Introdução a classificação de dados

Universidade Federal do Espírito Santo
Programa de pós-graduação em informática

André Pacheco

Sumário

- O problema de classificação de dados
- *Clusterização*
- Classificadores
- Elencos de classificadores
- Exemplos de aplicação

Objetivo

- Apresentar os conceitos básicos relacionados ao problema de classificação de dados.
  - Descrever métricas de desempenho da classificação.
- Apresentar algoritmos comumente utilizados para classificação.
- Discutir metodologias para melhorar a classificação.

Introdução

- Classificar dados ou objetos é tarefa tão corriqueira que desde criança realizamos classificações.

- Além disso é um problema que abrange as mais diversas áreas: biologia, medicina, automobilística, entretenimento, financeira etc.

- É um dos tópicos mais ativos na área de *machine learning*

O problema de classificação

- Problemas de classificação são comuns na prática. Exemplos não faltam:
  - Um médico quer determinar se um tumor é maligno ou benigno.
  - Um servidor de e-mails está interessado em classificar e-mails como spam ou não spam para os usuários.
  - Uma empresa quer classificar tipos de vinhos.
  - Um banco quer determinar se o cliente é apto ou não a receber determinado empréstimo.
  - Um biólogo deseja classificar espécies de plantas ou animais.

O problema de classificação

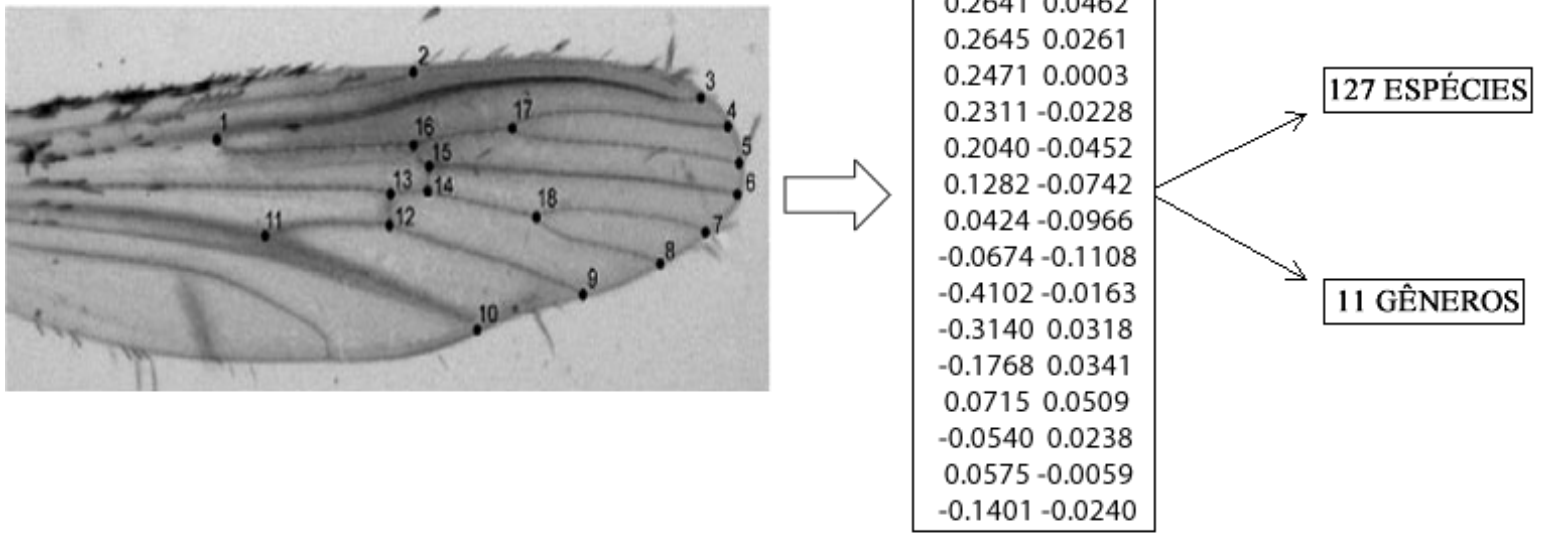
- O problema ocorre pois não sabemos a priori classe que o objeto pertence
  - Quando retirado um tumor de uma pessoa, o médico não sabe se o mesmo é benigno ou maligno, por exemplo.

- Mas como realizar a classificação? Baseado em que vamos classificar?

O problema de classificação

- O problema ocorre pois não sabemos a priori classe que o objeto pertence
  - Quando retirado um tumor de uma pessoa, o médico não sabe se o mesmo é benigno ou maligno, por exemplo.

- Mas como realizar a classificação? Baseado em que vamos classificar?



O problema de classificação

- De maneira geral é avaliado atributos do objeto/indivíduo e então, baseado nesses atributos, é realizada a classificação do mesmo.

- Os atributos, obviamente, dependem da aplicação desejada
  - No caso dos e-mails são avaliados as palavras do mesmo e classificado entre spam ou não.
  - No caso da classificação do vinho, são avaliadas textura, coloração, pH, dentre outros atributos.

O problema de classificação

- De maneira geral é avaliado atributos do objeto/indivíduo e então, baseado nesses atributos, é realizada a classificação do mesmo.

- Os atributos, obviamente, dependem da aplicação desejada
  - No caso dos e-mails são avaliados as palavras do mesmo e classificado entre spam ou não.
  - No caso da classificação do vinho, são avaliadas textura, coloração, pH, dentre outros atributos.

- Resumindo: o problema de classificação consiste em **determinar o rótulo** de algum objeto, baseado em um **conjunto de atributos extraídos** do mesmo.

O problema de classificação

- Apesar dos problemas de classificação serem bastantes comuns, há casos em que as classes não estão bem definidas.

- Por exemplo: imagine que desejamos classificar cartas de um baralho. Como deveríamos fazer?

O problema de classificação

- Apesar dos problemas de classificação serem bastantes comuns, há casos em que as classes não estão bem definidas.

- Por exemplo: imagine que desejamos classificar cartas de um baralho. Como deveríamos fazer?
  - Por cor?
  - Por naipe?
  - Por número?

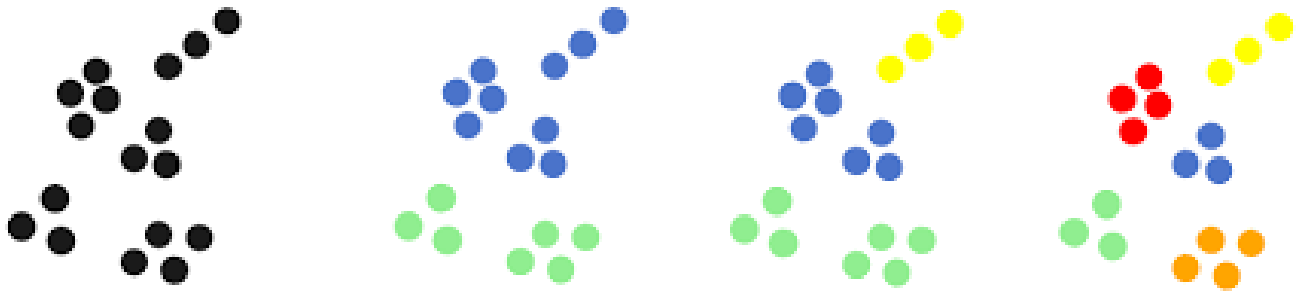
- Eis que surge o problema de ***clusterização.***

Clusterização

- Embora exista similaridade, classificar e *clusterizar* **são problemas diferentes.**

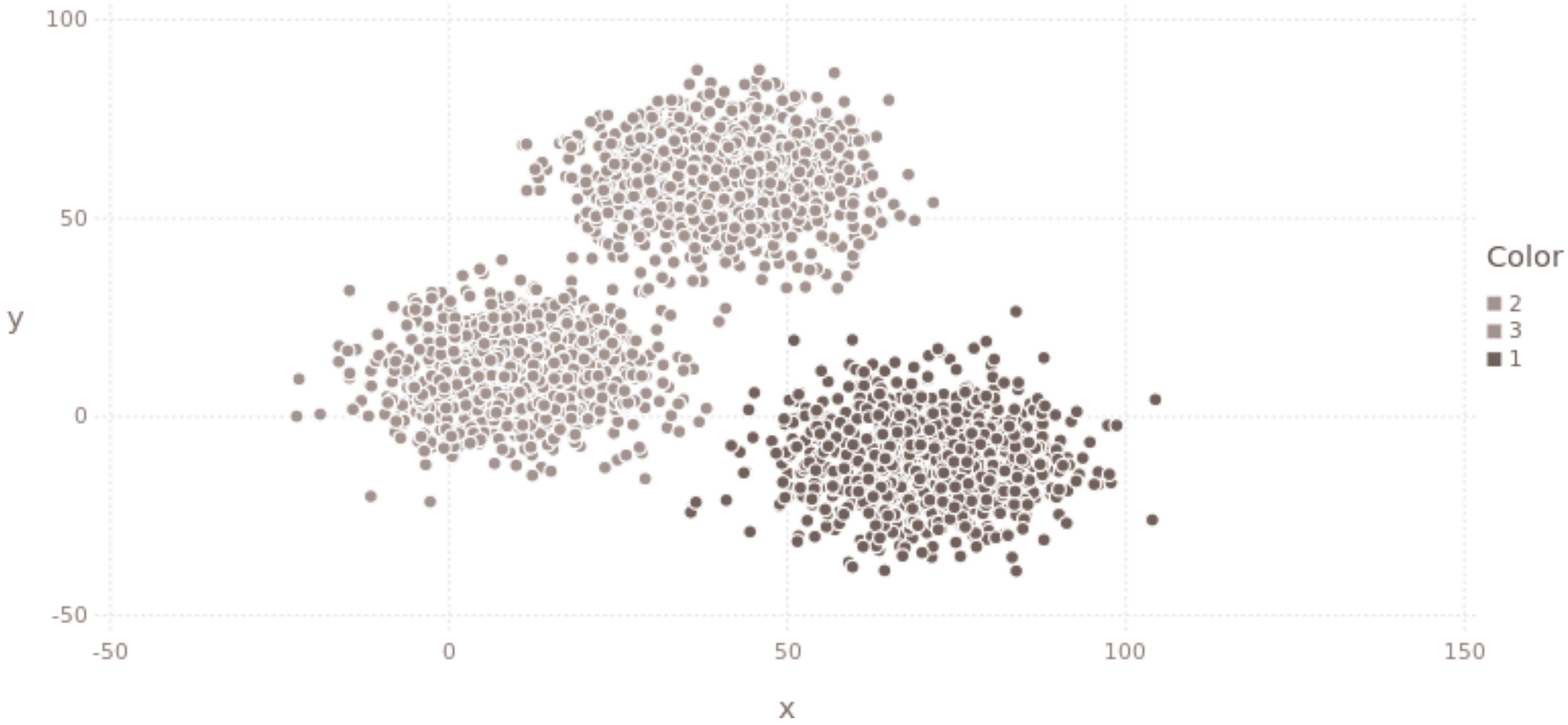
- O objetivo da *clusterização* é determinar a quantidade de rótulos existentes em um conjunto de dados.
  - Um dos algoritmos mais utilizados é o K-means, realizado através de medidas de distâncias.

- Portanto, a classificação já parte do princípio que o número de rótulos já é conhecido.
  - *Clusteriza* depois classifica.

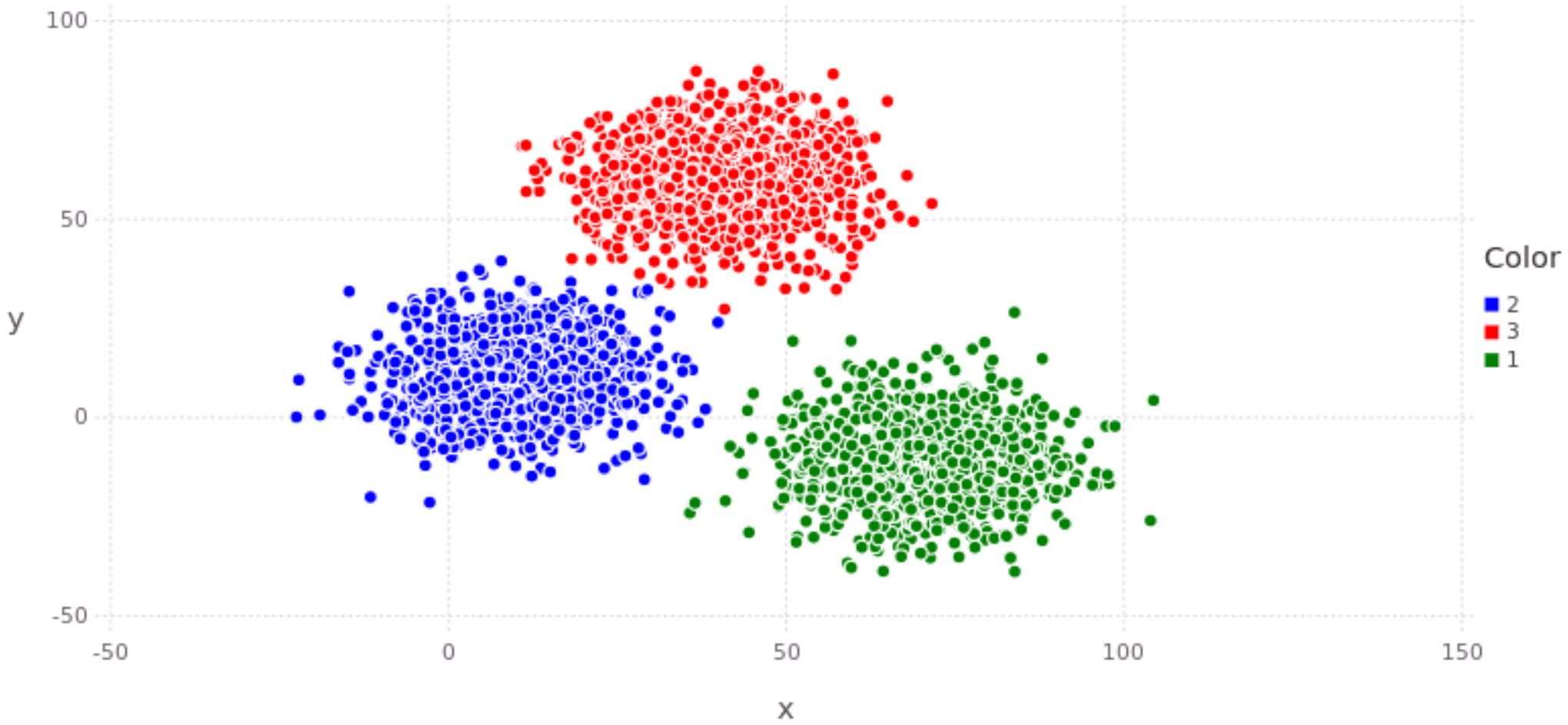


(a) Dados iniciais (b) Dois clusters (c) Três clusters (c) Cinco clusters

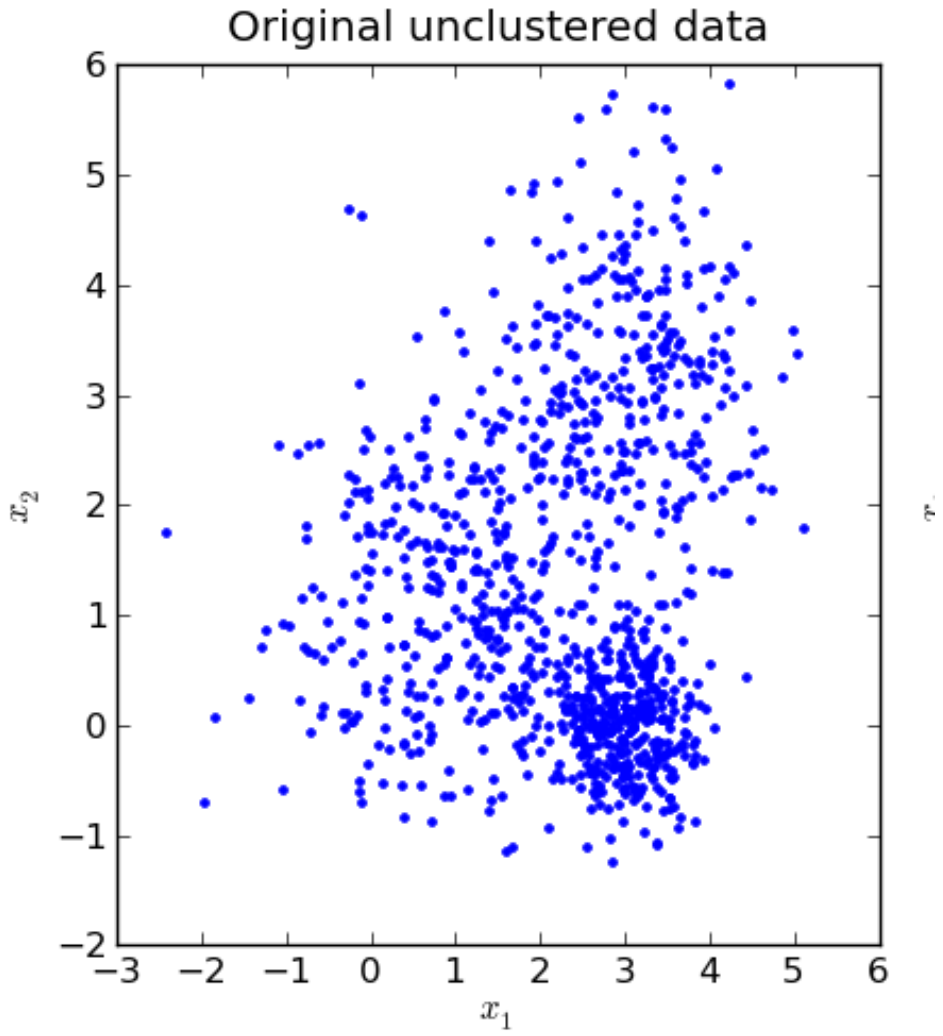
Clusterização



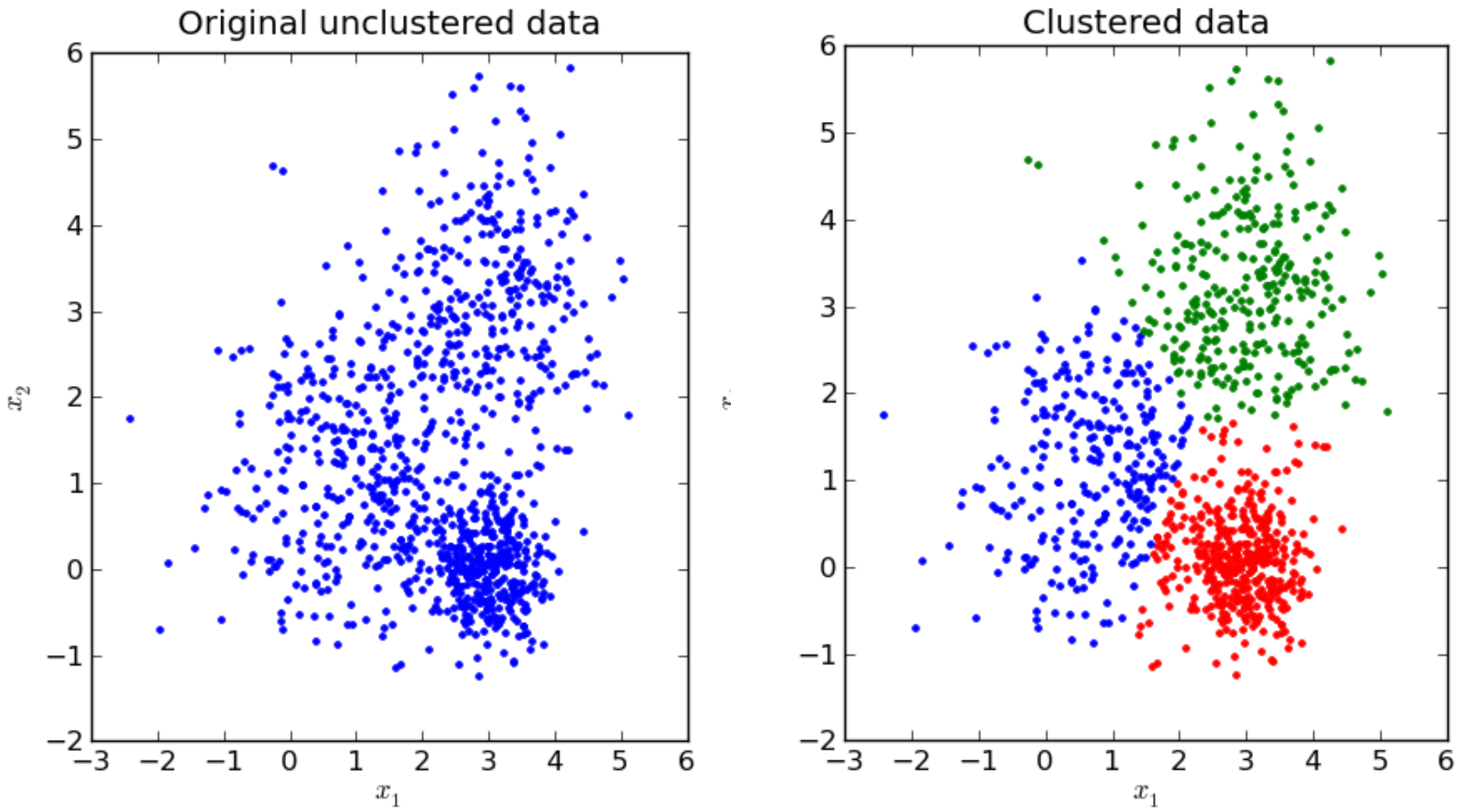
Clusterização



Clusterização



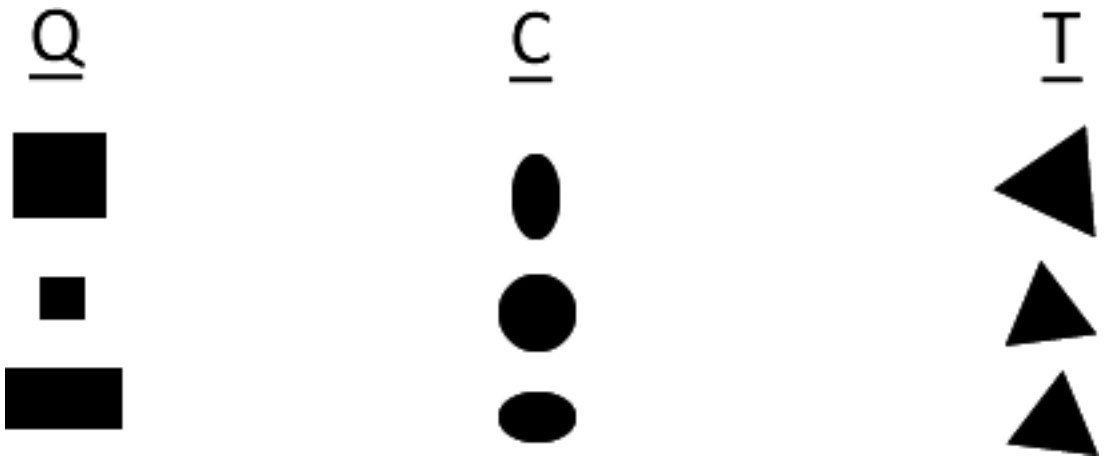
Clusterização



Classificadores

- Os algoritmos de classificação atuam baseado em exemplos com rótulos já conhecidos
  - Problema supervisionado
  - É necessário um conjunto de treinamento

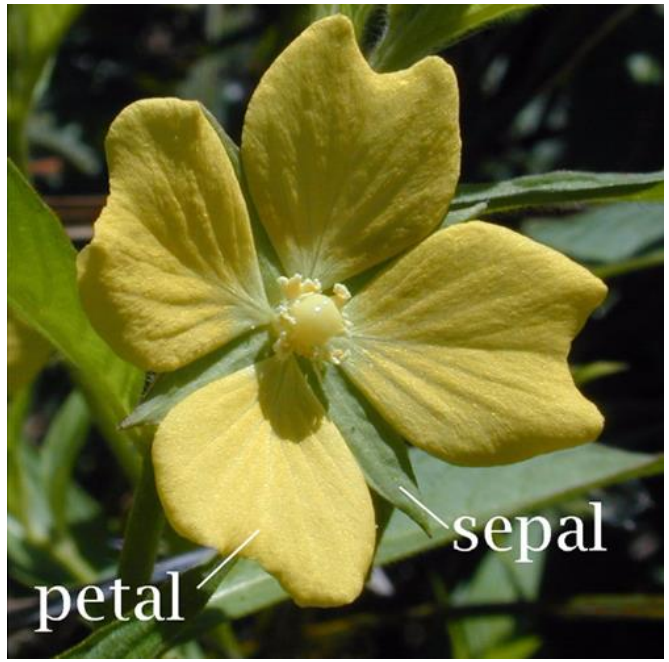
- Imagine que um algoritmo classifica 3 formas geométricas: quadrangular, triangular e circular. Um exemplo de conjunto de treinamento seria:



Exemplo de base de dados

- Um exemplo clássico é a base de dados de flores Iris.
  - 4 atributos de entrada
  - 3 classes de saída
  - 150 amostras

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Espécie
4.8	3.1	1.6	0.2	setosa
5.4	3.4	1.5	0.4	setosa
5.2	4.1	1.5	0.1	setosa
6.2	2.9	4.3	1.3	versicolor
5.1	2.5	3.0	1.1	versicolor
5.7	2.8	4.1	1.3	versicolor
6.3	3.3	6.0	2.5	virginica
6.4	2.8	5.6	2.2	virginica
6.3	2.8	5.1	1.5	virginica

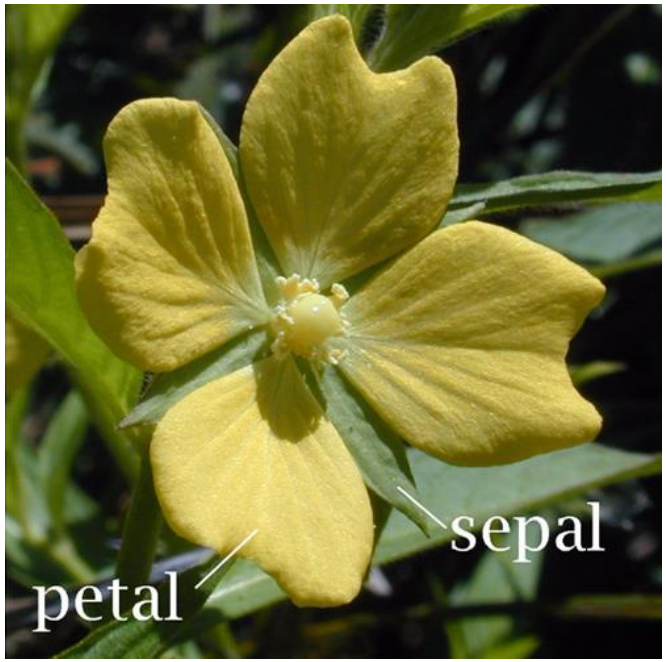


Exemplo de base de dados

- Um exemplo clássico é a base de dados de flores Iris.
  - 4 atributos de entrada
  - 3 classes de saída
  - 150 amostras

- Normalmente é utilizado 70% da base para treinamento e 30% para testes.

- Utiliza-se como métrica de desempenho a acurácia de classificação.



Classificadores

- Portanto, o objetivo de um algoritmo de classificação é **aprender**, a partir de um conjunto de treinamento, **uma correlação** entre os atributos de entrada e saída.
  - Dado uma nova **entrada não rotulada** o classificador seja capaz de **determinar o rótulo** vinculado a ela.

- Existem diversos algoritmos de classificação
  - Baseados em: redes neurais, árvores de decisão, em distâncias, em métodos bayesianos, etc.

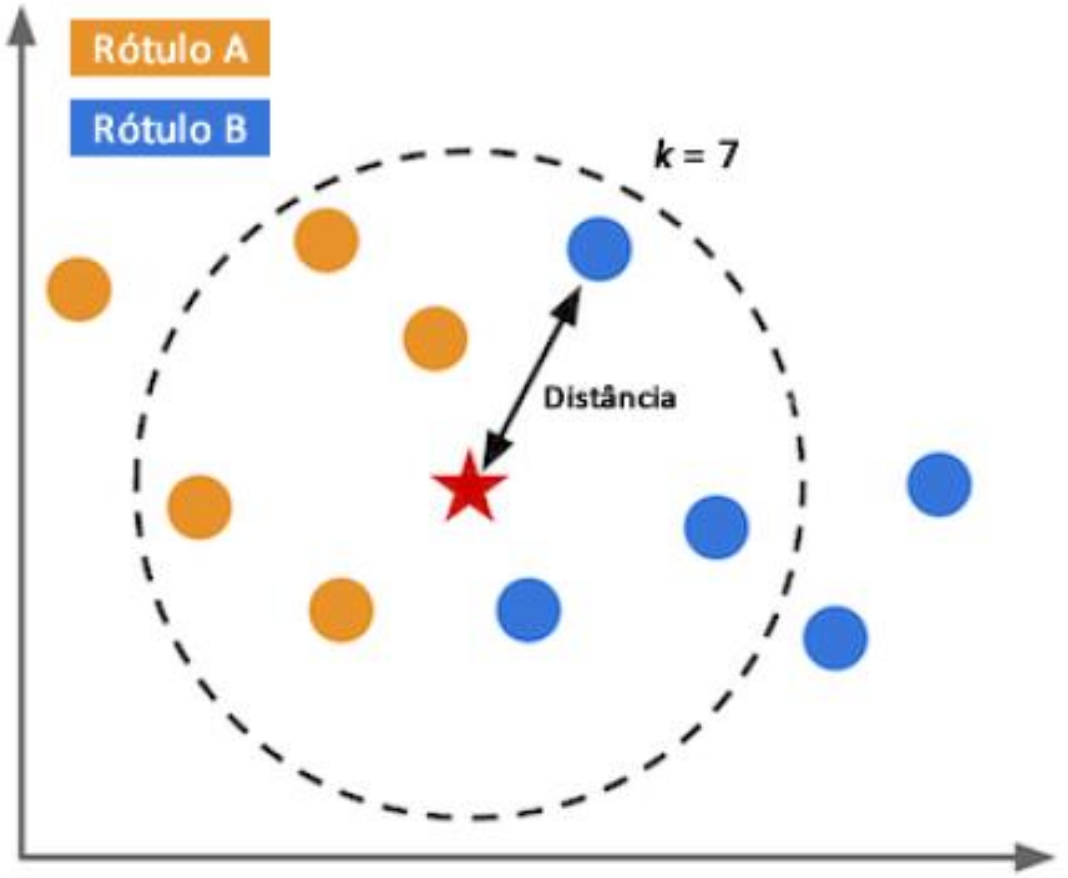
- Serão apresentados 3 algoritmos de classificação:
  - KNN – *K-nearest neighbors*
  - FNN – *Feedforward neural network*
  - ELM – *Extreme learning machine*

Classificadores - KNN

- O KNN é um dos algoritmos mais utilizados em problemas de classificação de dados.
  - Fácil entendimento.
  - Fácil implementação.
  - Baseado em distância.

- Escolhe-se o *k* vizinhos mais próximos da amostra em questão.

- Dois pontos chaves:
  - Escolha do valor de *k*
  - Métrica para distância

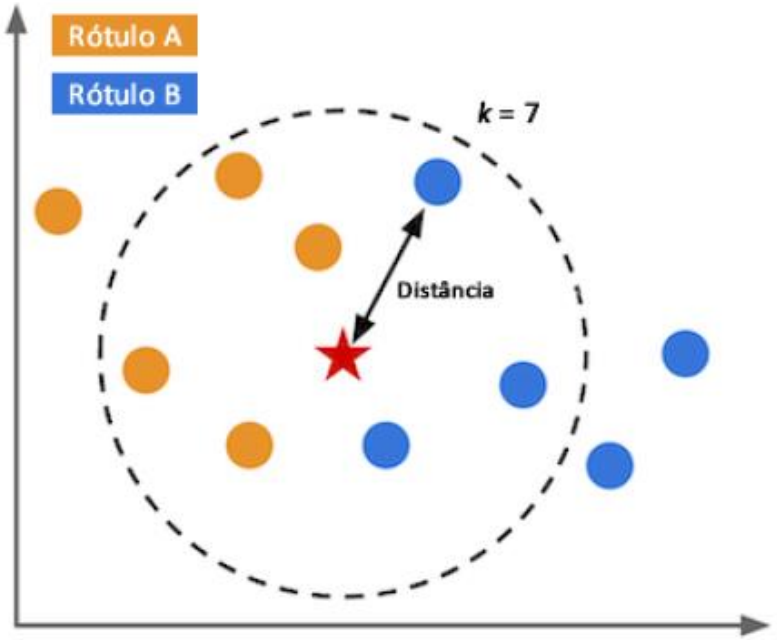


Classificadores - KNN

- A métrica de distância mais utilizada é a Euclidiana

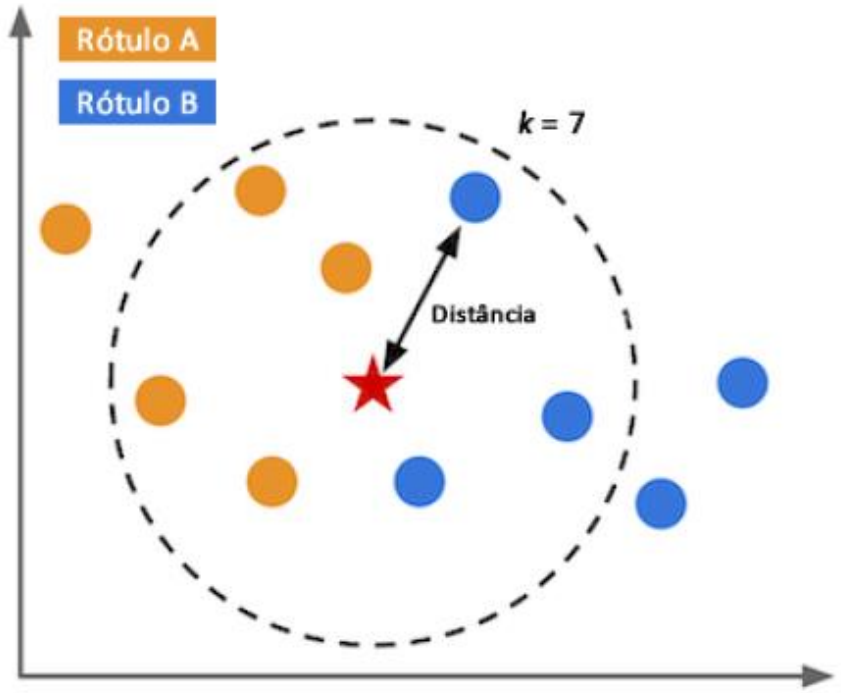
$$D = \sqrt{(p_1 - q_1)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

- O valor de *k* é escolhido de maneira empírica e depende da base de dados.
  - Busca exaustiva
  - *K* igual $\sqrt{|N|}$
  - *K* escolhido entre os números primos do intervalo $[3, \sqrt{|N|}]$



Classificadores - KNN

- Principais deficiências:
  - Sensível a escolha de *k*.
  - Calcular distância é custoso, pode tomar muito tempo se a base for grande!



Classificadores - KNN

- Pseudocódigo KNN:

Algoritmo: *K vizinhos mais próximos*

```
1 inicialização:
2       Preparar conjunto de dados de entrada e saída T;
3       Informar o valor de k;
4 para cada nova amostra faça
5       Calcular distância para todas as amostras de T;
6       Determinar o conjunto das k's distâncias mais próximas
7       O rótulo com mais representantes no conjunto dos k's vizinhos será o escolhido
8 fim para
9 retornar: conjunto de rótulos de classificação
```

Classificadores - KNN

- Desempenho do KNN para Iris:

Base de dados	Acurácia (%)	Acurácia (erros)	Desvio Padrão
Iris	95,55	2	0

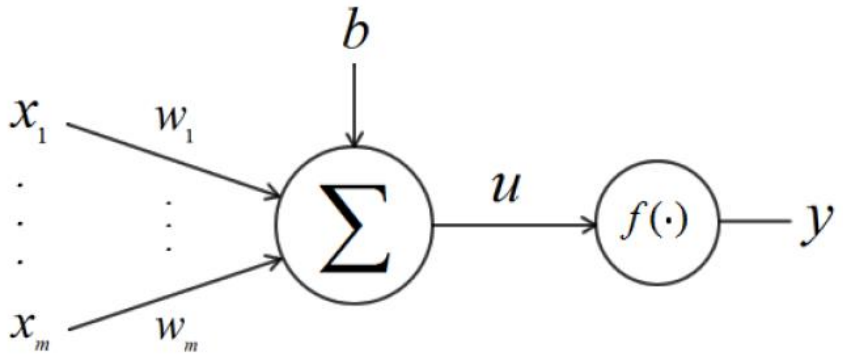
- Matriz de confusão:

	Virginica	Setosa	Versicolor
Virginica	**13**	0	0
Setosa	0	**13**	1
Versicolor	0	1	**17**

Classificadores - FNN

- Uma rede neural artificial é um sistema de processamento paralelo de informações constituído pela interconexão de unidades básicas de processamento, denominadas neurônios.
  - Inspirada na rede neural biológica.

- O modelo mais utilizado de neurônio é o *perceptron*:



$$\mapsto \quad y = f\left(\sum_{i=1}^{m} x_i w_i + b\right)$$

$$f(u) = \frac{1}{1 + e^{-u}}$$

Classificadores - FNN

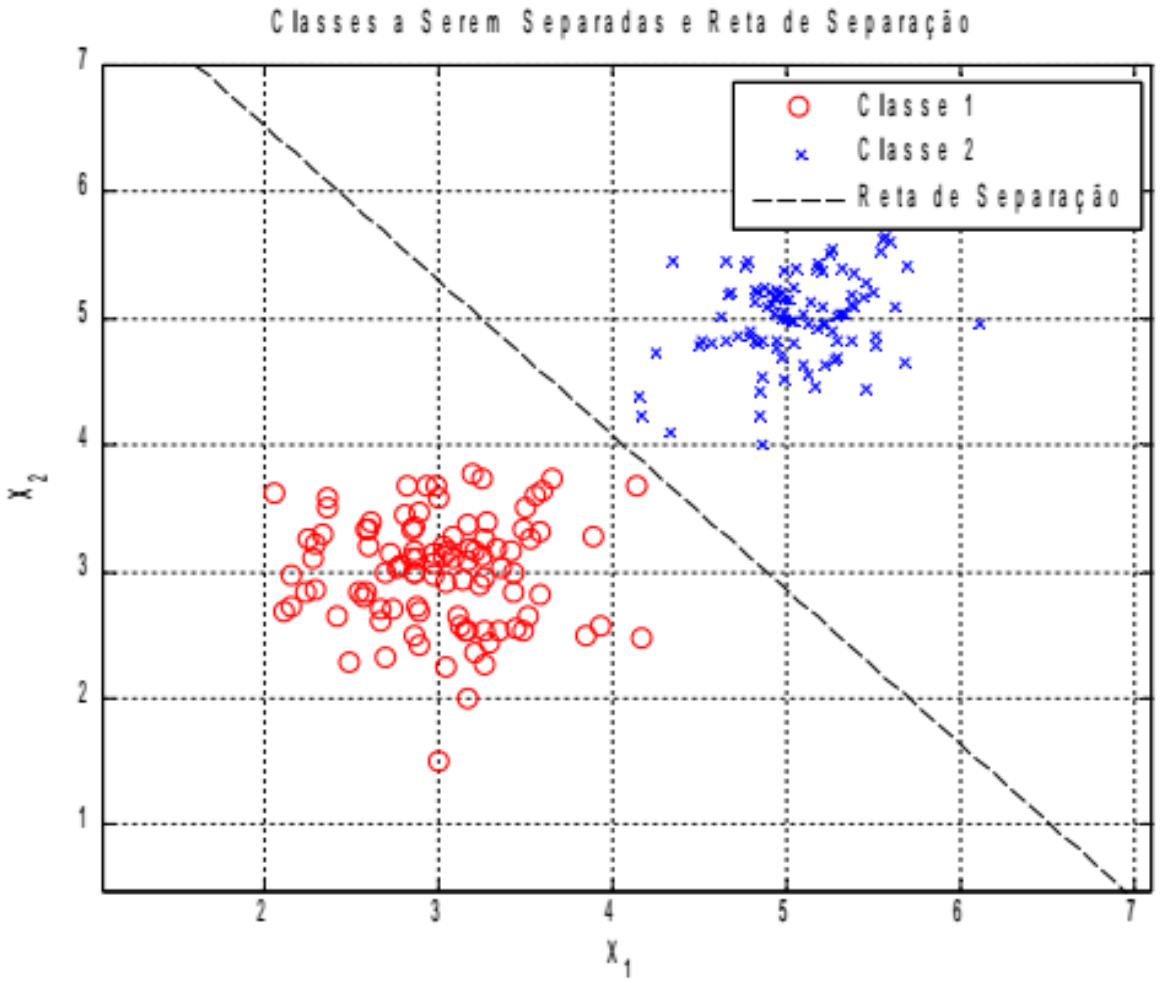
- Uma rede neural artificial é um sistema de processamento paralelo de informações constituído pela interconexão de unidades básicas de processamento, denominadas neurônios.
  - Inspirada na rede neural biológica.

- O modelo mais utilizado de neurônio é o *perceptron*:
  - Considerando apenas 2 entradas:

$$y = f(\sum_{i=1}^{m} x_i w_i + b) \qquad \mapsto \qquad y = x_1 w_1 + x_2 w_2 + b$$

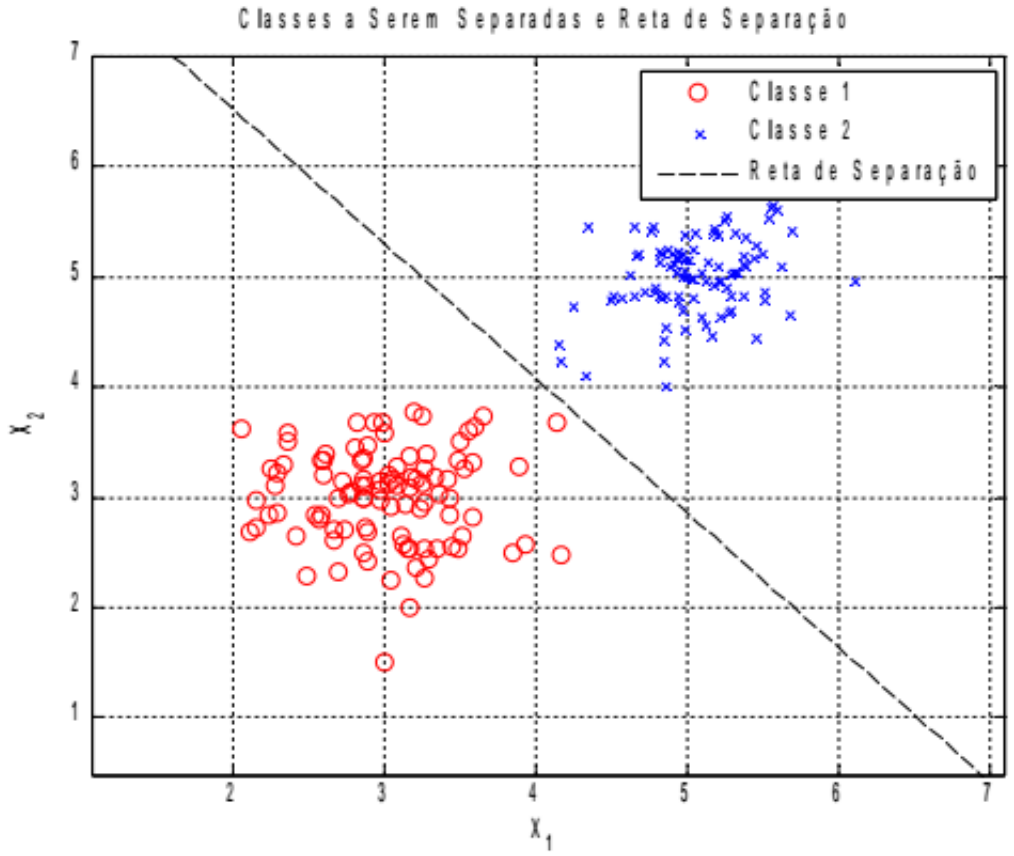
Classificadores - FNN

- Capacidade de um perceptron: problemas linearmente separáveis!



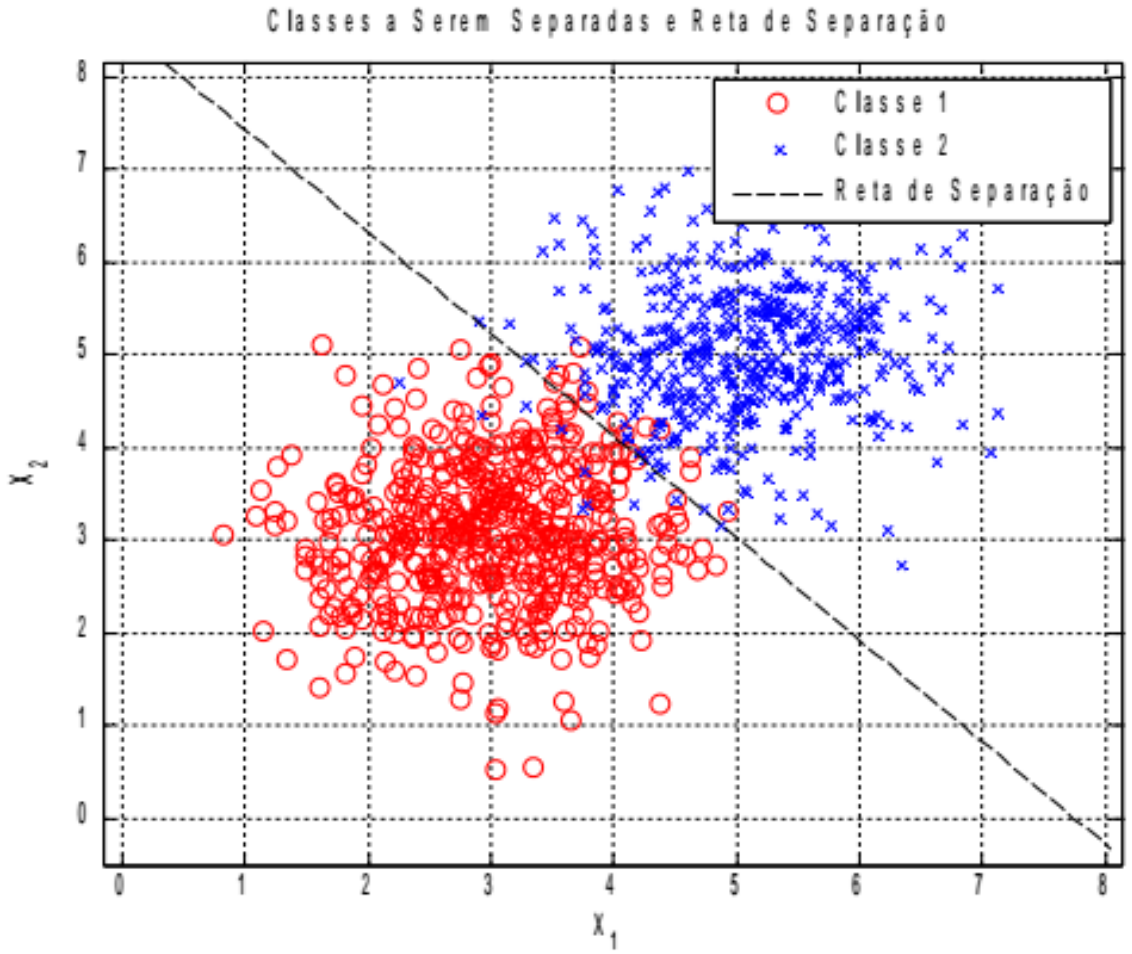
Classificadores - FNN

- Capacidade de um perceptron: problemas linearmente separáveis!
  - Neste caso, o que ocorre com o erro?



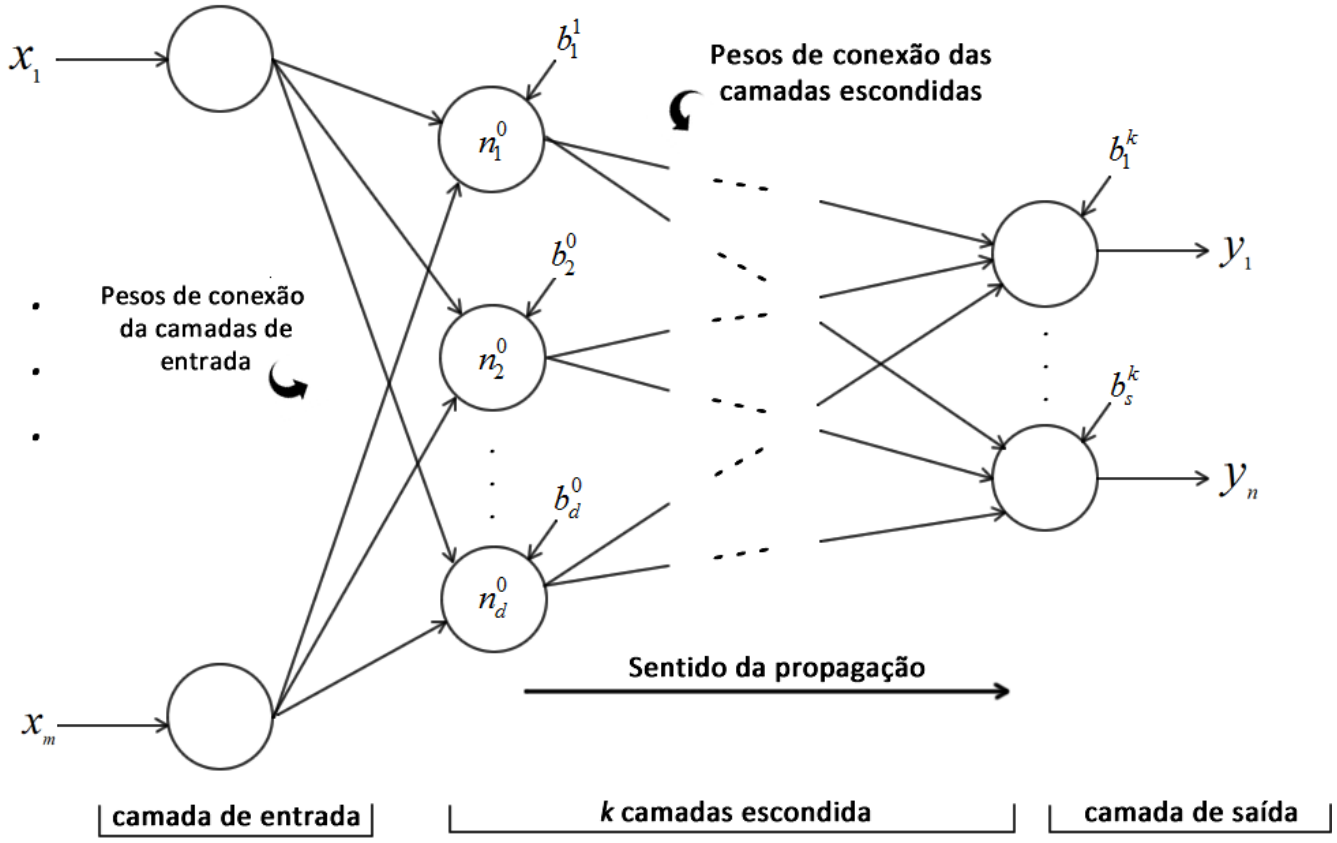
Classificadores - FNN

- Capacidade de um perceptron: problemas linearmente separáveis!
  - Neste caso, o que ocorre com o erro?



Classificadores - FNN

- Uma rede neural é formada por três camadas:
  - Entrada
  - Oculta(s)
  - Saída

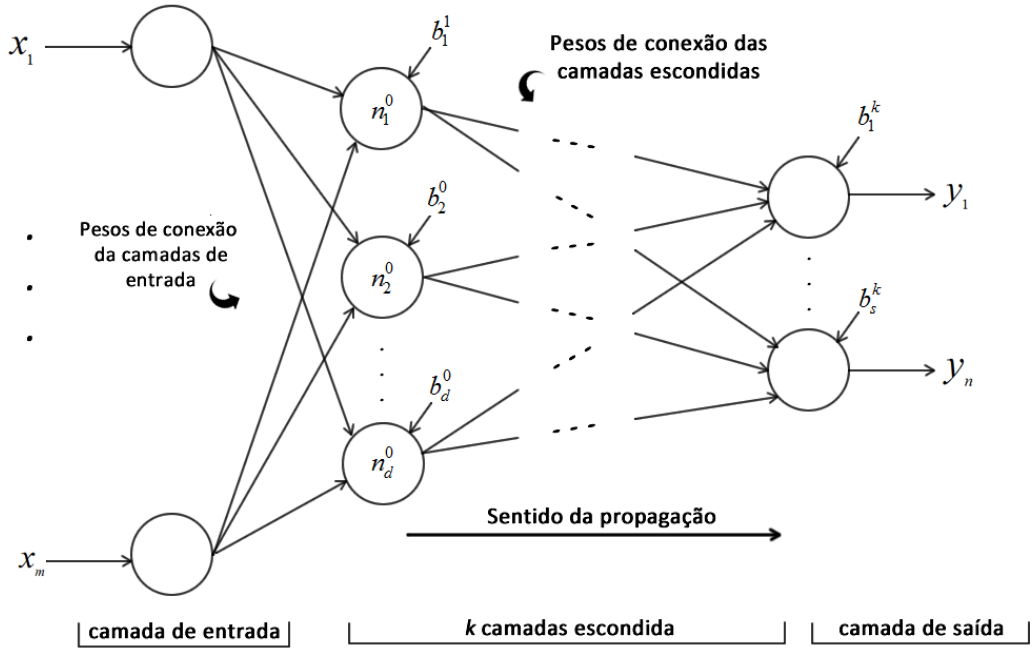


Classificadores - FNN

- Uma rede neural é formada por três camadas:
  - Entrada
  - Oculta(s)
  - Saída

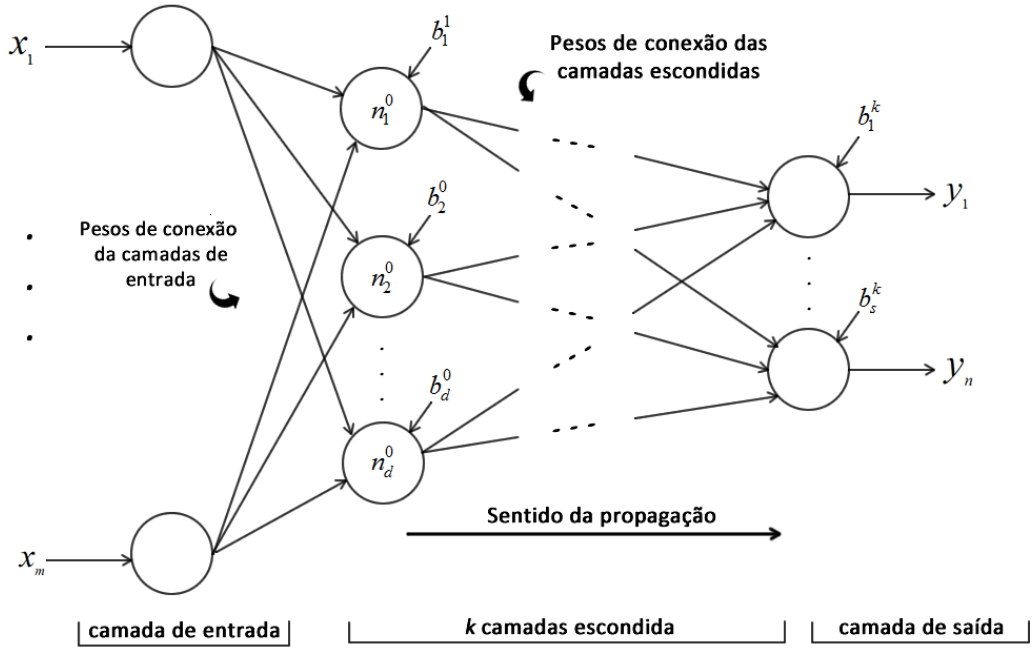
- Como modelar uma rede neural?

Classificadores - FNN



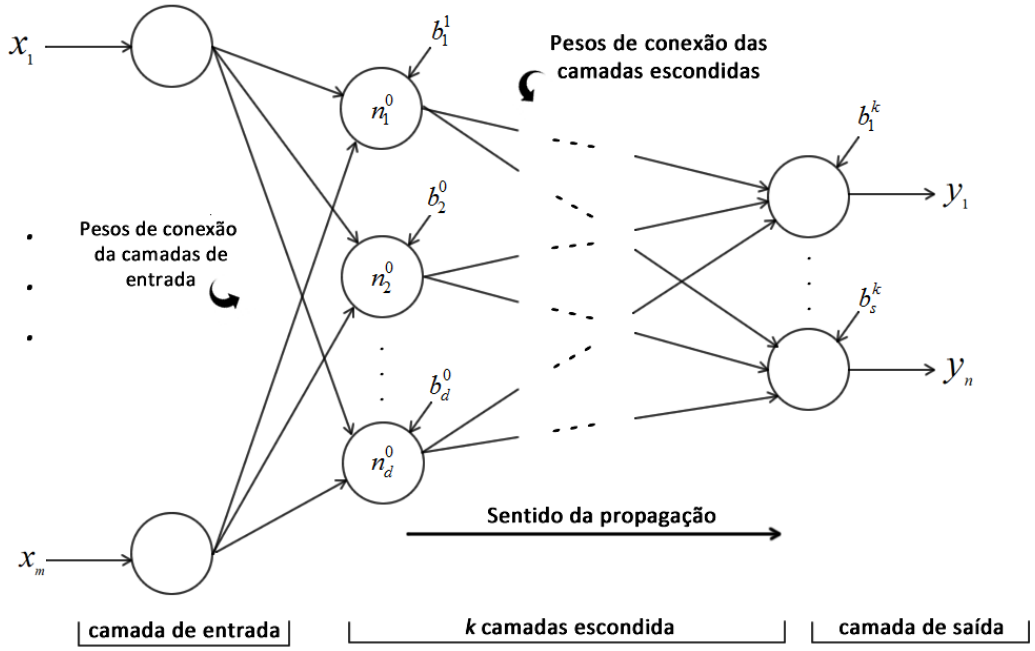
$$[n_1^0, n_2^0, \cdots, n_d^0] = [x_1, \cdots, x_m] \begin{bmatrix} w_{11}^0 & \cdots & w_{1d}^0 \\ \vdots & \ddots & \vdots \\ w_{m1}^0 & \cdots & w_{md}^0 \end{bmatrix} + [b_1^0, b_2^0, \cdots, b_d^0]$$

Classificadores - FNN



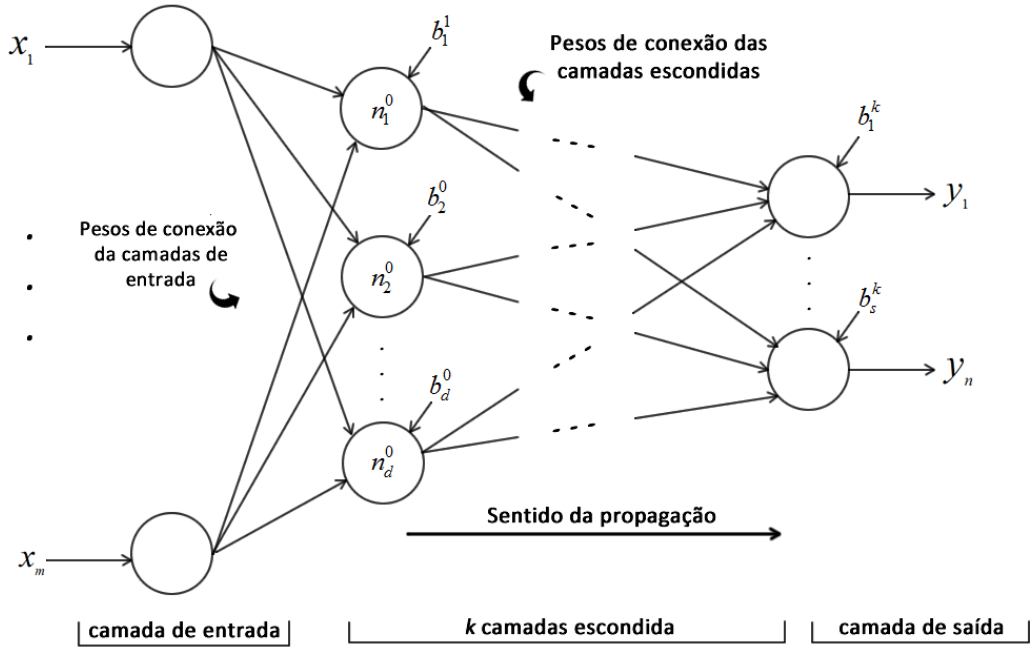
$$[n_1^0, n_2^0, \cdots, n_d^0] = [x_1, \cdots, x_m, 1] \begin{bmatrix} w_{11}^0 & \cdots & w_{1d}^0 \\ \vdots & \ddots & \vdots \\ w_{m1}^0 & \cdots & w_{md}^0 \\ b_1^0 & \cdots & b_d^0 \end{bmatrix}$$

Classificadores - FNN



- E o número de camadas e neurônios?

Classificadores - FNN



- E o número de camadas e neurônios?
  - Neurônios de entradas e saídas são determinados pelo problema.
  - Número de camada(s) ocultas e número de neurônios ocultos são determinados de maneira **empírica.**

Classificadores - FNN

- Como determinar os pesos e *bias* da rede neural?

Classificadores - FNN

- Como determinar os pesos e *bias* da rede neural?
  - É necessário um processo de **treinamento**

- O processo de treinamento mais utilizado é *backpropagation*, na qual a ideia é minimizar o erro entre saída obtida e a desejado por meio de um gradiente de maneira iterativa.

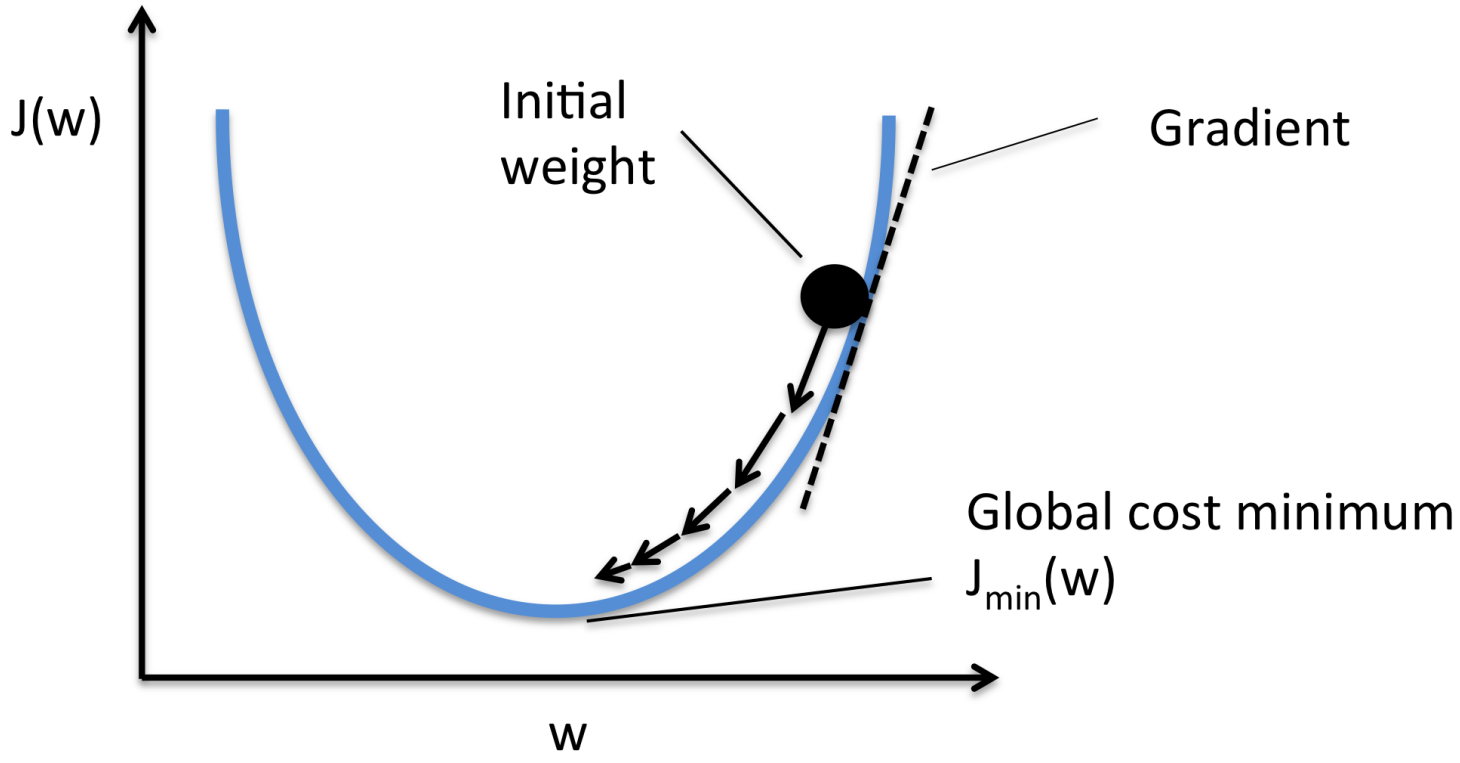
- Uma medida de erro comum: $E = \frac{1}{T}\sum_{a=1}^{T}(\hat{y}_t - y_t)^2$

- A partir do erro é gerada uma regra de atualização dos pesos e bias:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \Delta\mathbf{W}^t \quad \rightarrow \quad \Delta\mathbf{W} = -\eta\frac{\partial E}{\partial \mathbf{W}}$$

Classificadores - FNN

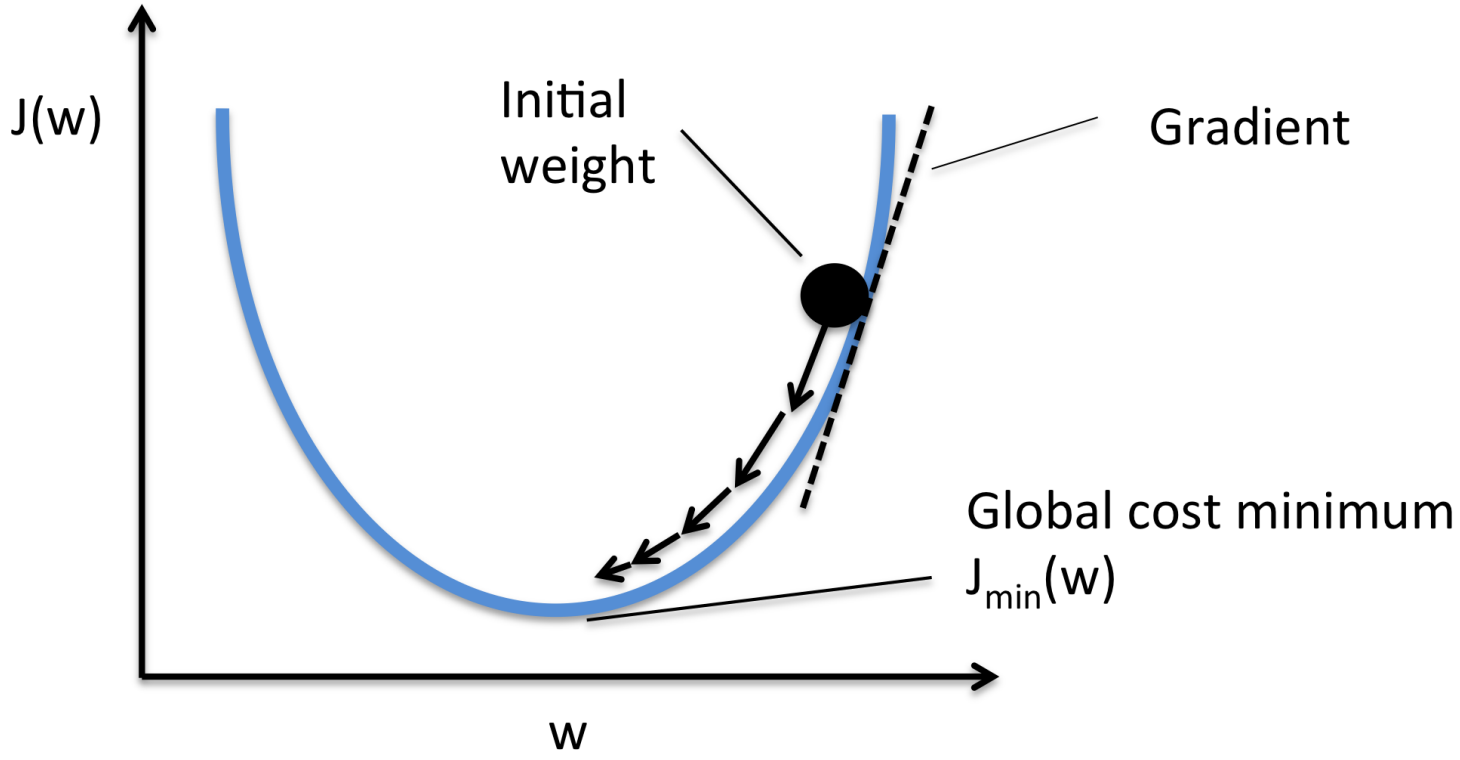
- A ideia do gradiente descendente:



$$\mathbf{W}^{t+1} = \mathbf{W}^{t} + \Delta\mathbf{W}^{t} \quad \rightarrow \quad \Delta\mathbf{W} = -\eta\frac{\partial E}{\partial \mathbf{W}}$$

Classificadores - FNN

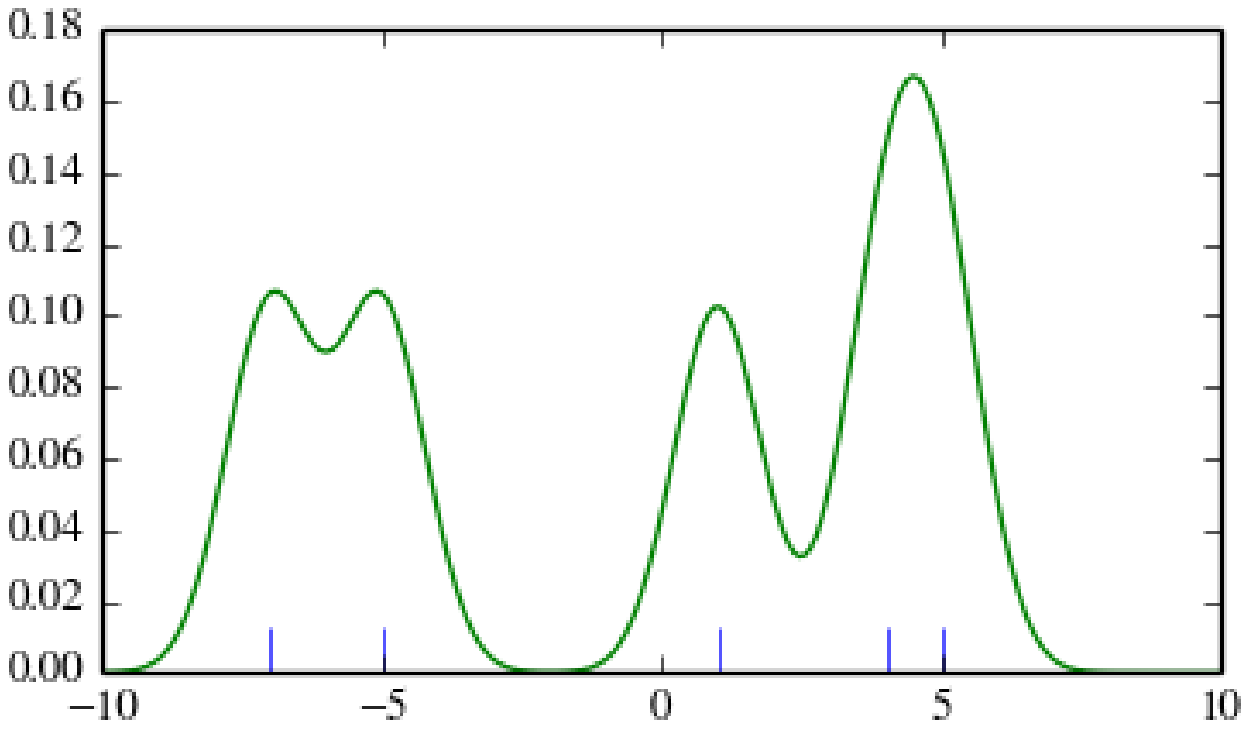
- A ideia do gradiente descendente:
  - Conseguem perceber a influência da taxa de aprendizado?



$$\mathbf{W}^{t+1} = \mathbf{W}^{t} + \Delta\mathbf{W}^{t} \quad \rightarrow \quad \Delta\mathbf{W} = -\eta\frac{\partial E}{\partial \mathbf{W}}$$

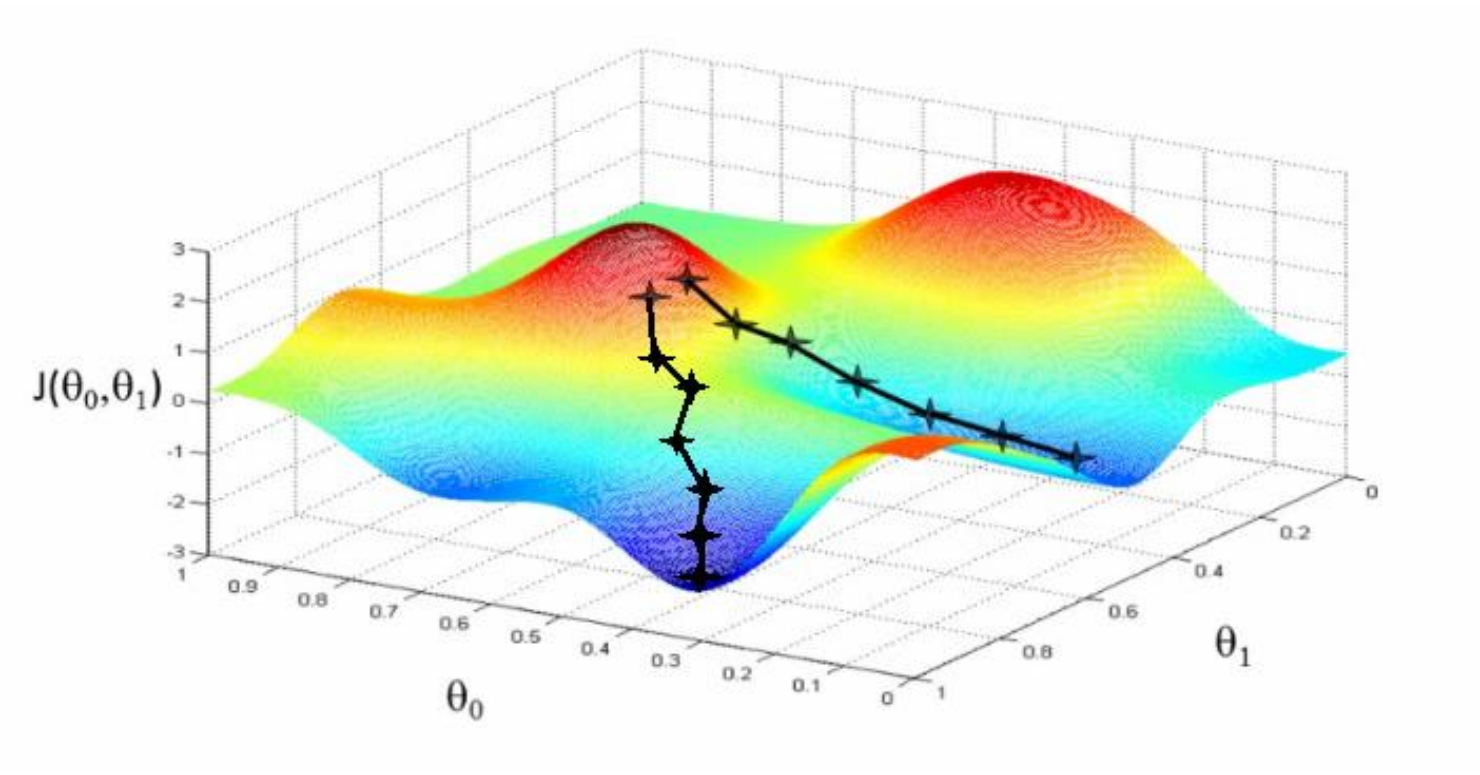
Classificadores - FNN

- A ideia do gradiente descendente:
  - Mínimos locais



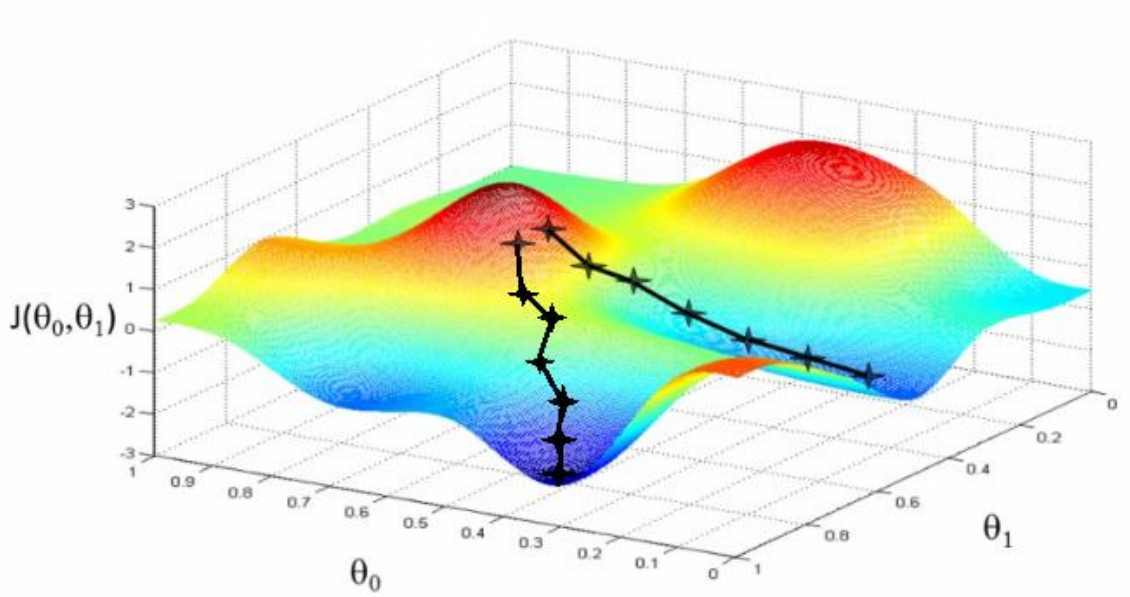
Classificadores - FNN

- A ideia do gradiente descendente:
  - O treinamento é **estocástico!**
  - Por conta disso, é executado várias vezes para cálculo estatístico.



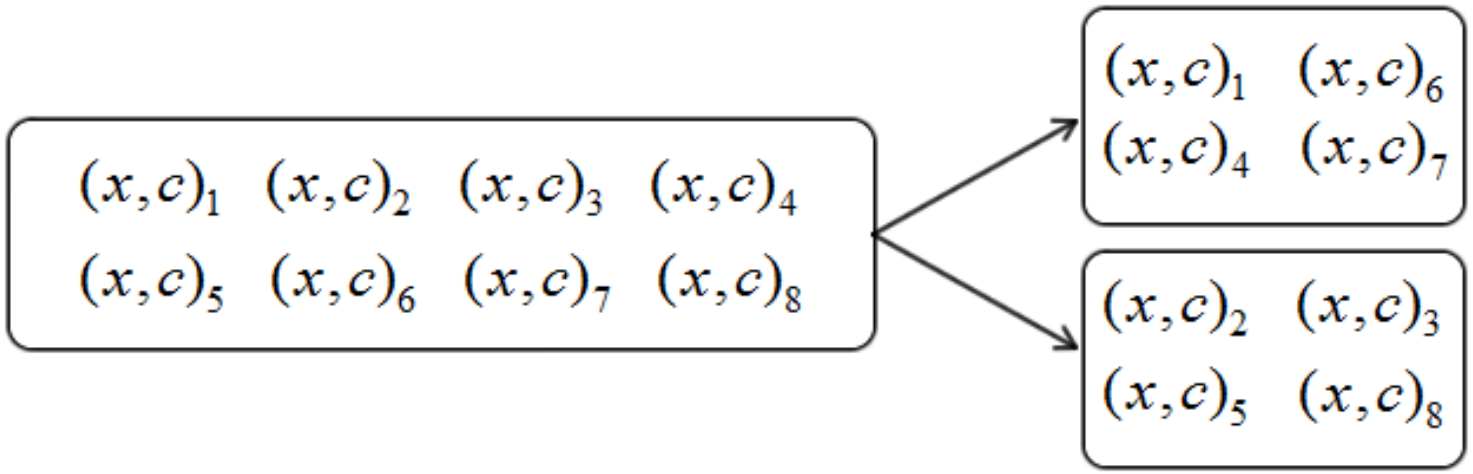
Classificadores - FNN

- Principais deficiências do *backpropagation*
  - Cai muito em mínimos locais, pois depende da inicialização.
  - É um algoritmo muito lento.



Classificadores - FNN

- Uma técnica comumente utilizada para auxiliar no treinamento de uma rede neural é o *mini-batch*
  - Agiliza o processo de treinamento.
  - Auxilia para evitar *overfiting.*



Classificadores - FNN

- Pseudocódigo da FNN:

Algoritmo : *backpropagation*

```
1  inicialização:
2      Preparar conjunto de dados de entrada e saída T ;
3      Informar número de camadas ocultas e neurônios para cada uma dela(s);
4      Informar η;
5  repita
6      para cada amostra do conjunto de treinamento T faça
7          para cada peso de conexão w faça
8              Calcular ΔW;
9              Atualizar W;
10         fim para
11     fim para
12 Até número de iterações pré-determinado ou erro mínimo satisfeito
13 retornar: pesos e bias treinados
```

Classificadores - FNN

- Desempenho da FNN para *Iris*:

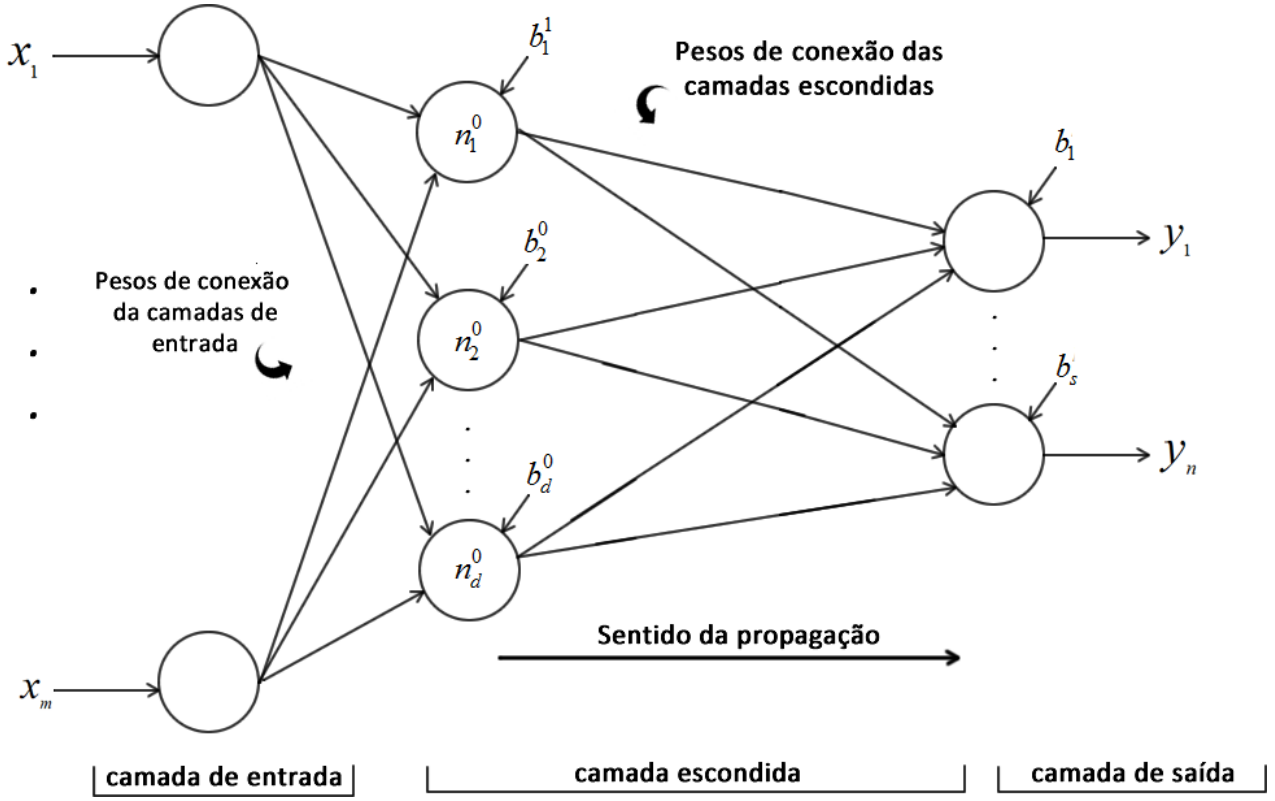
Base de dados	Acurácia (%)	Acurácia (erros)	Desvio Padrão (%)
Iris	95,70	1,93	1,42

- Matriz de confusão:

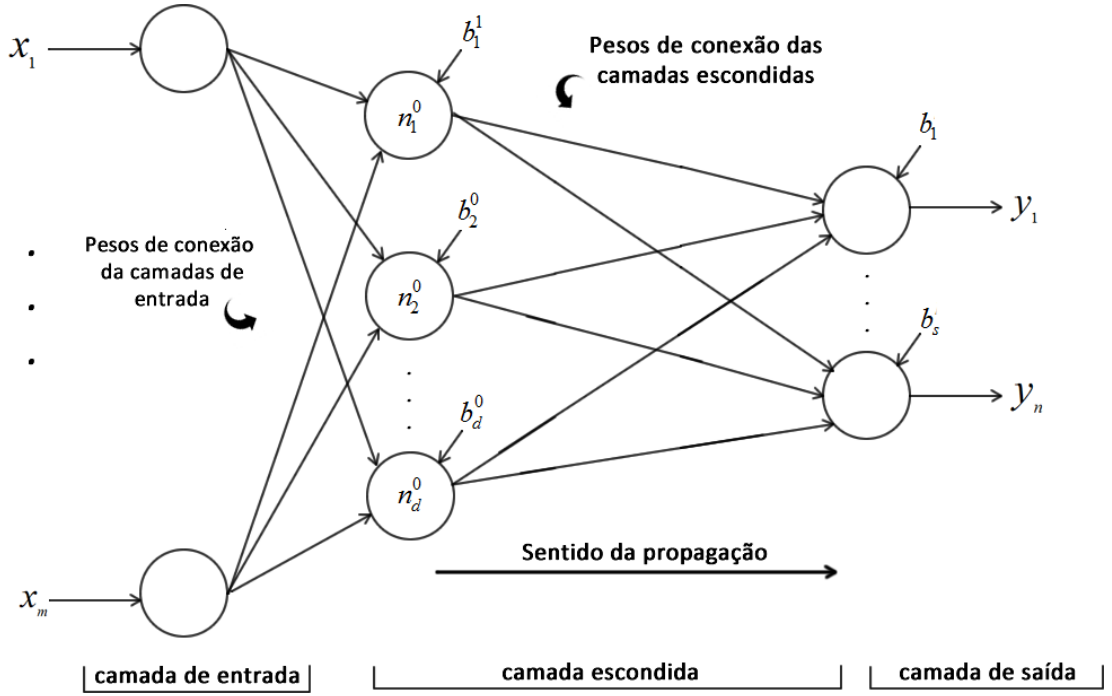
	Virginica	Setosa	Versicolor
Virginica	**13**	0	0
Setosa	0	**12,2667**	1,7333
Versicolor	0	0,2	**17,80**

Classificadores - ELM

- A máquina de aprendizado extremo nada mais é do que um algoritmo de aprendizado baseado em uma rede neural de apenas uma camada
  - Método não iterativo
  - Treinamento extremamente mais rápido do que o *backpropagation*



Classificadores - ELM



$$\mathbf{X} = [x_1, \cdots, x_m, 1] \qquad \mathbf{W} = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{md} \\ b_1 & \cdots & b_d \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1s} \\ \vdots & \ddots & \vdots \\ \beta_{d1} & \cdots & \beta_{ds} \end{bmatrix} \qquad \mathbf{Y} = [y_1, \cdots, y_s]$$

Classificadores - ELM

- As entradas, saídas, pesos e *bias* da rede são organizados na seguinte forma

$$\mathbf{X} = [x_1, \cdots, x_m, 1] \qquad \mathbf{W} = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{md} \\ b_1 & \cdots & b_d \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1s} \\ \vdots & \ddots & \vdots \\ \beta_{d1} & \cdots & \beta_{ds} \end{bmatrix} \qquad \mathbf{Y} = [y_1, \cdots, y_s]$$

- O objetivo do treinamento da ELM é determinar a matriz de pesos β. A matriz **W** é obtida de maneira aleatória e por meio dela e da entrada calcula-se a matriz **H**

$$\mathbf{H}^i = [x_1^i, \cdots, x_m^i, 1] \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{md} \\ b_1 & \cdots & b_d \end{bmatrix} \Rightarrow \mathbf{H} = \begin{bmatrix} f(H^1) \\ f(H^2) \\ \vdots \\ f(H^N) \end{bmatrix}_{N \times d}$$

Classificadores - ELM

- Uma vez determinada a matriz **H**, para se obter os pesos da matriz deve ser solucionado o seguinte sistema linear

$$\mathbf{H}\beta = \mathbf{Y} \rightarrow \beta = \mathbf{H}^{\dagger}\mathbf{Y}$$

onde $\mathrm{H}^{\dagger}$ é a inversa generalizada de Moore-Penrose $\mapsto H^{\dagger} = (H^{T}H + \lambda I) - 1$

- Deficiências do ELM:

Classificadores - ELM

- Uma vez determinada a matriz **H**, para se obter os pesos da matriz deve ser solucionado o seguinte sistema linear

$$\mathbf{H}\beta = \mathbf{Y} \rightarrow \beta = \mathbf{H}^{\dagger}\mathbf{Y}$$

onde $\mathrm{H}^{\dagger}$ é a inversa generalizada de Moore-Penrose $\mapsto H^{\dagger} = (H^{T}H + \lambda I) - 1$

- Deficiências do ELM:
  - Sensível a escolha da matriz **W.**
  - Caso a base de dados seja extremamente grande, calcular a inversa generalizada pode ser custoso.

Classificadores - ELM

- Pseudocódigo ELM:

Algoritmo: *treinamento ELM*

inicialização:
    Preparar conjunto de dados de entrada e saída T;
    Informar número de neurônios para camada oculta;
    Inicializar $\mathbf{W}$ de maneira aleatória;
para cada amostra de treinamento do conjunto T **faça**
    Calcular $\mathbf{H}^i$;
fim para
Obter a matriz $\mathbf{H}$ por meio de $\mathbf{H}^i$;
Obter a matriz β por meio do sistema;
retornar: pesos e bias treinados

Classificadores - ELM

- Desempenho da ELM para *Iris*:

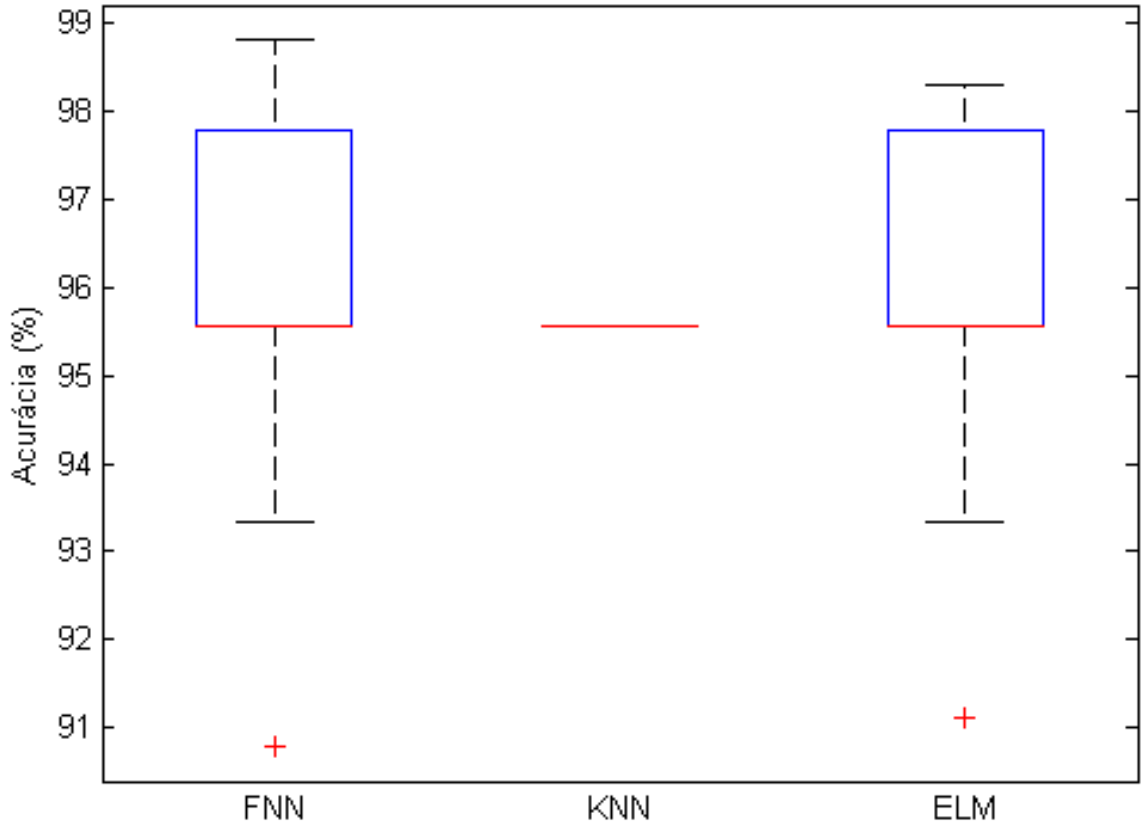
Base de dados	Acurácia (%)	Acurácia (erros)	Desvio Padrão (%)
Iris	95,92	1,83	1,75

- Matriz de confusão:

	Virginica	**Setosa**	**Versicolor**
Virginica	**13**	0	0
Setosa	0	**12,8333**	1,1667
Versicolor	0	0,667	**17,3333**

Classificadores - ELM

- Comparando desempenho dos classificadores por meio de um boxplot:



Elencos de classificadores

- Até então discutimos o uso de 3 classificadores de maneira individual.

- Por que não considerar o resultado de cada um deles?
  - Elencos de classificadores

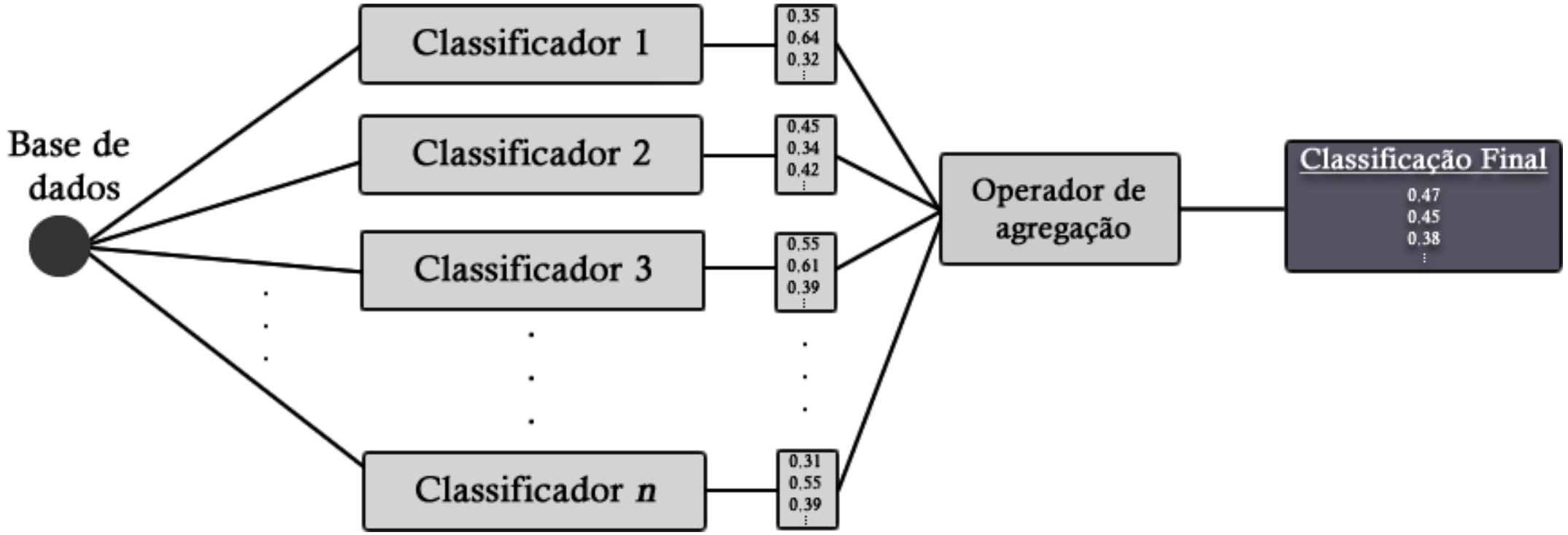
- Sistemas baseados em elencos são inspirados no comportamento humano em tomadas de decisões
  - Decisão de procedimentos médicos
  - Aplicações de investimento
  - Dentre outras

Elencos de classificadores

- Um classificador sozinho quase sempre não é capaz de identificar todas as correlações entre entrada e saída em uma base de dados.

- Portanto, um elenco de classificadores simples pode ser melhor do que um classificador complexo
  - Retorna uma classificação mais confiável
  - Não impede que o elenco tenha um resultado pior

Elencos de classificadores



Elencos de classificadores

- Metodologias de agregação
  - Voto majoritário
  - Media dos classificadores
  - Integral de Choquet

- Atualmente, a agregação de informação vem sendo utilizada em uma gama de problemas:
  - Classificação de dados
  - Predição de séries temporais
  - Tomada de decisão

Resultados experimentais – Grandes bases de dados

- Nesta parte do experimento foram utilizadas as seguintes bases de dados:

Base de Dados	# de amostras	# de atributos	# de classes
DNA	3186	180	3
Covtype	581012	54	7
Higgs	1000000	28	2
Isolet	7797	617	26
Susy	1000000	18	2

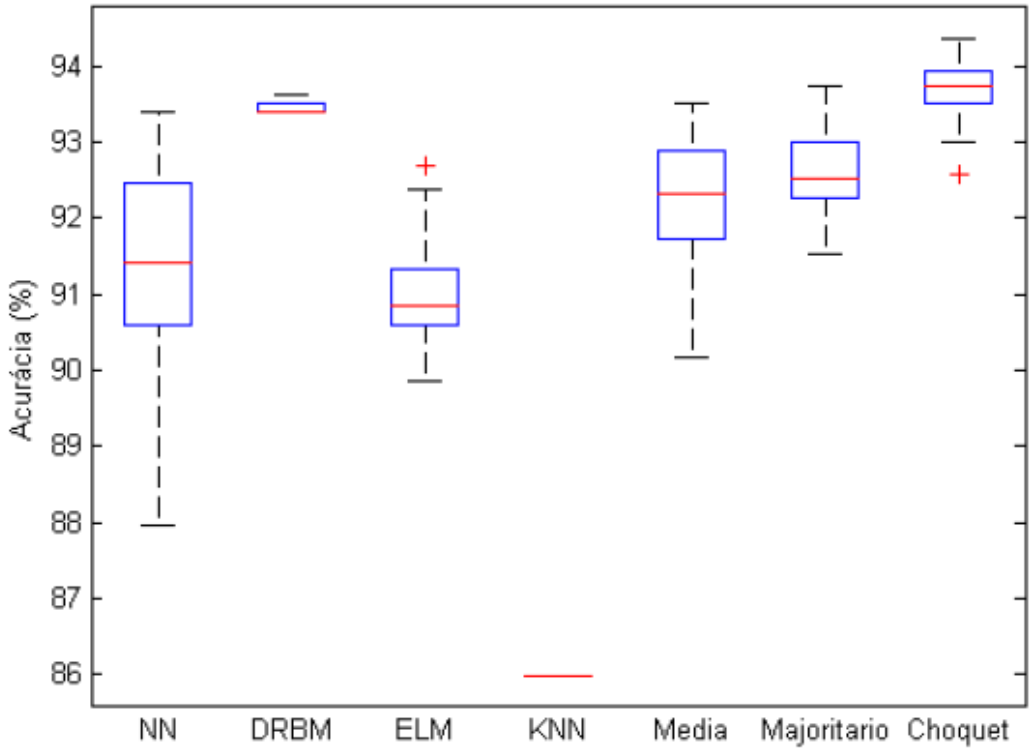
Resultados experimentais – Grandes bases de dados

- O desempenho dos classificadores para as pequenas bases:

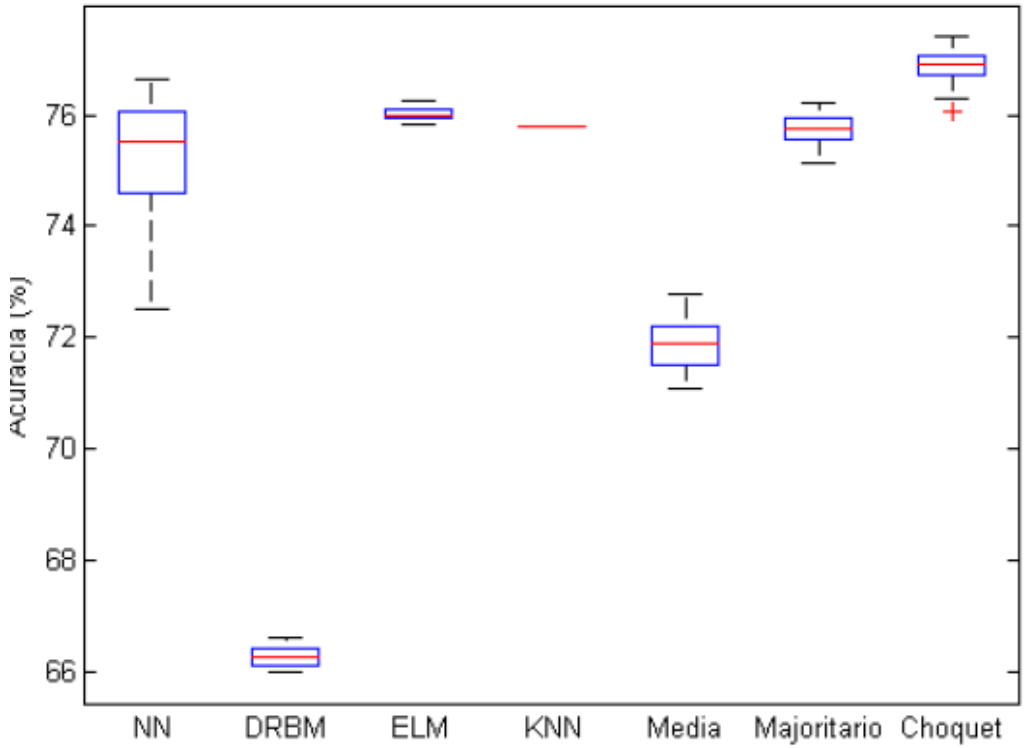
Desempenho em %							
Bases	Classificadores				Agregadores		
	NN	DRBM	ELM	KNN	Média	Major.	Choquet
DNA	$91,36 \pm 1,34$	$93,45 \pm 0,07^*$	$90,59 \pm 0,75$	$85,98 \pm 0$	$92.18 \pm 0,92$	$92.64 \pm 0,57$	$\mathbf{93.69 \pm 0.38}$
Covtype	$75,22 \pm 1,09$	$66,25 \pm 0,17$	$76,01 \pm 0,11^*$	$75,81 \pm 0$	$71,84 \pm 0,42$	$75,75 \pm 0,24$	$\mathbf{76,85 \pm 0,29}$
Higgs	$63,21 \pm 1,19$	$63,40 \pm 0,30$	$63,99 \pm 0,09^*$	$59,84 \pm 0$	$64,01 \pm 0,96$	$63,75 \pm 0,64$	$\mathbf{64,70 \pm 0,72}$
Isolet	$89,41 \pm 1,7$	$93,74 \pm 0,16^*$	$86,81 \pm 0,60$	$88,24 \pm 0$	$93,73 \pm 0,26$	$\mathbf{93.93 \pm 0,31}$	$93.75 \pm 0,16$
Susy	$78,14 \pm 0,65$	$76,39 \pm 0,32$	$\mathbf{79,39 \pm 0,29}$	$70,88 \pm 0$	$78,38 \pm 0,59$	$78,14 \pm 0,54$	$78,58 \pm 0,65$

Resultados experimentais – Grandes bases de dados

- Os *boxplots* do desempenho:



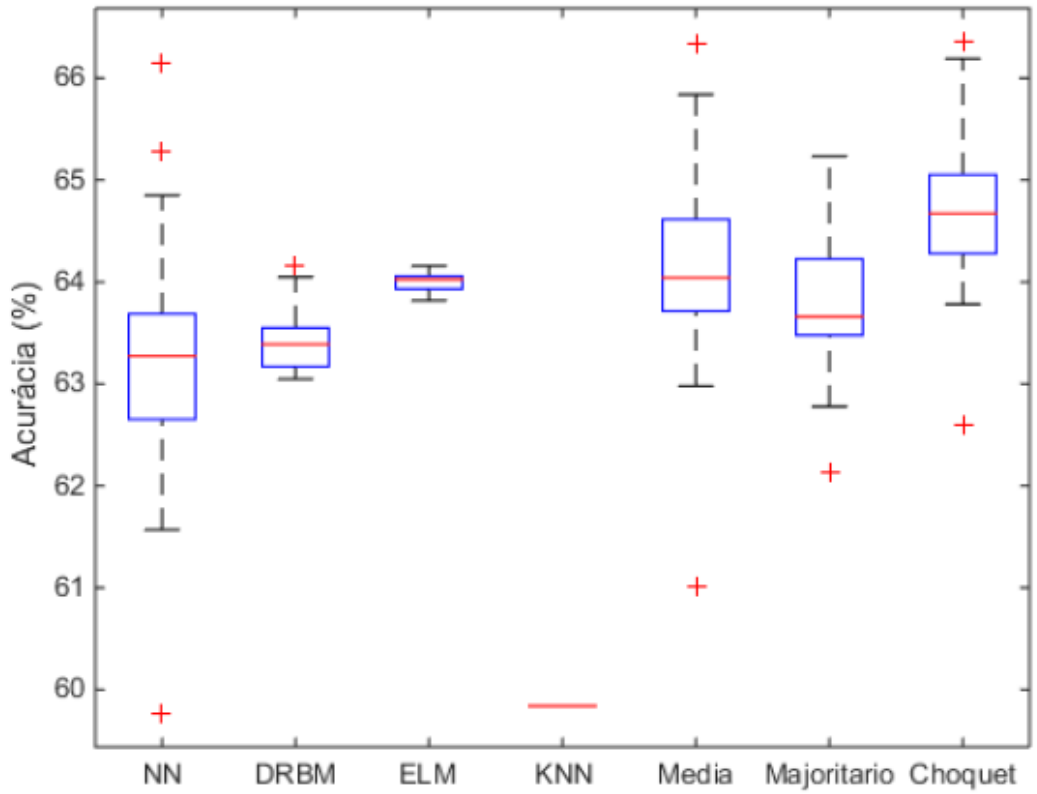
(a) *DNA*



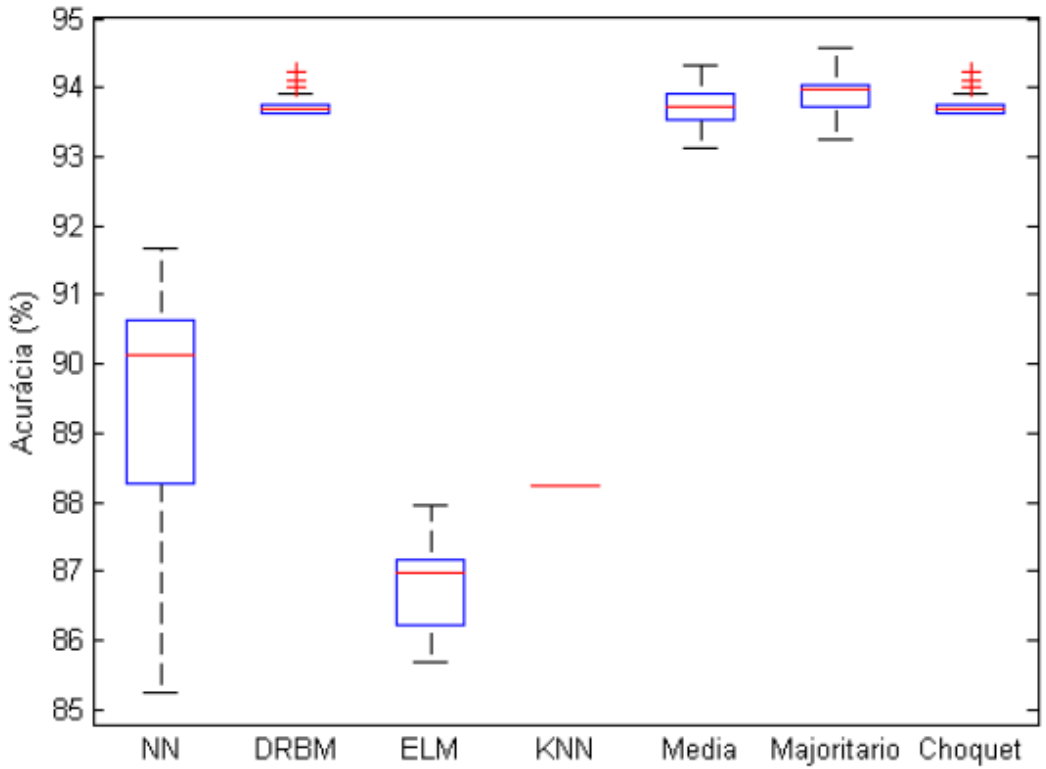
(b) *Covtype*

Resultados experimentais – Grandes bases de dados

- Os *boxplots* do desempenho:



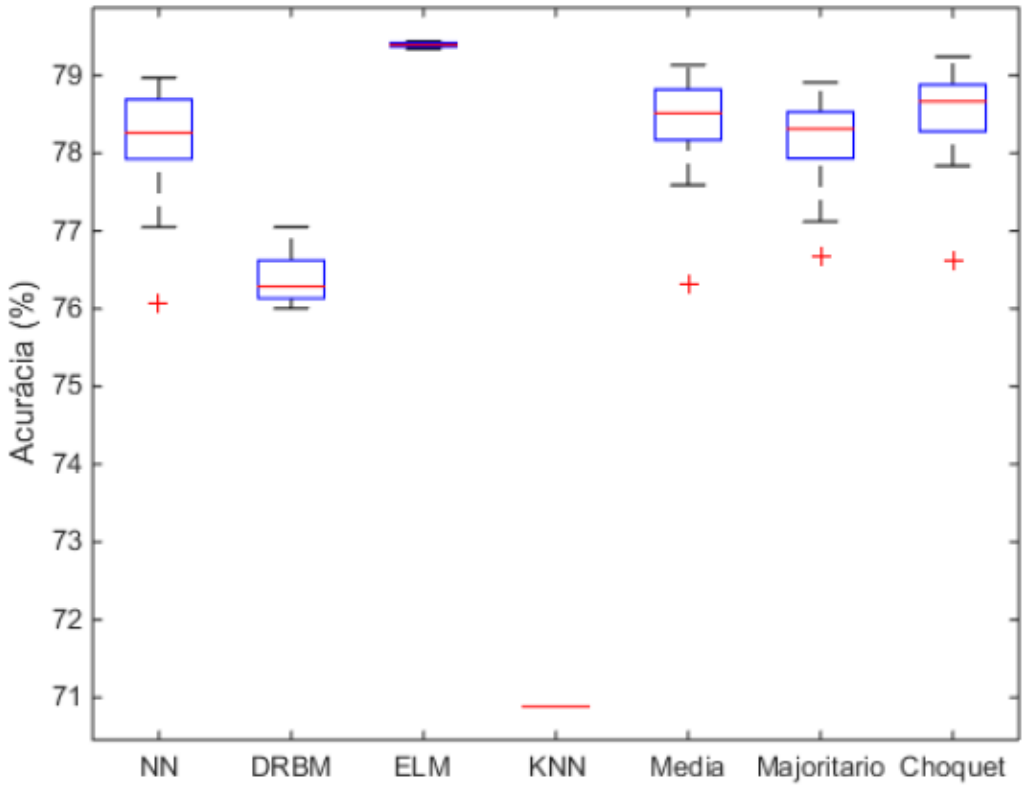
(c) *Higgs*



(d) *Isolet*

Resultados experimentais – Grandes bases de dados

- Os *boxplots* do desempenho:



(e) *Susy*

Resultados experimentais – Grandes bases de dados

- Para as grandes bases são apresentados os tempos computacionais de execução:

Tempos em seg.							
Bases	Classificadores				Agregadores		
	NN	DRBM	ELM	KNN	Média	Major.	Choquet
DNA	$255,33 \pm 12,32$	$184,85 \pm 56,39$	$0,15 \pm 0,08$	$1,52 \pm 0$	$0,07 \pm 0,0$	$0,1 \pm 0,0$	$0,84 \pm 0,01$
Covtype	$1426,5 \pm 157,2$	$127,91 \pm 19,25$	$24,51 \pm 0,60$	$2294,6 \pm 0$	$3,7 \pm 0,17$	$2,4 \pm 0,12$	$64,39 \pm 1,4$
Higgs	$2332,3 \pm 86,54$	$322,32 \pm 115,2$	$41,91 \pm 1,60$	$6748,3 \pm 0$	$5.7 \pm 0,19$	$10,1 \pm 0,25$	$99,4 \pm 1,8$
Isolet	$3440,4 \pm 462,9$	$2839,9 \pm 234,2$	$0,40 \pm 0,02$	$41,31 \pm 0$	$0.43 \pm 0,0$	$0,29 \pm 0,0$	$4,15 \pm 0,1$
Susy	$1712,9 \pm 49,75$	$608,15 \pm 96,68$	$42,97 \pm 1,14$	$5984,6 \pm 0$	$5,9 \pm 0,23$	$9,8 \pm 0,41$	$89,3 \pm 1,91$

Resultados experimentais – Grandes bases de dados

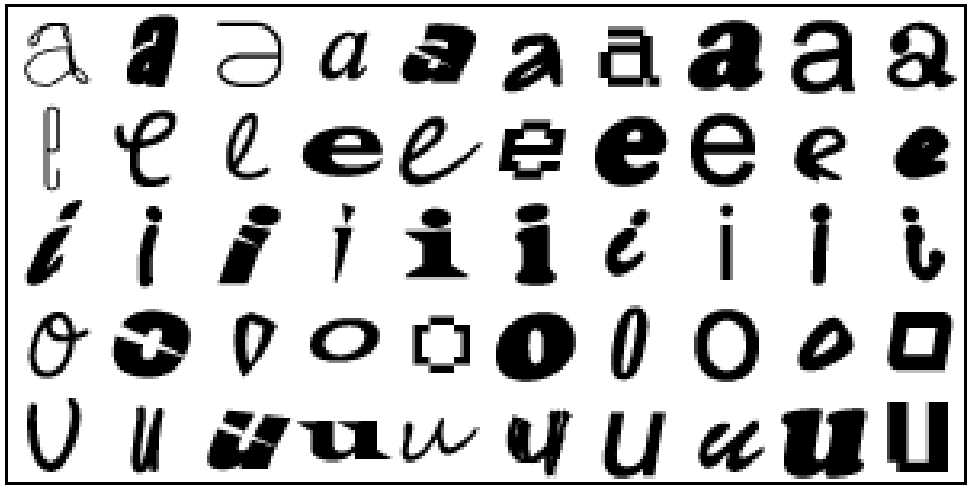
- Para complementar a comparação, foi aplicado o A-TOPSIS. O ranking com variação do peso da média e do desvio padrão do desempenho dos resultados:

Variação peso [media, desvio]	Ranking
[0.5, 0.5]	Choquet $\prec$ Maj. $\prec$ KNN $\prec$ DRBM $\prec$ ELM $\prec$ Media $\prec$ NN
[0.6, 0.4]	Choquet $\prec$ Maj. $\prec$ ELM $\prec$ DRBM $\prec$ KNN $\prec$ Media $\prec$ NN
[0.7, 0.3]	Choquet $\prec$ Maj. $\prec$ Media $\prec$ ELM $\prec$ DRBM $\prec$ KNN $\prec$ NN
[0.8, 0.2]	Choquet $\prec$ Maj. $\prec$ Media $\prec$ ELM $\prec$ NN $\prec$ DRBM $\prec$ KNN
[0.9, 0.1]	Choquet $\prec$ Maj. $\prec$ Media $\prec$ NN $\prec$ ELM $\prec$ DRBM $\prec$ KNN
[1.0, 0.0]	Choquet $\prec$ Maj. $\prec$ Media $\prec$ NN $\prec$ ELM $\prec$ DRBM $\prec$ KNN

Resultados experimentais – Estudo de caso

- Para realizar o estudo de caso foi criada uma base de dados com imagens de vogais utilizando diversas fontes de escritas diferentes

- Cada vogal, representada por uma imagem 30 x 30 pixels, possui 276 amostras de cada vogal totalizando 1380 amostras na base completa



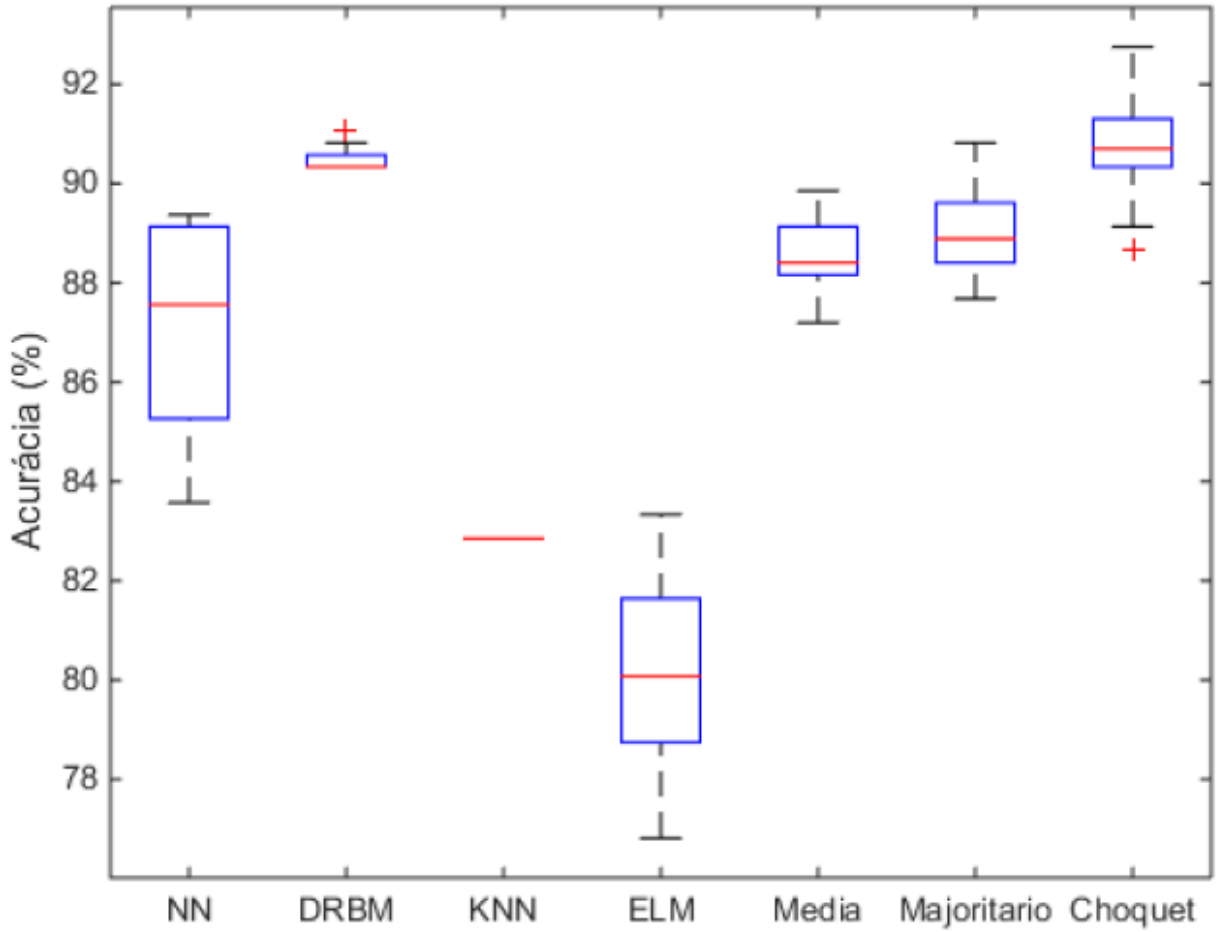
Resultados experimentais – Estudo de caso

- O desempenho dos classificadores para base de dados vogais:

Desempenho em %							
Base	Classificadores				Agregadores		
	NN	DRBM	ELM	KNN	Média	Major.	Choquet
Vogais	$86,90 \pm 2,19$	$90,46 \pm 0,18^{*}$	$80,04 \pm 1,72$	$82,85 \pm 0$	$88,56 \pm 0,74$	$89,09 \pm 0,85$	$\mathbf{90.75 \pm 0.88}$

Resultados experimentais – Estudo de caso

- O *boxplot* do desempenho base de dados vogais:



Resultados experimentais – Estudo de caso

- Para investigar um pouco mais a fundo o desempenho dos classificadores e a agregação via integral de Choquet, são apresentados as matrizes de confusão de cada um dos algoritmos:

Matriz de confusão NN - valores em %					
	a	e	i	o	u
a	**81.3889**	1.6270	2.6190	8.1349	6.2302
e	1.9907	**88.7500**	2.6389	6.3889	0.2315
i	2.2093	0.5039	**96.3178**	0.7752	0.1938
o	8.2479	3.0769	0.5983	**81.7521**	6.3248
u	7.6950	1.0638	0.4610	4.7163	**86.0638**

Matriz de confusão DRBM - valores em %					
	a	e	i	o	u
a	**89.6032**	1.1508	3.6111	4.2857	1.3492
e	2.2222	**90.7407**	4.2130	2.8241	0
i	2.2093	1.2016	**95.5814**	0.7364	0.2713
o	3.4615	1.3675	2.5641	**90.0855**	2.5214
u	6.9504	0.4255	2.0922	3.8652	**86.6667**

Resultados experimentais – Estudo de caso

- Para investigar um pouco mais a fundo o desempenho dos classificadores e a agregação via integral de Choquet, são apresentados as matrizes de confusão de cada um dos algoritmos:

Matriz de confusão ELM - valores em %					
	a	e	i	o	u
a	**71.9048**	3.0556	8.2143	9.2063	7.6190
e	4.0278	**76.2963**	7.5000	10.7870	1.3889
i	0.5426	0.6977	**98.4496**	0.1550	0.1550
o	7.6068	3.5470	7.0085	**78.9316**	2.9060
u	8.9716	1.6312	7.5887	7.5177	**74.2908**

Matriz de confusão KNN - valores em %					
	a	e	i	o	u
a	**59.5238**	3.5714	7.1429	16.6667	13.0952
e	0	**84.7222**	8.3333	5.5556	1.3889
i	0	0	**100.0000**	0	0
o	1.2821	5.1282	3.8462	**85.8974**	3.8462
u	6.3830	1.0638	3.1915	5.3191	**84.0426**

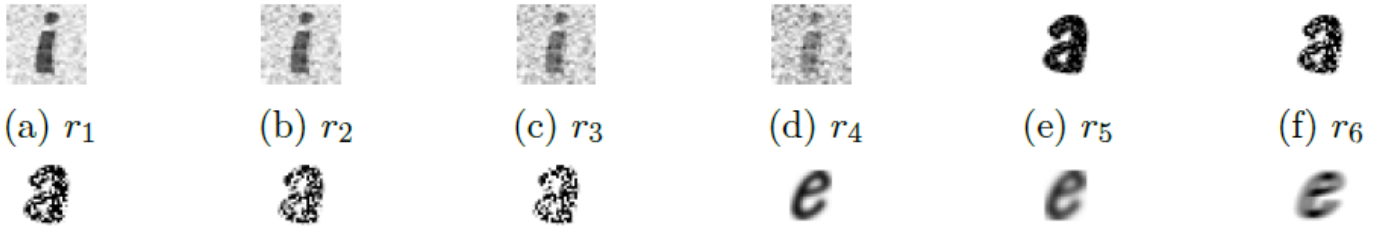
Resultados experimentais – Estudo de caso

- Para investigar um pouco mais a fundo o desempenho dos classificadores e a agregação via integral de Choquet, são apresentados as matrizes de confusão de cada um dos algoritmos:

Matriz de confusão integral de Choquet - valores em %					
	a	e	i	o	u
a	**87.4603**	1.1508	3.2143	5.4762	2.6984
e	1.6204	**90.6944**	3.9352	3.7037	0.0463
i	2.2093	0.9302	**96.6667**	0.1163	0.0775
o	3.1624	1.8803	1.8803	**89.5726**	3.5043
u	5.7447	0.7801	1.2411	3.1206	**89.1135**

Resultados experimentais – Estudo de caso

- Para verificar a robustez dos algoritmos, foi inserido ruído nas vogais:



(a) r_1 (b) r_2 (c) r_3 (d) r_4 (e) r_5 (f) r_6

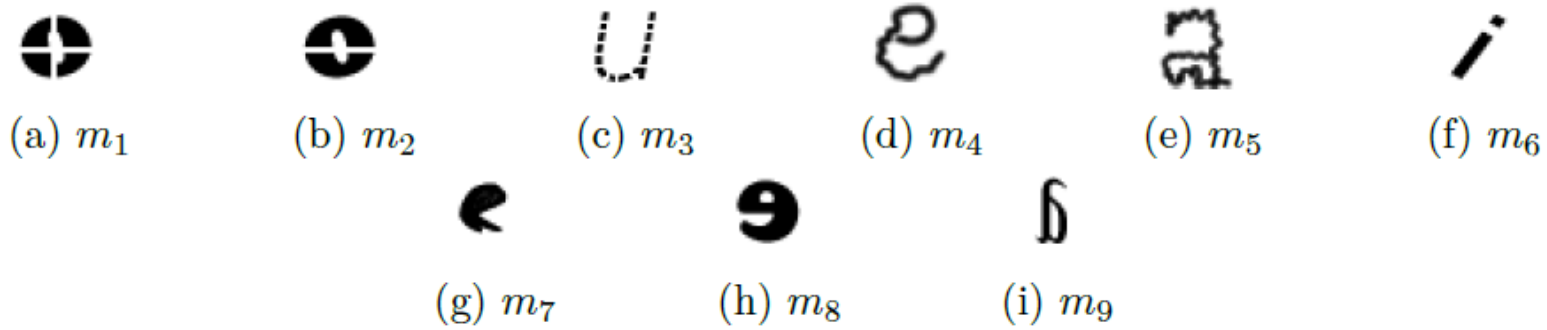
(g) r_7 (h) r_8 (i) r_9 (j) r_{10} (k) r_{11} (l) r_{12}

- A tabela a seguir descreve a classificação de cada classificador para cada amostra, bem como a agregação via Choquet:

Classificadores	Classificação das amostras com ruídos											
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
NN	i	i	i	i	a	a	a	a	o	e	e	e
DRBM	i	i	a	a	a	a	a	a	a	e	e	e
ELM	i	i	e	e	a	a	o	o	o	o	o	o
KNN	i	i	i	i	a	a	a	o	o	e	e	o
Choquet	i	i	i	i	a	a	a	a	a	e	e	e

Resultados experimentais – Estudo de caso

- Finalizando o estudo de caso, foram realizadas modificações nas vogais:



(a) m_1 (b) m_2 (c) m_3 (d) m_4 (e) m_5 (f) m_6

(g) m_7 (h) m_8 (i) m_9

- A tabela a seguir descreve a classificação de cada classificador para cada amostra, bem como a agregação via Choquet:

Classificadores	Classificação das amostras								
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
NN	e	e	**u**	a	u	**i**	**e**	a	i
DRBM	**o**	**o**	**u**	a	u	**i**	**e**	a	i
ELM	e	e	**u**	o	o	**i**	**e**	o	i
KNN	**o**	**o**	**u**	**e**	o	**i**	o	o	i
Choquet	**o**	**o**	**u**	**e**	u	**i**	**e**	o	i

Obrigado pela atenção

Universidade Federal do Espírito Santo
Programa de pós-graduação em informática

André Pacheco