

Introdução ao uso de redes neurais com Matlab

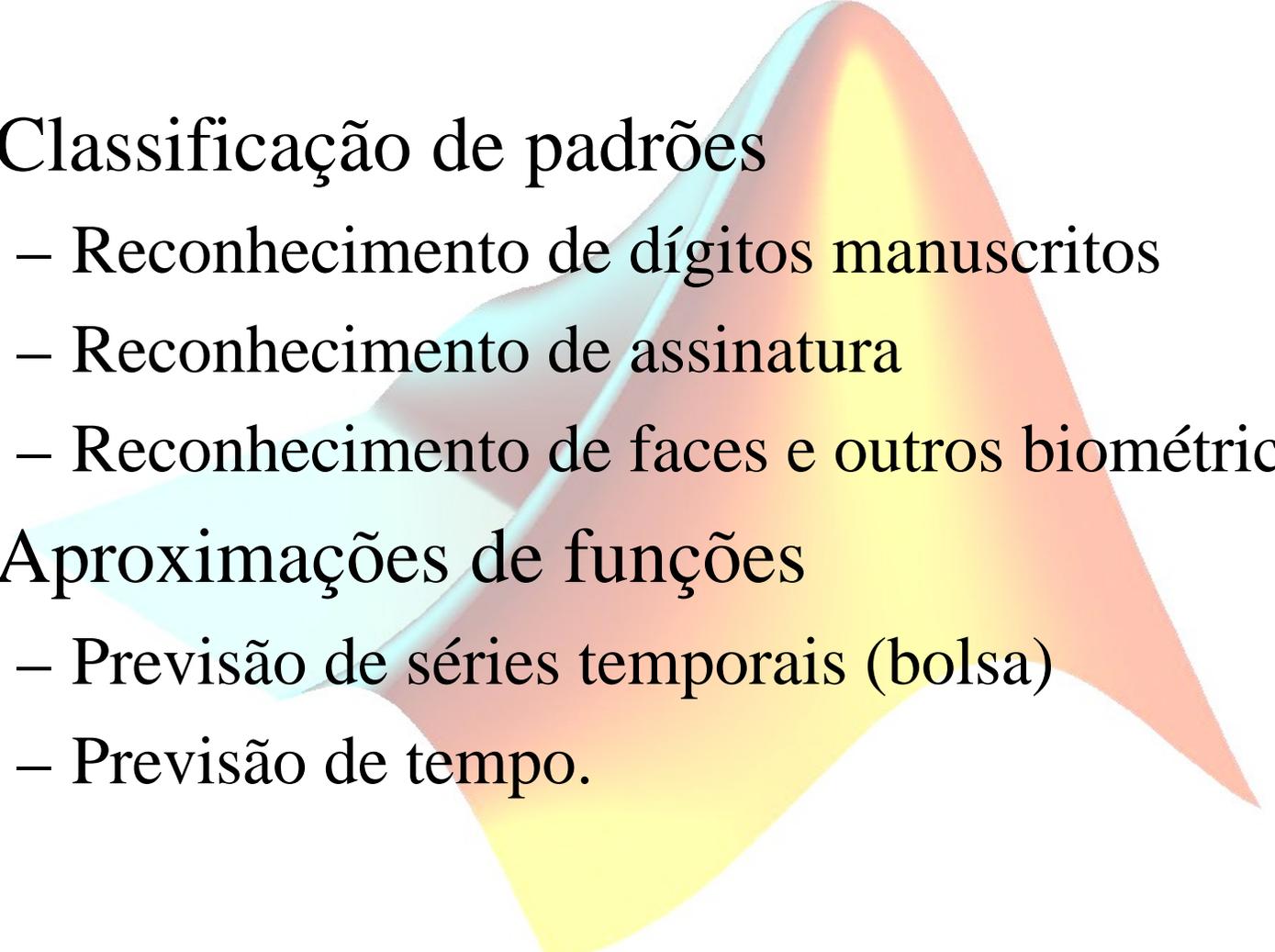
Adriano Martins Moutinho

Inteligência Computacional - 2004

Redes neurais

- Redes neurais são dispositivos matemáticos inspirados em neurônios biológicos que buscam, com isso, obter a capacidade de generalização destes neurônios.
- Redes neurais buscam obter a capacidade de reconhecimento e adaptação que a maioria dos algoritmos não possui.

Aplicações de redes neurais

- Classificação de padrões
 - Reconhecimento de dígitos manuscritos
 - Reconhecimento de assinatura
 - Reconhecimento de faces e outros biométricos.
 - Aproximações de funções
 - Previsão de séries temporais (bolsa)
 - Previsão de tempo.
- 

Uso de Matlab

- O matlab, com o toolbox versão 4.01 de fevereiro de 2001, tornou possível a integração de vários tipos de redes neurais com o ambiente matricial.
- A rede neural tornou-se uma estrutura (struct) de opções e configuração.
- Fácil configuração e exportação para outros ambientes.

Massa de dados

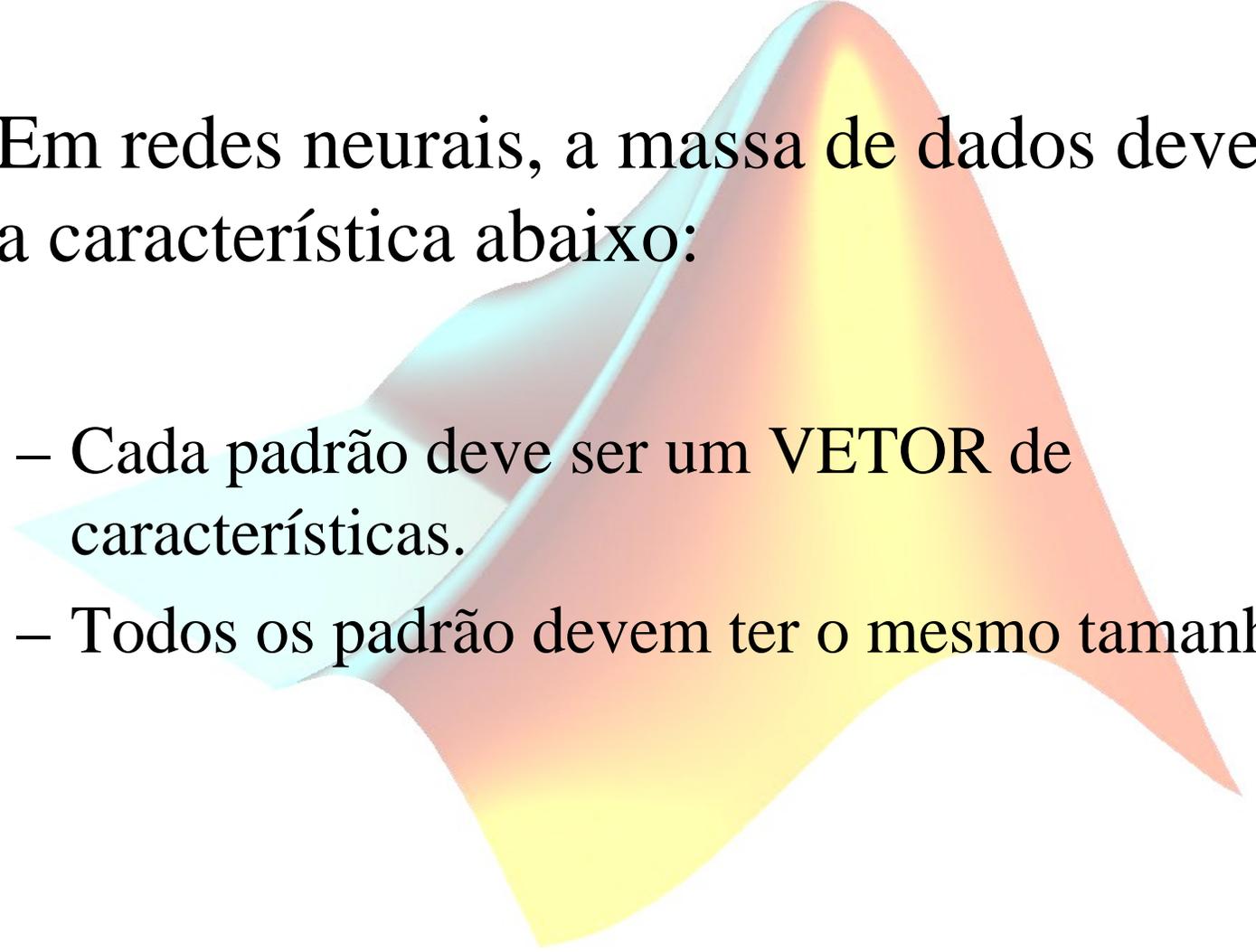
- A base do funcionamento do modelo neural é o treinamento, onde é apresentado à rede uma seqüência de padrões e a classe (resposta) a que estes pertence.
- Por exemplo, durante o treinamento é apresentado a rede neural o vetor de informações que codifica um dígito “1” e a resposta, ou seja, que o dígito corresponde ao “1”.

Massa de dados

- No matlab, a massa de dados corresponde a todos os vetores a serem classificados pela rede.

Formato: $\mathbf{m} \times \mathbf{n}$ - onde \mathbf{m} é o tamanho do vetor padrão e \mathbf{n} o número de padrões.

Características da Massa de dados

- Em redes neurais, a massa de dados deve ter a característica abaixo:
 - Cada padrão deve ser um VETOR de características.
 - Todos os padrão devem ter o mesmo tamanho.
- 

Características da Massa de dados

- Para possibilitar o treinamento, deve-se informar ao Matlab qual a classe de cada padrão a ser treinado. Esta matriz chamamos target ou alvo.

Formato: **m x n** - onde **m** é o número de classes e **n** o número de padrões.

Características da Massa de dados

- O vetor de alvo possui tantas colunas quantos padrões a serem treinados e tantas linhas quantas classes possuir o sistema.
- O vetor de alvo deve possuir um em todas as posições cujos padrões pertencerem a classe determinada, e zero nas demais.

Exemplo: $\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$ 2 classes
9 padrões

Exemplos de massa de dados

$x =$ (5 padrões com 4 características cada)

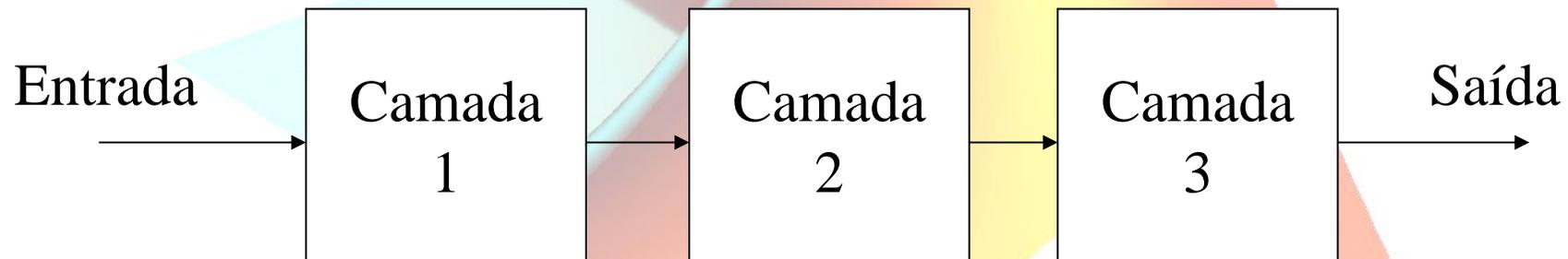
0.4046	0.3786	0.7010	0.8608	0.5947
0.9974	0.8479	0.6201	0.4031	0.9653
0.3764	0.9214	0.9331	0.7514	0.6914
0.6043	0.3494	0.1438	0.6035	0.4111

$t =$ (5 padrões em duas classes)

0	1	0	1	1
1	0	1	0	0

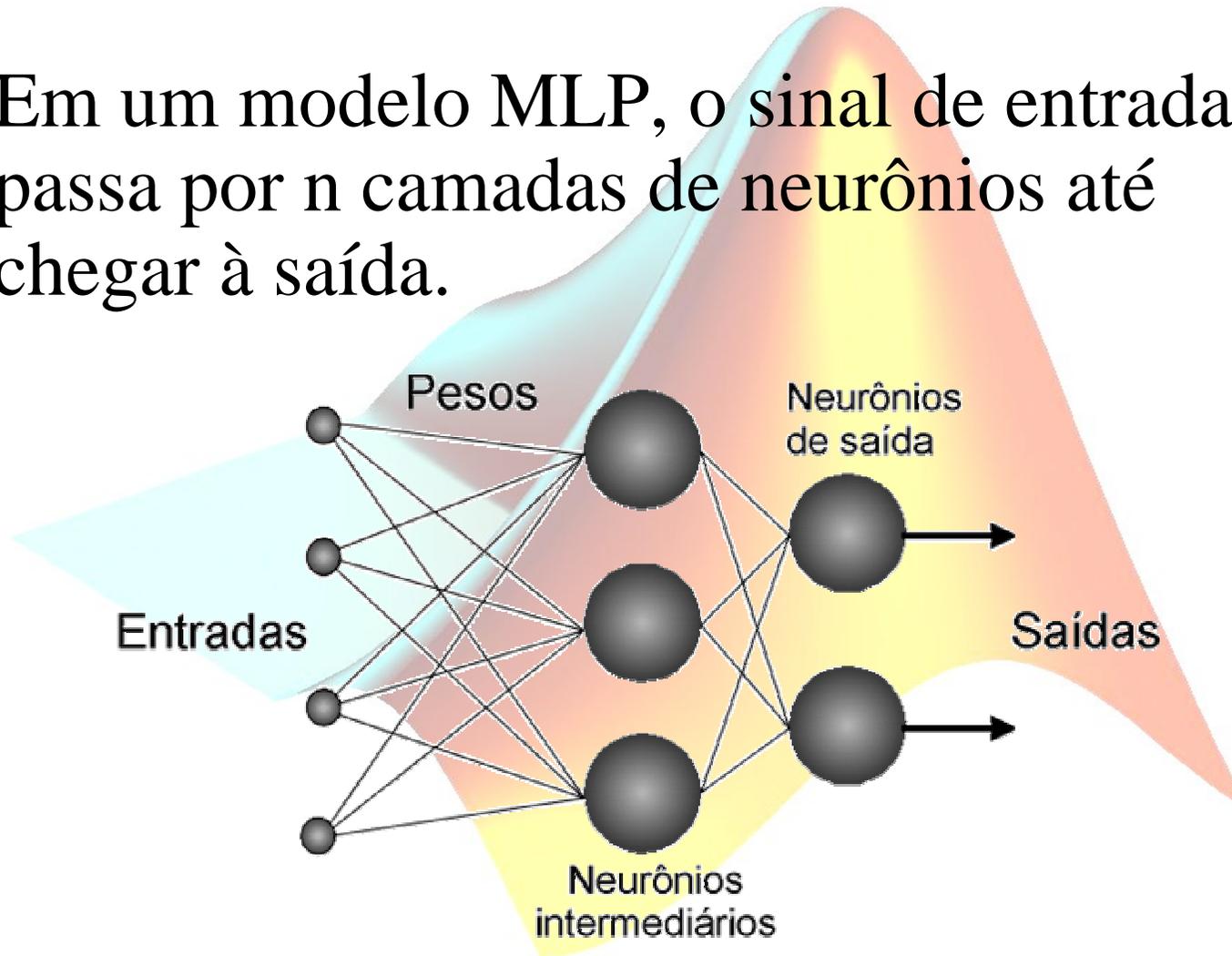
Criando redes neurais

- Existem vários tipos de rede neural. A mais comum é o modelo MLP (multi layer perceptron).



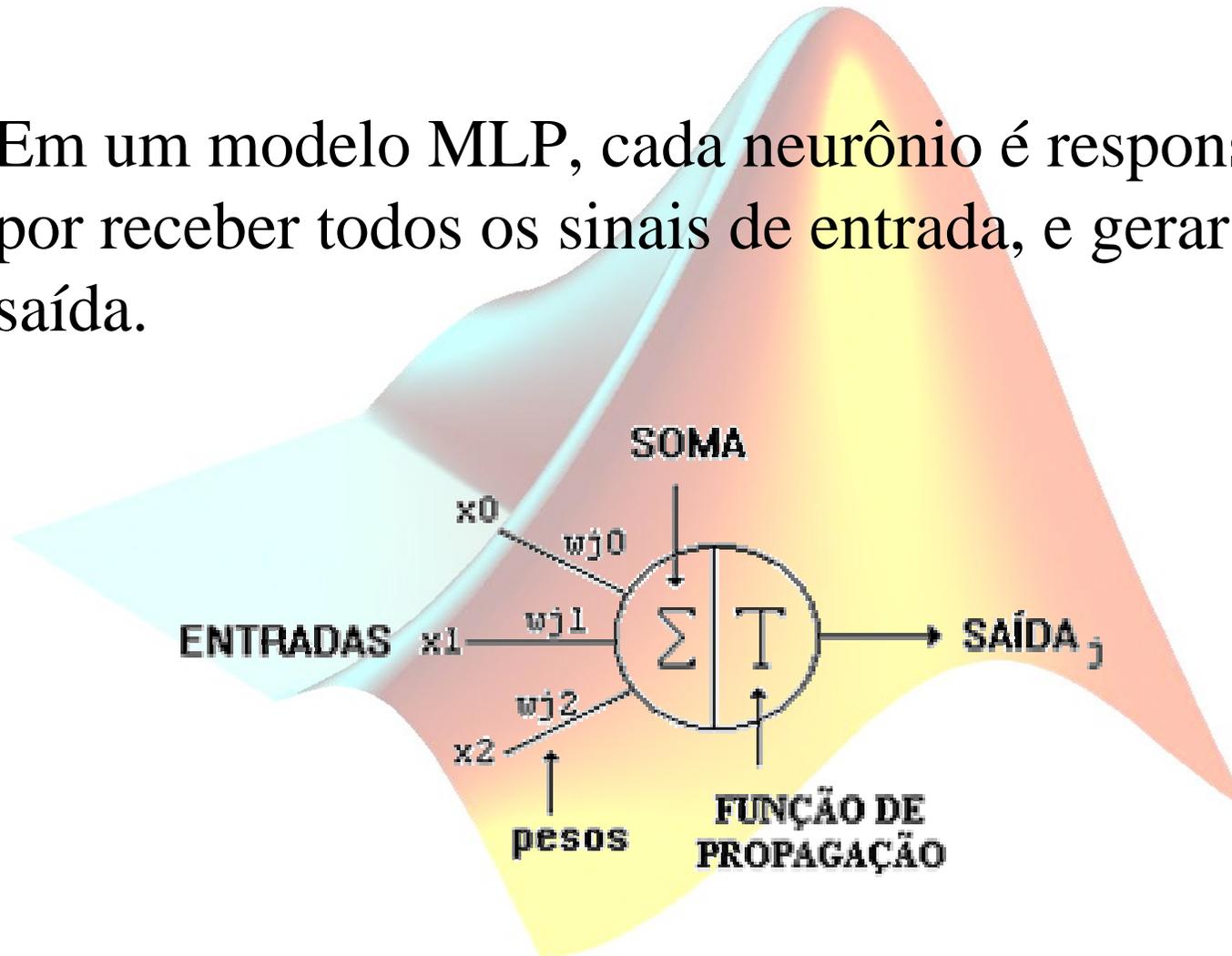
Modelo MLP

- Em um modelo MLP, o sinal de entrada passa por n camadas de neurônios até chegar à saída.



Modelo de cada neurônio

- Em um modelo MLP, cada neurônio é responsável por receber todos os sinais de entrada, e gerar uma saída.



Treinamento da rede neural

- Durante o treinamento, a rede neural receberá a matriz de dados e o alvo.
- Todos os pesos de todos os neurônios serão modificados para obter a classificação desejada.

Métodos de treinamento

- O Matlab possui um grande número de métodos de treinamento:

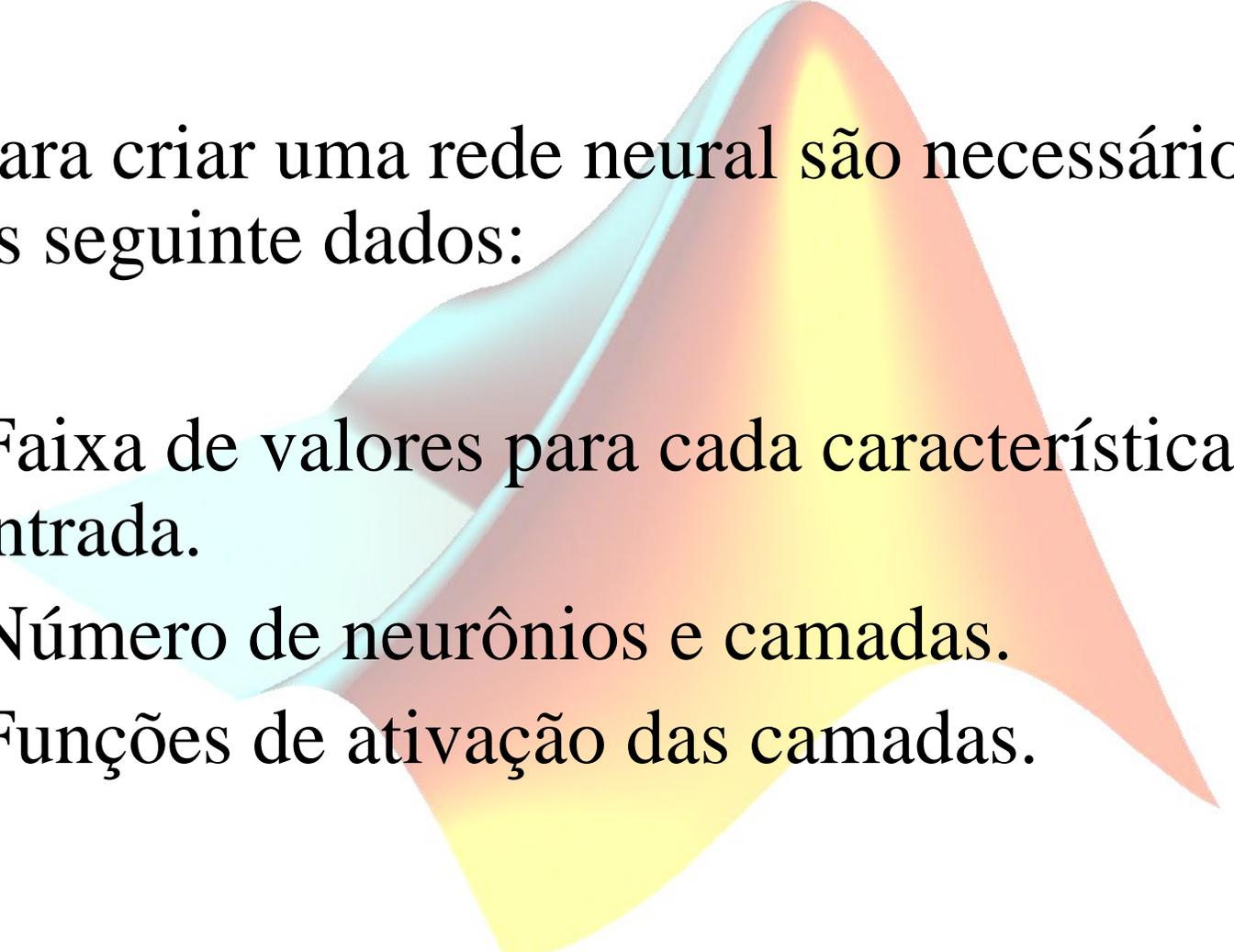
`traingd` → Clássico método de treinamento usando backpropagation e gradiente descendente

`traingdx` → Clássica adaptação do método usando gradiente descendente e taxa de aprendizado variável.

`traincgp` → Gradiente conjugado.

`trainrp` → Propagação resiliente.

Criando uma rede neural!

- Para criar uma rede neural são necessários os seguintes dados:
 - 1) Faixa de valores para cada característica da entrada.
 - 2) Número de neurônios e camadas.
 - 3) Funções de ativação das camadas.
- 

Exemplo

- Sendo \mathbf{p} a matriz de treinamento, que contém as características.

```
>>net = newff([0 1;0 1;0 1;0 1],[3 2],{'logsig'  
    'logsig'})
```

```
>>net = newff(minmax(p),[3 2],{'logsig' 'logsig'})
```

Variável de rede no Matlab

- A variável net, conforme criada no slide anterior, é uma estrutura.
- Exemplos de parâmetros:
 - >> net.numlayers → Número de camadas
 - >> net.performFcn: 'mse'
 - >> net.trainFcn: 'trainlm'
 - >> net.trainParam → Parâmetros de treinamento

Parâmetros de treinamento

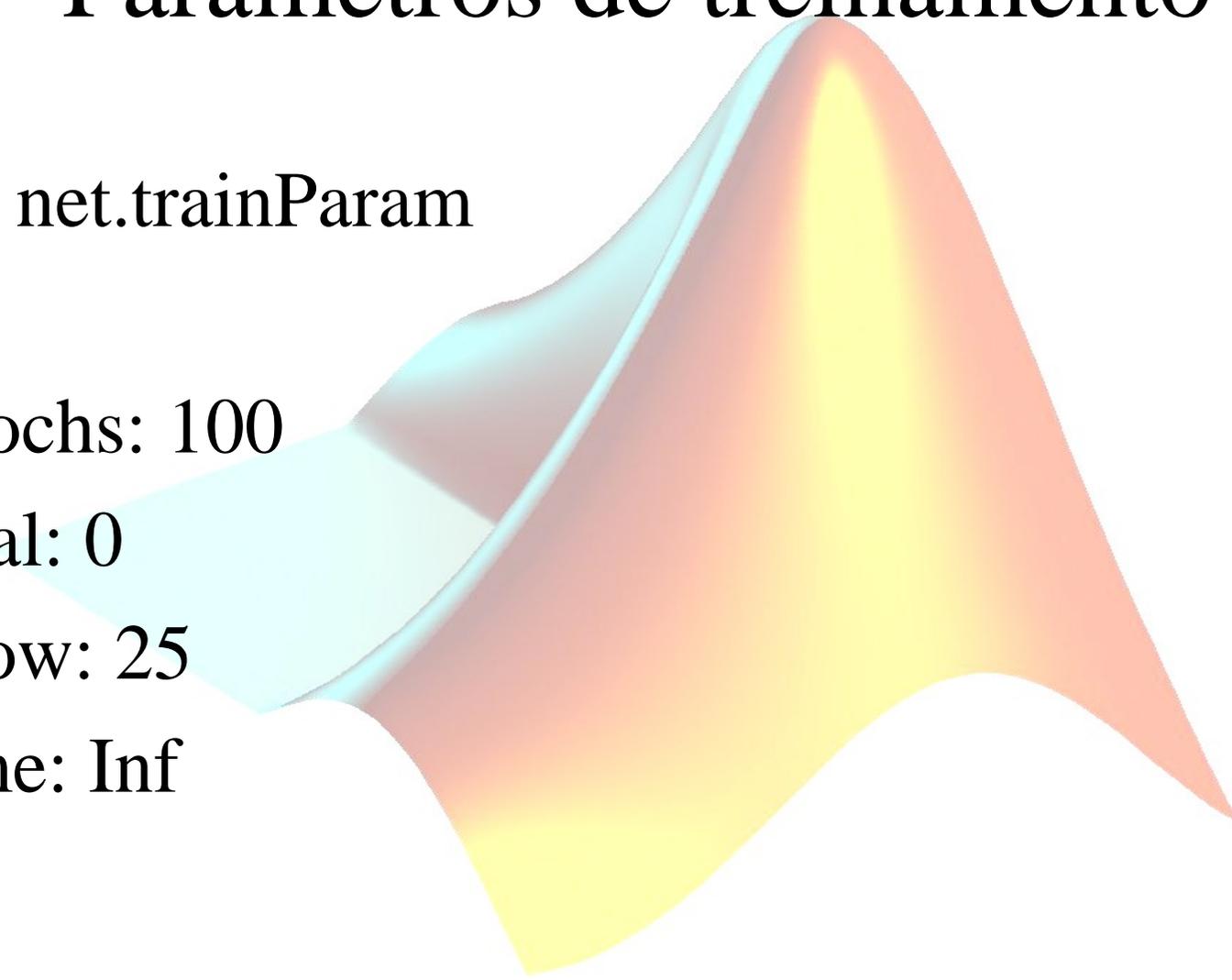
```
>> net.trainParam
```

```
epochs: 100
```

```
goal: 0
```

```
show: 25
```

```
time: Inf
```



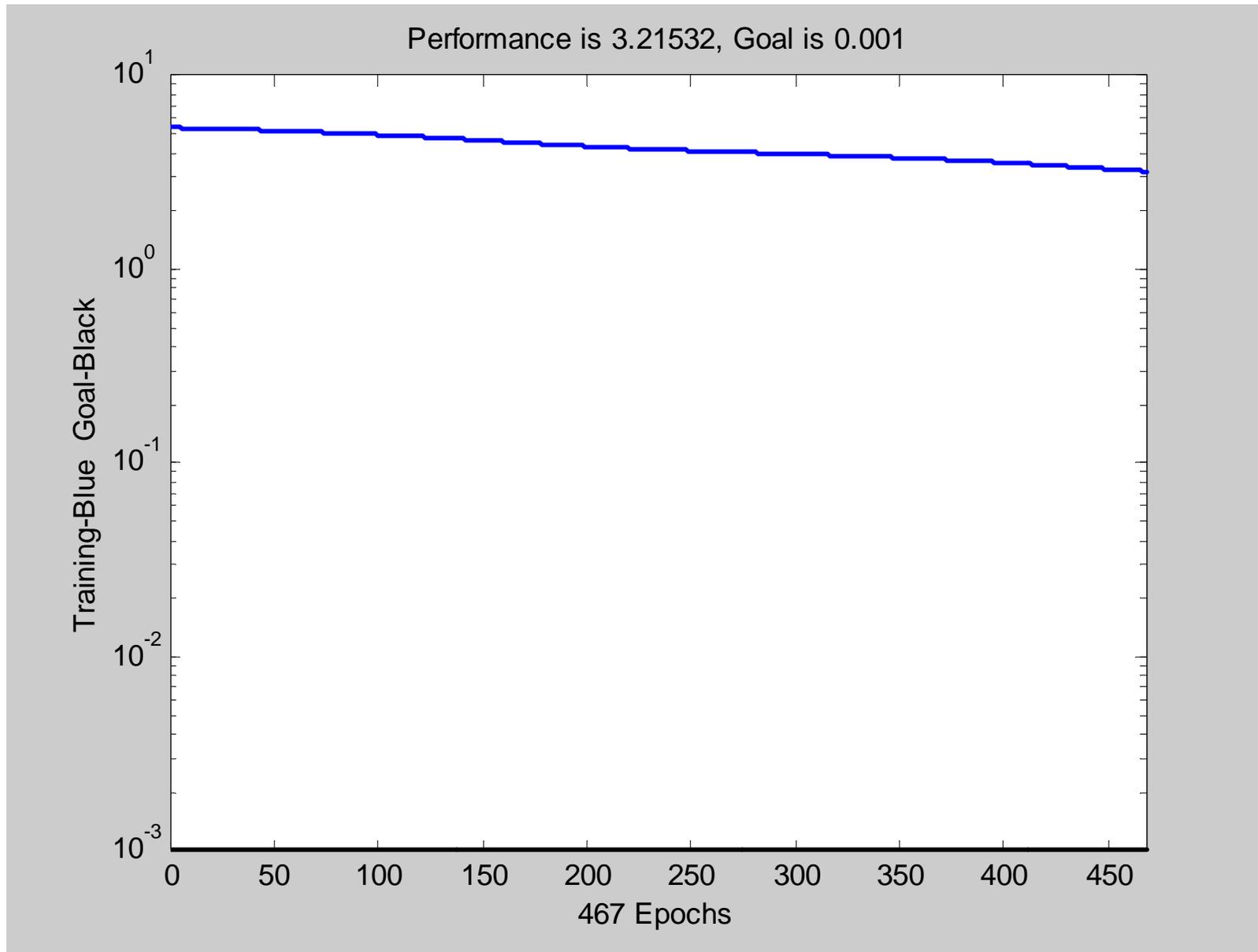
Treinamento!

- Para treinar a rede neural, são necessários:
 - 1) Variável net a ser treinada. Definindo-se a arquitetura (número de neurônios, número de camadas, função de treinamento...)
 - 2) Matriz de treinamento
 - 3) Matriz de target ou alvo.

Treinando...

- São necessários alguns ajustes antes de iniciar o treinamento:
 - `net.trainParam.epochs = 100000;`
 - `net.performFcn = 'sse';`
 - `net.trainParam.goal = 0.001;`
 - `net.trainParam.show = 1;`
 - `net.trainFcn = 'traingd';`
- `>> net = train(net,p,t);`
- Um gráfico aparecerá, mostrando a evolução do treinamento.

Introdução ao uso de redes neurais com Matlab



Usar a rede, depois de treinada:

- Depois de treinada, uma rede neural poderá ser testada para verificação do aprendizado.
- Deve-se verificar a saída da rede com novos padrões, que nunca antes foram apresentados à rede. O formato continua o mesmo do vetor de treinamento.

Simulação de rede:

- Para simular (usar) a rede, utilize o comando `sim`:

```
>>sim(net,p);
```

```
ans =
```

```
0.0526    0.0862    0.1160    0.2420    0.2529  
0.9983    0.9970    0.9982    0.9970    0.9970
```

Comandos adicionais

- `prestd` - Normaliza os dados para terem desvio padrão 1 e média zero.
- `premnmx` - Normaliza os dados para ter máximo de 1 e mínimo de -1.
- `vec2ind` - passa do formato padrão de alvo para mostrar o número da classe.
- `ind2vec` - passa do número da classe para formato padrão de alvo.