

Aprendizado Supervisionado de Máquina

Naive Bayes, MLP e Ensembles

João Carlos Xavier Júnior

jcxavier@imd.ufn.br

Naive Bayes

Teorema de Bayes

- **Thomas Bayes** foi um matemático inglês.



Teorema de Bayes

- ❑ Muitas vezes, uma informação é apresentada na forma de **probabilidade condicional**.
- ❑ Exemplo:
 - ❖ Qual a **probabilidade** de um evento ocorrer dada uma condição?
 - A probabilidade de um evento **B** ocorrer, sabendo qual será o resultado de um evento **A**.
- ❑ Esse tipo de problema é tratado usando o **Teorema de Bayes**.

Teorema de Bayes

- O Teorema de Bayes relaciona as probabilidades de **A** e **B** com suas respectivas probabilidades condicionadas:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}, \text{ para } P(B) > 0$$

- Onde:

- ❖ $P(A)$ e $P(B)$: probabilidades **a priori** de **A** e **B**;
- ❖ $P(B|A)$ e $P(A|B)$: probabilidades **a posteriori** de **B** condicional a **A**, e de **A** condicional a **B**, respectivamente.

Teorema de Bayes

□ Probabilidade a priori:

- ❖ Probabilidade dada **sem conhecimento** de qualquer outro evento.
- ❖ Qual probabilidade de tirar um **número par** em um dado?



Teorema de Bayes

□ Probabilidade a priori:

- ❖ Probabilidade dada **sem conhecimento** de qualquer outro evento.
- ❖ Qual probabilidade de tirar um **número par** em um dado?
- ❖ $P(\text{par}) = \{1, 2, 3, 4, 5, 6\} \Rightarrow 3,$
- ❖ Logo, $P(\text{par}) = 3/6 \Rightarrow 1/2$



Teorema de Bayes

- Probabilidade a posteriori:
 - ❖ É a probabilidade condicional que é atribuída quando um **evento relevante** é considerado.
 - ❖ Um dado lançado n vezes, veremos que a distribuição dos possíveis valores tende ao que foi inicialmente previsto a priori.
 - ❖ Para $n = 1000$, qual seria probabilidade a posteriori???

Teorema de Bayes

□ Probabilidade a posteriori:

- ❖ Para $n = 1000$, qual seria probabilidade a posteriori???
- ❖ $P(1) = 1000/6 \Rightarrow +/- 166$ vezes
- ❖ $P(2) = 1000/6 \Rightarrow +/- 166$ vezes
- ❖
- ❖ $P(6) = 1000/6 \Rightarrow +/- 166$ vezes

Teorema de Bayes

- A probabilidade **a posteriori** para um **padrão** pertencer a uma determinada **classe** pode ser calculado da seguinte forma:

$$Prob\ Posteriori = \frac{Prob\ Priori * Distrib\ Prob}{Evidencia}$$

Teorema de Bayes

□ Exemplo:

- ❖ Um médico sabe que a meningite causa torcicolo em 50% dos casos.
 - Probabilidade **a priori** de qualquer paciente ter meningite: $1/50.000$;
 - Probabilidade **a priori** de qualquer paciente ter rigidez de nuca: $1/20$.
- ❖ Se um paciente tem rigidez de nuca (evidência), qual será a probabilidade **a posteriori** de ele ter **meningite**?

Teorema de Bayes

□ Exemplo:

❖ Dado:

- M: meningite;
- R: rigidez no pescoço.

$$P(M|R) = \frac{P(R|M)P(M)}{P(R)} = \frac{0,5 * 1/50000}{1/20} \\ = 0,0002$$

Naive bayes

❑ Conjunto de Dados “Tempo”:

	Outlook	Temperature	Humidity	Windy	Play
D1	overcast	cool	normal	true	yes
D2	overcast	hot	high	false	yes
D3	overcast	hot	normal	false	yes
D4	overcast	mild	high	true	yes
D5	rainy	cool	normal	false	yes
D6	rainy	mild	high	false	yes
D7	rainy	mild	normal	false	yes
D8	sunny	cool	normal	false	yes
D9	sunny	mild	normal	true	yes
D10	rainy	cool	normal	true	no
D11	rainy	mild	high	true	no
D12	sunny	hot	high	false	no
D13	sunny	hot	high	true	no
D14	sunny	mild	high	false	no

Naive bayes

□ Probabilidades para o Conjunto:

Outlook			Temperature			Humidity			Windy			Play	
	<i>Yes</i>	<i>No</i>		<i>Yes</i>	<i>No</i>		<i>Yes</i>	<i>No</i>		<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Naive bayes

□ Para um novo dia:

Table 4.3 A new day.				
Outlook	Temperature	Humidity	Windy	Play
sunny	cool	high	true	?

$$P(\text{play} = \text{yes} \mid E) = \frac{P(E \mid \text{play} = \text{yes})P(\text{play} = \text{yes})}{P(E)}$$

$$\therefore P(\text{Outlook} = \text{sunny} \mid \text{play} = \text{yes}) = 2/9$$

$$P(\text{Temperature} = \text{cool} \mid \text{play} = \text{yes}) = 3/9$$

$$P(\text{Humidity} = \text{high} \mid \text{play} = \text{yes}) = 3/9$$

$$P(\text{Windy} = \text{true} \mid \text{play} = \text{yes}) = 3/9$$

$$P(\text{play} = \text{yes}) = 9/14$$

$$\therefore P(E \mid \text{play} = \text{yes}) * P(\text{play} = \text{yes}) = (2/9) (3/9) (3/9) (3/9)(9/14) = 0.0053$$

$$P(E \mid \text{play} = \text{no}) * P(\text{play} = \text{no}) = (3/5) (1/5) (4/5) (3/5)(5/14) = 0.0206$$

$$\therefore 0.0206 > 0.0053$$

\therefore For the new day, no is more likely than yes.

Naive bayes

□ Para um novo dia:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Verossimilhança para as duas classes:

Para “yes” = $2/9 * 3/9 * 3/9 * 3/9 * 9/14 = 0.0053$

Para “no” = $3/5 * 1/5 * 4/5 * 3/5 * 5/14 = 0.0206$

Convertendo para probabilidades por meio de normalização:

$P(\text{“yes”}) = 0.0053 / (0.0053 + 0.0206) = \mathbf{0.205 (20,5\%)}$

$P(\text{“no”}) = 0.0206 / (0.0053 + 0.0206) = \mathbf{0.795 (79,5\%)}$

Naive bayes

❑ Valores ausentes:

- ❖ Treinamento: excluir exemplo do conjunto de treinamento;
- ❖ Classificação: omitir atributo com valor ausente do cálculo;
- ❖ Exemplo:

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

Verossimilhança para "yes" = $3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$

Verossimilhança para "no" = $1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$

Chance ("yes") = $0.0238 / (0.0238 + 0.0343) = 41\%$

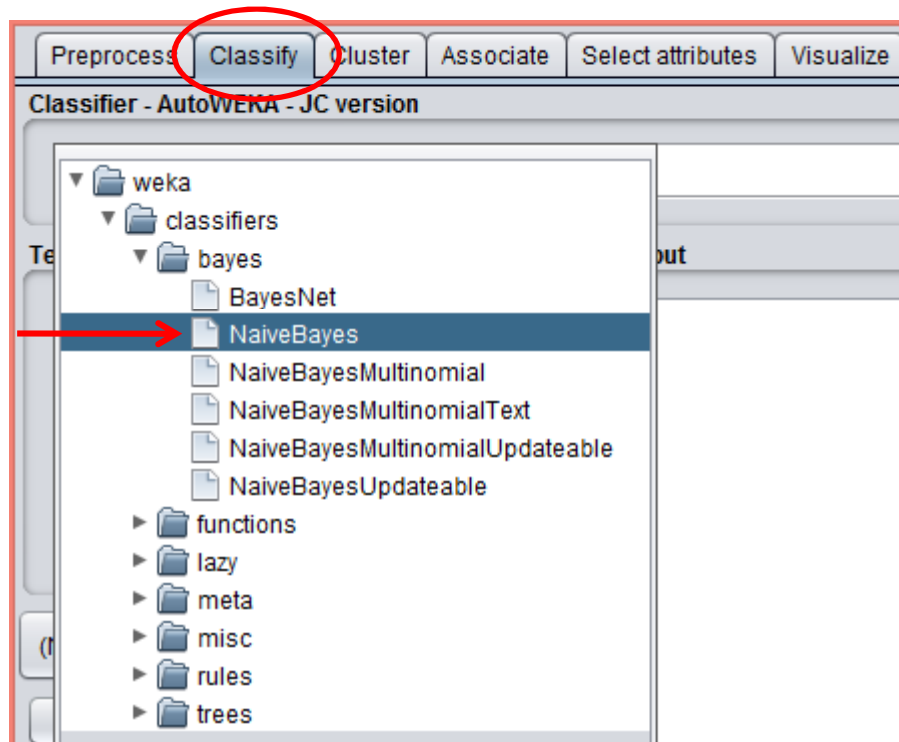
Chance ("no") = $0.0343 / (0.0238 + 0.0343) = 59\%$

Overview

- ❑ Junto com árvores de decisão e vizinhos mais-próximos, é um dos métodos de aprendizagem mais práticos.
- ❑ Quando usá-lo:
 - ❖ Quando se tem disponível um conjunto de treinamento **médio** ou **grande**.
 - ❖ Os atributos que descrevem as instâncias forem condicionalmente independentes.

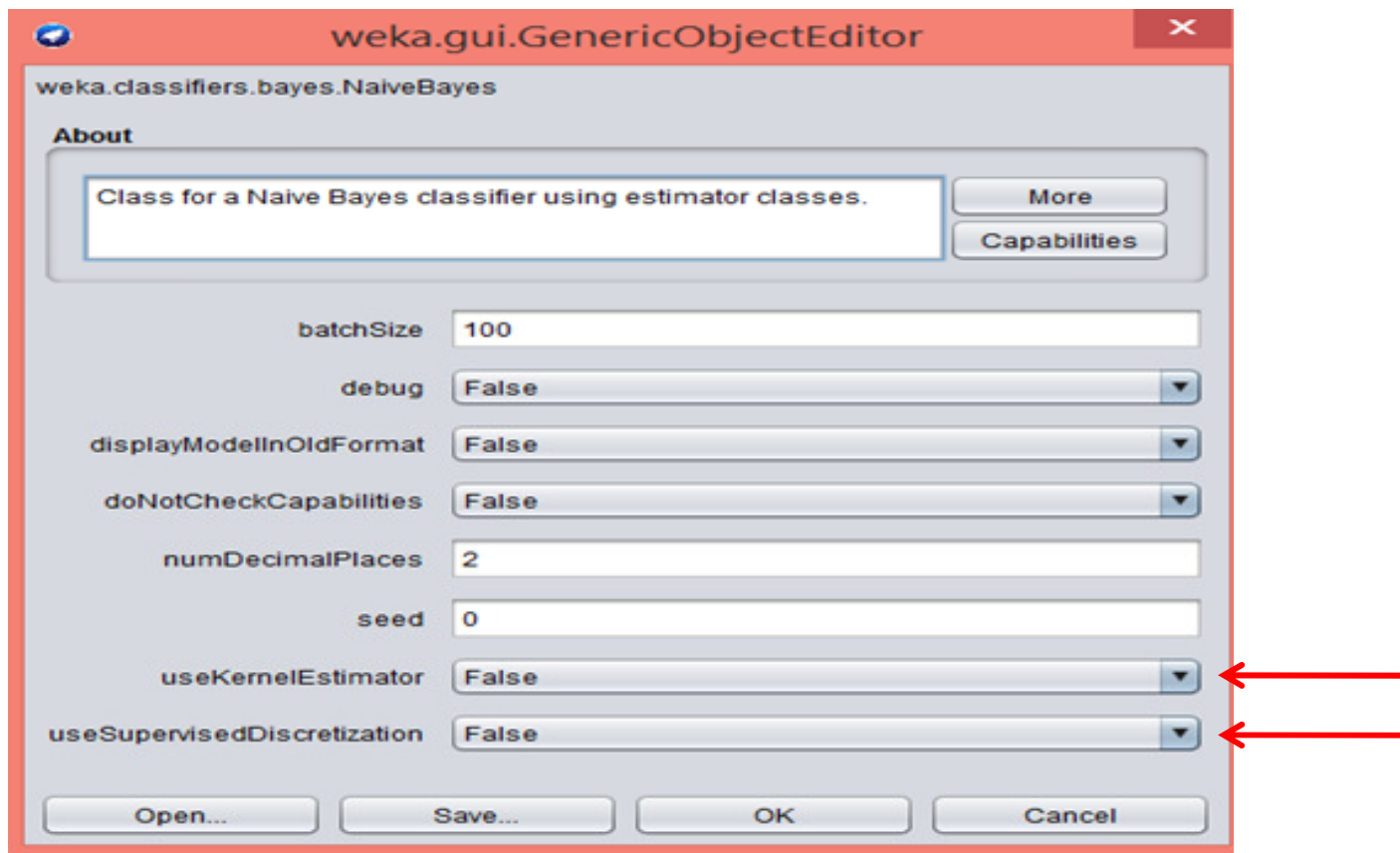
Naive Bayes

- ❑ Utilizando **Naive Bayes** (WEKA):



Naive Bayes

❑ Configurando o Naive Bayes:



Naive Bayes

□ Analisando os resultados....

```
Time taken to build model: 0.02 seconds
```

```
=== Stratified cross-validation ===
```

```
=== Summary ===
```

Correctly Classified Instances	290	82.6211 %
Incorrectly Classified Instances	61	17.3789 %
Kappa statistic	0.6394	
Mean absolute error	0.1736	←
Root mean squared error	0.3935	
Relative absolute error	37.7001 %	
Root relative squared error	82.0203 %	
Total Number of Instances	351	

```
=== Confusion Matrix ===
```

a	b	<-- classified as
181	44	a = g
17	109	b = b

Naive Bayes

❑ Analisando os resultados....

```
Time taken to build model: 0 seconds
```

```
=== Evaluation on test split ===
```

```
Time taken to test model on test split: 0.01 seconds
```

```
=== Summary ===
```

Correctly Classified Instances	85	80.9524 %
Incorrectly Classified Instances	20	19.0476 %
Kappa statistic	0.6194	
Mean absolute error	0.1802	←
Root mean squared error	0.4048	
Relative absolute error	37.434 %	
Root relative squared error	78.9467 %	
Total Number of Instances	105	

```
=== Confusion Matrix ===
```

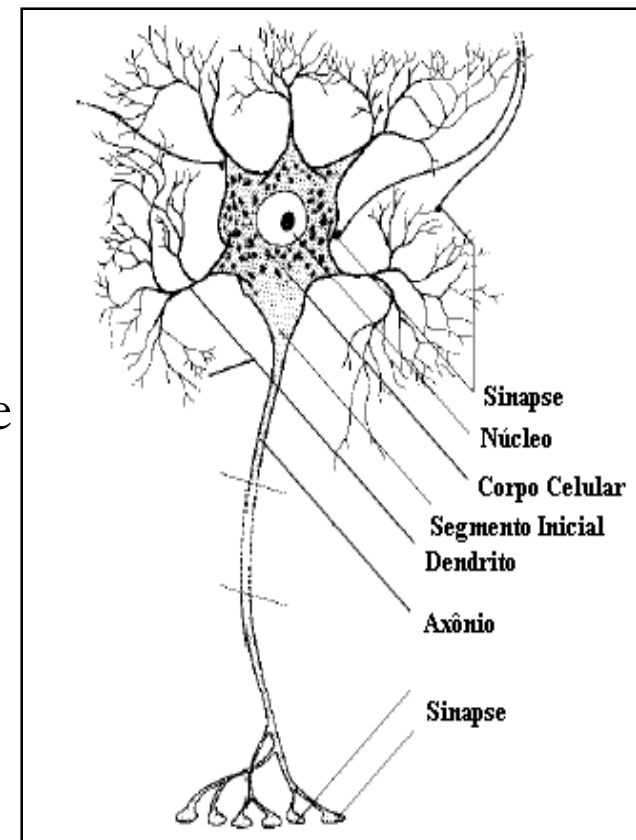
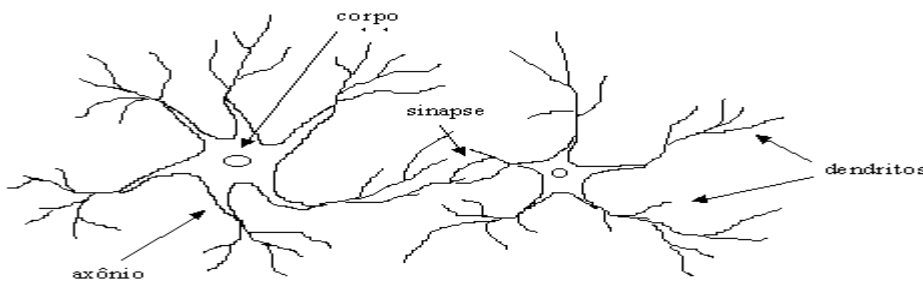
a	b	<-- classified as
45	13	a = g
7	40	b = b

Redes Neurais Artificiais

Redes Neurais Biológicas

□ Cérebro é extremamente eficiente:

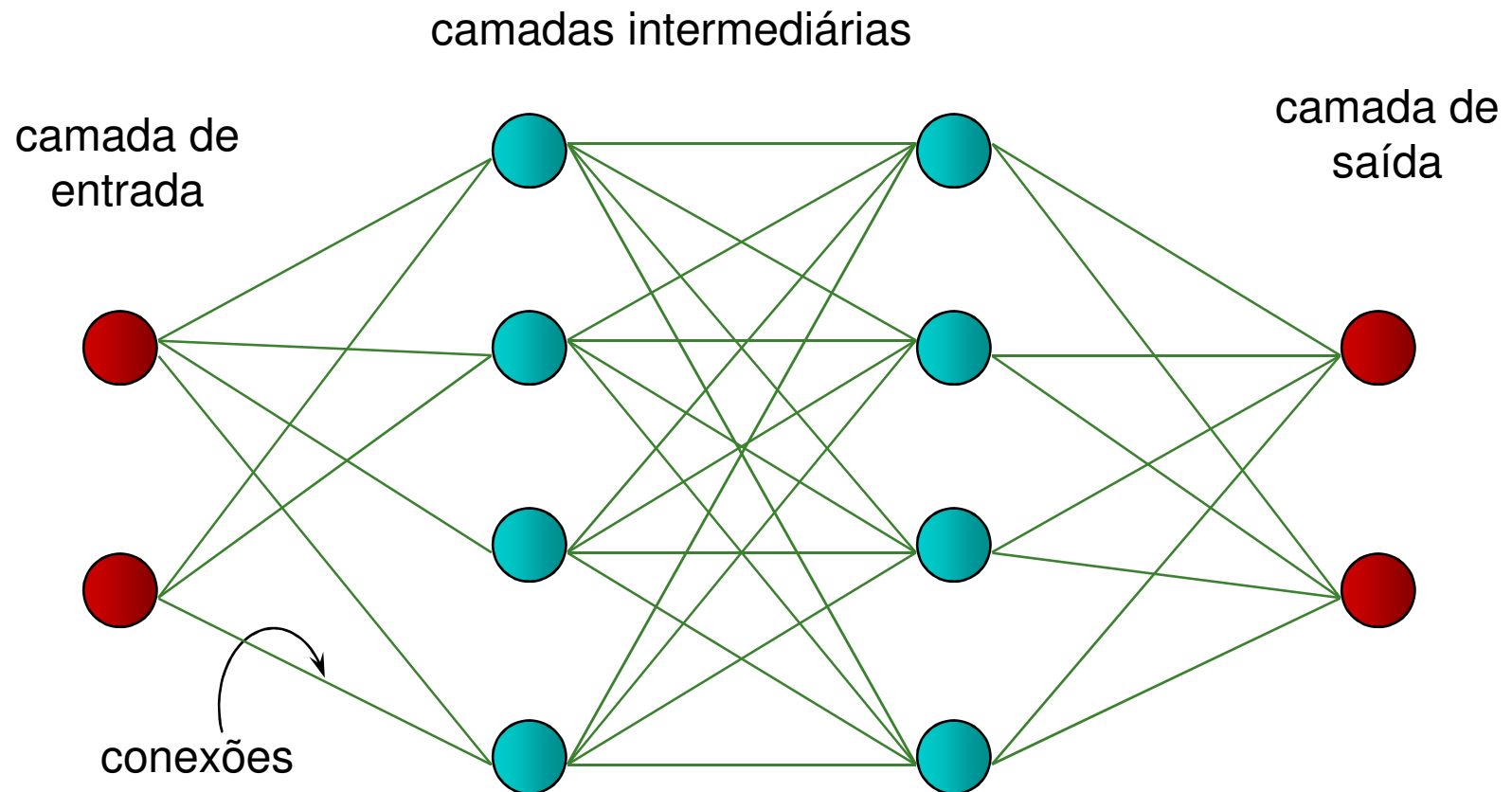
- Eficiência energética do cérebro é de aproximadamente 10^{-16} joules por operação por segundo.
- Nos melhores computadores é cerca de 10^{-6} joules por operação por segundo.



O que são Redes Neurais Artificiais

- ❑ **Redes Neurais Artificiais (RNA)** são modelos de computação com propriedades particulares.
 - Capacidade de se adaptar ou aprender;
 - Generalizar;
 - Agrupar ou organizar dados.

Redes Neurais Artificiais



Redes Neurais Artificiais

- ❑ **Modelos inspirados no cérebro humano:**
 - Compostas por várias unidades de processamento (“neurônios”);
 - Interligadas por um grande número de conexões (“sinapses”).

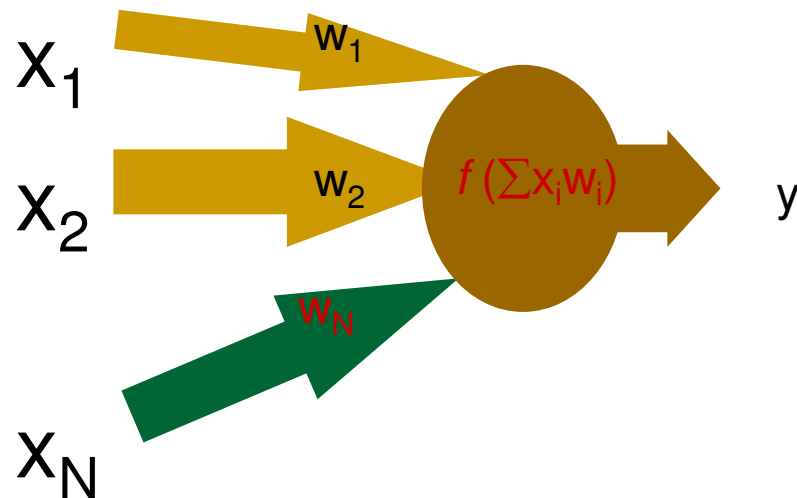
- ❑ **Eficientes onde métodos tradicionais têm se mostrado inadequados.**

Unidades de processamento: Neurônio de McCulloch e Pitts (MP)

❑ **Função:** receber entradas de conjunto de unidades **A**, computar função sobre entradas e enviar resultado para conjunto de unidades **B**.

❑ **Entrada total:**

$$u = \sum_{j=1}^N x_j w_j$$



Unidades de processamento

- Estado de ativação:
 - ❖ Representa o estado dos neurônios da rede.
 - ❖ Pode assumir valores:
 - Binários (0 e 1);
 - Reais.
 - ❖ Definido através de funções de ativação:
 - Adições;
 - Comparações;
 - Transformações matemáticas.

Funções de ativação

□ Funções mais comuns:

❖ Linear

$$a(t + 1) = u(t)$$

❖ Threshold ou limiar

$$a(t + 1) = \begin{cases} 1, & \text{se } u(t) \geq \theta \\ 0, & \text{se } u(t) < \theta \end{cases}$$

❖ Sigmóide

$$a(t + 1) = 1 / (1 + e^{-\lambda u(t)})$$

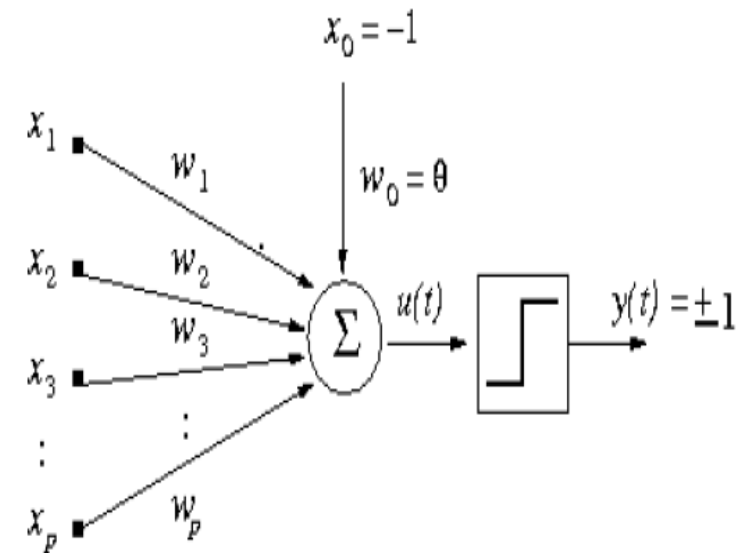
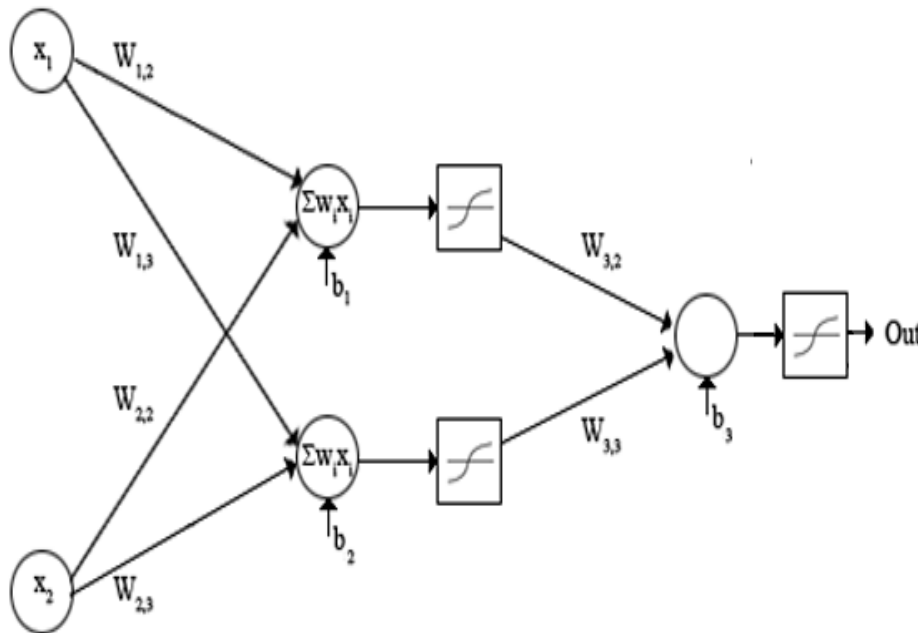
❖ tangente hiperbólica

$$a(t + 1) = \frac{(1 - e^{-\lambda u(t)})}{(1 + e^{-\lambda u(t)})}$$

Topologia

□ Número de camadas:

❖ Uma camada (Ex Perceptron, Adaline)



Perceptrons e Adalines

- ❑ Característica e limitação:
 - ❖ Representam uma superfície de decisão através de um hiperplano.
 - ❖ Resolvem apenas problemas linearmente separáveis.

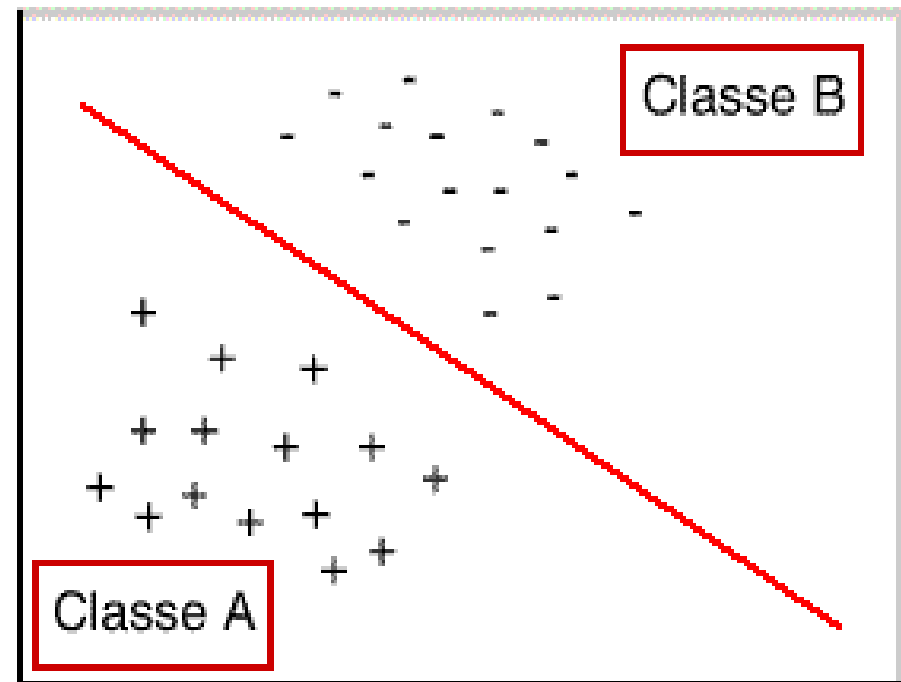


Fig. 1: Classes linearmente separáveis

Perceptrons e Adalines

- ❑ Problemas não linearmente separáveis:

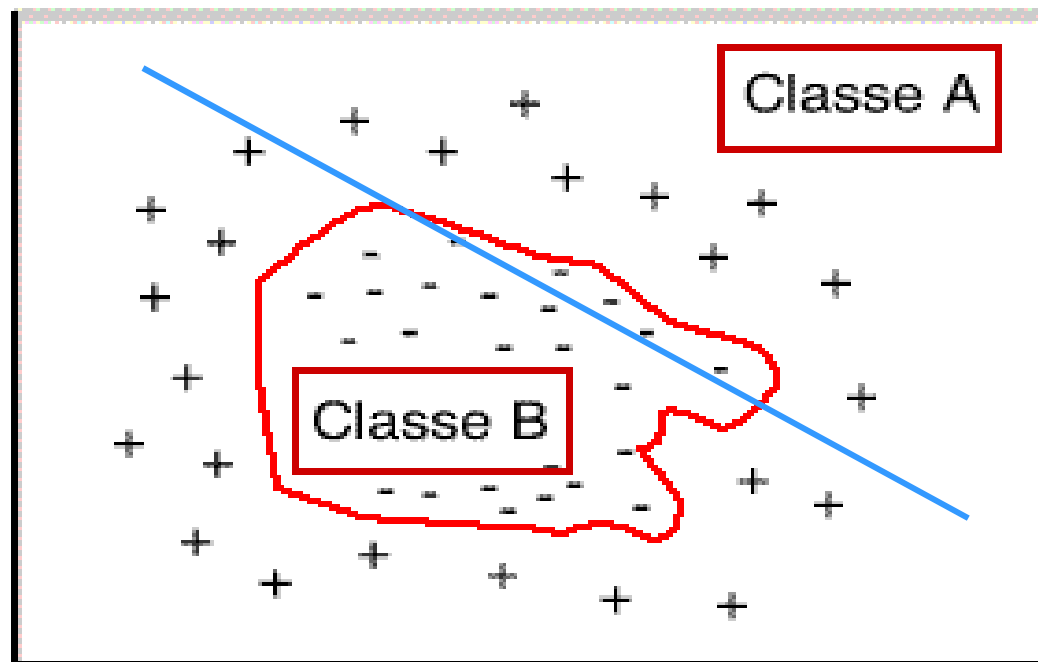
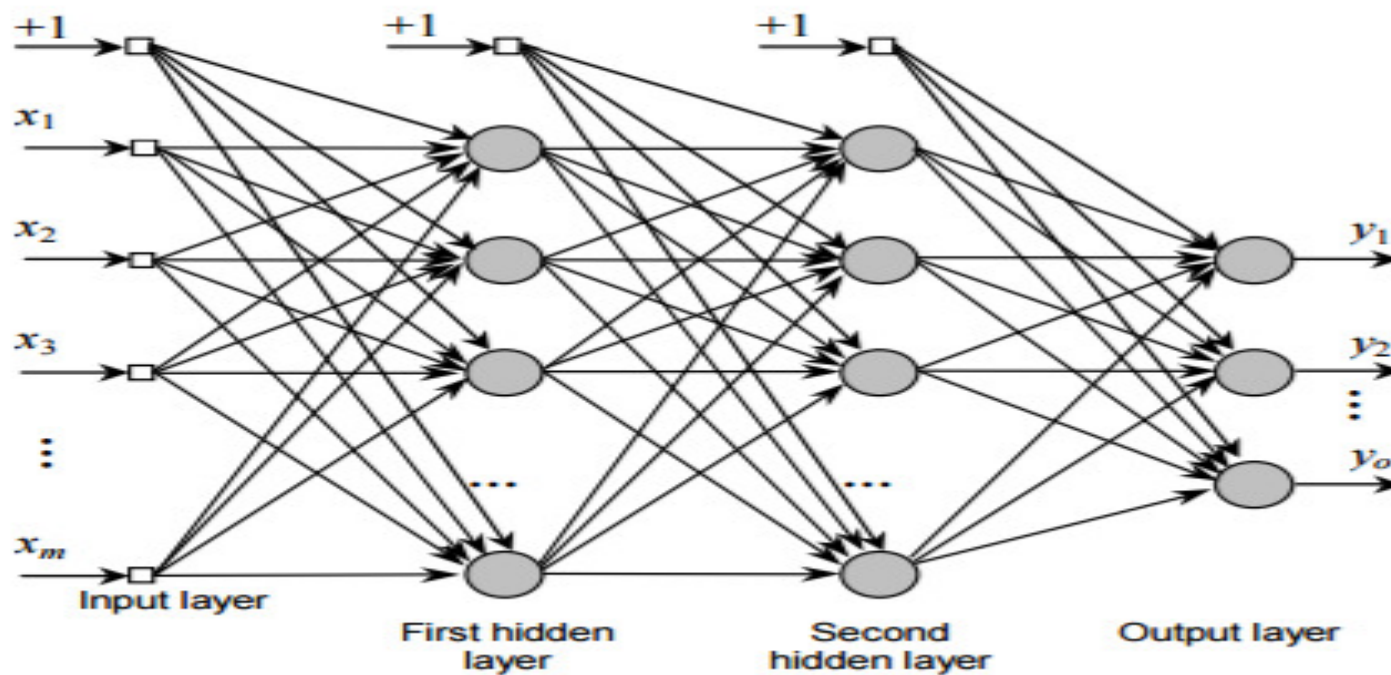


Fig. 2: Classes não linearmente separáveis

Topologia

□ Número de camadas:

❖ Multi-camadas (**Multilayer Perceptron**)



Redes Neurais Artificiais

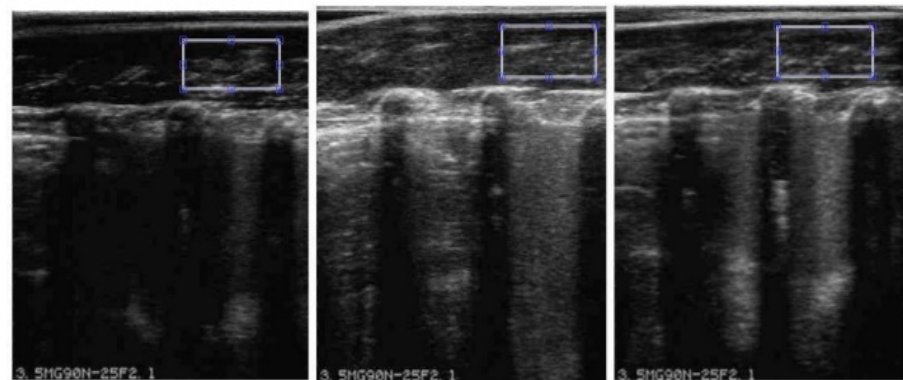
❑ Vantagens:

- ❖ Possuem boa generalização e tolerância a ruídos (dados ruidosos e aberrantes);
- ❖ Apresentam bom desempenho (baixa taxa de erros) quando utilizadas em **grande número de aplicações**.

Figura 06. Software utilizado no auxílio ao diagnóstico de osteoartrite de coluna lombar baseado em redes neurais artificiais (VERONEZI, et al., 2011).



Figura 07. Software que utiliza a tecnologia de ultrassom mais RNA para ser um classificador para estimativa de gordura intramuscular (CHICONINI, PACHECO, LULIO, SILVA & SILVA, 2017).



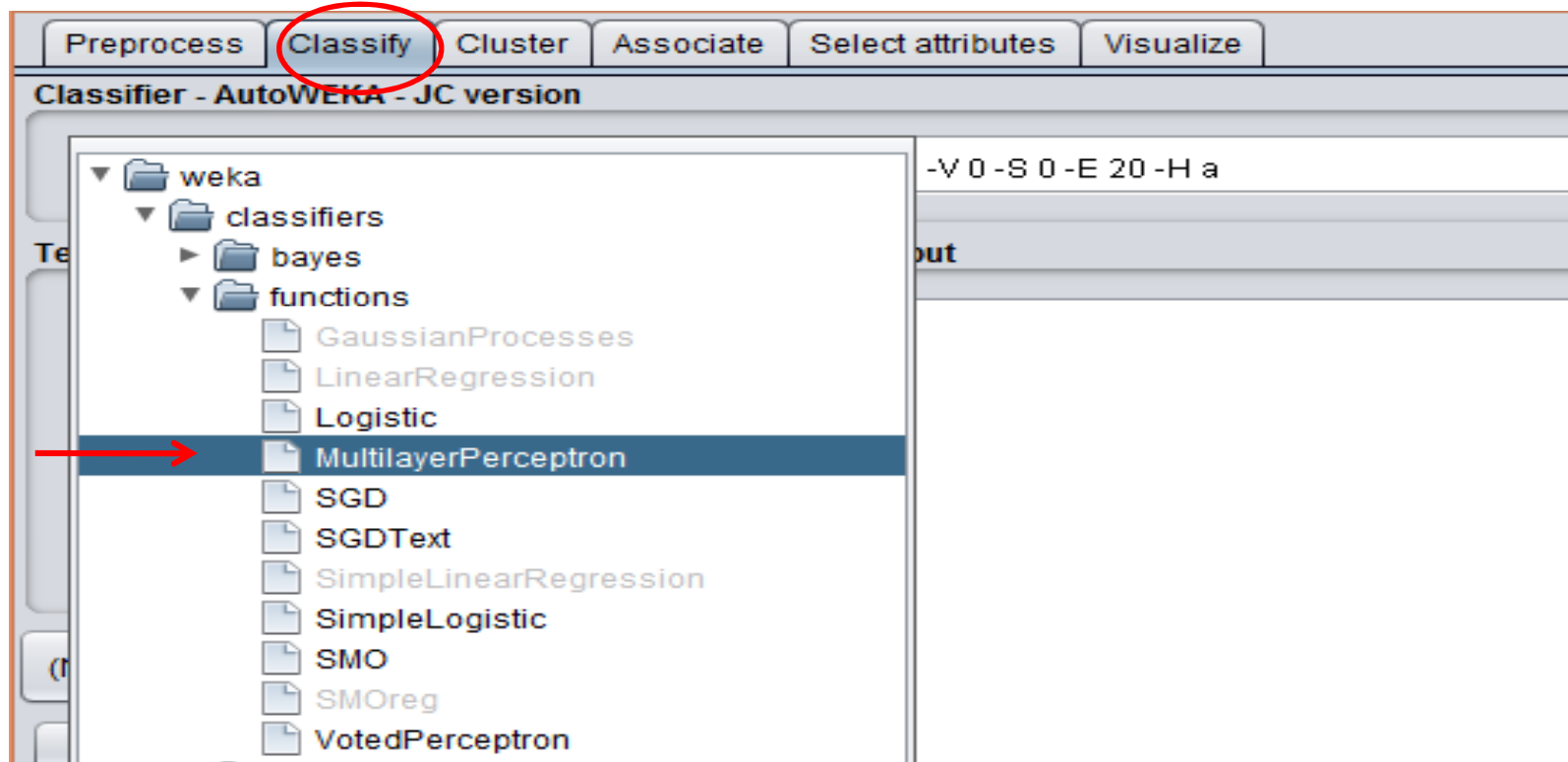
Redes Neurais Artificiais

❑ Desvantagens:

- ❖ Dificuldade de **entender** como e porque **as redes tomam** suas decisões;
- ❖ Dificuldade de **escolher** o melhor conjunto de parâmetros para a arquitetura da rede;
- ❖ **Alta complexidade** computacional do treinamento.

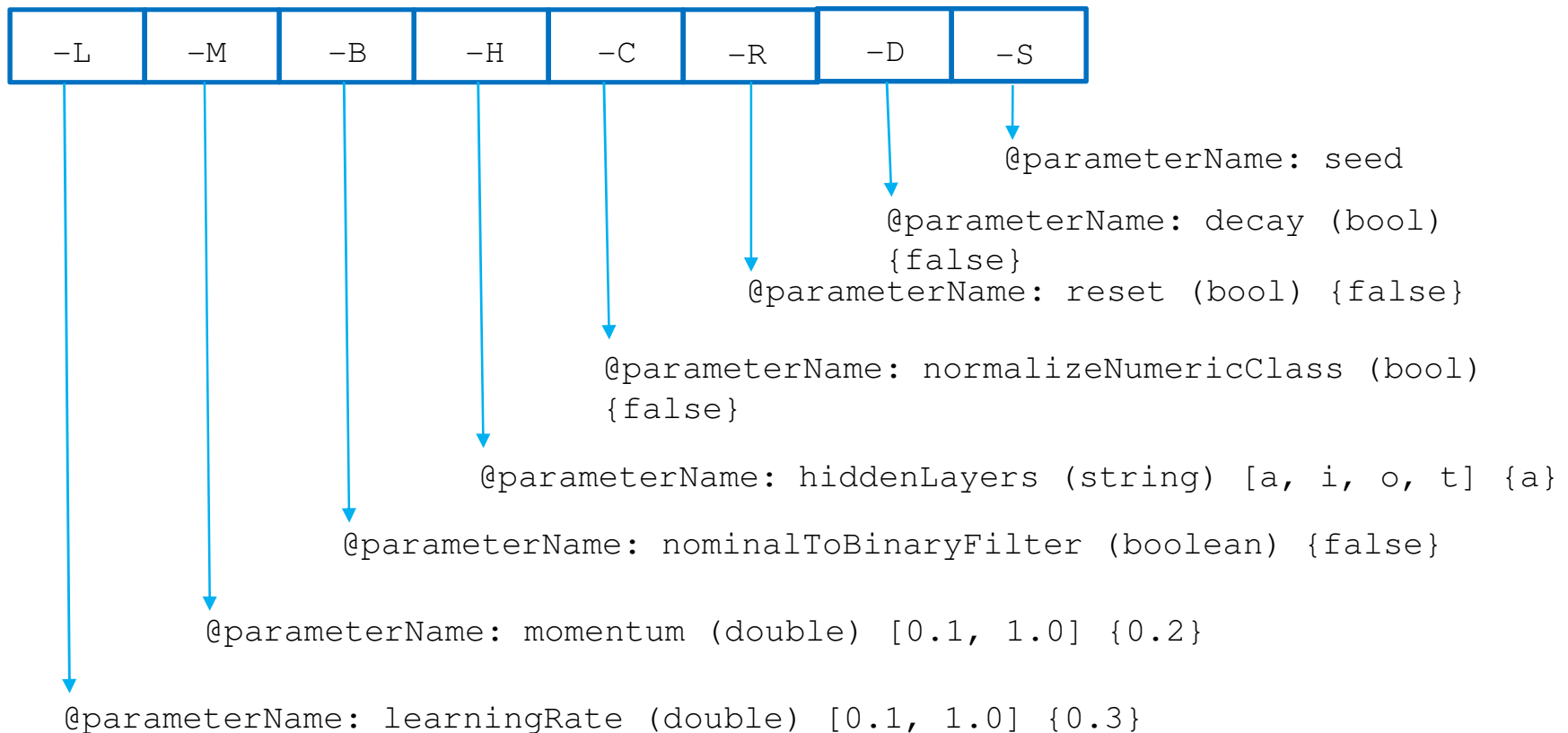
Multilayer Perceptron

- ❑ Utilizando **MLP** (WEKA):



Multilayer Perceptron

❑ Todos os parâmetros do **MLP** (WEKA):



Multilayer Perceptron

❑ Configurando o MLP:

weka.classifiers.functions.MultilayerPerceptron

About

A Classifier that uses backpropagation to classify instances. [More](#) [Capabilities](#)

GUI ☐

autoBuild ☒

batchSize

debug ☐

decay ☐

doNotCheckCapabilities ☐

hiddenLayers

learningRate

momentum

nominalToBinaryFilter ☐

normalizeAttributes ☒

normalizeNumericClass ☒

numDecimalPlaces

reset ☐

seed

trainingTime

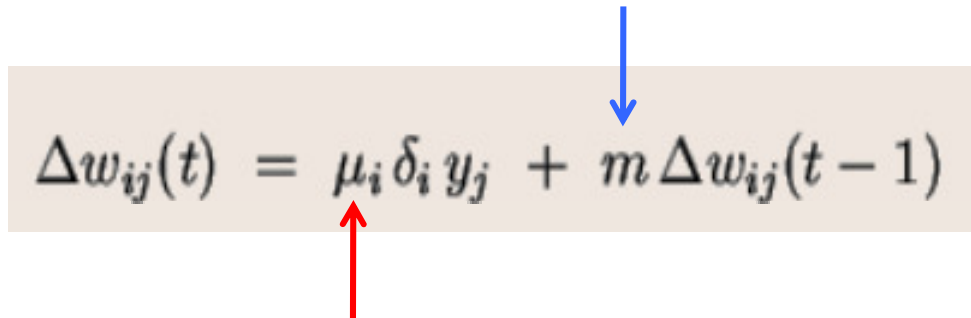
validationSetSize

validationThreshold

Multilayer Perceptron

□ Parâmetros:

- ❖ momentum: taxa aplicada sobre os pesos dos neurônios.

$$\Delta w_{ij}(t) = \mu_i \delta_i y_j + m \Delta w_{ij}(t-1)$$
A diagram showing the weight update equation $\Delta w_{ij}(t) = \mu_i \delta_i y_j + m \Delta w_{ij}(t-1)$ inside a light beige rectangular box. A red arrow points upwards from the text 'learningRate' below to the term μ_i in the equation. A blue arrow points downwards from the text 'momentum' above to the term m in the equation.

- ❖ learningRate: valor que atualiza os pesos dos neurônios.

Multilayer Perceptron

□ Parâmetro:

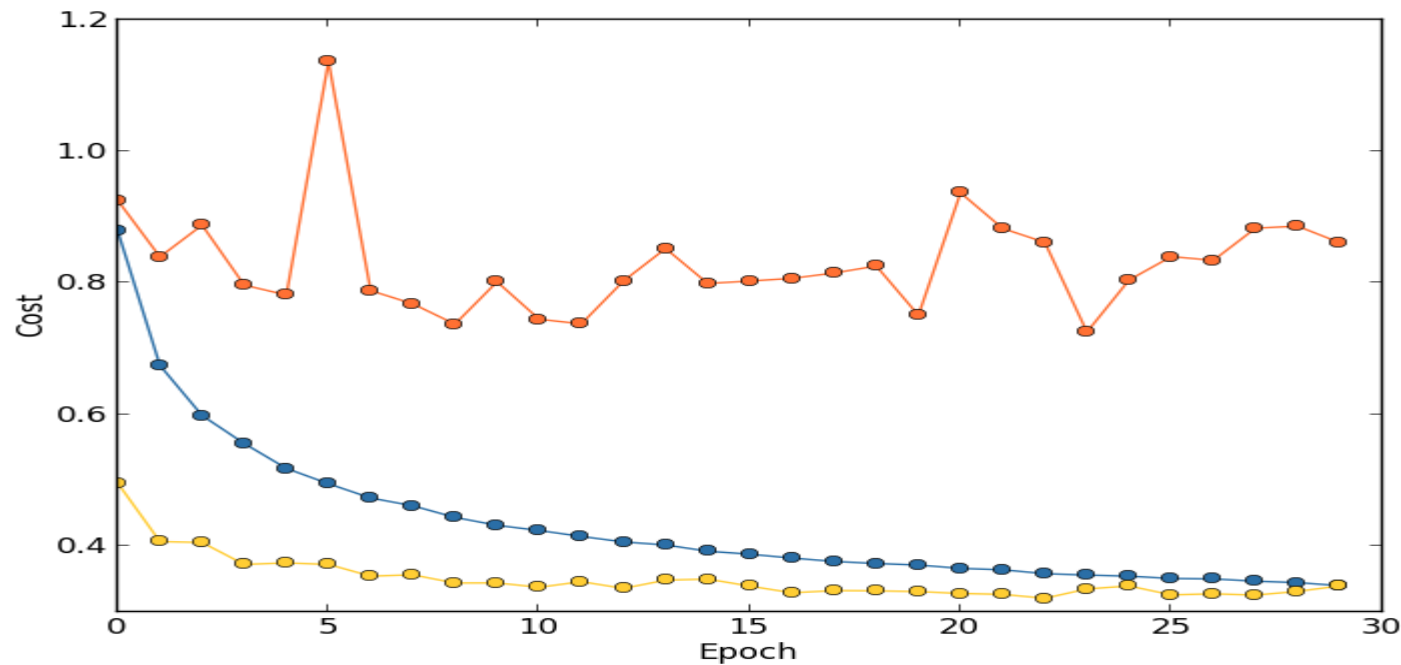
❖ hiddenLayers: número (n) de camadas escondidas, podendo ser:

- Qualquer valor inteiro maior ou igual a 1(*).
- Zero (sem hidden layers).
- Valores nominais (a, i, o, t):
 - $a = (\text{attrs.} + \text{classes})/2$;
 - $i = \text{attrs.}$;
 - $o = \text{classes}$;
 - $t = \text{attrs.} + \text{classes}$.

Multilayer Perceptron

❑ Parâmetro:

- ❖ trainingTime: número de épocas (ciclos) para treinamento da rede neural.



Multilayer Perceptron

□ Analisando os resultados....

Time taken to build model: 2.09 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	320	91.1681 %
Incorrectly Classified Instances	31	8.8319 %
Kappa statistic	0.8008	
Mean absolute error	0.0947	←
Root mean squared error	0.2798	
Relative absolute error	20.5688 %	
Root relative squared error	58.3128 %	
Total Number of Instances	351	

=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: ionosphere
Instances: 351
Attributes: 35

=== Confusion Matrix ===

a	b	<-- classified as
220	5	a = g
26	100	b = b

Comitê de Classificadores

Classifier Ensemble

Motivação

- ❑ Classificadores individuais:
 - ❖ Cada **modelo (classificador)** assume um conjunto de suposições (sujeito a “bias”);
 - ❖ Diferentes algoritmos podem convergir para **diferentes soluções**.
- ❑ Comitê de Classificadores:
 - ❖ Vários classificadores juntos podem **aumentar o desempenho** de sistemas de reconhecimento de padrões;
 - ❖ **Erros minimizados** através do uso de múltiplos classificadores ao invés de um único classificador.

Motivação

- ❑ O uso de **múltiplos classificadores** aparece na literatura com diferentes nomes:
 - ❖ Fusão de classificadores;
 - ❖ Combinação de classificadores;
 - ❖ Mistura de classificadores;
 - ❖ Pool;
 - ❖ Comitês;
 - ❖ Ensembles.

Combinação de Classificadores

□ Paralelo:

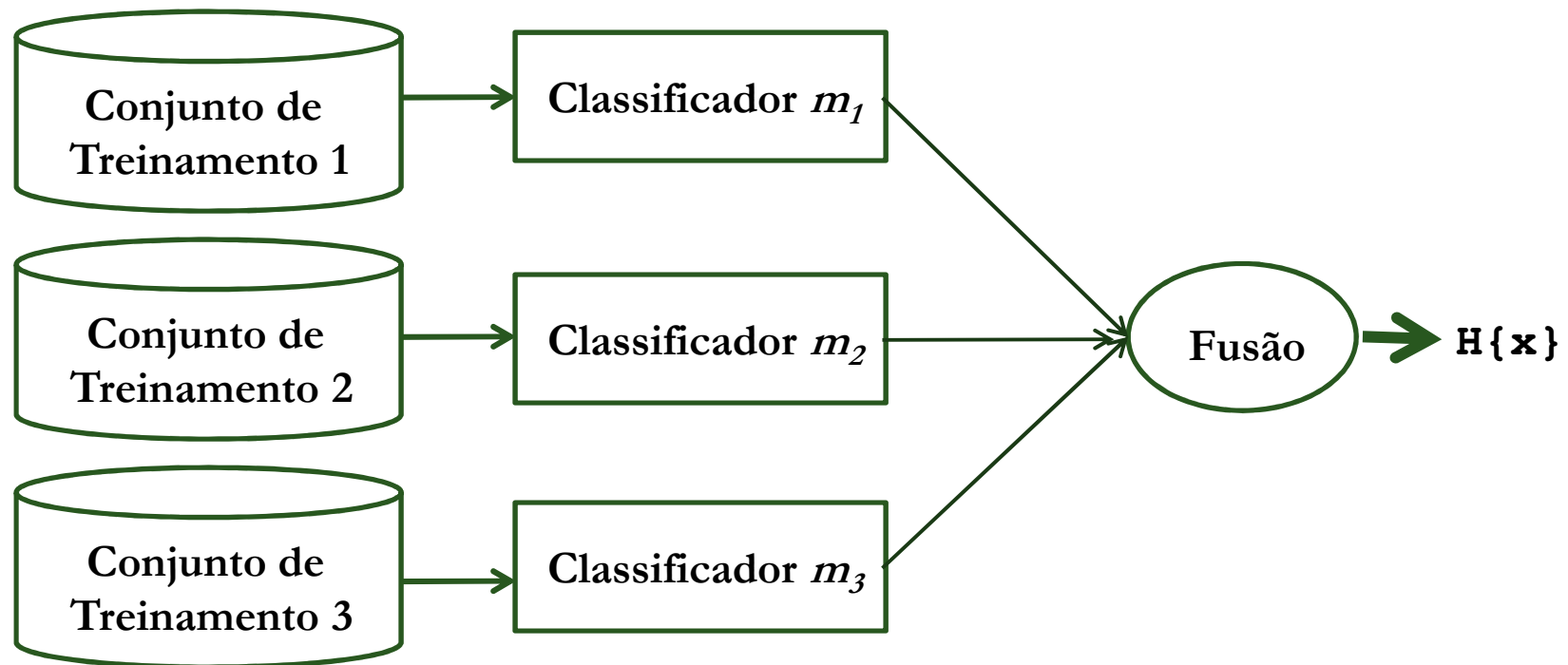
- ❖ Os classificadores $C_1 \dots C_n$ produzem **decisões** sobre um **padrão desconhecido**;
- ❖ Todas essas **decisões** são então enviadas para um **método de fusão** que produzirá o resultado final.

□ Serial:

- ❖ A cada estágio do sistema existe somente um classificador atuando no sistemas;
- ❖ Output de C_1 alimenta C_2 e assim por diante.

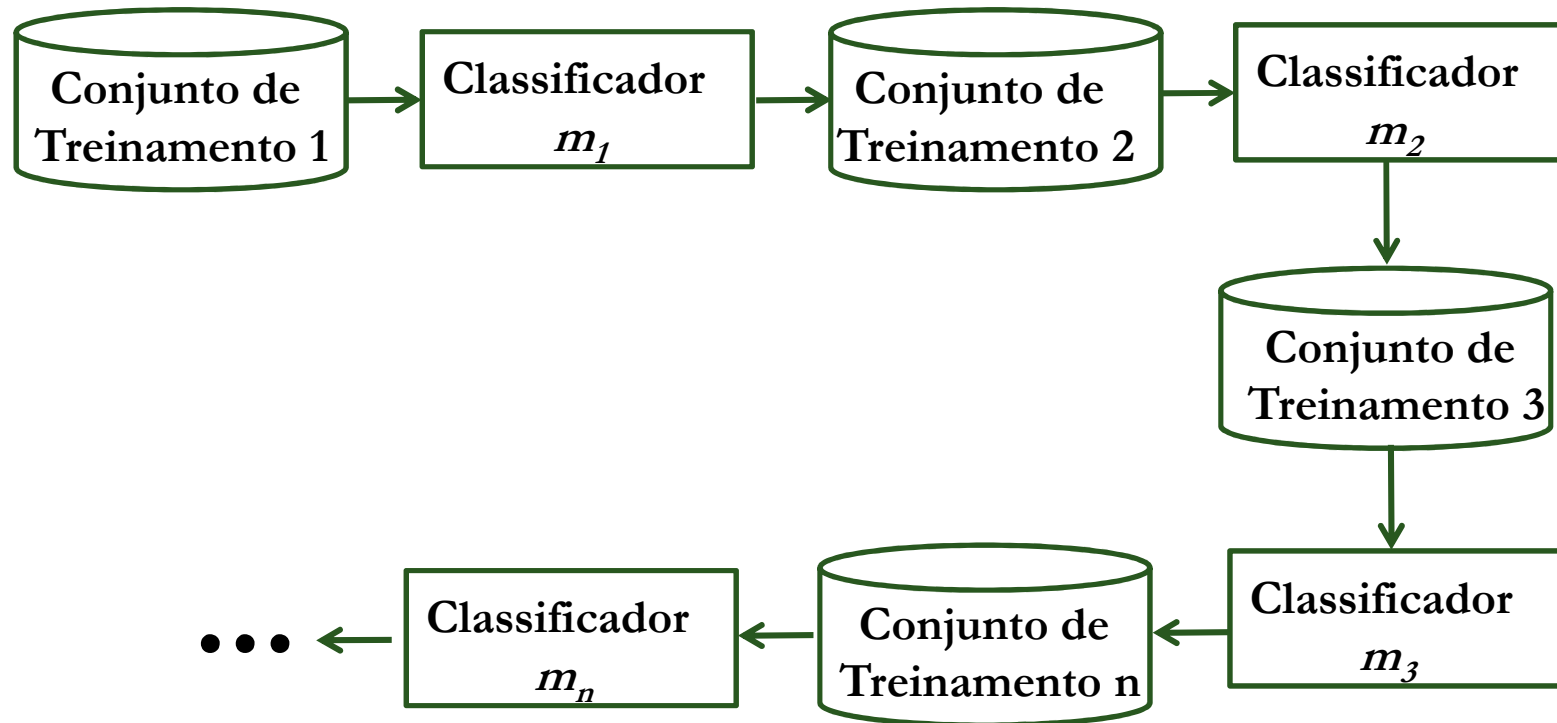
Arquitetura

- ❑ Paralelo: diferentes conjuntos de treinamento, mesmo classificador.



Arquitetura

- Serial: diferentes conjuntos de treinamento, mesmo classificador.



Ensembles

- ❑ Um conjunto de classificadores gerado automaticamente.
- ❑ **Melhor desempenho** do que um classificador único.
- ❑ Baseado na idéia de **diversidade**:
 - ❖ Mais diversidade gerando melhor desempenho.
- ❑ Métodos:
 - ❖ Bagging;
 - ❖ Boosting.

Bagging [Breiman, 1996]

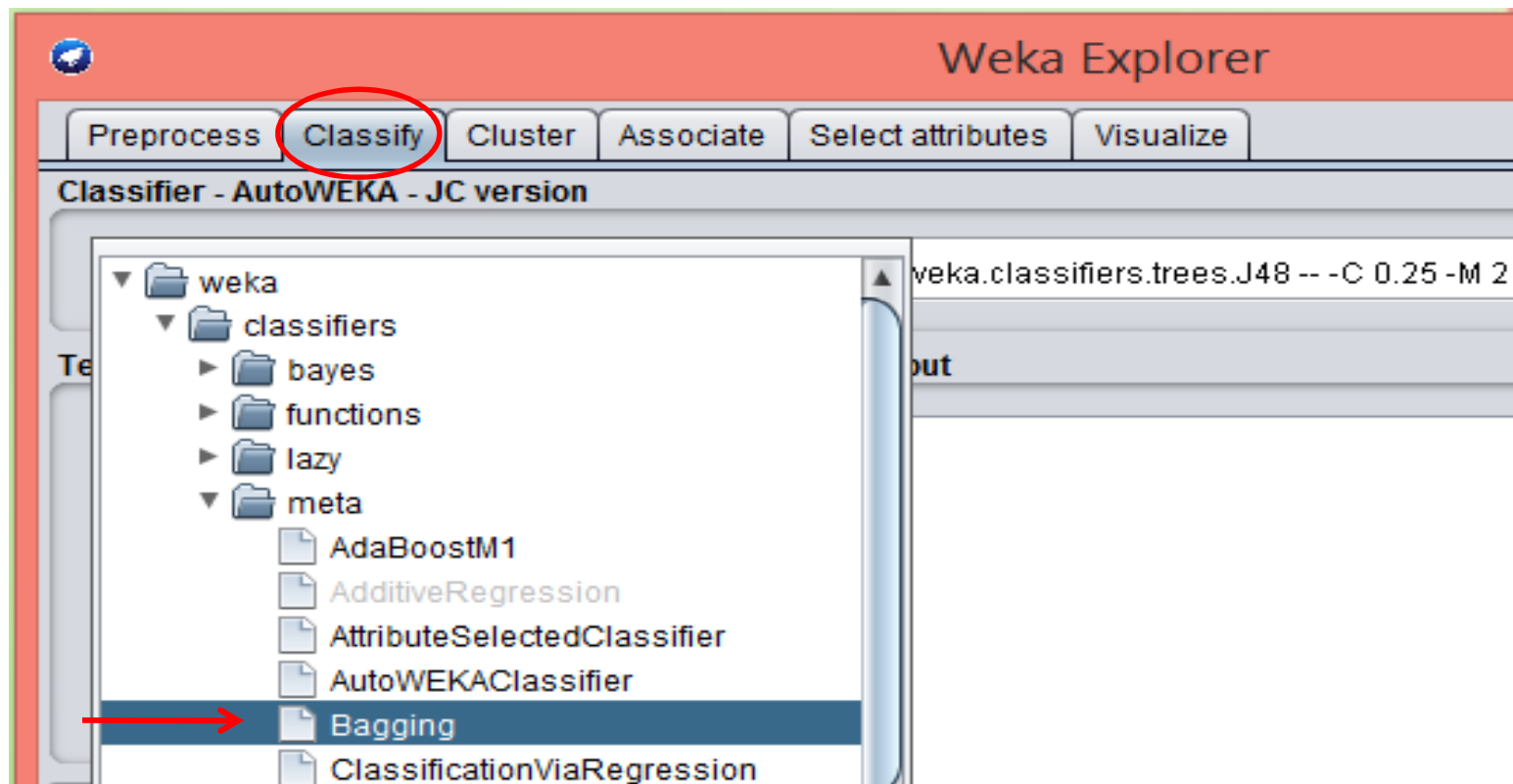
- ❑ Cada classificador é treinado usando-se uma distribuição específica dos dados.
- ❑ Utiliza **múltiplas versões** de um conjunto de treinamento.
 - ❖ Instâncias selecionadas aleatoriamente, havendo a possibilidade de repetição.
 - ❖ Mesmo número de instâncias do conjunto original.
- ❑ Os classificadores componentes tem todos a mesma forma geral (todos NN, ou todos Árvores de Decisão).

Bagging [Breiman, 1996]

- ❑ Naturalmente paralelizável.
- ❑ Voto majoritário (fusão).
- ❑ Robusto **a ruídos** nos dados.

Bagging [Breiman, 1996]

- Utilizando **Bagging** (WEKA):



Bagging [Breiman, 1996]

❑ Configurando o Bagging:

The screenshot shows the 'weka.gui.GenericObjectEditor' window for the 'weka.classifiers.meta.Bagging' class. The 'About' section describes it as a 'Class for bagging a classifier to reduce variance.' with 'More' and 'Capabilities' buttons. The configuration parameters are as follows:

Parameter	Value
bagSizePercent	100
batchSize	100
calcOutOfBag	False
classifier	Choose J48 -C 0.25 -M 2
debug	False
doNotCheckCapabilities	False
numDecimalPlaces	2
numExecutionSlots	1
numIterations	3
outputOutOfBagComplexityStatistics	False
printClassifiers	False
representCopiesUsingWeights	False
seed	1
storeOutOfBagPredictions	False

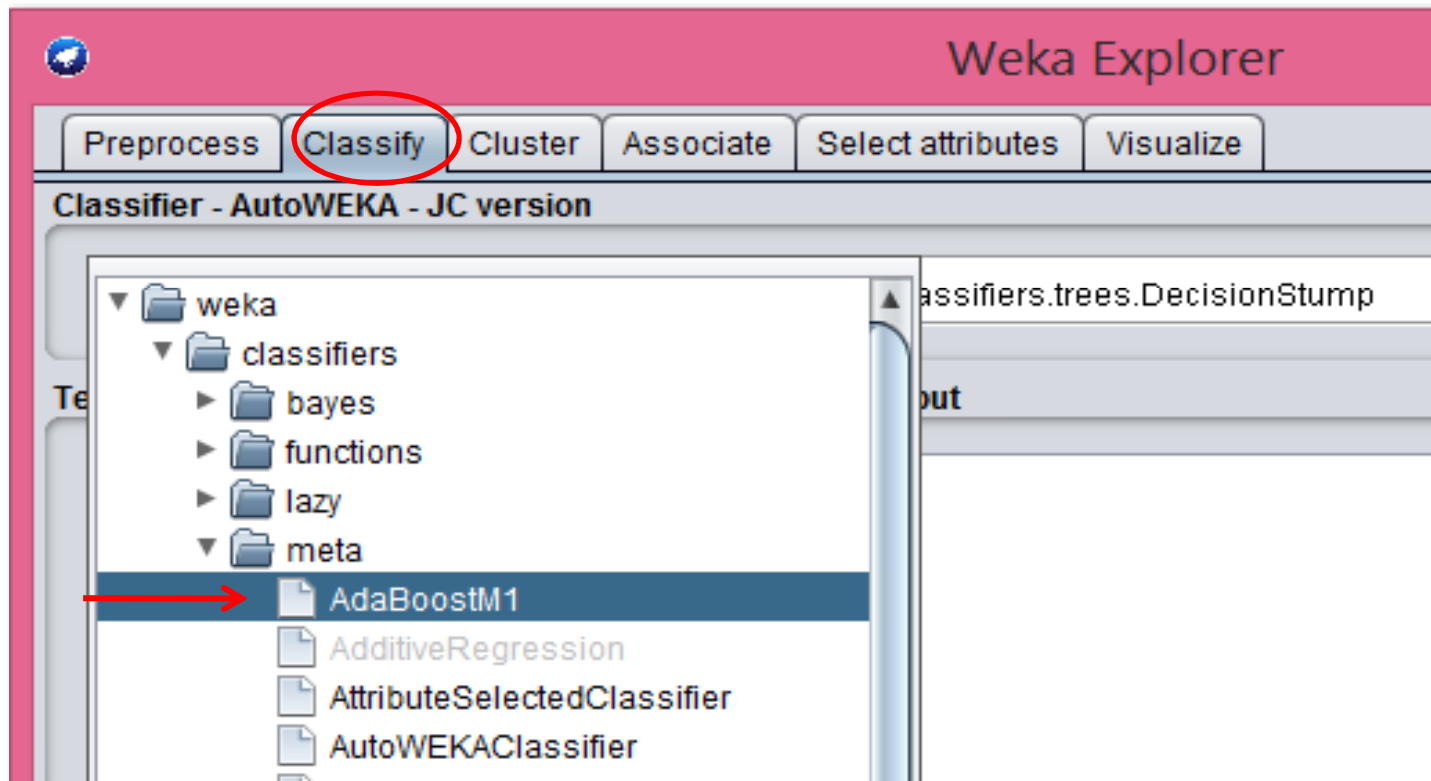
At the bottom are buttons for 'Open...', 'Save...', 'OK', and 'Cancel'. Red arrows point to the 'bagSizePercent', 'classifier', and 'numIterations' fields.

Boosting [Schapire, 1990]

- Induz sequencialmente um conjunto de classificadores.
- **Melhor desempenho** do que um classificador único.
 - ❖ O classificador corrente depende dos anteriores tendo maior foco no erro destes últimos.
 - ❖ Distribuição do conjunto de treinamento é modificada (erro anterior).
 - ❖ Instâncias incorretamente preditas anteriormente são escolhidas com maior frequência / ponderadas com maior peso.

Boosting [Schapire, 1990]

- ❑ Utilizando **Boosting** (WEKA):



Boosting [Schapire, 1990]

❑ Configurando o **Boosting**:

