

# Técnicas de Redes Neurais para o Problema de Reconhecimento de Caracteres: Um Estudo Comparativo

Marcelo Corrêa Horewicz, Cairo Lúcio Nascimento Jr. e Thiago Marques Silva

Instituto Tecnológico da Aeronáutica – Praça Marechal Eduardo Gomes, 50 - Vila das Acácias. CEP 12228-900 – São José dos Campos – SP.

**Resumo** — Este artigo apresenta o resultado da comparação do desempenho de dois tipos de classificadores neurais com aprendizagem supervisionada para o problema de reconhecimento de caracteres. As técnicas de redes neurais utilizadas foram: rede neural tipo *feedforward Perceptron* Multicamada (MLP) com 1 camada escondida sendo treinada com o algoritmo *backpropagation* e a rede *Support Vector Machines* (SVM). Os classificadores foram implementados no MATLAB 6.5.1, e como medida de desempenho foi utilizada a taxa de acertos da classificação na fase de testes. Para este caso específico o SVM apresentou melhor resultado.

**Palavras-chaves** — Inteligência Artificial, Redes Neurais, Classificação de padrões, MLP, SVM.

## I. INTRODUÇÃO

A discriminação de dados em categorias e classes é denominada de classificação. No caso em que tem-se apenas duas classes, o problema de classificação é conhecido como binário. No entanto, vários problemas reais envolvem a discriminação de mais do que duas classes, sendo referenciados como problemas multiclasse.

Nos problemas de classificação, o principal objetivo é desenvolver um classificador, a partir de um conjunto de dados conhecidos, que seja capaz de determinar corretamente a classe de novos exemplos não conhecidos a priori.

Uma rede neural artificial (RNA) é um modelo computacional abstrato do cérebro humano. Análogo ao cérebro, uma RNA é composta de neurônios artificiais, chamadas de unidades de processamento, e uma grande interconexão entre estas unidades. Estas redes possuem a habilidade de aprender com exemplos e a generalizar, ou seja, produzir saídas razoáveis para novas entradas que não foram utilizadas no processo de treinamento. As RNAs possuem alguns aspectos que o diferenciam de outras estruturas computacionais que são: aprendizagem por exemplos, habilidade de generalização, a não-linearidade de unidades de processamento, adaptação a novas situações, interconexões paralelas e tolerância a falhas.

No problema de classificação de padrões, os parâmetros considerados relevantes ao problema são representados numericamente e fornecidos como entrada à camada de entrada de uma rede (que pode ou não ser neural).

Estes valores numéricos são processados nas diversas camadas intermediárias (ou ocultas) utilizando-se, para isto, regras de combinação e de ativação. Na camada de saída é gerado o valor  $out_i$  proveniente da função de saída  $F$  que se refere ao valor da classificação encontrada. Em uma rede com aprendizagem supervisionada este valor encontrado é comparado com o valor desejado e, desta forma, é calculado um erro de estimação, que normalmente é feito ser quadrático. Assim, os diversos pesos da rede (que são iniciados com valores pré-estabelecidos) são atualizados através de uma regra de aprendizagem.

Neste artigo é apresentado o resultado da comparação do desempenho de dois tipos de classificadores neurais com aprendizagem supervisionada para o problema de reconhecimento de 16 caracteres (0-9, A-F) tipo BITMAP em grade  $7 \times 7$ , caracterizando um problema multiclasse.

Nas seções II e III são apresentados os classificadores neurais utilizados neste trabalho: a rede neural tipo *feedforward Perceptron* Multicamada (MLP) com 1 camada escondida sendo treinada com o algoritmo *backpropagation* e a rede *Support Vector Machines* (SVM) utilizando a estratégia decomposicional. Esta estratégia decompõe o problema multiclasse em subproblemas binários.

Na seção IV são mostrados os experimentos e resultados numéricos alcançados e, por fim são relatadas algumas conclusões do trabalho.

Como exemplos de comparação do desempenho destes classificadores, em [5] eles são comparados para o problema de classificação de duas espirais, e em [6] para a verificação de faces.

## II. REDE FEEDFORWARD PERCEPTRON MULTICAMADAS (MLP)

As redes neurais podem ser ditas como sendo aproximadores universais de dados medidos num espaço multidimensional. Eles realizam dois tipos de aproximação: global e local. O exemplo mais importante de rede global é o *perceptron multicamada* (MLP) empregando funções de ativação sigmoidais para a ativação dos neurônios[5].

Em uma RNA tipo *feedforward* uma unidade de processamento envia sua saída apenas para as unidades das quais ela não recebe nenhuma entrada direta ou indiretamente, implementando um mapeamento estático do seu espaço de entrada para o seu espaço de saída [1].

Os neurônios em uma rede MLP são dispostos em camadas, contando da camada de entrada (o conjunto de nós

de entrada), passando pelas camadas escondidas, até chegar na camada de saída. Ela é uma rede neural global com função de ativação sigmoideal dos neurônios.

O algoritmo mais comumente utilizado para atualizar os pesos em uma MLP é o *backpropagation* que utiliza a regra delta generalizada cuja função de custo a ser minimizada é uma função de erro definida pela soma dos erros quadráticos[2]:

$$E = \frac{1}{2} \sum_p \sum_{i=1}^k (y_i^p - out_i^p)^2 \quad (1)$$

onde E é a medida do erro total, p é o número de padrões, k é o número de unidades de saída,  $y_i$  é a i-ésima saída desejada e  $out_i$  é a i-ésima saída gerada pela rede. Esta equação define o erro total cometido pela rede, ou a quantidade em que, para todos os padrões p de um dado conjunto, as saídas geradas pela rede diferem das saídas desejadas.

A regra delta generalizada requer que as funções de ativação utilizadas pelas unidades de processamento sejam contínuas, diferenciáveis e, geralmente, não-decrescentes da entrada total recebida pelo neurônio[2].

A unidade do tipo sigmóide ou tangente hiperbólica é o mais comumente utilizado em aplicações de redes neurais por atenderem aos requisitos da regra delta generalizada.

Pela Fig.1, tem-se que a regra de combinação comumente utilizada é:

$$net_i = \sum_j w_{ij} x_j + bias_i \quad (2)$$

$w_{ij}$  é o peso vindo da unidade j para a unidade i;

$x_j$  é o valor da entrada j;

$bias$  ou valor do limiar é um outro peso vindo de uma unidade cuja saída é sempre 1.

Com isto para a unidade tipo sigmóide temos:

$$out_i = sig(net_i) = \frac{1}{1 + e^{(-net_i)}} \quad (3)$$

sendo sig(.) a função *signal*.

E para a unidade tipo tangente hiperbólica temos:

$$out_i = \tanh(net_i) = 2sig(2net_i) - 1 \quad (4)$$

A saída da unidade tipo sigmóide varia entre 0 e 1, enquanto que a saída da unidade tipo tangente hiperbólica varia entre -1 e 1.

A regra delta generalizada sugere que a variação dos pesos seja definida de acordo com o gradiente descendente do erro com relação ao peso[2]:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j(t) x_i(t) \quad (5)$$

sendo:

$$\delta_j = (d_j - out_j) f'(net_j) \quad (6)$$

o erro se a unidade de processamento j estiver na camada de saída.

$$\delta_j = f'(net_j) \sum_i \partial_i w_{ij} \quad (7)$$

o erro se a unidade de processamento j estiver em uma das camadas intermediárias.

$\eta$  é a taxa de aprendizado da rede.

### III. SUPPORT VECTOR MACHINES (SVM)

O mais representativo exemplo de rede neural local é o *Support Vector Machine* (SVM). É uma rede neural com duas camadas sendo a primeira uma camada escondida de unidades radiais e, uma segunda com um neurônio de saída. A arquitetura e a aprendizagem dos parâmetros da rede são tratados com funções de kernel em vez de processamento direto de sinais das camadas escondidas como no MLP [5].

O SVM é uma máquina linear que trabalha num espaço de características de alta dimensão formada pelo mapeamento não-linear de um vetor de entrada de dimensão N em um espaço de características de dimensão K, sendo  $K > N$ , através de um mapeamento  $\Phi(x)$  [3,4].

As SVMs são baseadas na teoria de aprendizado estatístico introduzido por Vapnik [7]. Ele implementa um mapeamento não-linear executado por um produto interno kernel (polinomial, gaussiano, sigmoideal, entre outros) escolhido a priori dos dados de entrada para um espaço característico de alta-dimensão, em que um hiperplano ótimo é construído para separar os dados linearmente em duas classes. Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço característico é aquele que apresenta a máxima margem de separação. Para dados de treinamento em que as amostras das diversas classes apresentam superposição (dados não separáveis), uma generalização deste conceito é utilizada, através de um tratamento estatístico.

Embora o SVM separe os dados linearmente em duas classes, o reconhecimento de mais do que duas classes é possível utilizando a estratégia decomposicional que decompõe o problema multiclasse em subproblemas binários. Existem várias técnicas que são utilizadas para resolver problemas de multiclases, as mais conhecidas são a “one against all” e “one against one”.

No decorrer desta seção, primeiramente será mostrada a utilização de SVMs na obtenção de fronteiras lineares (SVMs lineares) para a separação de dados pertencentes a duas classes, posteriormente será apresentada uma extensão para fronteiras não-lineares (SVMs não-lineares) e, por fim, será apresentada a técnica para problemas multiclases utilizada neste trabalho que foi a “one against one”.

#### A. SVMs Lineares

Os SVMs lineares definem fronteiras lineares sobre conjuntos de dados gerais para a separação de dados pertencentes a duas classes, caracterizando um problema de classe binária.

Quando o aprendizado supervisionado é aplicado ao problema de classificação, as amostras de treinamento são formadas pelo conjunto de dados de entrada associados às suas correspondentes respostas pré-classificadas. Após o

treinamento, o objetivo é classificar novas amostras, ainda não rotuladas.

Considere o conjunto de dados  $T$  de treinamento [3]:

$$(x_i, y_i) \quad 1 \leq i \leq N, x_i \in \mathbb{R}^m, y_i \in \{+1, -1\} \quad (8)$$

onde  $x_i$  é o dado de entrada para a amostra  $i$  e  $y_i$  é a correspondente resposta desejada.

$T$  é linearmente separável se é possível separar os dados das classes  $+1$  e  $-1$  por um hiperplano.

Classificações binárias são frequentemente realizadas pelo uso de funções  $g: x \in \mathbb{R}^m \rightarrow \mathbb{R}$  com a seguinte estratégia: as amostras são designadas para a classe positiva, se  $g(x) \geq 0$ , e caso contrário, para a classe negativa.

A superfície de decisão será representada por um hiperplano na forma:

$$g(x) = (w^T x) + b = 0, \quad (9)$$

onde  $w \in \mathbb{R}^m$  é o vetor de pesos, e  $b \in \mathbb{R}$  é o bias.

Assim podemos aplicar a seguinte estratégia de decisão:

$$(w^T x_i) + b \geq 0 \quad \text{para } y = +1 \quad (10.a)$$

$$(w^T x_i) + b < 0 \quad \text{para } y = -1 \quad (10.b)$$

Para descrever o lugar geométrico dos hiperplanos separadores será utilizada a seguinte forma canônica onde o vetor  $w$  e o escalar  $b$  são re-escalados de tal maneira a atender as desigualdades:

$$(w^T x_i) + b \geq +1 \quad \text{para } y = +1 \quad (11.a)$$

$$(w^T x_i) + b \leq -1 \quad \text{para } y = -1 \quad (11.b)$$

Ou de forma mais compacta:

$$y_i [(w^T x_i) + b] \geq 1, \forall (x_i, y_i) \in T \quad (12)$$

Para um dado vetor de pesos  $w$  e bias  $b$ , a separação entre o hiperplano  $g(x) = (w^T x_i) + b = 0$  e o dado de entrada mais perto é chamada de margem de separação denotada por  $\rho$ . Sempre que for possível obter um  $\rho > 0$  existirão infinitos hiperplanos, dentre os quais se busca um hiperplano particular em que a margem de separação  $\rho$  é maximizada.

De acordo com esta condição, a superfície de decisão é dita ser o hiperplano, sendo que os dados de treinamento que se encontram à distância  $\rho$  do hiperplano ótimo são chamados de vetores de suporte (*support vectors*).

Os vetores de suporte exercem um papel importante nas operações deste tipo de aprendizagem de máquina. Em termos conceituais, eles são os pontos que se encontram mais perto da superfície de decisão e, portanto, são os de classificação mais difícil. Como tal, eles têm uma relação direta com a localização da superfície de decisão.

O hiperplano ótimo definido por (12), apresenta um vetor de pesos  $w$  que leva à máxima separação entre as amostras positivas e negativas. Esta condição ótima é alcançada minimizando a norma euclidiana do vetor de pesos  $w$  levando ao seguinte problema de otimização[3]:

$$\text{Minimizar} \quad Q(w, b, \xi) = \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^N \xi_i \right) \quad (13)$$

com as restrições:

$$y_i [(w^T x) + b] \geq 1 - \xi_i \quad \xi_i \geq 0, \forall i=1, \dots, N \in T \quad (14)$$

A constante  $C$  é um termo de regularização que impõe um peso à minimização da complexidade do modelo.  $\xi_i$  são variáveis de folga utilizados para suavizar as margens do classificador linear, permitindo que alguns dados permaneçam entre os hiperplanos e também que ocorra alguns erros de classificação para evitar o superajustamento (*overfitting*).

O problema de otimização obtido é quadrático e pode ser solucionado com a introdução de uma função Lagrangiana como os multiplicadores de Lagrange  $\alpha_i$ .

Desta forma é possível obter o seguinte problema de otimização conhecido como forma dual:

$$\text{Maximizar} \quad Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (15)$$

$$\text{Com as restrições:} \begin{cases} 0 \leq \alpha_i \leq C, & \forall i = 1, \dots, N \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (16)$$

A Fig. 1. apresenta uma visão geométrica da construção do hiperplano ótimo para um espaço bi-dimensional, além da interpretação dos vetores-suporte.

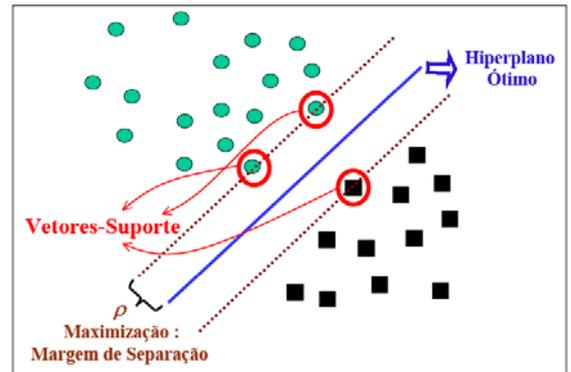


Fig. 1. O hiperplano ótimo separando os dados com a máxima margem  $\rho$ , os vetores-suporte (support vectors) e uma distribuição dos dados no  $\mathbb{R}^2$ .

## B. SVMs Não Lineares

Na maioria dos casos reais não é possível separar os dados de treinamento por um hiperplano. Nestes casos é preciso realizar um mapeamento de seu espaço original das entradas para um espaço de maior dimensão denominado de espaço de características (*feature space*) [3].

Seja  $\Phi: X \rightarrow \Gamma$  um mapeamento em que  $X$  é o espaço de entradas e  $\Gamma$  é o espaço de características.  $\Phi$  é determinante para que o conjunto de treinamento em  $\Gamma$  possa ser separado linearmente, podendo ser computacionalmente bastante custosa dependendo da dimensão de  $\Gamma$ .

Todas as operações de aprendizagem, treinamento e teste utilizando funções de kernel no SVM satisfazem as condições de Mercer[7]. A função kernel é utilizada no SVM não linear

para o cálculo de produtos escalares entre dados no espaço de características:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (17)$$

As funções de kernel mais comumente utilizados são: Polinomiais, RBF (*Radial Basis Function*) ou Gaussianos e os Sigmoidais.

Neste trabalho foi utilizada a função de kernel RBF com parâmetro  $\sigma$ :

$$K(x, x_i) = \exp\left(\frac{-\|x - x_i\|^2}{\sigma^2}\right) \quad (18)$$

Desta forma pode-se alterar (15) para o caso não linear:

$$\text{Maximizar } Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \quad (19)$$

com as mesmas restrições de (16).

A solução de (19) leva ao vetor de pesos ótimo e a saída  $y(x)$  da rede SVM[5]:

$$w_{opt} = \sum_{i=1}^{N_s} \alpha_{oi} y_{oi} \Phi(x_{oi}) \quad (20)$$

$$y(x) = \sum_{i=1}^{N_s} \alpha_{oi} y_{oi} k(x_{oi}, x) + b \quad (21)$$

sendo  $N_s$  o número de vetores de suporte. O valor de  $y(x)$  maior do que 0 é associado com 1 (membro de uma classe particular) e menor do que 0 com -1 (membro da classe oposta).

### C. Algoritmo de Decomposição *One-Against-One*

Suponha uma entrada  $\Gamma_{XY} = \{(x_1, y_1), \dots, (x_l, y_l)\}$  de vetores de treinamento  $x_i \in X \subset \mathbb{R}^n$  e correspondentes saídas  $y_i \in Y = \{1, 2, \dots, c\}$ . O objetivo é treinar as regras de múltipla classificação (*multi-class*)  $q: X \subset \mathbb{R}^n \rightarrow Y$  baseado na estratégia do voto majoritário. A decomposição OAO transforma o problema de múltiplas classificações em séries  $g = (c-1)/2$  de subtarefas que podem ser treinadas por um SVM binário.

Define-se então o conjunto  $\Gamma^j_{XY} = \{(x'_1, y'_1), \dots, (x'_l, y'_l)\}$  contendo os vetores de treinamento  $x_i \in \Gamma^j = \{i: y_i = y^1 \vee y_i = y^2\}$  e as saídas modificadas como:

$$y'_i = \begin{cases} 1 & \text{para } y_j^1 = y_i, \quad i \in \Gamma^j \\ 2 & \text{para } y_j^2 = y_i, \end{cases} \quad (22)$$

O conjunto  $\Gamma^j_{XY}$ ,  $j = 1, \dots, g$  é construído para todas  $g = (c-1)/2$  combinações de classes  $y^1_j \in Y$  e  $y^2_j \in Y \setminus \{y^1_j\}$ . O SVM binário  $q_j$ ,  $j = 1, \dots, g$  são treinados nos  $\Gamma^j_{XY}$ .

## IV. EXPERIMENTOS E RESULTADOS

Neste trabalho foi realizada uma comparação quantitativa de desempenho do MLP e do SVM para o problema de reconhecimento de imagens ruidosas de 16 caracteres (0-9, A-F) tipo BITMAP em grade 7x7. A implementação dos classificadores foi realizada no MATLAB 6.5.1.

O conjunto de padrões utilizados para a validação da técnica são os 16 ilustrados na Fig. 2.

Dividiu-se a etapa de simulações em duas fases: fase de treinamento e fase de testes.

Na fase de treinamento cada *pixel* de cada imagem foi considerada como uma entrada, resultando em conjuntos de imagens de 49 pixels (7x7) como entrada para o treinamento da rede. Como a rede foi utilizada para reconhecer imagens ruidosas, ela foi treinada com versões ruidosas da imagem original, as quais foram apresentadas aleatoriamente ordenadas. O ruído utilizado foi do tipo flip-flop, ou seja, associa-se uma probabilidade de inversão para cada pixel da imagem.

Ao fim do treinamento as redes devem ser capazes de associar a cada padrão de entrada uma unidade de saída.

Depois que as redes foram treinadas passou-se para a fase de testes, onde conjuntos de dados de entrada que não foram treinados foram apresentados às redes. A taxa de acertos da classificação mostra o desempenho das redes e sua capacidade de generalização.

### A. Simulação com a Rede MLP

Para o conjunto de dados de treinamento a rede neural artificial tipo *feedforward* MLP com 1 camada escondida treinada com o algoritmo *backpropagation* foi configurada da seguinte forma: a camada de entrada possuiu 49 unidades, pois cada *pixel* de cada padrão foi considerada como uma entrada (7x7); a camada escondida foi composta de 8 unidades; e a camada de saída 16 unidades, correspondendo ao número de classes dos padrões de entrada, sendo que somente uma unidade de saída deve ser ativada de cada vez, indicando o padrão de entrada classificado. A função de ativação das camadas escondida e de saída escolhida foi a tangente hiperbólica.

Assim, as entradas da rede foram as linhas concatenadas do próprio bitmap (na forma bipolar: -1 ou 1) que se desejava reconhecer. A rede foi treinada de modo que cada unidade de saída codificasse uma imagem armazenada. A camada escondida codifica de modo aleatório os mesmos padrões de entrada.

A inicialização da matriz de pesos foi aleatória. O algoritmo *backpropagation* utilizou a mesma taxa de aprendizado constante para as unidades de saída e da camada escondida.

Ao fim do treinamento a rede foi capaz de associar a cada padrão de entrada uma unidade de saída; para fazê-lo, considerou-se um valor de limiar (0.35) acima do qual deve estar a saída da unidade selecionada. Os casos de imagens mal reconhecidas foram aqueles em que não houve apenas uma unidade com saída acima do valor de limiar.

O treinamento se realizou em 200 iterações com 30 épocas totalizando 6000 amostras de treinamento.

As taxas de acerto foram calculadas sobre 100 amostras ruidosas para cada um dos 16 padrões, ou seja em 1600 amostras de testes diferentes dos treinados.

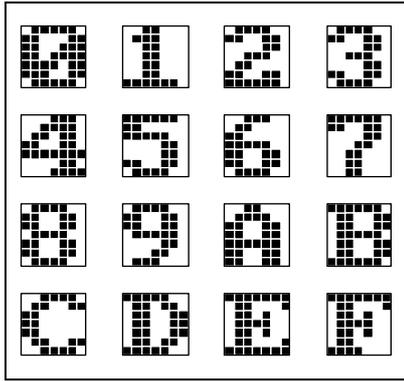


Fig. 2. 16 caracteres (0-9, A-F) tipo BITMAP em grade 7X7.

A Tabela I mostra o resultado da taxa de acerto da rede MLP treinada em função do nível de ruído. O nível de ruído foi alterado entre 0 e 50% com passo de 5%.

TABELA I TAXA DE ACERTOS DA REDE FEEDFORWARD MLP

Ruído (%)	Erro Total	Acerto Total	Taxa acerto (%)
0	0	1600	100
5	34	1566	97.8750
10	185	1415	88.4375
15	473	1127	<b>70.4375</b>
20	738	862	53.8750
25	978	622	38.8750
30	1218	382	23.8750
35	1343	257	16.0625
40	1486	114	7.1250
45	1531	69	4.3125
50	1573	27	1.6875

### B. Simulação com a Rede SVM

O algoritmo *One-Against-One*, do *Statistical Pattern Recognition Toolbox* do MATLAB[4], foi utilizado para implementar o treinamento do SVM neste trabalho. A função Kernel utilizada foi a gaussiana ( $\sigma = 1$ ) e o valor da constante C foi 10. Foram realizados 100 testes em cada padrão de treinamento e verificado quantas vezes o SVM acertou corretamente a classificação da imagem ruidosa fornecida.

A Tabela II mostra o resultado da taxa de acerto da rede SVM treinada em função do nível de ruído. O nível de ruído foi alterado entre 0 e 50% com passo de 5%.

TABELA II TAXA DE ACERTOS DA REDE SVM

Ruído (%)	Erro Total	Acerto Total	Taxa acerto (%)
0	0	1600	100
5	3	1597	99.8125
10	9	1591	99.4375
15	36	1564	97.7500
20	107	1493	93.3125
25	223	1377	86.0625
30	476	1124	<b>70.2500</b>
35	796	804	50.2500
40	1087	513	32.0625
45	1336	264	16.5000
50	1499	101	6.3125

A Fig. 3. apresenta o teste de generalização, comparando o desempenho duas redes neurais estudadas através da taxa de acertos de classificação na fase de testes.

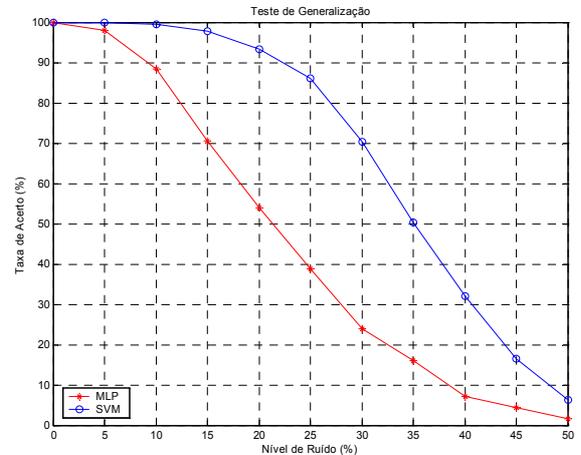


Fig. 3. Teste de Generalização dos classificadores.

## V. CONCLUSÕES

O experimento numérico mostrou que tanto o classificador neural MLP quanto o classificador neural SVM tiveram uma boa capacidade de generalização para baixos níveis de ruído (acima de 15%), num problema de classificação multiclasse.

No entanto, para o problema de classificação de caracteres ruidosos apresentados neste trabalho, o classificador SVM utilizando a estratégia *one against one* se mostrou bastante superior à rede MLP, tendo altas taxas de acerto mesmo para níveis de ruído de 30%.

## REFERÊNCIAS

- [1] Nascimento Júnior, Cairo Lúcio; Yoneyama, Takashi. "Inteligência Artificial em Controle e Automação", Editora Wdguard Blücher Ltda, 2000.
- [2] Braga, Antônio de Pádua; Ludermir, Teresa Bernarda; Carvalho, André Carlos Ponce de Leon Ferreira. Redes Neurais Artificiais - Teoria e aplicações, LTC – Livros Técnicos e Científicos Editora S.A., 2000.
- [3] Burges, Christopher J.C., A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2, 121-167 (1998).
- [4] Franc, V., Hlavac, V., "Statistical Pattern Recognition Toolbox User's Guide", 24 Junho, 2004.
- [5] Osowski, Stanislaw; Siwek, Krzysztof; Markiewicz, Tomasz. MLP and SVM Networks – a Comparative Study, Proceedings of the 6<sup>th</sup> Nordic Signal Processing Symposium – NORSIG 2004, June 9-11, 2004. Espoo, Finland.
- [6] Cardinaux, Fabien; Marcel, Sébastien. Face Verification using MLP and SVM, IDIAP Research Report, May 2002. Martigny, Valais, Switzerland.
- [7] Vapnik, V. Statistical Learning Theory, Wiley, N.Y., 1998.