Reconhecimento de Palavras Isoladas utilizando: HMM Discreto e Redes Neurais

Durante o decorrer da disciplina de TP-510 (Processamento de Voz), foram propostos aos alunos a tarefa de implementar um sistema de reconhecimento de palavras isoladas para automação bancária através das seguintes técnicas: HMM (Modelos Ocultos de Markov) e Redes Neurais. A seguir serão apresentadas as especificações utilizadas na implementação de cada sistema e os respectivos resultados.

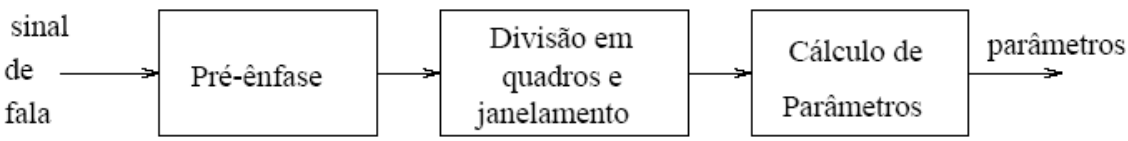
1. HMM Discreto

1.1 Aquisição do Sinal de Voz

A aquisição do sinal de voz é realizada utilizando-se um microfone conectado a uma pequena mesa de som, cuja saída está ligada a uma placa de som instalada em um computador pessoal. A placa de som transforma o sinal analógico em amostras digitais a uma taxa de 8000 amostras por segundo com resolução de 16 bits por amostra.

1.2 Análise do Sinal

A análise do sinal consiste em extrair os parâmetros acústicos da fala (mel_cepstrais, delta-mel-cepstrais e delta-delta-mel-cepstrais), e quantizar vetorialmente estes parâmetros. A seguir serão apresentadas um pouco mais detalhadamente as tarefas que devem ser realizadas para que se possa analisar o sinal de voz. A sequencia de operações necessárias para a obtenção dos parâmetros é mostrado na figura abaixo.



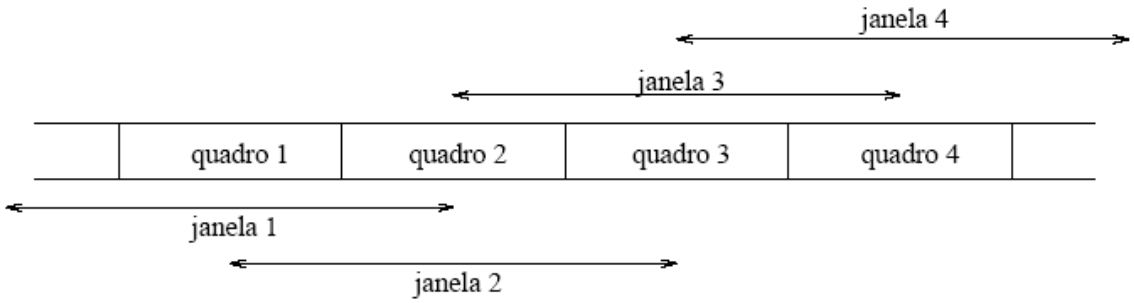
1.2.1 Extração dos Parâmetros

Para a obtenção dos parâmetros acústicos deve-se primeiro pré-enfatizar o sinal, janelá-lo e por último realizar a análise espectral de cada janela.

O sinal é pré-enfatizado através de um filtro FIR passa altas de primeira ordem com função de transferência dada por $1\text{-}0{,}95\,z^{-1}$. O objetivo da pré-ênfase é o de eliminar uma tendência espectral de aproximadamente -6dB/oitava na fala irradiada pelos lábios. Esta distorção espectral não traz informação adicional e portanto pode ser eliminada através do uso do filtro de pré-ênfase, que tem resposta de aproximadamente +6dB/oitava.

Como é sabido o sinal de voz é não estacionário, e portanto é necessário trabalhar com "pequenas porções" ou frames do sinal para que se possa razoavelmente assumir que este frame é razoavelmente estacionário. Aqui definimos frame como sendo a multiplicação de uma janela discreta com as amostras do sinal de voz.

No presente trabalho utilizamos frame de 160 amostras (eqüivale a 20ms da locução) sendo que o deslocamento do frame é de 80 amostras, para obtermos o frame multiplicamos 160 amostras do sinal de voz pela janela de Hamming com tamanho de 160 amostras, faz-se isto para suavizar a amplitude do sinal janelado em seus extremos. A figura abaixo ilustra o processo de superposição de janelas.



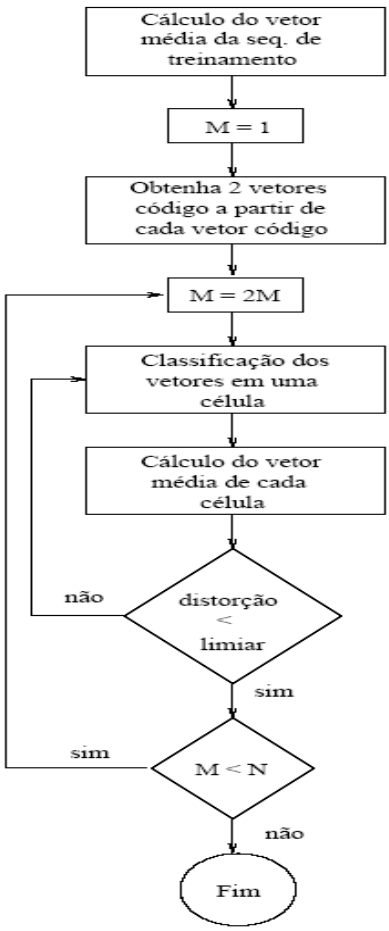
O método de análise espectral utilizado é o de análise espectral por banco de filtros. Inicialmente calcula-se a energia da janela sob análise (utiliza-se o quadrado do modulo da FFT da janela em questão), na sequencia este sinal é filtrado por um banco de filtros triangulares na escala mel, em seguida calcula-se o logaritmo da energia obtida na saída dos filtros e, por fim, calcula-se a IDCT (Inverse Discrete Cosine Transform) destes valores, e assim obtêm-se os parâmetros mel-cepstrais. Nesta implementação foram calculados 12 coeficientes mel-cepstrais por janela.

Além dos parâmetros mel-cepstrais, utilizou-se também os parâmetros diferencias (delta_mel_cepstrais e delta-delta-mel-cepstrais), obtidos através dos parâmetros mel-cepstrais. Os parâmetros diferencias foram usados no intuito de melhorar a caracterização das variações temporais do sinal de fala, ou seja, melhor o desempenho do sistema de reconhecimento de palavras isoladas.

1.2.2 Quantização Vetorial

Esta é a ultima etapa da fase de análise do sinal de voz. A quantização vetorial é realizada em duas fases: uma responsável pela geração do codebook, e outra responsável pela quantização de uma dada locução.

A geração do codebook é feita utilizando-se o algoritmo Lindo-Buzo-Gray (LBG) na versão “splitting”. O algoritmo LBG tem como objetivo gerar um número N de vetores, de forma que estes possam representar uma grande gama de vetores amostra com a menor distorção possível. A figura abaixo apresenta um diagrama do procedimento do algoritmo LBG.



Na fase de quantização vetorial, cada amostra de entrada do quantizador é comparada com cada vetor do codebook usando como medida de distorção a Distancia Euclidiana, o vetor código que resultar em uma menor distorção será escolhido para representar o vetor amostra. Na presente implementação gerou-se um codebook de 256 vetores.

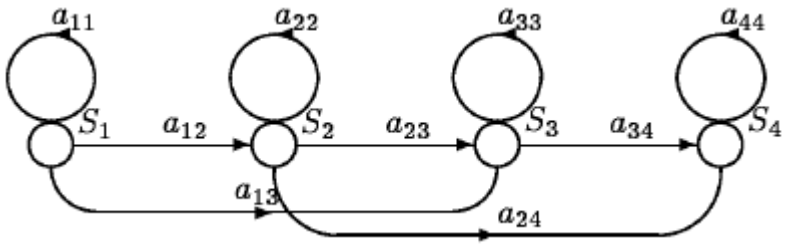
Nesta implementação utilizou-se o método mais simples de quantização vetorial de uma sequencia de vetores, que é o método chamado de *full search*, ele compara cada vetor da sequencia de entrada com todos os codewords armazenados no codebook. O codeword mais "semelhante" é assumido como representante do vetor em questão.

1.3 Treinamento

Neste item descreveremos como serão treinados os modelos HMM´s das palavras que serão utilizadas para o reconhecimento. O treinamento dos modelos é feito em duas partes, na primeira são inicializados os modelos e a segunda consiste propriamente no treinamento do modelo.

1.3.1 Inicialização

A topologia do HMM utilizada é a ‘’left-rigth’’, assim, inicializamos a matriz de transição de estados com probabilidades de transição equiprováveis. O modelo ’left-rigth’’ pode ser visualizado na figura abaixo.



A densidade de probabilidade de emissão de um símbolo num dado estado de um modelo é inicialmente calculada da seguinte forma, divide-se cada locução de treinamento em n partes iguais (de mesmo tamanho), onde n é definido como sendo o número de estados do modelo. Desta forma, cada estado do modelo é associado à n-ésima parte da locução, em seguida, faz-se uma contagem dos símbolos que ocorrem em cada uma das n partes da locução, esta sequencia é realizada para todas as locuções de

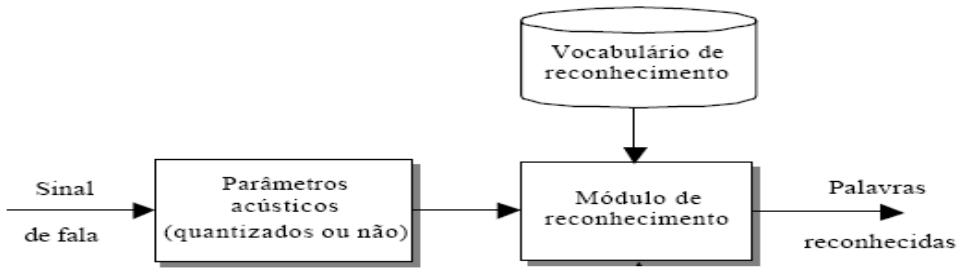
treinamento, após o término da contagem de ocorrências de um símbolo em um dado estado para todas as locuções de treinamento divide-se o número de ocorrências de cada símbolo num dado estado pelo número total de ocorrências de símbolos para aquele estado (soma-se o número de ocorrências de todos os símbolos para um dado estado). Após isto, tem-se o vetor probabilidade de emissão de um símbolo em um dado estado, e assim pode-se iniciar o processo de treinamento dos modelos.

1.3.2 Treinamento

Após a inicialização, vem o treinamento dos modelos, para isto é utilizado o algoritmo de Baum-Welch. Este algoritmo é finalizado quando a diferença relativa entre a probabilidade média da época atual e a probabilidade média da época anterior for menor que 0.001.

1.4 Reconhecimento

Dada uma locução de entrada, o sistema de reconhecimento de fala tem como objetivo gerar uma palavra que mais se assemelhe com a locução de entrada através de um processo de busca. Este processo de busca é realizado comparando-se as probabilidades resultantes da apresentação da locução de entrada a cada um dos modelos das palavras que estão no vocabulário utilizado pelo sistema. O reconhecimento é realizado levando-se em consideração: os parâmetros acústicos das locuções a reconhecer, os modelos HMM já treinados e o vocabulário do sistema. De posse da maior probabilidade decide-se qual foi a locução de entrada e apresenta-se a palavra reconhecida pelo sistema. A figura abaixo representa o sistema de reconhecimento de palavras isoladas.



1.4.1 Vocabulário de Reconhecimento

O vocabulário do sistema de reconhecimento implementado consiste de 11 palavras, que são: atendimento, bloqueio, cartão, cheque, conta, empréstimo, investimento, poupança, saldo, seguro e transferência.

1.5 Avaliação do Sistema

Primeiramente, realizamos testes utilizando somente os parâmetros mel-cepstrais, após concluídos estes testes, realizamos outros, utilizando-se os parâmetros mel-cepstrais, delta-mel-cepstrais e delta-delta-mel-cepstrais. Nestes testes foram utilizadas 220 locuções, pronunciadas por 4 locutores masculinos e 1 feminino, com cada um deles pronunciando 11 palavras distintas, sendo que cada palavra foi repetida 4 vezes. Após os testes realizados obtivemos os seguintes resultados:

Parâmetro	% Erro
mel	2,27
mel, dmel, ddmel	0

Como observado da tabela acima, o conjunto de parâmetros mel, dmel e ddmel apresenta melhores resultados em comparação com os resultados obtidos com a utilização de somente os parâmetros mel. No caso dos parâmetros dmel e ddmel utilizou-se K=5, onde K está associado ao número de quadros adjacentes empregados no cálculo dos parâmetros diferencias. A partir dos resultados obtidos acima e das especificações dadas durante o transcorrer deste trabalho, podemos definir o modelamento final para o HMM discreto.

Especificações do sistema utilizado:

- Aplicação: Sistema Bancário.
- Banco de Dados:

    Locutores: 15 sendo 10 para treinamento e 5 para reconhecimento.

    Palavras:

        Total: 11

- Atendimento
- Bloqueio
- Cartão
- Cheque
- Conta
- Empréstimo
- Investimento
- Poupança
- Saldo
- Seguro
- Transferência

Repetições por palavra: 4 por locutor.

- Parâmetros utilizados: mel, dmel e ddmel (12 coeficientes cada), e K=5.
- Codebook: foi obtido um codebook contendo todos os parâmetro (mel, dmel, ddmel) com 256 vetores.
- Algoritmo de treinamento: Baum-Welch.
- Vocabulário de treinamento: 440 palavras (pronunciadas por 9 homens e 1 mulher).
- Vocabulário de reconhecimento: 220 (4 homens e 1 mulher).

1.6 Bibliografia

YNOGUTI, C. A – "Reconhecimento de Fala Contínua usando Modelos Ocultos de Markov". Tese de Doutorado, UNICAMP, Campinas, Maio 1999.

LINDO Y., BUZO, A., GRAY R. M. – "An Algorithm for Vector Quantizer Design". IEEE Transactions on Communications, COM-28(1), January 1980.

L. R. RABINER and B. H. JUANG - "Fundamentals of Speech Recognition". Englewood Cliffs, Prentice-Hall, 1993.

2. Redes Neurais

O programa utilizado para reconhecimento através do uso de Redes Neurais foi uma alteração na simulação desenvolvida pela aluna de Iniciação Científica, Poliana Magalhães. Inicialmente, o banco de dados utilizado foi de:

- 40 locuções para treinamento (vocabulário “atendimento” e “bloqueio”).
- 20 locuções para reconhecimento.

Foram utilizados os parâmetros mel das locuções “atendimento” e “bloqueio” com entrada do programa.

Especificações do sistema utilizado:

Arquitetura da Rede Neural:

- Número de neurônios da camada de Entrada (nEscondida): 5.
- Número de neurônios da camada de Saída (nSaida): 2.
- Número de sinais de entrada: 720.

Constantes da Rede Neural:

- Constante de não-saturação do neurônios da camada escondida(b1): 0,0002.
- Constante de não-saturação do neurônios da camada de saída(b2): 1.
- Fator de aprendizagem da camada escondida (eta1): 0,05.
- Fator de aprendizagem da camada de saída (eta2): 0,01.
- Número de épocas: 100.

Resultado

- Erro: nulo.
- Número de neurônios saturados: 7 (figura 1).

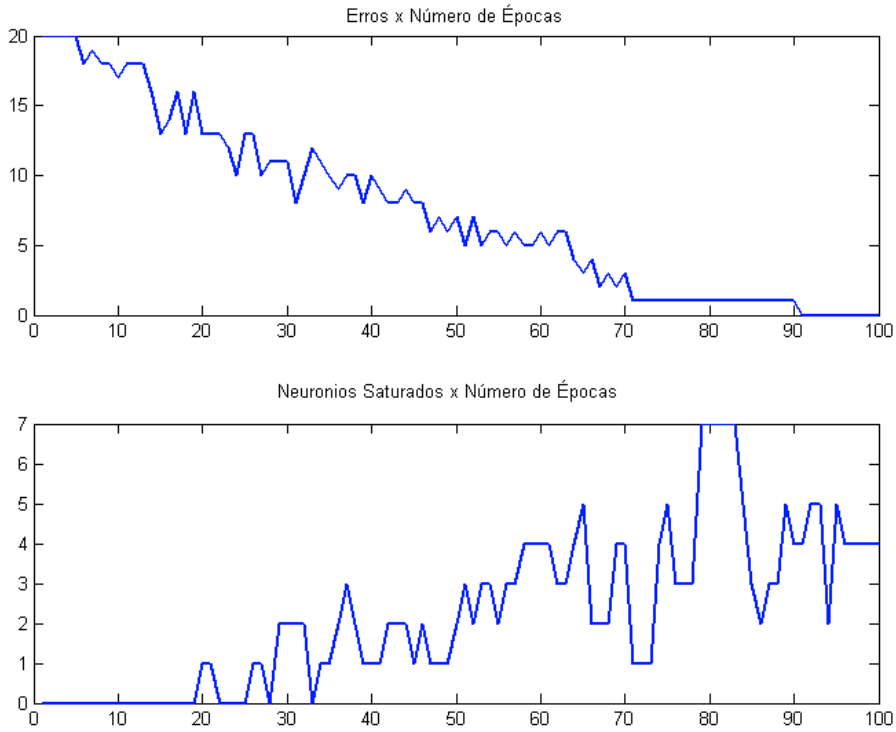


Figura 1 – Erro nulo e 7 neurônios saturados

Utilizou-se posteriormente todo o banco de dados, 440 locuções para treinamento e 220 locuções para reconhecimento) e as seguintes especificações do sistema utilizado:

Arquitetura da Rede Neural:

- Número de neurônios da camada de Entrada (nEscondida): variável (vide tabela 1).
- Número de neurônios da camada de Saída (nSaida): 11.
- Número de sinais de entrada: 720.

Constantes da Rede Neural:

- Constante de não-saturação do neurônios da camada escondida(b1): 0,0002.
- Constante de não-saturação do neurônios da camada de saída(b2): 1.
- Fator de aprendizagem da camada escondida (eta1): 0,2555.
- Fator de aprendizagem da camada de saída (eta2): 0,08888.
- Número de épocas: 50.

Resultados:

nEscondida	Erros	Neurônios Saturados	Figura
50	45	6	2
60	44	6	3
70	41	6	4
80	45	6	5
90	41	5	6
110	41	5	7
150	40	5	8
210	45	6	9

Tabela 1

Alterando *b1* e *b2* pode-se modificar o número de neurônios saturados (terceira coluna da tabela 1). Modificando o valor dos fatores de aprendizagem (eta1 e eta2) na faixa de variação de 0,05 a 0,8, pode-se minimizar os erros, já que este possui uma relação de compromisso com os mesmos.

Assim, ocorrerá um valor ótimo de eta1 e eta2 onde ter-se-á erro nulo. Observa-se que mesmo com erro nulo pode-se ter neurônios saturados (figura 1).

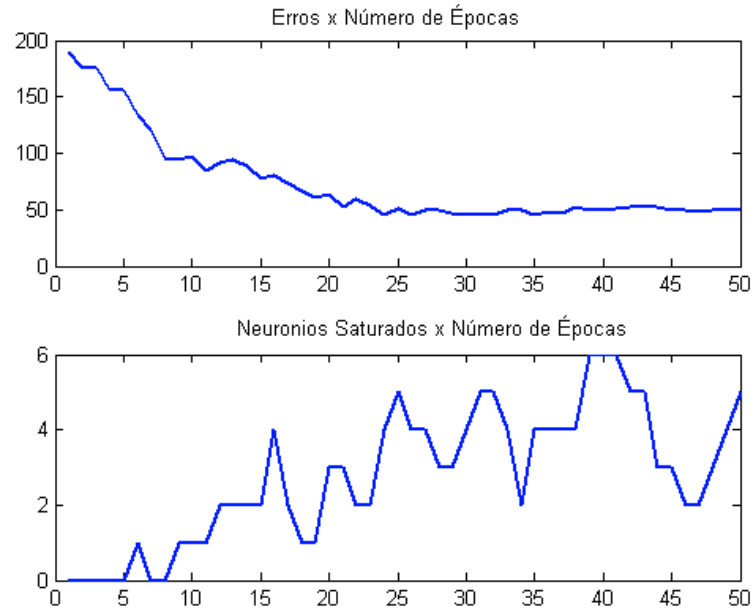


Figura 2 – 45 erros e 6 neurônios saturados

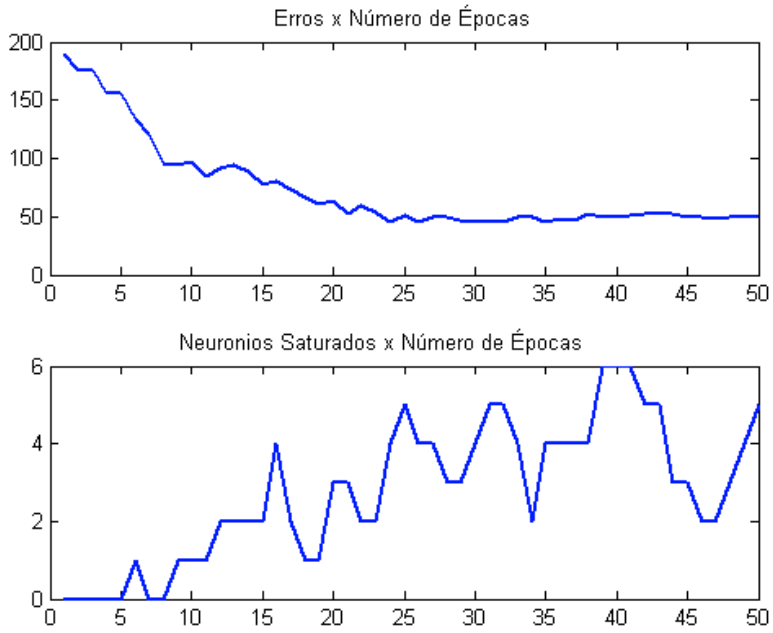


Figura 3 – 44 erros e 6 neurônios saturados

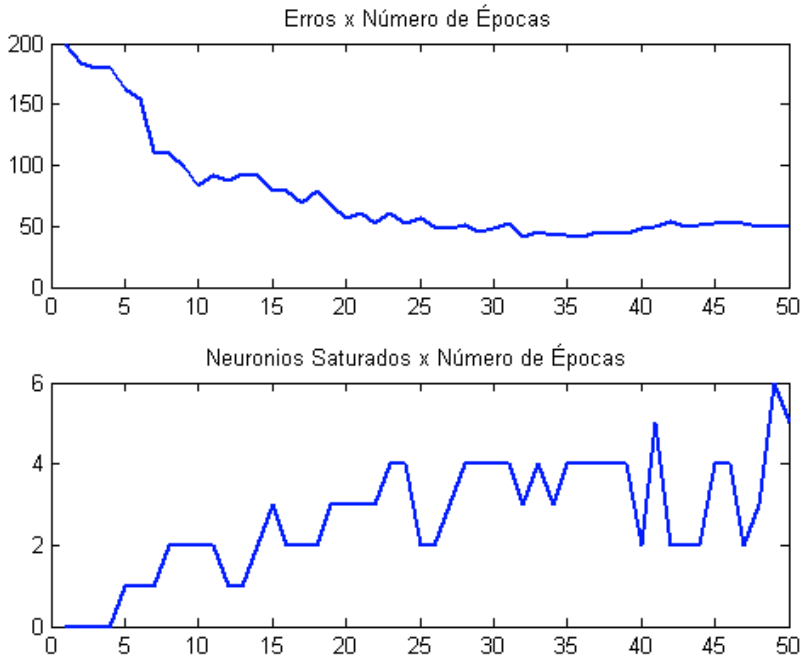


Figura 4 – 41 erros e 6 neurônios saturados

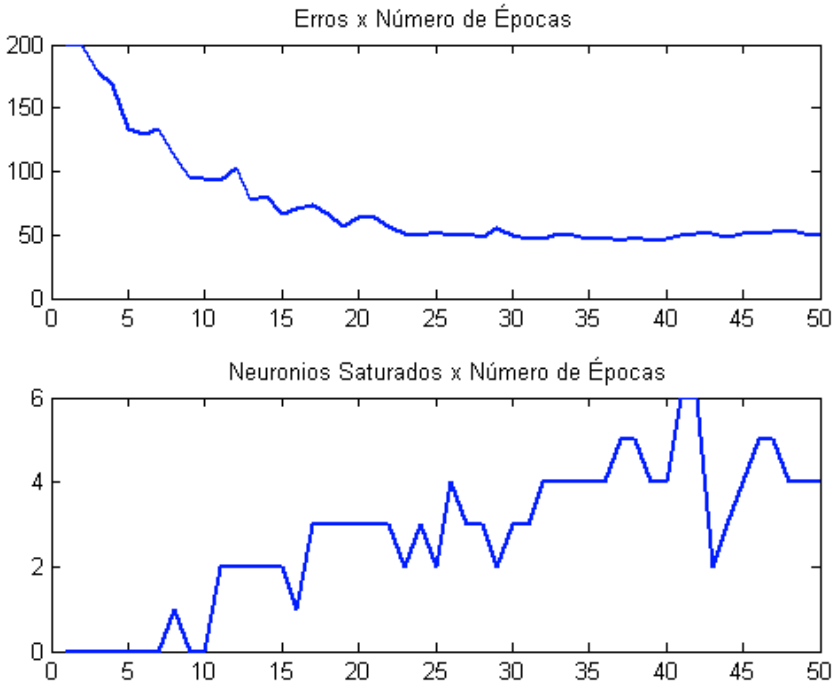


Figura 5 – 45 erros e 6 neurônios saturados

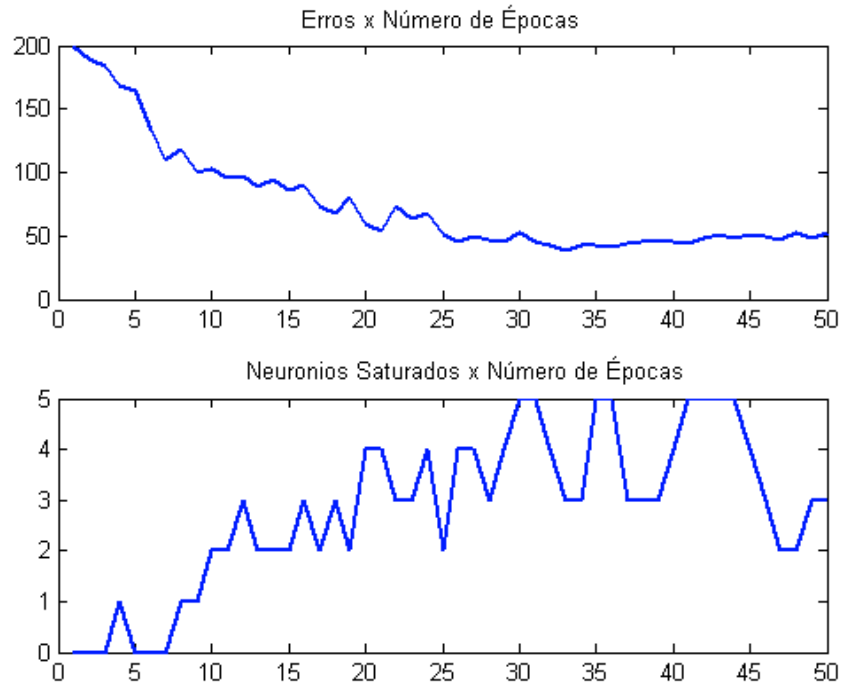


Figura 6 – 41 erros e 5 neurônios saturados

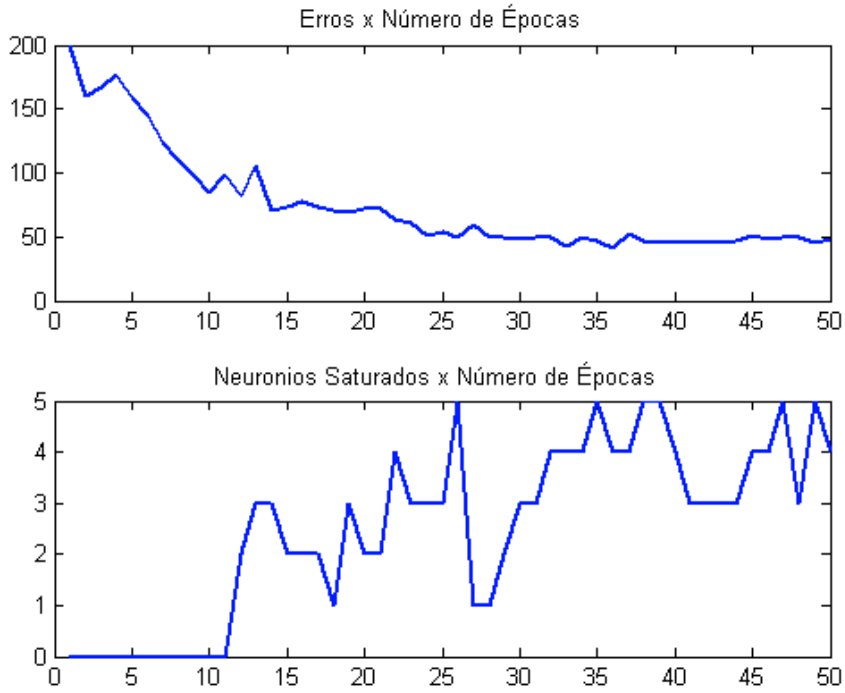


Figura 7 – 41 erros e 5 neurônios saturados

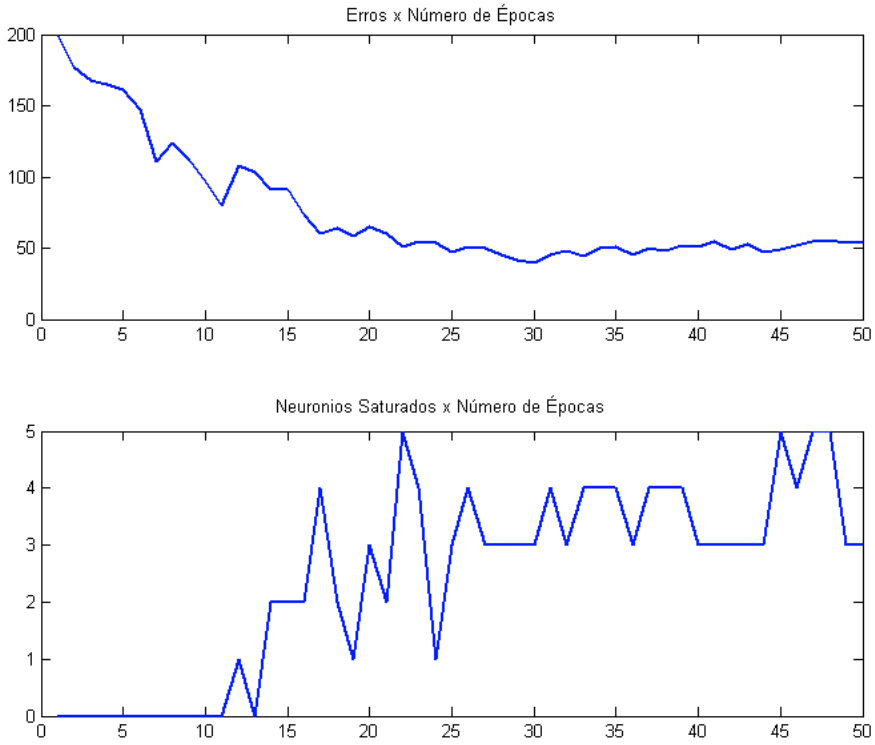


Figura 8 – 40 erros e 5 neurônios saturados

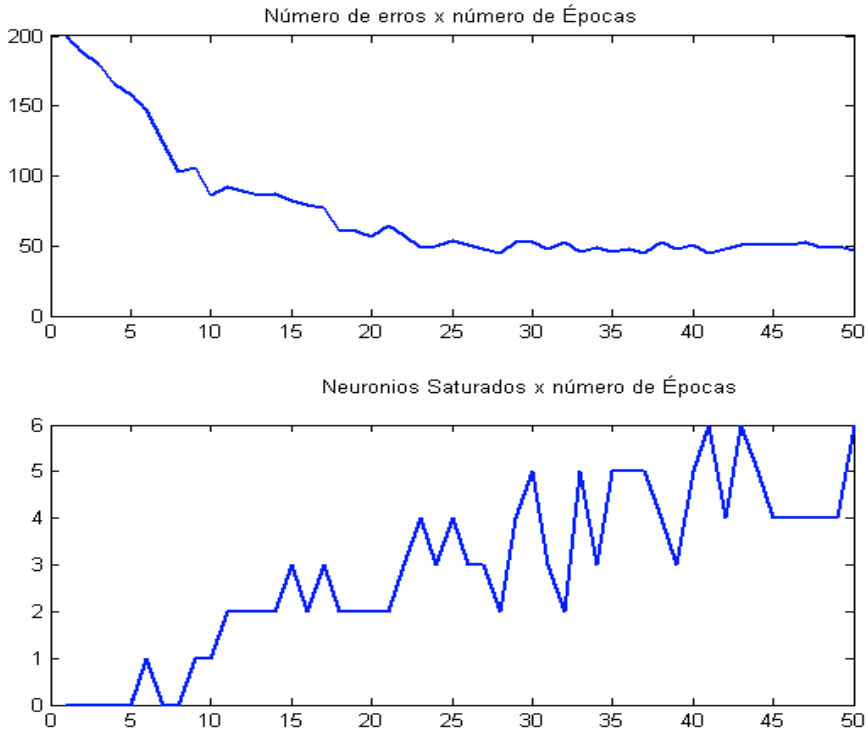


Figura 9 – 45 erros e 6 neurônios saturados