada de entrada, neurônios da camada oculta, neurônios da camada de saída, taxa de aprendizagem e taxa de inércia.
- **Configurar pesos da RNA:** Por padrão, estes pesos foram deixados fixos no programa, para evitar maiores dificuldades. Sendo definido a conversão da letra para seu valor na tabela ASCii. Os valores adotados para as taxas de aprendizagem e inércia irão interferir diretamente no processo de atualização dos pesos.

Uma taxa alta de aprendizagem pode fazer o processo oscilar em torno da solução procurada. Por isso, usa-se um determinado valor para a taxa de inércia, com o objetivo de evitar tais oscilações e atingir a solução em menor tempo.

5.1.2.2 Treinar a RNA

Este caso de uso está disponível para os dois atores e compreende ao processo de treinamento ou aprendizado. No qual é apresentado à rede MLP, em várias épocas ou iterações, um subconjunto definido de valores de entrada e de saída correlacionados. Ou seja, é apresentado um conjunto de palavras de exemplo para que a rede neural identifique e aprenda as regras do plural.

Nesta apresentação, o valor obtido na saída é comparado com o valor desejado ou "alvo". Através de "backpropagation" (retropropagação), as conexões sinápticas são atualizadas uma a uma, de forma a minimizar o erro e encontrar um mínimo global para a rede MLP. Desta forma, a rede é capaz de generalizar soluções para o problema em foco.

5.1.2.3 Escrever Palavra

Compreende a entrada de uma palavra, pelo ator Professor, na rede neural. Esta entrada individual de palavras se fez necessária por este trabalho ser específico para a morfologia da palavra. Inicialmente foi pensado em fazer a leitura de um texto, mas este procedimento foi cancelado na implementação porque uma análise sintática e uma análise semântica tornam-se necessárias para identificar o real sentido da palavra no texto. Por exemplo, a palavra "casa", isoladamente é um substantivo, porém, no contexto de uma frase no texto poderia ser um verbo: "Ela casa com ele."

Contudo, a arquitetura necessária para efetuar a leitura de um texto e executar a análise léxica, análise sintática e análise semântica estão prontas neste trabalho.

5.1.2.4 Usar e Testar a RNA

Este compreende o processo de teste da RNA. Após a fase de treinamento a rede MLP estará apta (caso tenha convergido adequadamente) para executar a

tarefa esperada. Apresenta-se então, um subconjunto de valores de entrada diferente do subconjunto de treinamento, e a partir dos estados atuais das sinapses a rede MLP processa-se estes valores e se produz uma saída compatível, conforme o "conhecimento adquirido".

5.2 Desenvolvimento do Protótipo

Após análise dos trabalhos de SILVA (1999) e VALDAMERI (1997) e dos resultados obtidos com os mesmos, optou-se pela utilização dos componentes de modelo de rede MLP com aprendizado de retro-propagação de erro.

A escolha deste modelo de RNA deu-se principalmente pela grande capacidade de generalização e na sua rápida operacionalização.

5.2.1 Ferramentas utilizadas

A ferramenta de programação visual utilizada para o desenvolvimento do protótipo foi o Delphi, porque sua característica Rapid Application Development ou Desenvolvimento Rápido de Aplicações (RAD), torna-se um diferencial em relação a outras linguagens de programação para este tipo de trabalho, onde foi avaliado o comportamento da RNA e não questões de desenvolvimento de software.

Para armazenamento dos dados foi utilizado o banco de dados Interbase, criado pela Interprise Corporation em 2000. A ferramenta IBExpert foi utilizada para manipular o banco de dados e possibilita a análise estruturada de uma aplicação, fornecendo assim os passos e um estudo antecipado dos itens que serão necessários para a implementação do protótipo.

5.2.2 Implementação do banco de dados

A primeira etapa da implementação proposta neste trabalho foi a estruturação da base de dados. Para este protótipo optou-se pela criação de apenas um banco de dados, onde todas as estruturas necessárias para a aplicação foram criadas e atualizadas. A base de conhecimento é apresentada no Apêndice A.

Esta estrutura tem o objetivo de armazenar informações das áreas de estudos suportadas pelo PluralRNA e contempla o cadastramento de Regras do Plural e a Arquitetura da RNA.

É importante salientar que todas as estruturas do modelo de dados foram construídas para este protótipo, mesmo as que não estão sendo utilizadas no momento. Isto permitirá que, em trabalhos futuros, sejam necessários poucos acertos para a continuação e melhoria deste trabalho.

5.2.2.1 Estrutura de dados construídas

Neste protótipo foi necessário utilizar poucas tabelas. A tabela CONFIGURACÃO é utilizada para armazenar a configuração atual da RNA e possui os atributos descritos no capítulo 4, são eles: neurônios de entrada (configurado para três), neurônios ocultos (configurado para quatro), neurônios de saída (configurado para seis), número de iterações, taxa de aprendizagem e taxa de inércia. Estes últimos três parâmetros serão melhor detalhados na implementação das telas.

A tabela LETRA_PESO é utilizada para representar o valor de cada letra do alfabeto e o seu valor da tabela ASCII respectivamente. Como por exemplo, letra A tem valor na tabela ASCII igual a 65, letra B, valor 66, e assim por diante.

A tabela PALAVRA_CLASSIFICACAO contém as três classificações quanto a sílaba tônica, são elas: oxítone, paroxítone e proparoxítone.

A tabela SINGULAR_PLURAL é utilizada para o cadastramento dos treinamentos a serem executados na RNA. Nesta tabela o usuário cadastra a palavra no singular, a classificação da palavra (oxítone, paroxítone e proparoxítone) e a palavra no plural, a partir deste cadastro o sistema identifica a regra do plural que está sendo usada nesta combinação e manda esta informação para a RNA. A informação quanto a regra que está sendo usada está armazenada na tabela NEURONIO_SAIDA, onde nesta estão as letras que serão adicionadas e o quanto vai ser necessário alterar na palavra de resposta, além de informar qual o neurônio deve ser “ativado” na saída da RNA. O processo inverso ocorre quando do teste da RNA, temos o neurônio de saída e a partir dele, quais as letras devem ser adicionadas na nova palavra.

As tabelas CONHECIMENTO e UTILIZACAO são utilizadas para gerar os logs necessários para avaliação deste modelo.

E por fim, as tabelas PALAVRA e PALAVRA_CLASSE representam o dicionário léxico com base no LMCPC - Léxico Multifuncional Computadorizado do Português Contemporâneo (NASCIMENTO, 2004) que foi importado para dentro deste protótipo. Esta importação compreende a conversão do arquivo texto do dicionário para o presente banco de dados.

5.2.3 Implementação do aplicativo

A segunda etapa da implementação proposta neste trabalho foi construir as interfaces do aplicativo. Os algoritmos de separação de sílaba e o descobre sílaba tônica são apresentados no Apêndice B. O código fonte do aplicativo é apresentado no Apêndice C.

Para construir o modelo computacional, que validasse a eficiência da rede neural utilizando-se a topologia MLP, o programa desenvolvido possui uma interface que permite a manipulação dos dados referentes à configuração dos parâmetros da rede MLP, uma interface que permite o treinamento da rede e o teste da rede com a entrada de palavra no singular.

5.2.3.1 Configurações da rede neural

A figura 8 mostra a interface destinada a informação dos dados de configuração da arquitetura da rede MLP.



Figura 8 – Tela de configuração da rede neural

Fonte: Próprio, 2008

Nesta opção estão disponíveis a visualização ou alteração de algumas variáveis. Estas variáveis definem como será o treinamento da rede MLP. São elas:

- 1) **Números de Épocas:** é o número de iterações de treinamento da rede MLP.
- 2) **Neurônios Camada Entrada:** No modelo deste trabalho foi definido a entrada de três informações: a penúltima letra da palavra, a última letra e sua classificação quanto a sílaba tônica.
- 3) **Neurônios Camada Oculta:** quantidade de neurônios que irão constituir a camada oculta da rede MLP, ou seja, a camada ligando os neurônios de entrada aos de saída.
- 4) **Neurônios Camada Saída:** No modelo deste trabalho foi identificado seis saídas possíveis, que formam o plural das palavras, de acordo com cada regra são elas: S, ES, igual, IS, S (menos) e EIS.
- 5) **Taxa de Aprendizagem:** índice que define a atualização das sinapses (variando de 0% a 100%), ou seja, quanto será alterado do valor atual da sinapse na próxima etapa.

- 6) **Taxa de Inércia:** índice que define a inércia ou Momentum da aprendizagem (variando de 0% a 100%), ou seja, quanto do valor atual da sinapse irá persistir na próxima época.

Os valores adotados para as taxas de aprendizagem e inércia irão interferir diretamente no processo de atualização dos pesos. Uma taxa alta de aprendizagem pode fazer o processo oscilar em torno da solução procurada. Por isso, usa-se um determinado valor para a taxa de inércia, com o objetivo de evitar oscilações e atingir a solução em menor tempo. Após exaustivos testes, o melhor valor encontrado para a taxa de aprendizagem e a taxa de inércia, respectivamente, foram 90% e 50%.

5.2.3.2 Conjuntos de entrada

A figura 9 mostra a interface destinada ao conjunto de entrada da rede MLP.



Figura 9 – Tela do conjunto de entrada (singular x plural)
Fonte: Próprio, 2008

Esta opção permite ao professor interferir indicando a palavra no singular, classificação da palavra e a palavra no plural. Uma observação cabe para o campo Classificação: a palavra no singular será automaticamente classificada pelo sistema, pois este algoritmo para identificar a tonicidade fonética da palavra foi adicionado a mais neste trabalho.

Analisando somente esta tela dá a entender que o sistema usa uma tabela com a palavra no singular e seu plural, processo o qual não permitiria ao sistema pluralizar palavras diferentes das que não estão no conjunto de entrada, além do fato de que, para ter o plural de todas as palavras, teria que alimentar uma base de dados enorme com todas as palavras.

5.2.3.3 Treinamento

A figura 10 mostra a interface destinada ao treinamento da rede MLP.

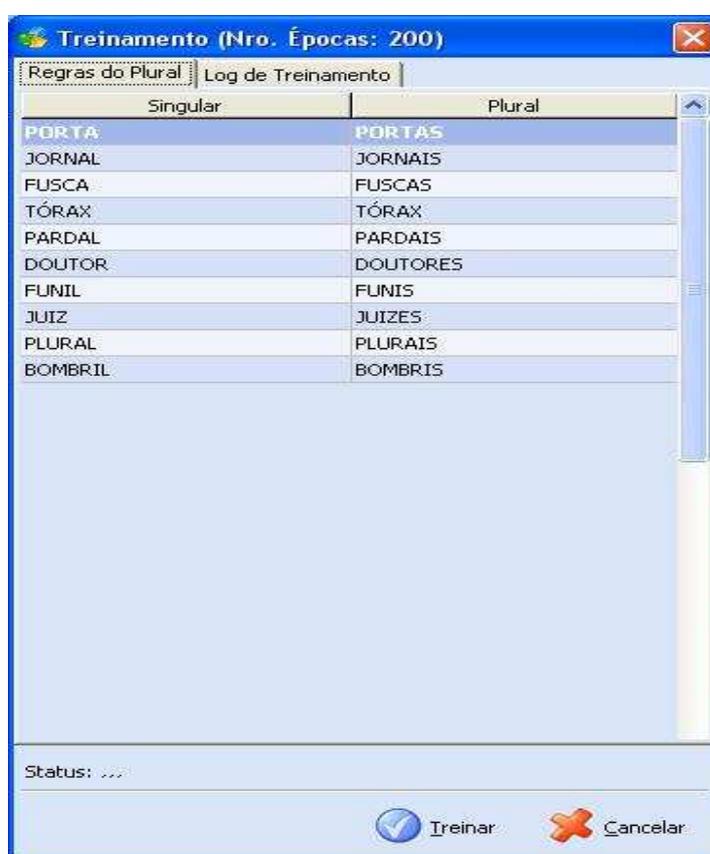


Figura 10 – Tela de treinamento da rede MLP

Fonte: Próprio, 2008

Esta opção executa o treinamento da rede MLP que acontece em um loop que conta até o número de épocas especificado na configuração da rede e um loop interno que varre a tabela de treinamento do primeiro registro até o final da tabela. Então para cada época, são apresentadas à rede todas as amostras do conjunto de treinamento, uma a uma.

Para cada conjunto de treinamento o algoritmo captura a penúltima e última letra da última sílaba. E executa o algoritmo mostrado no Apêndice B para separar

sílabas e para encontrar a sílaba tônica. Estas são informações para alimentar a entrada da rede (os três neurônios da entrada).

Além disso, compara a palavra no singular com a palavra no plural e identifica a diferença entre elas. Esta diferença é a indicação de qual neurônio deve ser ativado na saída da rede (qual dos seis neurônios da saída deve ser identificado).

Após o treinamento será mostrado um log, conforme mostra a figura 11, onde são mostrados o erro ou custo total que a rede apresenta para o conjunto de treinamento processado. Estes valores são discutidos na avaliação deste trabalho.

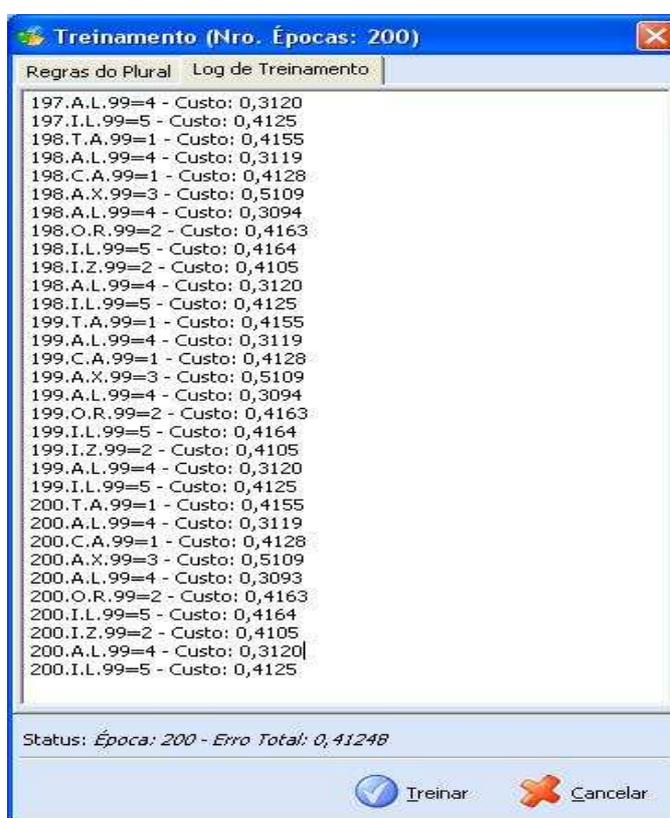


Figura 11 – Tela de log do treinamento da rede MLP

Fonte: Próprio, 2008

5.2.3.4 Pluralizar uma palavra

A figura 12 mostra a interface destinada ao teste da rede MLP.

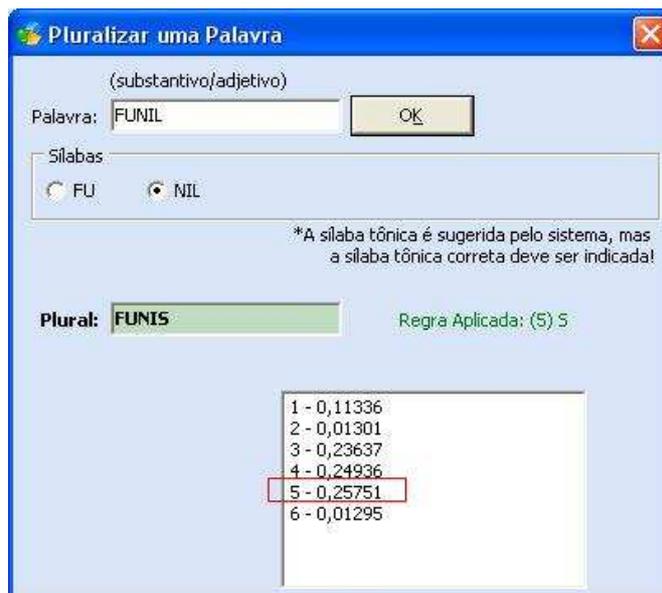


Figura 12 – Tela de teste da rede MLP
Fonte: Próprio, 2008

Esta opção executa o teste da rede MLP. O teste acontece a partir da entrada de uma palavra no singular e uma consulta ao dicionário léxico com base no LMPCP garante que a palavra seja um substantivo ou um adjetivo.

Para cada palavra digitada, o algoritmo existente no botão OK, captura a penúltima e última letra da última sílaba e é executado o algoritmo mostrado no Apêndice B para separar sílabas e encontrar a sílaba tônica. O resultado é mostrado na caixa Sílabas, onde é possível a intervenção do professor para indicar a sílaba tônica. Estas três informações são enviadas para a entrada da rede (os três neurônios da entrada).

A partir do momento que é indicado a sílaba tônica a rede neural é testada. A regra aplicada, ou seja, o neurônio ativado na saída da rede é mostrado no campo “Regra Aplicada”. Fazer o plural da palavra é alterar a sua terminação, última sílaba. Esta modificação da terminação acontece quando aplicado a regra do plural ,conforme a tabela NEURONIO_SAIDA.

Exemplo para a palavra “DOUTOR”. A regra diz que os substantivos terminados em “r” fazem o plural de acréscimo de “es”. O algoritmo fará a entrada na rede com três informações:

- 1) penúltima letra: o (valor ASCII = 79)
- 2) última letra: r (valor ASCII = 82)
- 3) classificação: oxítone (valor conforme tabela CLASSIFICACAO = 1)

Levando em consideração que a rede neural já tenha sido previamente treinada, com outras palavras da mesma regra, este teste mostrará o resultado no campo “Plural”: DOUTORES, e no campo “Regra Aplicada”: (2) ES.

5.3 Considerações sobre o capítulo

Esse capítulo apresentou a implementação do protótipo PluralRNA proposto para avaliação do modelo e da arquitetura da rede neural artificial de multi camadas. Além do detalhamento das funcionalidades desse protótipo, como por exemplo, a configuração da arquitetura da rede e as fases de treinamento e teste.

Foi descrito também como foi construída a base de dados, quais ferramentas e tecnologias foram utilizadas para a implementação do protótipo. Foram descritas como são utilizadas as estruturas e informações da base de dados para que fosse possível chegar a uma resposta para o usuário.

No próximo capítulo serão apresentadas as avaliações feitas sobre o modelo computacional adotado neste trabalho: rede neural artificial de multi camadas aplicada na morfologia da língua portuguesa. Os resultados apresentados a seguir somente foram possíveis a partir do estudo feito e da utilização do protótipo desenvolvido.

6 ANÁLISE DOS RESULTADOS

Neste capítulo, são apresentadas as simulações propostas para testar a rede neural e também os seus resultados. Os cenários serão detalhados, visto que a variação principal é a regra do plural em questão. É importante também ressaltar que, para cada cenário definido nas simulações, são simuladas algumas palavras com relação à regra utilizada. Dessa forma, há uma resposta no log de treinamento para cada palavra e cada época de treinamento.

6.1 Treinamento X Aprendizado

Segundo Medeiros (2003) quando estruturamos uma rede MLP para um objetivo específico, geralmente, os valores assumidos pelos pesos são aleatórios ou zerados. Quando mencionado “aprendizado”, não se está sendo tão abrangentes quanto ao aprendizado referente ao ser humano. Em uma rede MLP, o processo de aprendizado se restringe a uma espécie de treinamento por inibição ou reforço, onde se “penaliza” a rede se a mesma estiver fora do padrão requerido ou a “elogia” quando está dentro do padrão.

O aprendizado especifica valores para os pesos de forma a dar à rede MLP um sentido de processamento. Ao entrar com dados em uma rede não treinada a saída será aleatória e não previsível. Na medida em que é mostrado à rede um padrão de entrada (exemplos de palavras no singular e no plural) também é mostrado como deve ser o resultado na saída da rede. Para cada amostra apresentada, temos que indicar qual o resultado pretendido à saída (chamado par amostra-resultado).

Analogamente ao exemplo do adestramento de um animal, é preciso fazer com que um exercício, em uma rede MLP, seja repetido inúmeras vezes para que seja

fixado, pois não será na primeira apresentação do par amostra-resultado que a rede irá aprender efetivamente uma classificação (BILOBROVEC, 2004).

Foi preciso, muitas vezes, apresentar o conjunto de amostras-resultado (ou conjunto de treinamento) para a rede começar a gerar o plural das palavras conforme mostrado abaixo no Gráfico 1. Estas repetições são denominadas de épocas de treinamento.

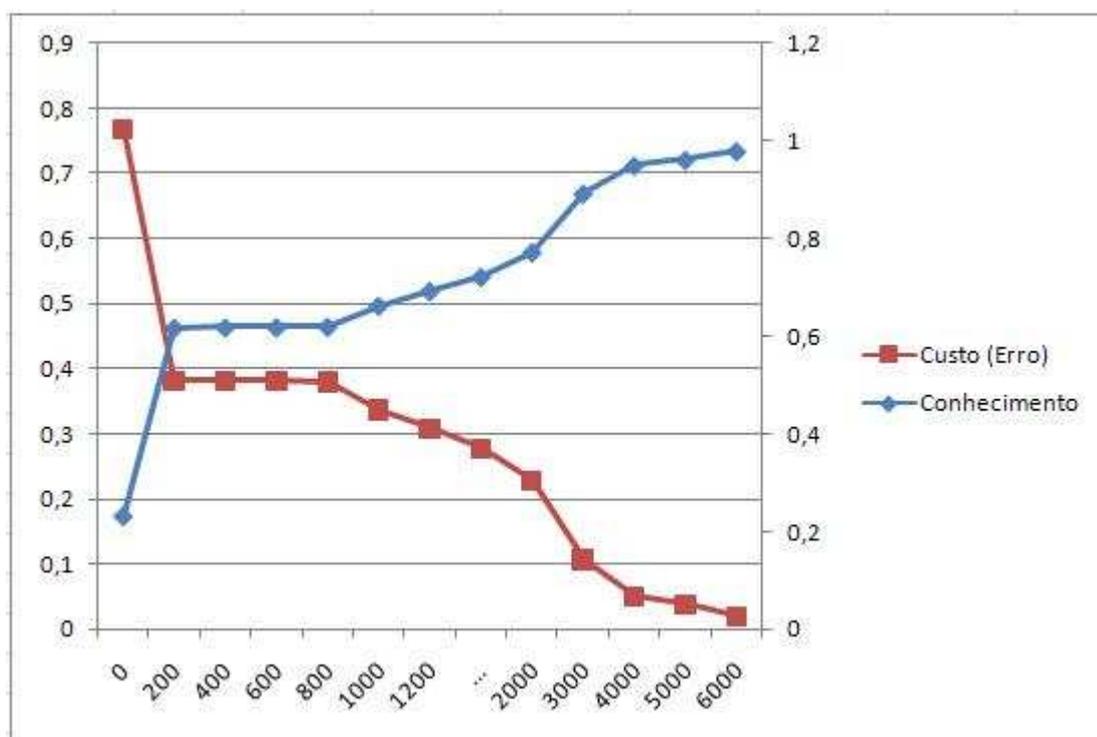


Figura 13 - Relação entre Treinamento X Aprendizado.

Fonte: Próprio, 2008.

Como observado na figura 1 a partir da época de treinamento 200 a rede passa a apresentar conhecimento de uma determinada regra e o custo obtido pelo retorno do backpropagation começou a ser minimizado. Se o erro é diminuído, a obtenção de dados próximos aos desejados é a consequência e a rede fica apta para executar o processo para o qual foi projetada.

6.2 Simulação de uso do Plural RNA

A seguir é mostrado uma simulação de uso do protótipo PluralRNA. Nesta simulação foi apresentada à rede MLP o seguinte conjunto de entrada (executado treinamento com os seguintes pares de amostra-resultado).

Tabela 4 – Simulação: Conjunto de treinamento

<i>Palavra no Singular</i>	<i>Classificação</i>	<i>Palavra no Plural</i>	<i>Regra Utilizada</i>
casa	paroxítona	casas	Como regra geral, o plural é feito pelo acréscimo do s à forma do singular.
jornal	oxítona	jornais	Os substantivos terminados em al, el, ol e ul fazem o plural trocando o l por is.
anil	oxítona	anis	Os substantivos, quando são oxítonos, trocam o l por s.
lápiz	paroxítona	lápiz	Os substantivos terminados em s, quando não são oxítonos, não mudam de forma no plural.

Fonte: Próprio, 2008

Na fase de treinamento, para cada amostra apresentado, a saída respectiva sempre é marcada como 0 ou 1 (ativa ou desativa o neurônio de saída). Altos valores de convergência obtidos na fase de aprendizagem aproximaram os neurônios de saída destes valores para as amostras do conjunto de treinamento. Para amostras diferentes do conjunto de treinamento apresentadas à rede, esta classificou as palavras conforme a proximidade de uma amostra do conjunto de treinamento.

Um conjunto que ilustra isso é o da figura 13. Esta amostra não pertence ao conjunto de treinamento, porém o neurônio vencedor teve um valor de probabilidade de 0,2513, resultando na regra do plural aplicada (neurônio de saída número 4, resultando na regra errada).

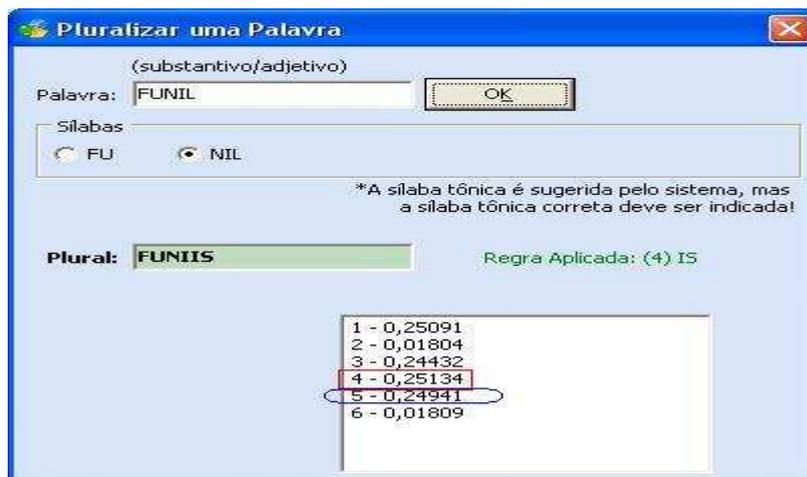


Figura 14 - Tela de simulação: apresentação de erro

Fonte: Próprio, 2008

Como inicialmente os resultados obtidos neste teste não foram satisfatórios, incluímos esta amostra de palavra no conjunto de treinamento. Realizando um novo treinamento, a classificação teve uma margem de representatividade maior para esta amostra e outras próximas.

Após este novo treinamento o teste apresentou um resultado positivo e é mostrado na figura 14.

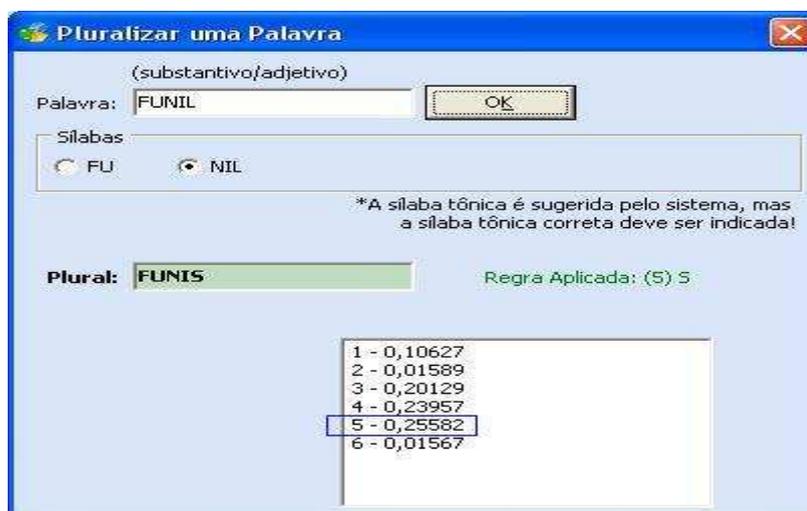


Figura 15 - Tela de simulação: apresentação de acerto

Fonte: Próprio, 2008

Em virtude do processo de treinamento da rede MLP não se dar de forma única, valores ligeiramente diferentes podem ser encontrados para várias execuções do treinamento, diferentes dos explanados no trabalho. Porém, apresentou aproximação muito satisfatória dos exemplos.

O Gráfico 2 demonstra o log de aprendizado para as quatro regras. Na horizontal é representado o número de épocas de treinamento num intervalo de 0 a 6000 épocas. Na vertical está o valor de conhecimento adquirido, num intervalo de 0 a 1, onde zero significa nada de conhecimento e um significa 100% de conhecimento.

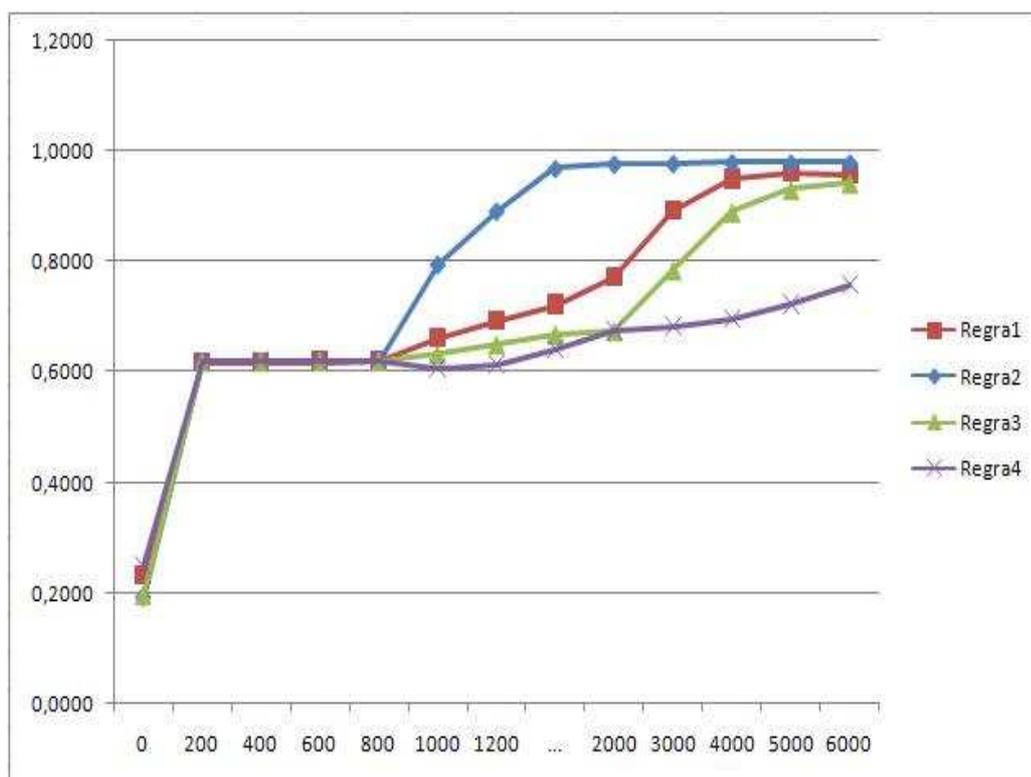


Figura 16 – Log de aprendizado para quatro regras.
Fonte: Próprio, 2008.

Nas primeiras mil épocas de treinamento a rede MLP não apresentava correta saída para nenhuma das quatro regras, a partir de 1200 épocas a regra 2 começou a ser acertada, mas ainda a rede errava as outras regras. Portanto, o perfeito acerto das quatro regras somente foi possível após 6000 épocas de treinamento. Outros trabalhos apresentaram esta mesma necessidade de tempo de aprendizado, entre eles os trabalhos de Valdameri (1997), Silva (1999) e Osório (2000).

6.3 Simulação para todas as regras do plural

Como visto na simulação anterior quanto mais conhecimento para apresentar para a rede MLP, mais longo vai ser o período para alcançar o ideal aprendizado.

Esta simulação foi feita e a rede MLP alcançou bom resultado após 15.000 épocas de treinamento para um conjunto de par amostra-resultado de 16 diferentes regras do plural, que demandou certo esforço, porém após a rede teve 100% de acerto, o que comprovou a eficácia desta técnica.

Contudo algumas observações devem ser feitas para cada regra do plural. E são elas:

- a) *Como regra geral, o plural é feito pelo acréscimo do s à forma do singular.*
Na regra geral não foi encontrada nenhuma dificuldade, pois, a rede MLP foi treinada para ativar a saída um e após este treinamento a rede respondeu corretamente;
- b) *Os substantivos terminados em al, el, ol e ul fazem o plural trocando o l por is:* Para algumas palavras a rede MLP foi capaz de resolver. O neurônio quatro é ativado e resultado correto, como por exemplo: *jornal = jornais, azul = azuis*. Porém, para palavras onde sai de sílaba simples e vai para sílaba com ditongo aberto (oi, ei), a rede MLP não é capaz de resolver, por exemplo: *anzol = anzóis, pastel = pastéis*;
- c) *Os substantivos terminados em il fazem o plural de duas formas:*
 - a. *Quando são oxítonos, trocam o l por s:* Não foi encontrado problemas. O neurônio de saída cinco é ativado e o resultado é obtido sem problemas;
 - b. *Quando são paroxítonos, trocam o il por eis:* Não foi encontrado problemas. O neurônio de saída seis é ativado e o resultado é obtido corretamente;

É importante destacar que o sucesso alcançado neste caso somente foi possível mediante a avaliação da sílaba tônica, processo que não havia sido pensado no início.

- d) *Os substantivos terminados em r e z fazem o plural pelo acréscimo de es:* O neurônio de saída é o número dois e para a grande maioria das palavras relacionadas com esta regra a rede MLP mostrou-se como ótima solução (*pastor = pastores, perdiz = perdizes*). Porém, para palavras onde ocorre mudança na quantidade de sílabas e a nova palavra formada recebe o acento a rede MLP não foi capaz (*juiz = juízes*). Este processo no qual demandaria um controle sobre as regras de acentuação e tonicidade, não foi levado em conta neste estudo;

- e) *Os substantivos terminados em n fazem o plural pelo acréscimo de es ou s:* Para esta regra foi assumido como saída o acréscimo do s. Existem algumas variações entre gramáticos da língua portuguesa sobre as regras o que o caso aqui (alguns indicam acréscimo de es e outros acréscimo de s). Como não foi objetivo deste trabalho estudar o porquê desta diferença, assumiu-se pelo mais simples (acréscimo de s) (neurônio de saída número um). Após a rede convergir o plural das palavras foi gerado sem problemas. Exceto para casos onde acontece a mudança de acentuação e tonicidade da sílaba como, por exemplo: *pólen = polens, hífen = hífens*;
- f) *Os substantivos monossílabos e os oxítonos terminados em s fazem o plural pelo acréscimo de es:* A mesma situação enfrentada anteriormente, ocorre mudança de acentuação e tonicidade. Além de que, em algumas palavras acrescentar sílaba a mais, por exemplo: *mês = meses*;
- g) *Quando não são oxítonos, os substantivos terminados em s não mudam de forma no plural:* Nenhuma dificuldade encontrada, pois, a rede MLP foi treinada para ativar a saída três e após este treinamento a rede respondeu corretamente (*lápiz = lápis, tênis = tênis*);
- h) *Os substantivos terminados em x não mudam de forma no plural:* Nenhuma dificuldade encontrada, pois, a rede MLP foi treinada para ativar a saída três e após este treinamento a rede respondeu corretamente (*tórax = tórax*);
- i) *Os substantivos terminados em ão podem formar o plural de três maneiras:*
- a. *Pelo acréscimo de s;*
 - b. *Pela transformação do ao em ães;*
 - c. *Pela transformação do ao em ões.*

A maneira escolhida para gerar o plural das palavras terminadas em ão foi a primeira (a) (irmão = irmãos), mas alguns problemas de concordância foram identificados, como por exemplo: alemão = alemãos. Não existe uma concordância entre gramáticos sobre esta regra o que não será discutido neste trabalho (conforme visto no item 3.3.2, regra 8). Para fins de avaliar a rede MLP foi definido a regra A como verdade e o neurônio de saída é o número um (acréscimo de s). Resultado: a rede aprendeu esta regra e teve bons resultados, levando em conta que algumas palavras foram geradas de forma estranha (leão = leãos). Para o caso de comporta as transformações propostas pelas regras b e c seria necessário o acréscimo de

mais dois neurônios na camada de saída (ães e ões) e uma definição de qual regra utilizar, pois a entrada na rede seria a mesma ão.

6.4 Considerações sobre o capítulo

Nesta fase de avaliação a participação de um professor do curso de graduação em Letras do Unilasalle foi fundamental para uma correta avaliação das questões referentes à língua portuguesa, no caso o co-orientador deste trabalho o professor Celso Antunes.

De um modo geral a rede neural MLP (aprendizado supervisionado) apresentou-se como ótima solução em generalizar o caso do plural das palavras e apresentou o comportamento esperado.

Algumas questões foram identificadas como o caso das regras de acentuação e a mudança na tonicidade da palavra, o que de certo modo dificultou o resultado 100%. Mas algumas melhorias foram adicionadas ao trabalho, como por exemplo, o algoritmo para identificar a sílaba tônica da palavra e o algoritmo para separação de sílabas, que não haviam sido previstos para implementação.

No próximo capítulo serão mostrados os trabalhos relacionados a esta monografia e no último capítulo será feito o fechamento do trabalho, explanando as conclusões observadas depois do desenvolvimento dele, assim como, as suas contribuições, limitações e possibilidades de trabalhos futuros.

7 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os trabalhos relacionados a esta monografia, suas vantagens e desvantagens, suas limitações e contribuições.

Com o objetivo de entender e analisar os diversos problemas sobre uma rede neural aplicada a língua portuguesa, foram analisadas diversas iniciativas que abordam o assunto. Pesquisadores fizeram progressos significativos em demonstrar o poder de redes neurais em executar tarefas cognitivas. Aqui estão três conhecidas experiências que têm os incentivado a acreditar que redes neurais são bons modelos de inteligência humana.

7.1 NETtalk

Um dos mais atraentes destes esforços é o trabalho sobre uma rede que pode ler um texto em Inglês, chamado NETtalk (SEJNOWSKI e ROSENBERG, 1987). O treinamento para NETtalk foi uma grande base de dados que consiste em palavras em inglês juntamente com o seu correspondente fonético como saída, onde, foi escrita em um código adequado para uso com um sintetizador de discurso. Uma seqüência de fala do NETtalk se mostra bastante interessante de ser escutado. Na primeira saída é um ruído aleatório, posteriormente, o som soa como um ruído mais definido, e mais tarde como se ele estivesse falando em inglês (discurso, que é formado por sons que lembram palavras do Inglês). No final da formação, o NETtalk faz um bom trabalho de pronunciar o texto que lhe é dado. Além disso, esta capacidade generaliza bem textos que não foram apresentados no treinado da rede (SEJNOWSKI e ROSENBERG, 1987).

7.2 Parallel Distributed Processing

Outro influente modelo conexionista foi uma rede formada por Rumelhart e McClelland (1986) para prever sentenças dos verbos em Inglês no passado. A tarefa é interessante, porque apesar que a maioria dos verbos em inglês (os verbos regulares) formam o verbo no passado, adicionando o sufixo “- ed”, muitos dos mais freqüentes são verbos irregulares (“is” / “was”, “come” / “came”, “go” / “went”). O treinamento primeiro foi formado sobre um conjunto contendo um grande número de verbos irregulares, e mais tarde em um conjunto de 460 verbos regulares. A rede conseguiu aprender dos 460 verbos em cerca de 200 rodadas de formação, e ela generalizava bastante para verbos que não foram treinados. Ele ainda mostrou uma boa valorização das “regularidades” a serem encontradas entre os verbos irregulares (“send” / “sent”, “build” / “built”; “blow” / “blew”, “fly” / “flew”) (RUMELHART e MCCLELLAND, 1996).

Durante a aprendizagem, uma vez que o sistema foi exposto ao conjunto de treinamento contendo mais verbos regulares, no qual tinha uma tendência para over-training, de combinar as duas formas regulares e irregulares: (“break” / “broked”, em vez de “break” / “broke”). Isso foi corrigido com mais formação. É interessante notar que as crianças são conhecidas por exibir a mesma tendência para “over-training” durante a aprendizagem de línguas.

No entanto, existe grande debate sobre se Rumelhart e McClelland's, seja este um bom exemplo de como os humanos realmente aprendem o processo das terminações dos verbos. Por exemplo, Pinker & Prince (1988) salientam que o modelo faz um mau trabalho de generalizar a questão com alguns verbos regulares. Eles acreditam que este é um sinal de uma falha básica nos modelos do conexionismo.

A rede pode ser boa em fazer associações e padrões correspondentes, mas têm limitações fundamentais em dominar regras gerais, tais como a formação do verbo no passado (Past Tense). Estas queixas levantam uma questão importante para os modelos do conexionismo, a saber, se pode generalizar redes adequadamente a dominar tarefas cognitivas que envolvem regras. Apesar de Pinker e Prince's não acreditarem, muitos conexionistas acreditam que a generalização da aprendizagem ainda é possível (Niklasson e Van Gelder 1994) (PINKER e PRINCE, 1988) (NIKLISSON e VAN GELDER, 1994).

7.3 Distributed Representations, Simple Recurrent Network, and Grammatical Structure

Os trabalhos de Elman's (1991) sobre redes que podem aprender as estruturas gramaticais, apresentam importantes implicações para o debate sobre redes neurais, caso possam aprender a dominar as regras. Elman treinou uma simples rede para prever a próxima palavra em um grande corpo de sentenças em Inglês. As sentenças foram formadas a partir de um simples vocabulário de 23 palavras usando um subconjunto da gramática do Inglês (ELMAN, 1991). A gramática, embora simples, colocava um duro teste para a conscientização lingüística. Permitiu que uma ilimitada formação de cláusulas relativas ao mesmo tempo formasse acordo entre o substantivo e o verbo. Assim, por exemplo, na frase:

Any man that chases dogs that chase cats... runs.

O singular “man” deve concordar com o verbo “runs”, apesar de intervindo os substantivos no plural (“dogs”, “cats”), que podem causar a intenção de “run”. Uma das características importantes do modelo de Elman é a utilização de conexões recorrentes. Os valores que estão ocultos das unidades são guardados em um contexto, a serem enviados de volta para o nível de entrada para a próxima ronda de transformação. Os valores voltam a partir de camadas de entrada, com uma forma rudimentar de memória da seqüência de palavras na frase de entrada. Rede Elman's exige uma apreciação da estrutura gramatical das frases que não foram inclusas no treinamento.

Prever a próxima palavra numa frase em Inglês é, evidentemente, uma tarefa impossível. Mas após o treinamento intensivo, Elman foi capaz de produzir redes que exibiu perfeito desempenho sobre esta medida, incluindo não apenas o conjunto do treinamento.

Apesar deste desempenho ser impressionante, há ainda um longo caminho a percorrer na formação de redes que podem processar linguagem. Além disso, foram levantadas dúvidas sobre o significado dos resultados de Elman. Por exemplo, Marcus (1998, 2001) argumenta que as redes de Elman's não são capazes de generalizar este desempenho de frases formadas a partir de um novo vocabulário (MARCUS, 1998), (MARCUS, 2001). Isto, ele alega, é um sinal de que modelos

conexionistas simplesmente associam instâncias, e são incapazes de dominar verdadeiramente regras abstratas. Por outro lado, Phillips (2002) argumenta que as arquiteturas clássicas não são melhores a este respeito (PHILIPS, 2002). A suposta impossibilidade de modelos em generalizar o desempenho desta, formou-se um tema importante de discussão.

8 CONCLUSÃO

Ao longo deste trabalho, foram analisados três algoritmos de aprendizagem, mas somente um se mostrou viável e foi aplicado para viabilizar a geração do plural dos substantivos e adjetivos da língua portuguesa.

Atualmente as diversas áreas de Inteligência Artificial são muito pesquisadas e existe uma grande dificuldade no que diz respeito à aplicação destas técnicas. Os resultados aqui apresentados são de grande relevância, pois possibilitam que outros projetos sejam norteados por este estudo.

Durante a coleta dos resultados, ficou evidente o bom desempenho da rede MLP com algoritmo de aprendizagem supervisionado. Em alguns dos testes realizados a RNA aprendeu 100% das regras submetidas, contudo em algumas circunstâncias a RNA não obteve sucesso em sua totalidade e precisou de maior tempo de treinamento. Este processo de treinamento e obtenção de melhores resultados foi um processo exaustivo, pois, a rede neural precisou de muitas etapas para aprender.

Outro fato importante, que não foi previsto na fase de pesquisa, mas foi resolvido durante a implementação foi identificar a sílaba tônica das palavras. Como visto, quando temos o acento presente ficava fácil a classificação, porém quando não temos tornou-se um complicador. Para resolver foi elaborado um algoritmo que identifica a sílaba tônica das palavras, aumentando em muito os resultados de acerto da rede neural.

Portanto, de um modo geral os resultados obtidos com as simulações foram que a rede neural atende a quase todas as necessidades, observando apenas algumas limitações.

8.1 Limitações

O fato da rede neural trabalhar com pesos (números) traz algumas limitações, como por exemplo, o trabalho de transformar a palavra após saber qual a saída da rede foi ativada. Foi o caso para as regras de acentuação que adicionam ou removem acento da palavra pluralizada, pois ocorre mudança na tonicidade da palavra ou mudança na quantidade de sílabas da palavra. Mesmo sabendo o neurônio correto de saída da rede, a correta transformação da palavra não foi possível, devido a não aplicação destas regras.

8.2 Trabalhos futuros

Existem ainda várias áreas em que a tomada de decisões seria de grande apoio e várias áreas em que as redes neurais poderiam auxiliar. Apresenta-se como uma boa solução para os problemas de generalização e de reconhecimento de padrão. Também mostrou-se eficiente para o uso como maneira de representar o conhecimento e aprendizado.

Seria interessante continuar a pesquisa para um nível mais alto (leitura de um texto), efetuando uma análise sintática, semântica e até pragmática, porém focando na aprendizagem. Imagine, por exemplo, uma rede neural capaz de reconhecer palavras em um texto, saber o seu sentido na frase. Dessa forma, seria possível analisar questões mais específicas do conexionismo, ou seja, o modelo neural seria mais fortemente colocado à prova.

Também é pertinente o estudo de técnicas que possam contribuir de outra forma contribuir para uma redução no tempo de treinamento da rede. Já existem hoje alguns algoritmos elaborados com esta finalidade, mas que não são aplicados na Língua Portuguesa, sugerindo assim que esse campo de investigação tem potencial para essas aplicações.

REFÊRENCIAS BIBLIOGRÁFICAS

BILOBROVEC, Marcelo; MARÇAL, Rui F. Martins, PILATTI, Luiz Alberto (2004) – **Aplicações de redes neurais em simulações**. In: XI SIMPEP – Simpósio de Engenharia de Produção, Anais, Bauru, SP.

BITTENCOURT, Guilherme, **Inteligência artificial: ferramentas e teorias**. Série Didática, 3 ed. 2006.

BORGES, José Antônio, **Sistema de síntese de fala para português. sintetizador DOSVOX do NCE/UFRJ**. Rio de Janeiro, 2007.

CEGALLA, Domingos Paschoal. **Novíssima gramática da língua portuguesa**. 33 ed. Companhia Editora Nacional: São Paulo, 1998.

CONRADS, P. A.; ROEHL Jr., E. A.; **Comparing physics-based and neural network models for simulating salinity, temperature, and dissolved oxygen in a complex, tidally affected river basin**. Estados Unidos, 1999. Disponível em: <http://smig.usgs.gov/SMIG/features_0302/beaufort.html>. Acesso em: Março de 2008.

ELMAN, J. L. **Distributed representations, simple recurrent network, and grammatical structure**, Touretzky, 1991.

FERNANDES, Ana Maria da Rocha, **Inteligência artificial: noções gerais**. Florianópolis, 2005.

GARGON, Michel. **Processamento da linguagem natural**. Tese de Mestrado, PUC-SP. Campinas, SP, 2000.

GOUVEIA, Paulo D. F. **Divisão silábica automática do texto escrito e falado**. PROPOR 2000 – 5º Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada, Anais, São Paulo, SP, 2000.

HANSELMAN, D; LITTLEFIELD, B. **MATLAB 5: versão do estudante: guia do usuário**. MAKRON Books: São Paulo, SP, 1999.

HAYKIN, S. **Neural networks: a comprehensive foundation**. New York: Macmillan College Publishing Company, 1994.

HRUSCHKA Jr.. **Propagação de evidências em redes bayesianas: diagnóstico de doenças pulmonares**. Dissertação de mestrado, UnB, 1997.

JUDE (UML Modeling Tool), 2007. Disponível em: <<http://jude.change-vision.com/>>

JUNIOR, Ney José Van Laren. **Uma proposta de processamento de linguagem natural para o bonobot**. Monografia para título em Bacharel, UNILASALLE-RS, Canoas, 2004.

KOEHLER, Cristiane; VICARI, Rosa Maria; FLORES, Cecília Dias. **Mineração de redes bayesianas a partir de base de dados médicos: proposta de algoritmo**. In: IX CIBIS - Congresso Brasileiro de Informática em Saúde, Anais, Ribeirão Preto, SP, 2004.

KOVACS, Zsolt László. **Redes neurais artificiais: fundamentos e aplicações: um texto básico**. 3 ed. Livraria da Física: São Paulo, 2002.

LUDWIG JR, Oswaldo; MONTGOMERY, Eduardo. **Redes neurais: fundamentos e aplicações com programas em C**. Ciência Moderna: Rio de Janeiro, 2007.

LUGER, George F. **Inteligência artificial: estruturas e estratégias para a solução de problemas complexos**. Porto Alegre, 2004.

LUDWIG, O. J., Castro Lima, A.C., SCHNITMAN L., Souza J. A. M. F. **Supervised methods for feature extraction**. 6th Portuguese Conference on Automatic Control. Faro: Portugal, 2004.

LUDWIG, O. J., Castro Lima, A.C., SCHNITMAN L., Souza J. A. M. F. **Entropy analysis applied to nfir models**. American Control Conference – ACC 2005. Portland, USA, 2005.

MAGALHÃES, Thereza Cochar; CEREJA, William Roberto, **Gramática reflexiva: texto, semântica e interação**, 1999.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 4. ed. São Paulo: Atlas, 2001.

MARCUS, G. **Rethinking eliminative connectionism**, Cognitive Psychology, 1998.

MARCUS, G. **The algebraic mind**. Mass MIT Press: Cambridge, 2001.

MEDEIROS, Luciano Frontino de. **Redes neurais em Delphi**. Visual Books: Florianópolis, 2003.

NASCIMENTO, Maria Fernanda Bacelar do. et al. **Léxico multifuncional computadorizado do português contemporâneo**. Lisboa: Universidade de Lisboa, 2000. Disponível em <http://www.clul.ul.pt/sectores/projecto_lmcp.html>. Acesso em: Março de 2008

NGUYEN, D. e WIDROW, B. **Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptatives weights**, Standford University, Standford, CA, 1990.

NIKLISSON, L., e VAN GELDER, T. **On being systematically connectionist**. Mind and Language. 1994.

OMG (Object Management Group) - UML (Unified Modeling Language), 2006.
Disponível em: < <http://www.uml.org/>>.

OSÓRIO, Fernando; BITTENCOURT, João Ricardo. Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processo de imagens. In: **I Workshop de Inteligência Artificial**, Anais, Santa Cruz do Sul, 2000.

OSORIO, Fernando Santos. **Um estudo sobre reconhecimento visual de caracteres através de redes neurais**. Dissertação de Mestrado, CPGCC, UFRGS, Porto Alegre, 1991.

PEARL J, **Bayesian network**, MIT Encyclopedia of the Cognitive Sciences, 1997.

PEARL J. **Probabilistic reasoning in intelligent systems: networks of plausible inference**. Morgan Kaufmann Publishers: San Francisco, 1988.

PHILIPS, S. **Does classicism explain universality?** Mind and Machines, 2002.

PINKER, S., e PRINCE, A. **On language and connectionism: analysis of a parallel distributed processing model of language acquisition**, Cognition, 23. 1988.

RUMELHART, D.; MCCLELLAND, J. **Parallel distributed processing**. Mass MIT Press: Cambridge, 1986. vol I.

SCHREIBER, Jacques N. Carlota. **Análise do tempo de navegação na composição de um modelo para hipermídia adaptativa**. Tese de Doutorado, PPEP, UFSC, Florianópolis, 2003.

SEJNOWSKI, T.; ROSENBERG C. **Parallel networks that learn to pronounce english text**, Complex System, 1987.

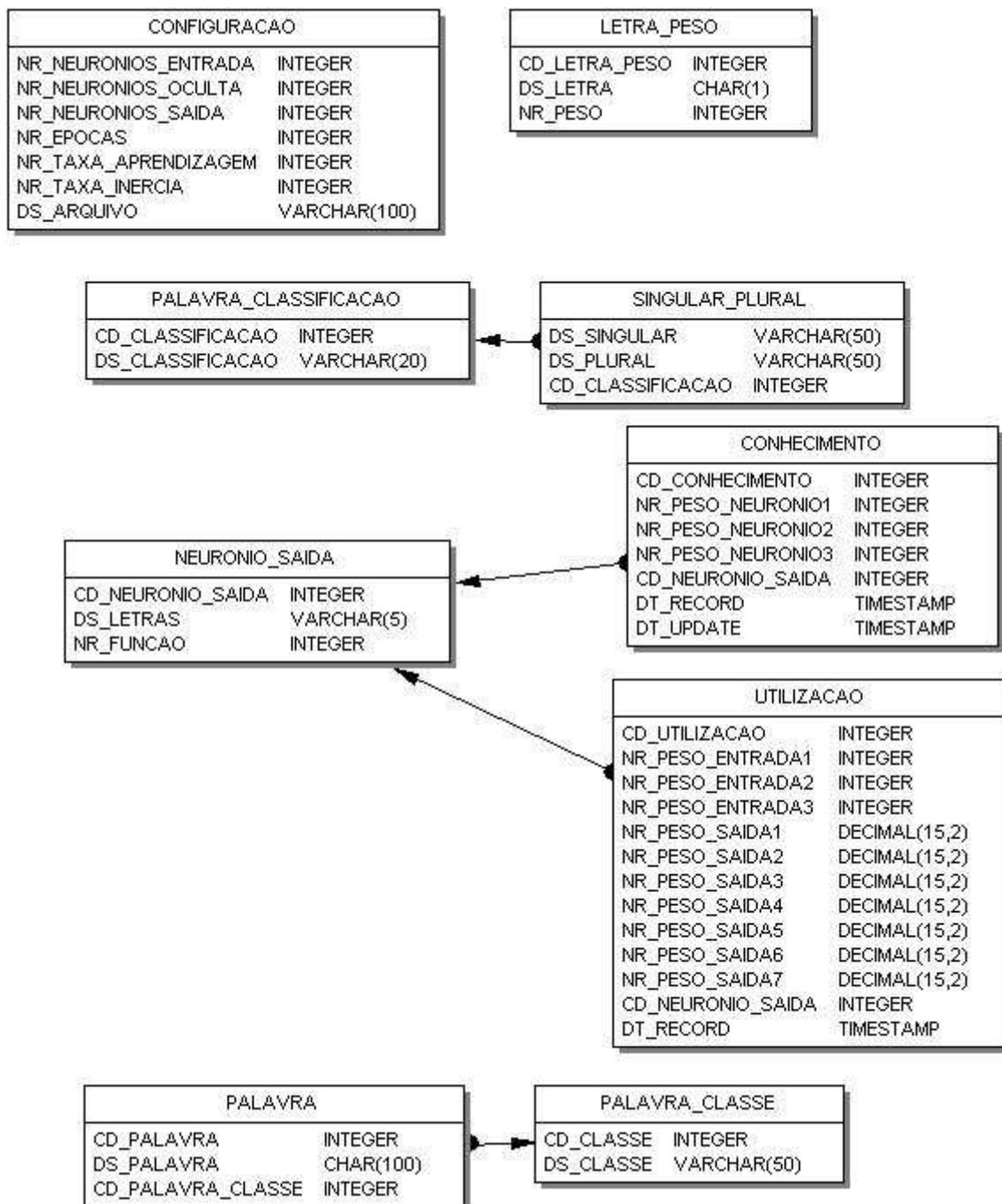
SILVA, Alex Sandro da. **Protótipo de software para classificação de impressão digital**. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 1999. 70 f.

TUFANO, Douglas, **Gramática e Literatura Brasileira: Curso Completo**. Moderna: São Paulo, 1995

VALDAMERI, Alexander R. **Redes neurais aplicadas ao sistema de informações do jogo de empresas virtual**. 1997. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

YONG, C. K. e LIM, C. M.; **An integrated water quality monitoring system using artificial neural networks**. Singapura, 2001. Disponível em: <http://www.np.edu.sg/~yck/nn_waterquality.pdf>. Acesso em: Março de 2008.

APÊNDICE A – Diagrama ER da base de conhecimento



APÊNDICE B – Algoritmos: Separar Sílabas e Identificar a Sílabas Tônicas

```

unit uFuncoes;

interface

uses SysUtils;

function Ilf( Expressao : Variant; ParteTRUE, ParteFALSE : Variant ) : Variant;
function Contem( letra : String; texto : String ) : Boolean;
function Right( Source : String; Lengths : Integer ) : String;
function DescobreTonica( palavra : String ) : Integer;
function Separar_Silabas( _word : String ) : String;

//var

implementation

function Ilf( Expressao : Variant; ParteTRUE, ParteFALSE : Variant ) : Variant;
begin
  if Expressao then
    Result := ParteTRUE
  else
    Result := ParteFALSE;
end;

function Contem( letra : String; texto : String ) : Boolean;
begin
  result := false;

  if (Pos( letra, texto ) > 0) then result := true;
end;

function Right( Source : String; Lengths : Integer ) : String;
begin
  Result := Copy( Source, (Length(Source) - Lengths) + 1, Lengths );
end;

function DescobreTonica( palavra : String ) : Integer;
var
  p      : Integer;
  c      : Char;
  posAcento : Integer;
  estado  : Integer; { Indica estado corrente do diagrama }
const
  vogais : set of char = ['a','e','i','o','u','y','w','ü'];
  acentos : set of char = ['á','â','ã','ä','é','ê','í','ó','ô','õ','ú'];
begin
  descobreTonica := 0;
  palavra        := ansiLowerCase (palavra);
  posAcento      := 0;

  for p := Length(palavra) downto 1 do
  begin
    c := palavra[p];

    if c in acentos then
      begin
        descobreTonica := p;
        Exit;
      end;

    if (posAcento = 0) and (c in vogais) then
      posAcento := p;
    end;

  if posAcento = 0 then Exit; // palavra não tem vogais

  estado := 0;
  p      := length (palavra);

```

```

while p > 0 do
begin
  c := palavra[p];

  case estado of
    0 : case c of
      'a', 'e', 'o':
        begin
          descubreTonica := p;
          estado := 1;
        end;
      'u':
        begin
          descubreTonica := p;
          estado := 10;
        end;
      'i':
        begin
          descubreTonica := p;
          estado := 20;
        end;
      's' : estado := 0;
      'm' : estado := 40;
    else
      estado := 30;
    end;

    {--- palavras com última vogal 'a', 'e', 'o' ---}
    1 : if c = 'u' then
      estado := 4
    else
      if c in vogais then
        begin
          descubreTonica := p;
          p := 0;
        end
      else estado := 2;

    2 : case c of
      'i', 'u':
        begin
          descubreTonica := p;
          estado := 3;
        end;
      'a', 'e', 'o':
        begin
          descubreTonica := p;
          p := 0;
        end;
    end;

    3 : begin
      if c in ['a', 'e', 'o'] then
        begin
          descubreTonica := p;
          p := 0;
        end
      else
        if c = 'u' then
          estado := 5
        else p := 0;
      end;

    4 : begin
      if c in ['g', 'q'] then
        begin
          descubreTonica := p + 2;
          estado := 2;
        end
      else
        begin
          descubreTonica := p + 1;
          p := 0;
        end;
      end;
  end;
end;

```



```

else if Pos(LETRA,'BCDFGHJLMNPQRSTVXZKYW'+Çç')>0 then
  _word2 := _word2 + 'C'
  else _word2 := _word2 + '?';
end;

_point := Pos('VCV',_word2);

while (_point <> 0) do
begin
  _word := Copy(_word ,1,_point)+'-'+RIGHT(_word ,Length(_word )-_point);
  _word2 := Copy(_word2,1,_point)+'-'+RIGHT(_word2,Length(_word2)-_point);
  _point := Pos('VCV',_word2);
end;

_point := Pos('VCCV',_word2);

while _point <> 0 do
begin
  C1 := UPPERCase(Copy(_word,_point+1,1));
  C2 := UPPERCase(Copy(_word,_point+2,1));

  if ( ( not Contem(C1,'JLHMNQRSXZ') ) and Contem(C2,'LR') ) or ( ( Pos(C1,'CLNPST')>0) and (Pos(C2,'H')>0) ) ) then
  begin
    // V-CCV
    _word := Copy(_word ,1,_point)+'-'+RIGHT(_word ,Length(_word )-_point);
    _word2 := Copy(_word2,1,_point)+'-'+RIGHT(_word2,Length(_word2)-_point);
  end
  else
  begin
    // VC-CV
    _word := Copy(_word ,1,_point+1)+'-'+RIGHT(_word ,Length(_word )-( _point+1));
    _word2 := Copy(_word2,1,_point+1)+'-'+RIGHT(_word2,Length(_word2)-( _point+1));
  end;
  _point := Pos('VCCV',_word2);
end;

_point := Pos('VCCCV',_word2);
while _point <> 0 do
begin
  C1 := UPPERCase(Copy(_word,_point+2,1));
  C2 := UPPERCase(Copy(_word,_point+3,1));

  if ( (NOT Contem(C1,'JLHMNQRSXZ') ) and Contem(C2,'LR') ) or ( Contem(C1,'CLNPST') and (C2='H')) then
  begin
    _word := Copy(_word ,1,_point+1)+'-'+RIGHT(_word ,Length(_word )-( _point+1));
    _word2 := Copy(_word2,1,_point+1)+'-'+RIGHT(_word2,Length(_word2)-( _point+1));
  end
  else
  begin
    _word := Copy(_word ,1,_point+2)+'-'+RIGHT(_word ,Length(_word )-( _point+2));
    _word2 := Copy(_word2,1,_point+2)+'-'+RIGHT(_word2,Length(_word2)-( _point+2));
  end;
  _point := Pos('VCCCV',_word2);
end;

_point := Pos('VCCCCV',_word2);
while _point <> 0 do
begin
  _word := Copy(_word ,1,_point+2)+'-'+RIGHT(_word ,Length(_word )-( _point+2));
  _word2 := Copy(_word2,1,_point+2)+'-'+RIGHT(_word2,Length(_word2)-( _point+2));
  _point := Pos('VCCCCV',_word2);
end;

_point := Pos('VV',_word2);
while _point <> 0 do
begin
  if not Pos('!'+UPPERCase(Copy(_word,_point
,2))+'!', '!A!AO!ÃO!ãO!AU!E!EU!OE!ÕE!õE!O!OO!OU!'+iif(_point<>1,iif(Pos(UPPERCase(Copy(_word,_point-
1,1)), 'GQ')>0,'UA!ÛA!üA!UE!ÛE!üE!U!Û!ü!UO!ÛO!üO!UU!ÛU!üU!',','))>0 then
  begin
    _word := Copy(_word ,1,_point)+'-'+RIGHT(_word ,Length(_word )-_point);
    _word2 := Copy(_word2,1,_point)+'-'+RIGHT(_word2,Length(_word2)-_point);
  end;

  if Pos('VV',RIGHT(_word2,Length(_word2)-_point)) <> 0 then

```

```
    _point := _point + Pos('V',RIGHT(_word2,Length(_word2)-_point))  
  else _point := 0;  
end;
```

```
  result := (_word);  
end;
```

```
end.
```