



Politecnico  
di Bari

# Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione  
Corso di Laurea Triennale in Ingegneria dei Sistemi Medicali



DIPARTIMENTO DI  
INGEGNERIA ELETTRICA  
E DELL'INFORMAZIONE

## Bioinformatics and Big Data Analytics

### Neural Networks

*Eng. Nicola **Altini**, Ph.D. Student*

*Eng. Giacomo Donato **Cascarano**, Ph.D. Student*

*Prof. Eng. Vitoantonio **Bevilacqua**, Ph.D.*



Anno Accademico 2019/2020



apulian  
bioengineering  
company

# Classification

---

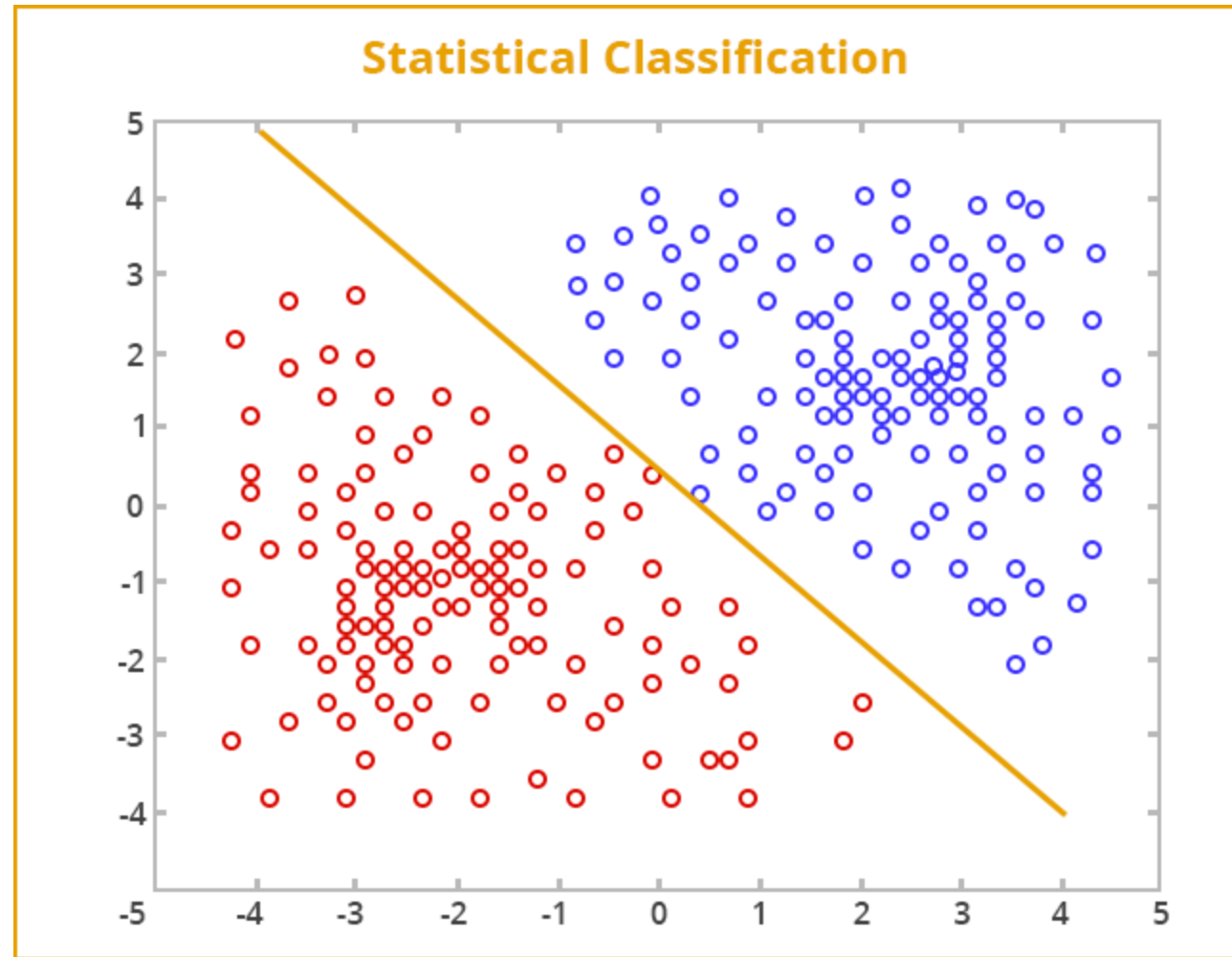
- Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.
- Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.).
- Classification is an example of pattern recognition.

# Classification

---

- Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or features. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.
- An algorithm that implements classification, especially in a concrete implementation, is known as a **classifier**. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category.

# Classification



# Confusion Matrices

---

- Say we test some people for the presence of a disease. Some of these people have the disease, and our test correctly says they are positive. They are called **true positives (TP)**. Some have the disease, but the test incorrectly claims they don't. They are called **false negatives (FN)**. Some don't have the disease, and the test says they don't – **true negatives (TN)**. Finally, there might be healthy people who have a positive test result – **false positives (FP)**.
- These can be arranged into a 2×2 contingency table (confusion matrix), conventionally with the test result on the vertical axis and the actual condition on the horizontal axis.

# Confusion Matrices

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	<b>Sensitivity</b> $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	<b>Specificity</b> $\frac{TN}{(TN + FP)}$
		<b>Precision</b> $\frac{TP}{(TP + FP)}$	<b>Negative Predictive Value</b> $\frac{TN}{(TN + FN)}$	<b>Accuracy</b> $\frac{TP + TN}{(TP + TN + FP + FN)}$



# Classification Metrics

Metric	Definition
<i>Accuracy</i>	$\frac{TP + TN}{TP + TN + FP + FN}$
<i>Precision</i>	$\frac{TP}{TP + FP}$
<i>Recall</i>	$\frac{TP}{TP + FN}$
<i>Jaccard Similarity Index</i>	$\frac{TP}{TP + FP + FN}$
<i>Miss Rate</i>	$\frac{FN}{FN + TP}$
<i>F – Measure</i>	$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$



# ROC Curves

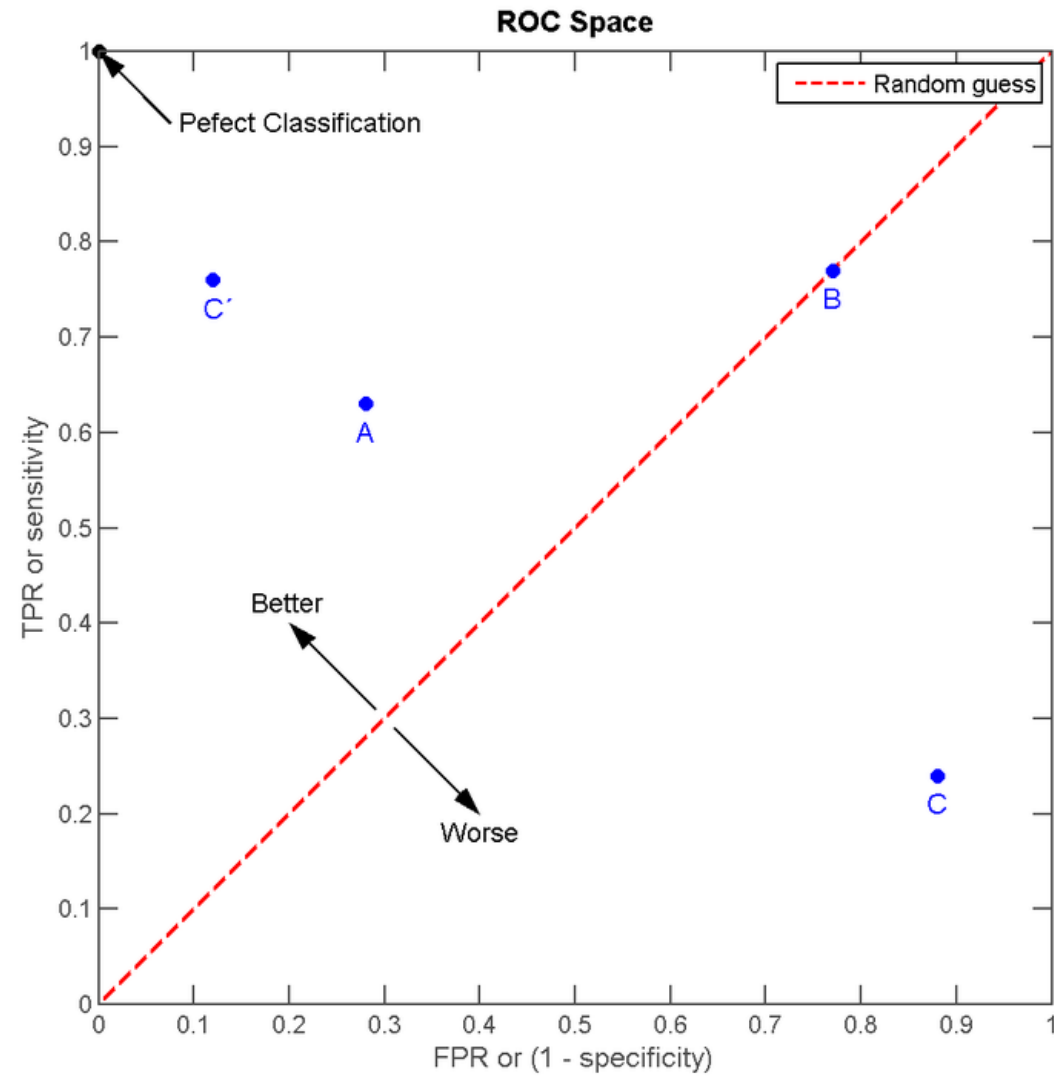
---

- A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
- The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as probability of false alarm.

- $$TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$$

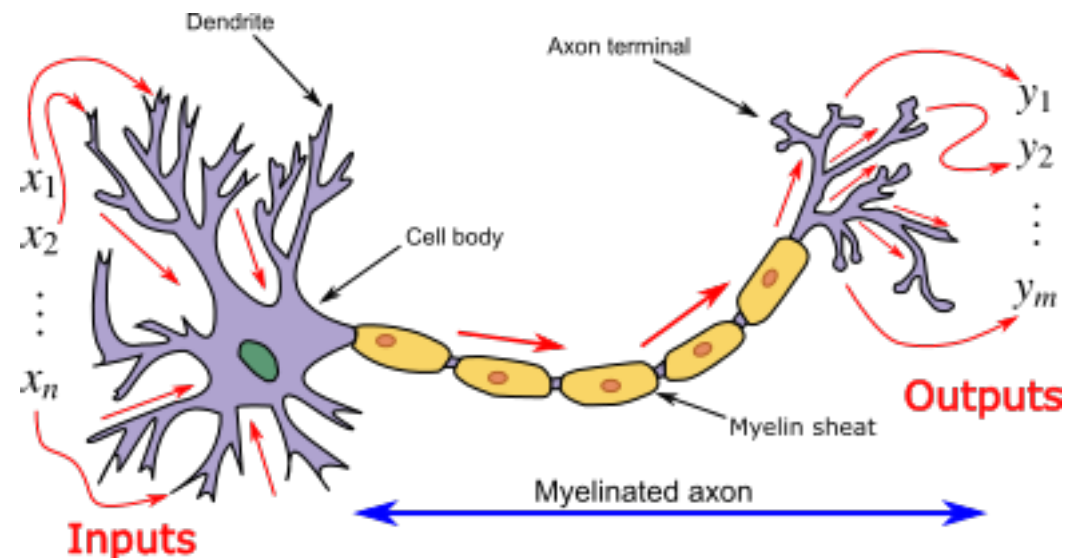
- $$FPR = \frac{FP}{N} = \frac{FP}{TN+FP}$$

# ROC Curves



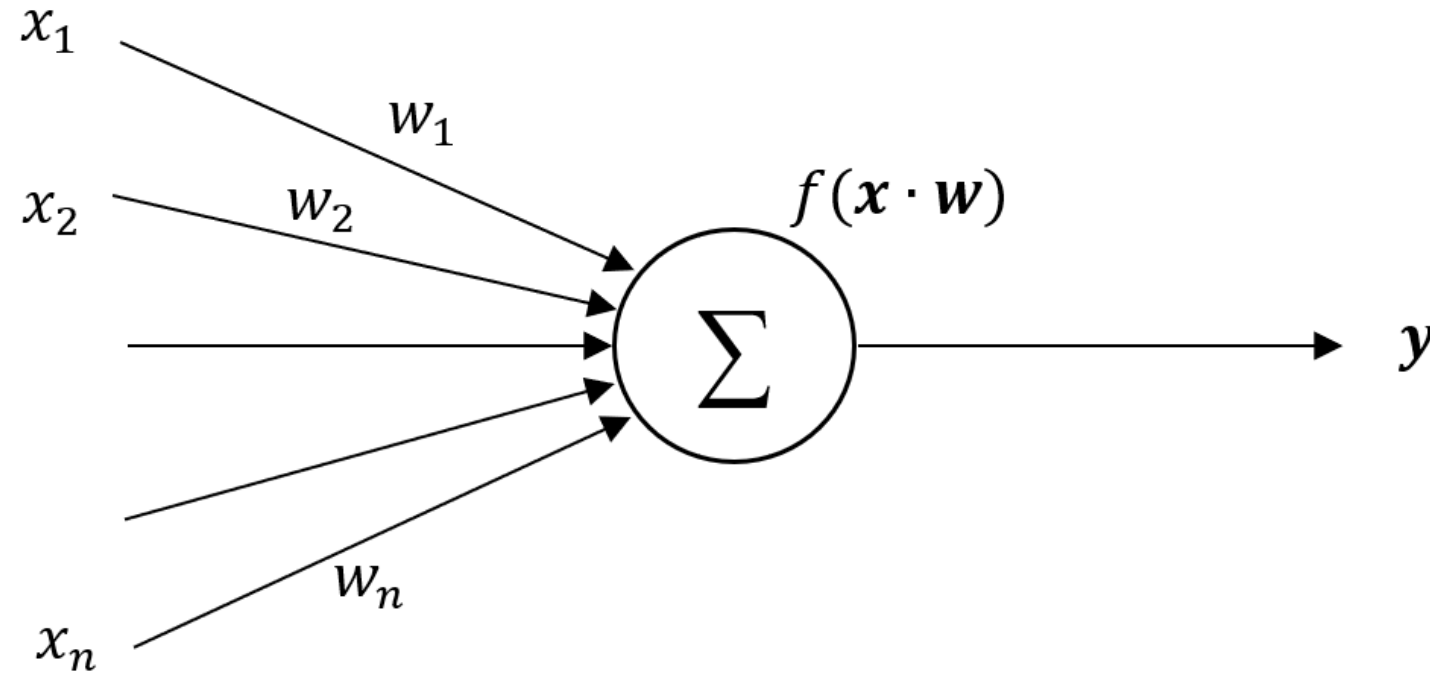
# Perceptron

- The perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class.
- The perceptron model stems from pioneering work on neural simulation conducted over 75 years ago.



# Perceptron

---



# Perceptron

---

- We denote the inputs by  $x_1, x_2, \dots, x_n$  and the weights by  $w_1, w_2, \dots, w_n$ .
- In vectorized form:  $\mathbf{x} = [x_1, x_2, \dots, x_n]$  and  $\mathbf{w} = [w_1, w_2, \dots, w_n]$ .
- The input to the neuron is:

$$s = \sum_{i=1}^n x_i w_i = \mathbf{x} \cdot \mathbf{w}$$

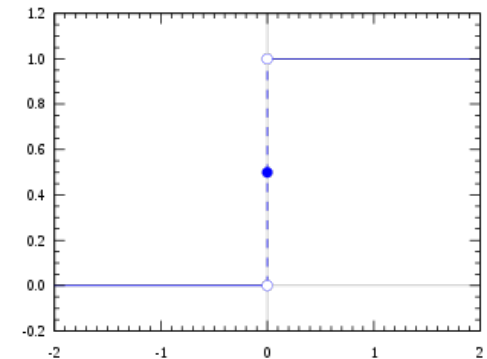
- or, more generally, considering a threshold  $\theta$  associated with the neuron:

$$s = \sum_{i=1}^n x_i w_i - \theta = \mathbf{x} \cdot \mathbf{w} - \theta$$

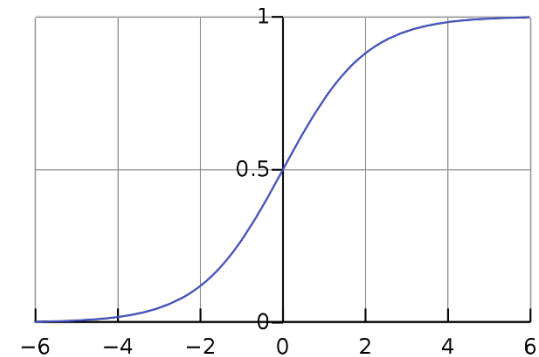
# Activation functions

- A neuron has an associated transfer (activation) function  $f(s)$  which provides the output.
- Possible examples of activation function are:

- Heaviside step function:  $f(s) = \begin{cases} 0 & \text{if } s \leq 0 \\ 1 & \text{if } s > 0 \end{cases}$



- Sigmoid function:  $f(s) = \frac{1}{1 + e^{-s}}$



# Artificial Neural Networks

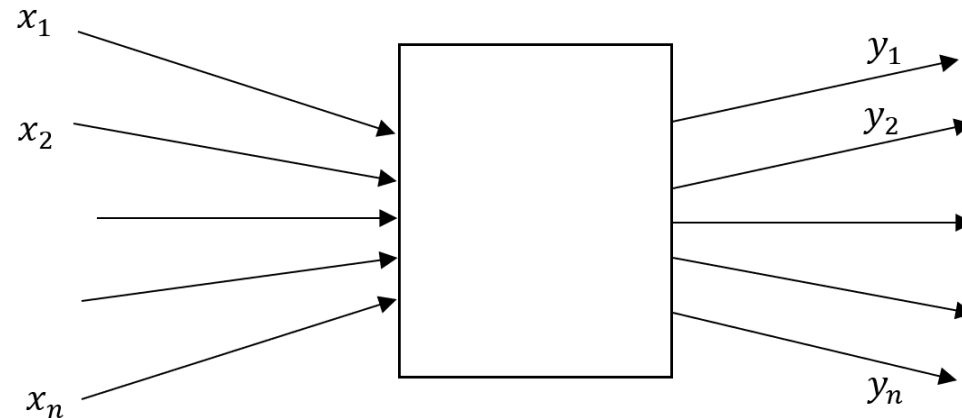
---

- Artificial Neural Networks (ANNs) have seen extensive application for some decades and represent a valuable tool in various fields where it is difficult to describe the task algorithmically, particularly speech and visual pattern recognition.
- Most neural approaches rely on combinations of elementary processors – neurons – each of which takes several inputs and produces a single output.
- Each input is associated with a weight, and the output is usually a function of the weighted sum of inputs. This output function can be continuous or discrete, depending on the specific network in use.

# Artificial Neural Networks

---

- The general idea behind the construction of networks of these neurons lies in their interconnection (i.e., the output of one neuron turns into the input of other neurons). This concept is related to the high level of interconnection of elementary neurons found in the brain, which is assumed to explain the damage resistance and recall capabilities of humans.





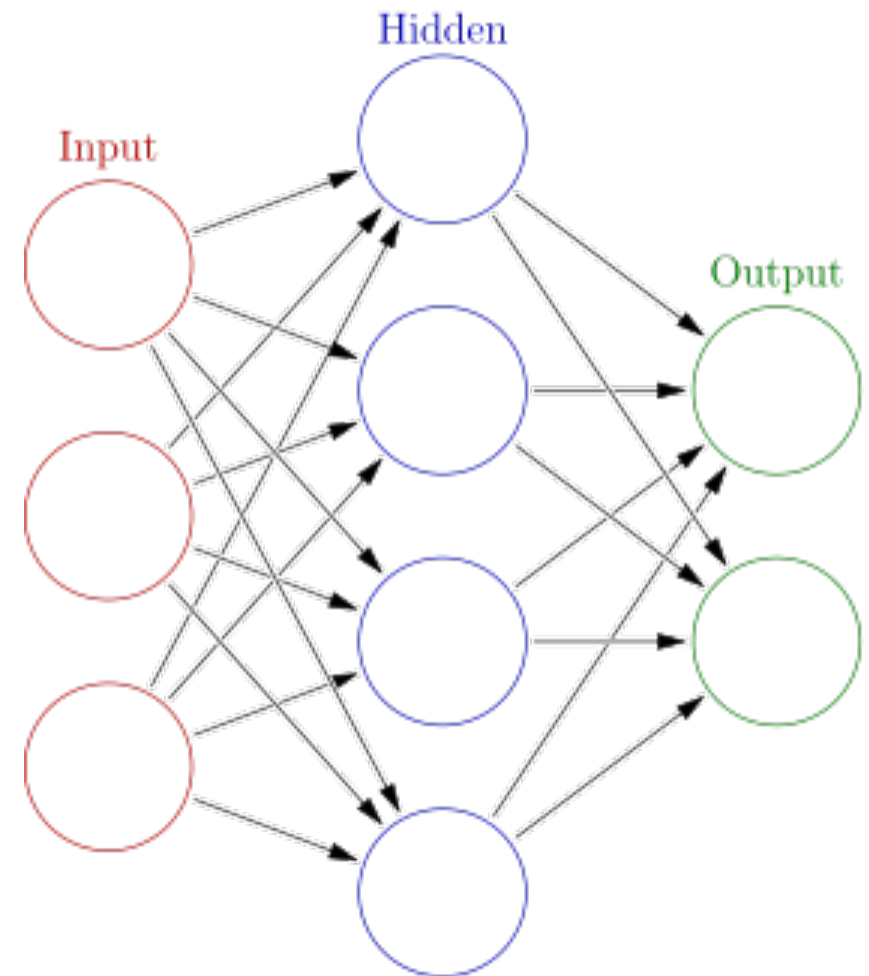
# Artificial Neural Networks

---

- The network topology is specified from what lies between: there could be many heavily interconnected neurons, or some structured (as layered) interconnection, or, pathologically, nothing (in this case inputs are connected straight to outputs).
- We are interested in multilayer networks.
- One of the most popular techniques for training multilayer neural networks rely on gradient descent in error – the backpropagation algorithm or generalized delta rule.

# Artificial Neural Networks

- An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain.
- Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.



# ANNs in MATLAB

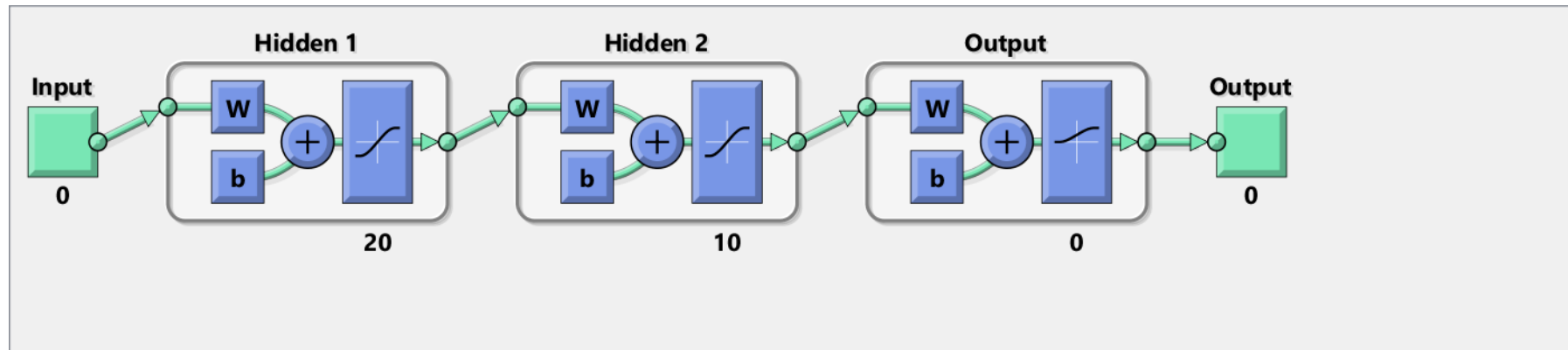
---

```
patternnet(hiddenSizes, trainFcn, performFcn)
```

- Pattern recognition networks are feedforward networks that can be trained to classify inputs according to target classes. The target data for pattern recognition networks should consist of vectors of all zero values except for a 1 in element  $i$ , where  $i$  is the class they are to represent.
- Arguments:
  - `hiddenSizes` Row vector of one or more hidden layer sizes (default = 10)
  - `trainFcn` Training function (default = 'trainscg')
  - `performFcn` Performance function (default = 'crossentropy') and returns a pattern recognition neural network.

# ANNs in MATLAB

```
% Hidden Layers  
hiddenLayerSize = [20,10];  
  
% Training Function - help nntrain  
trainFcn = 'traingdx'; % traingda, traingdm, traingd  
  
% Creazione rete  
net = patternnet(hiddenLayerSize, trainFcn);
```



# ANNs in MATLAB

---

```
% Dataset split
net.divideFcn = 'dividerand';
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio    = 30/100;
net.divideParam.testRatio   = 0/100;

% Stop criteria
net.trainParam.epochs = 5000;
%net.trainParam.max_fail = 20;
%net.trainParam.min_grad = 0;%10e-5;

% Error function
net.performFcn = 'mse';

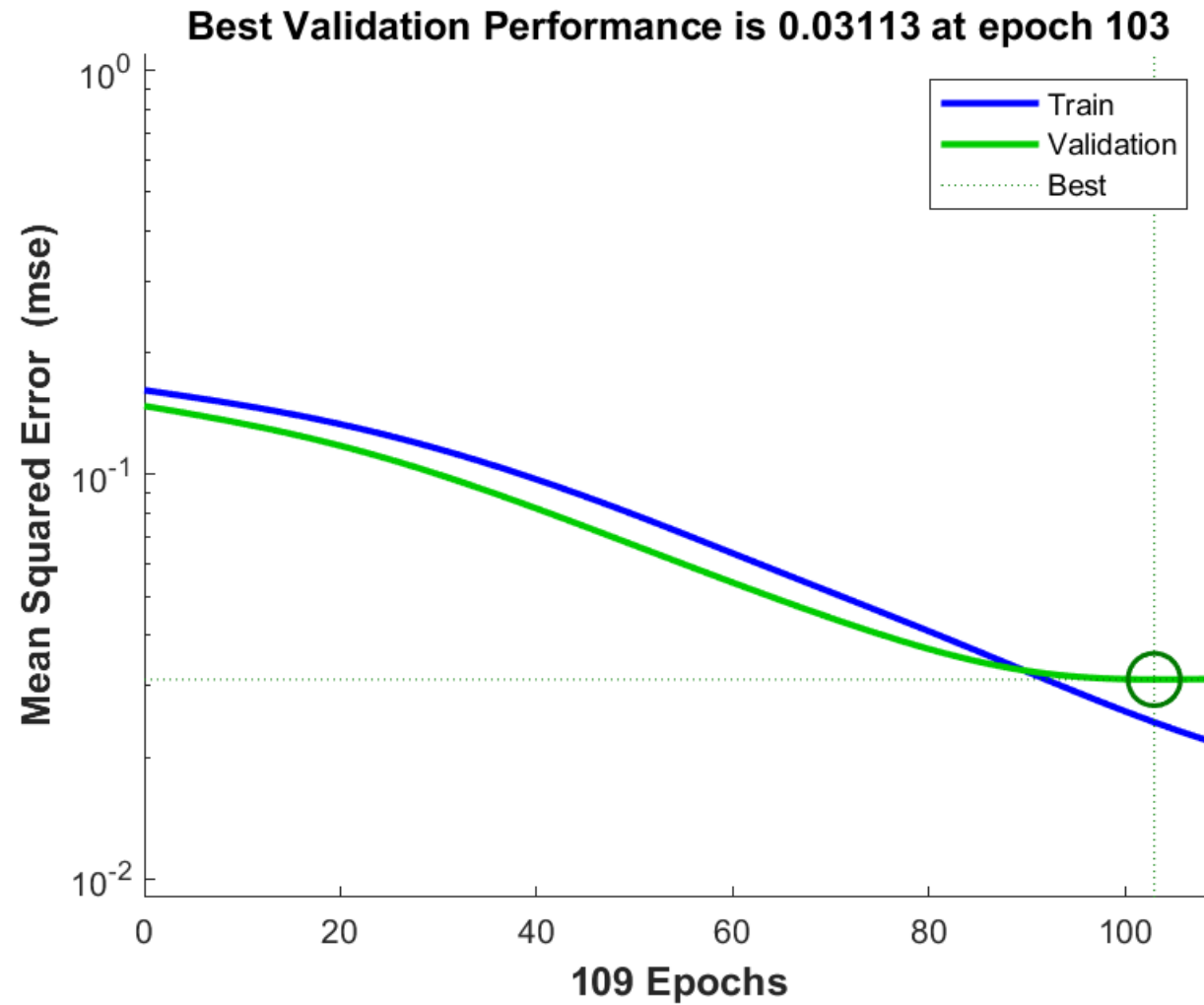
% Activation function
net.layers{end}.transferFcn = 'logsig';
```

# ANNs in MATLAB

---

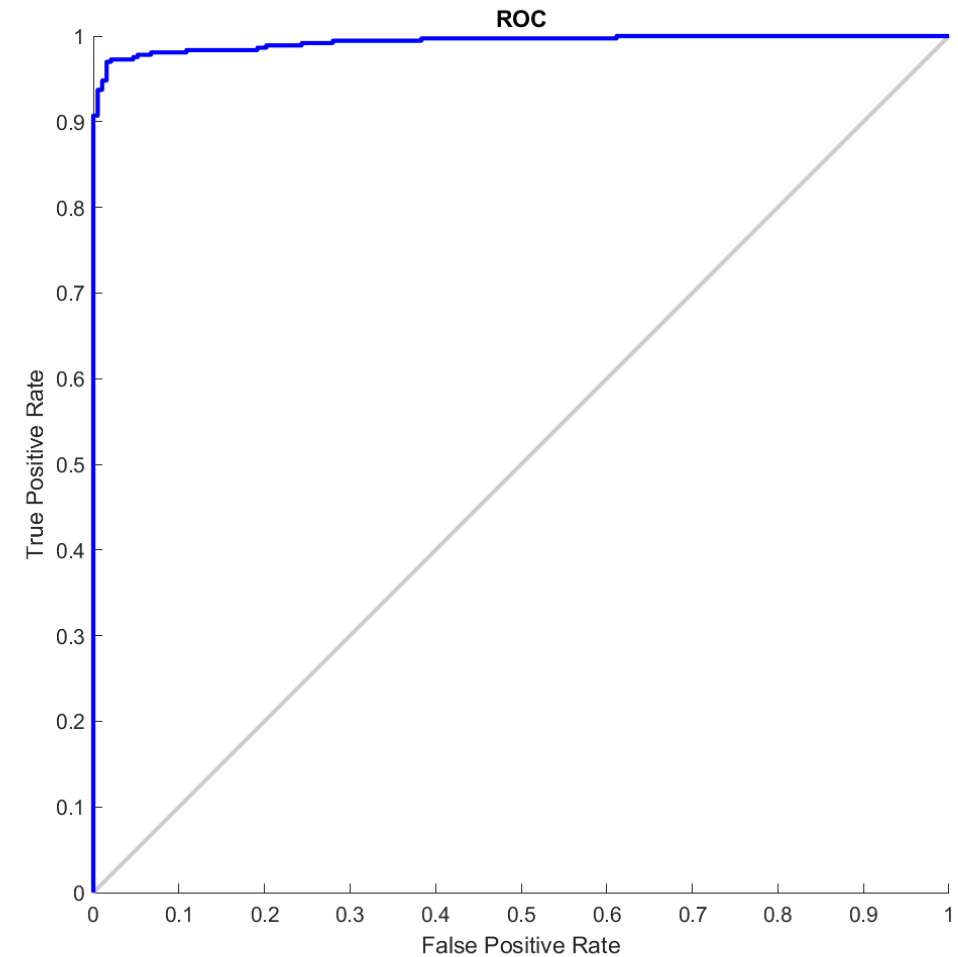
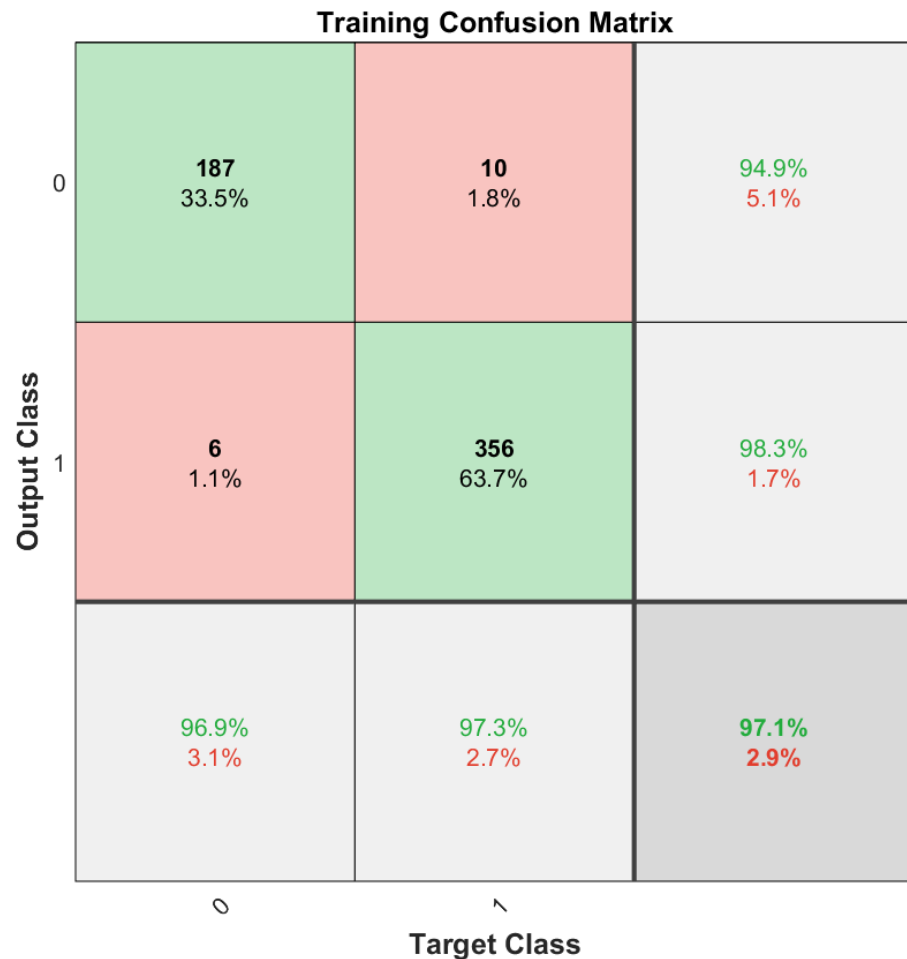
```
% Network initialization  
rng(0)  
net = configure(net,x_train,t_train);  
net = init(net);  
init_LW = net.LW;  
init_IW = net.IW;  
  
% Network training  
[net,tr] = train(net,x_train,t_train);  
y_train = net(x_train);  
figure, plotperform(tr)
```

# ANNs in MATLAB



# ANNs in MATLAB

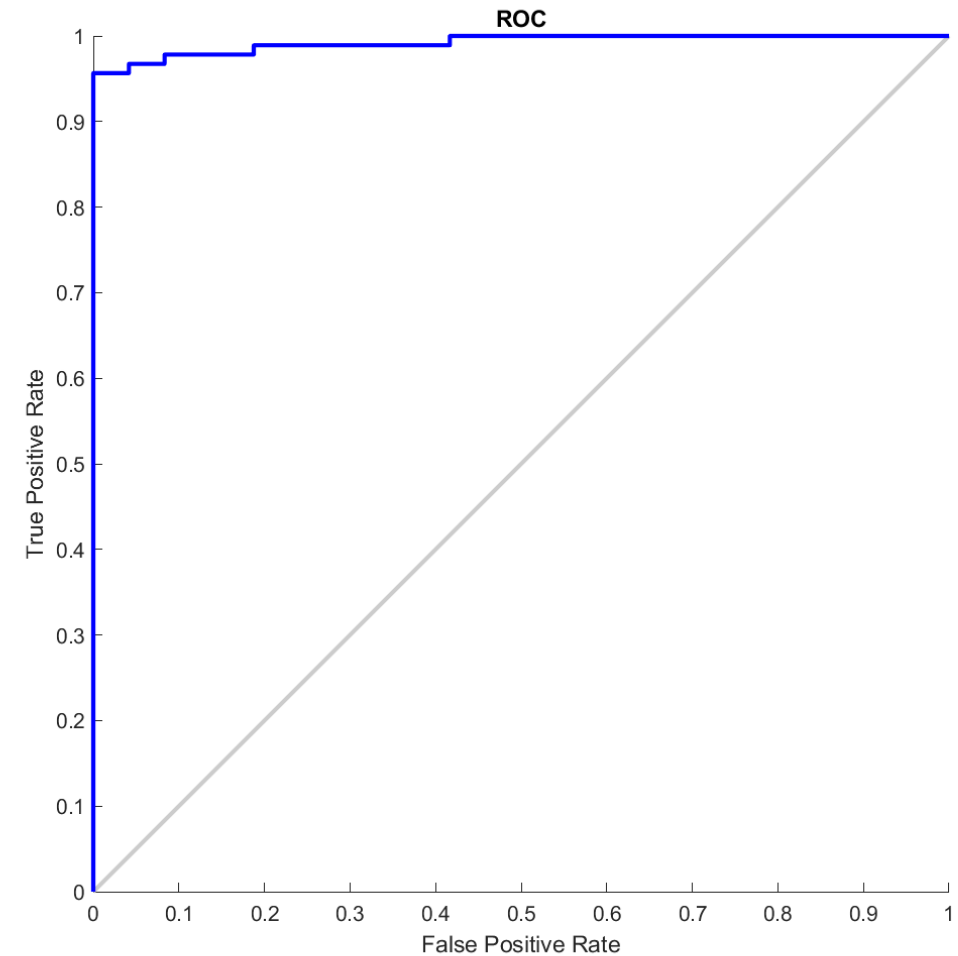
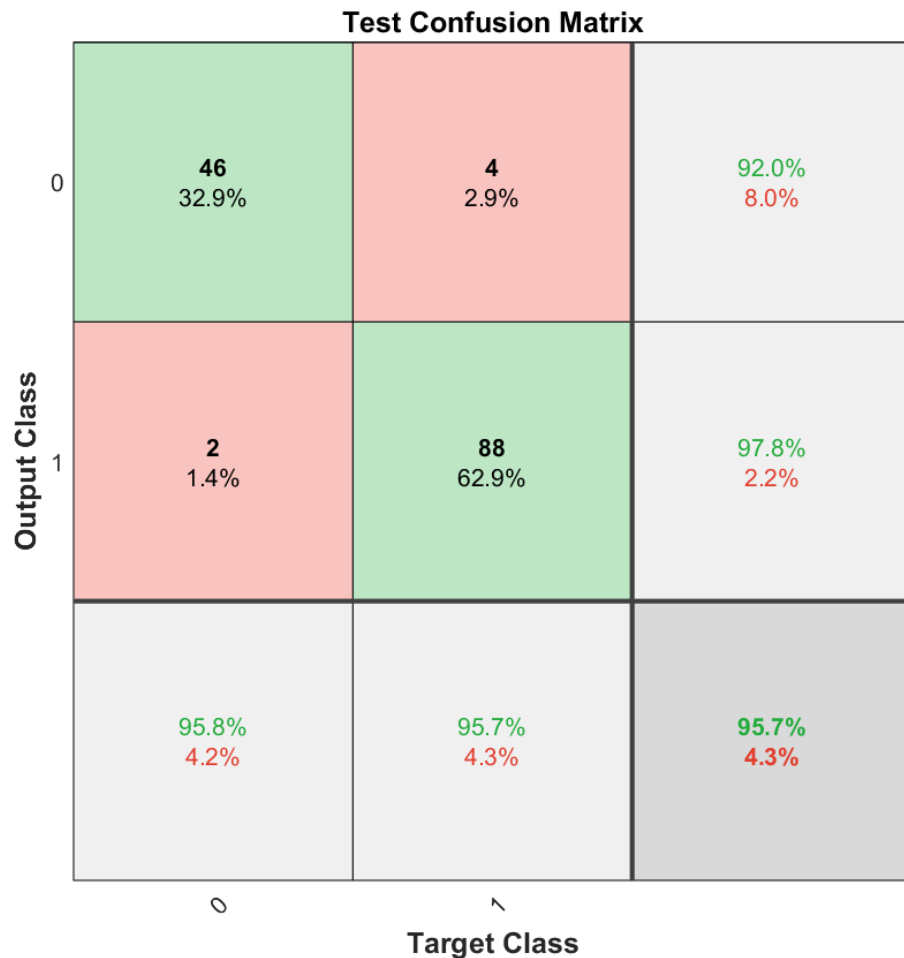
```
figure, plotconfusion(t_train,y_train),title('Training Confusion Matrix')  
figure, plotroc(t_train,y_train)
```





# ANNs in MATLAB

```
figure, plotconfusion(t_test,y_test),title('Test Confusion Matrix')  
figure, plotroc(t_test,y_test)
```



# References

---

- MATLAB Documentation

- *Pattern recognition network.*

URL: <https://www.mathworks.com/help/deeplearning/ref/patternnet.html>

- *Plot classification confusion matrices.*

URL: <https://www.mathworks.com/help/deeplearning/ref/plotconfusion.html>

- *Plot network performances.*

URL: <https://www.mathworks.com/help/deeplearning/ref/plotperform.html>

- *Plot receiver operating characteristics.*

URL: <https://www.mathworks.com/help/deeplearning/ref/plotroc.html>