UNIVERSITÀ DEGLI STUDI DI MILANO

Neural Networks for Classification

Ruggero Donida Labati

Dipartimento di Tecnologie dell'Informazione
via Bramante 65, 26013 Crema (CR), Italy
ruggero.donida@unimi.it

Classification

- Classification is one of the most frequently encountered decision making tasks of human activity.
- A classification problem occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object.

G.P. Zhang, "Neural networks for classification: a survey," in *IEEE Transactions on Systems, Man, and Cybernetics*, Part C: Applications and Reviews, vol.30, no.4, pp.451- 462, November 2000.

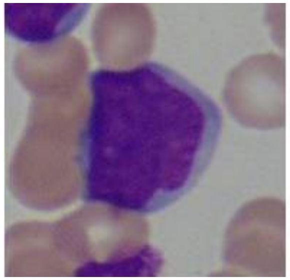
Classification with NN

- Neural networks have emerged as an important tool for classification.
- Advantages:
  - NN are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model
  - NN are universal functional approximators in that neural networks can approximate any function with arbitrary accuracy
  - NN are non-linear models, which makes them flexible in modeling real world complex relationships

G.P. Zhang, "Neural networks for classification: a survey,“ in *IEEE Transactions on Systems, Man, and Cybernetics*, Part C: Applications and Reviews, vol.30, no.4, pp.451- 462, November 2000.

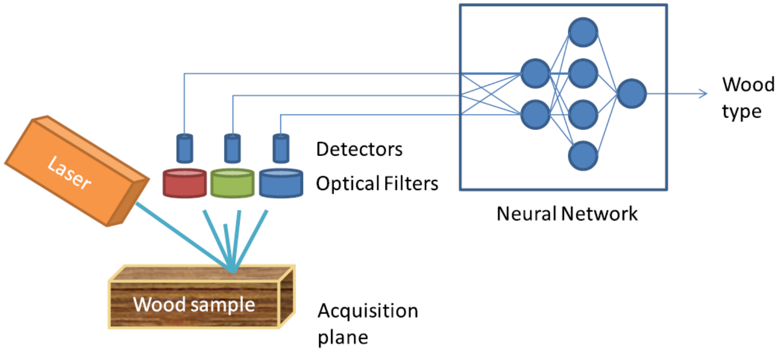
Examples of classification with NN

- Some our works
  - Acute Limphoblastic Leucemia
    - "healty cell" & "lymphoblast“



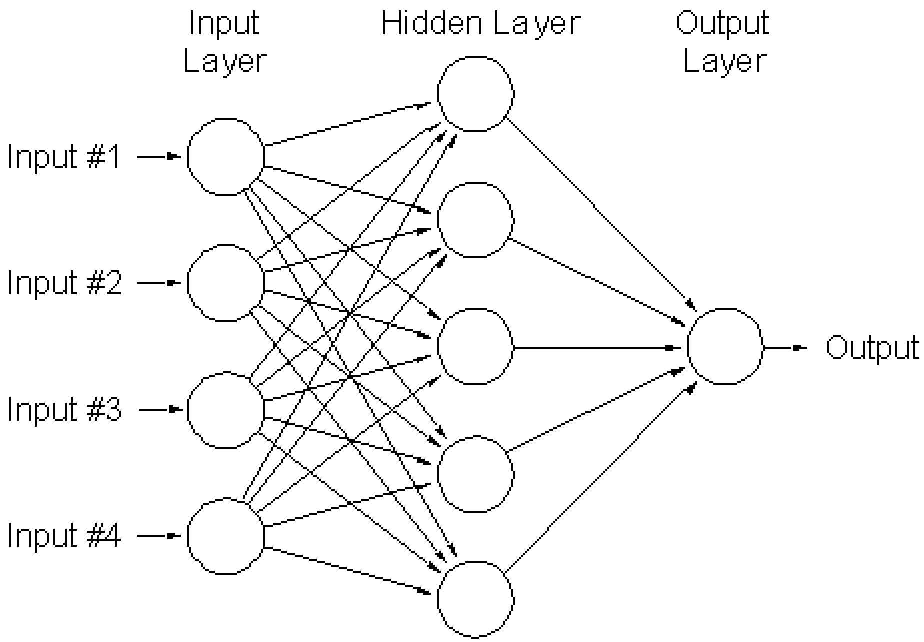
  - Wildfires
    - "Smoke frame" & "not smoke frame"

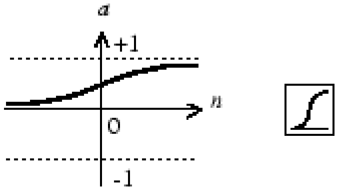
  - Wood
    - 21 classes



Classification with NN in Matlab

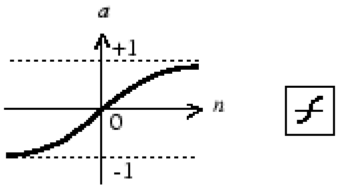
- We will use:
  - Neural Network Toolbox
  - Feedforward Neural Networks





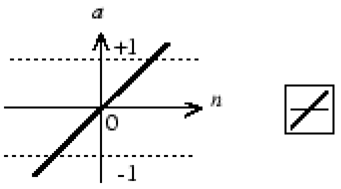
$a = logsig(n)$

Log-Sigmoid Transfer Function



$a = tansig(n)$

Tan-Sigmoid Transfer Function

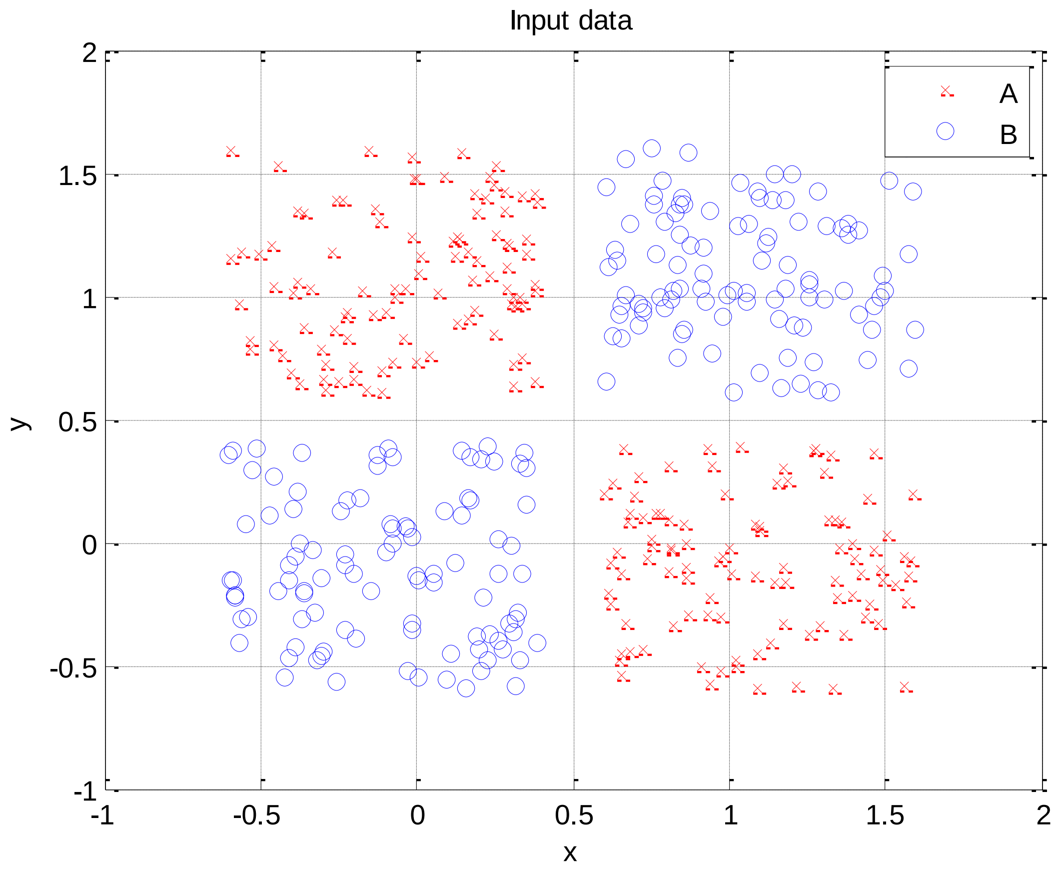


$a = purelin(n)$

Linear Transfer Function

Exercise 1 (1)

- Two classes classification of these data



Exercise 1 (2)

- We will
  - train a neural classifier
  - evaluate the obtained results in a graphical mode
  - evaluate the obtained error (mean, standard deviation)
  - change the parameters of the neural network
  - change the number of the input points
  - add noise to the input data

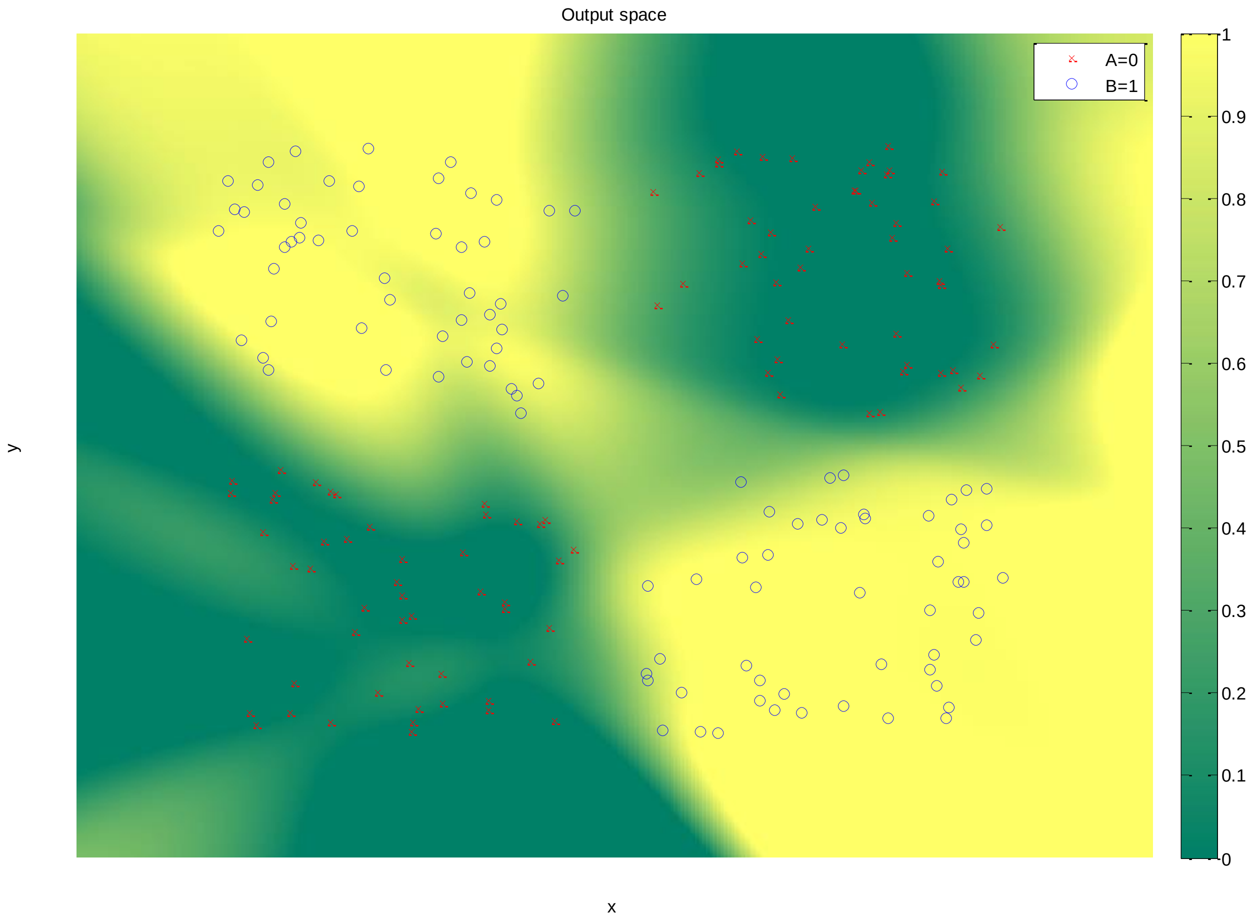
Exercise 1 (3)

- Example of results

TRAINING
n. of el. = 200
n. of errors = 7
mean error = 0.035000
std error = 0.184241

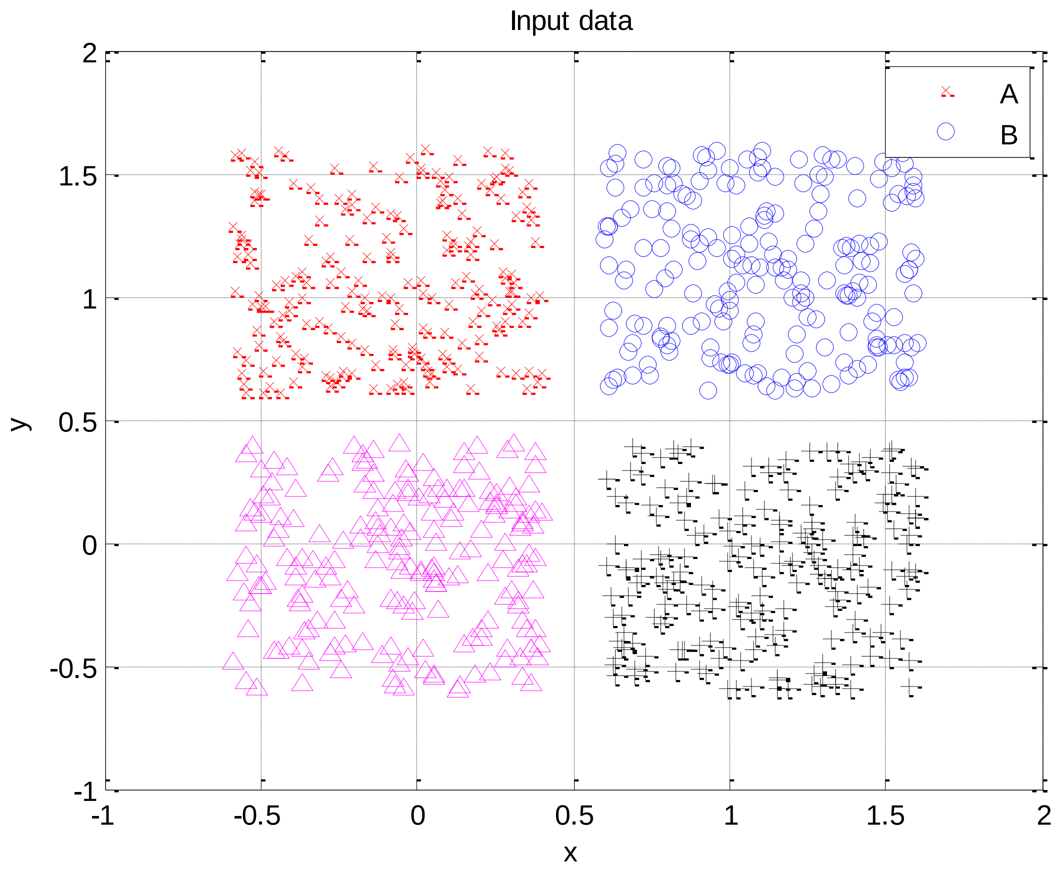
TESTING
n. of el. = 200
n. of errors = 2
mean error = 0.010000
std error = 0.099748

TOTAL
n. of el. = 400
n. of errors = 9
mean error = 0.022500
std error = 0.148489



Exercise 2 (1)

- Four classes classification of these data



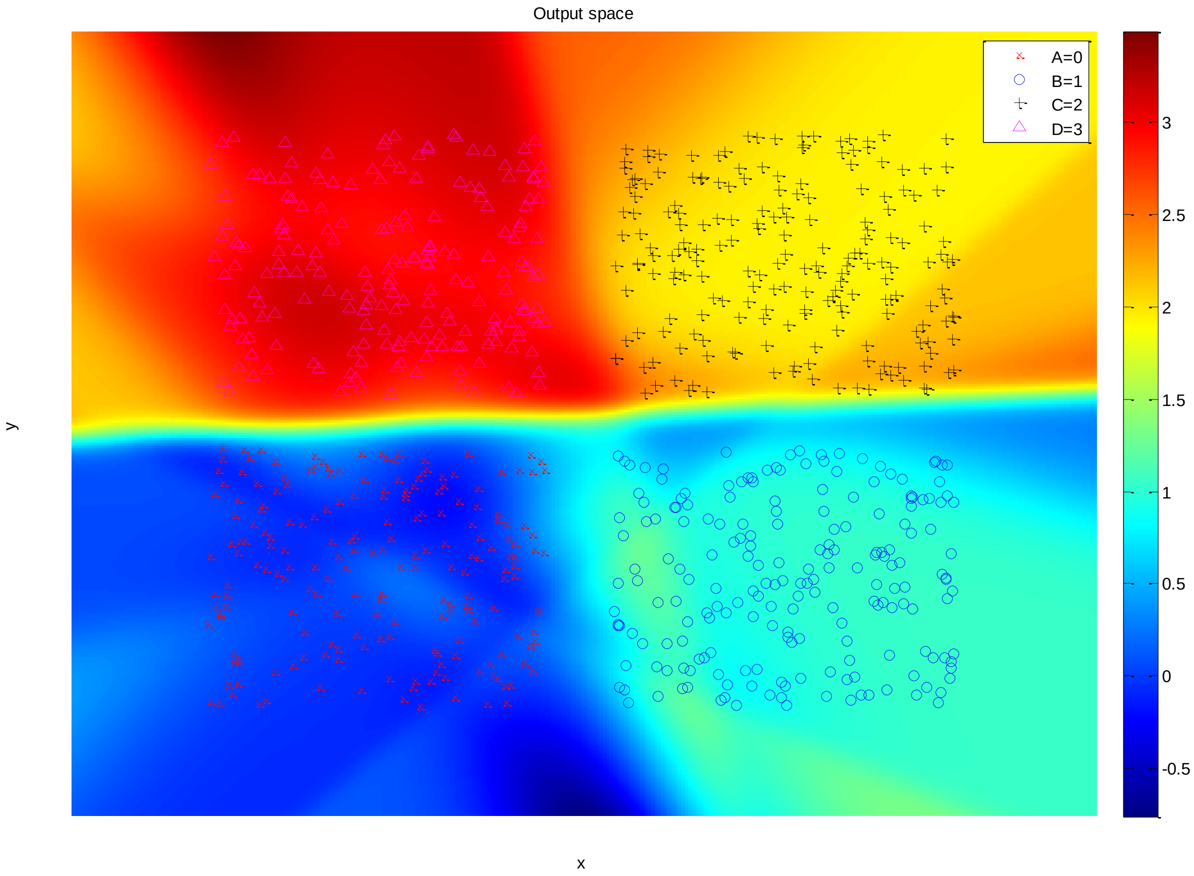
Exercise 2 (2)

- Example of results

TRAINING
n. of el. = 400
n. of errors = 2
mean error = 0.005000
std error = 0.070622

TESTING
n. of el. = 400
n. of errors = 1
mean error = 0.002500
std error = 0.050000

TOTAL
n. of el. = 800
n. of errors = 3
mean error = 0.003750
std error = 0.061161



Confusion matrix (1)

- A confusion matrix is a visualization tool typically used in supervised learning
- Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class

	box	clap	wave	jog	Run	Walk
Box	100	0	0	0	0	0
Clap	0	94	6	0	0	0
Wave	0	1	99	0	0	0
Jog	0	0	0	91	7	2
Run	0	0	0	10	89	1
Walk	0	0	0	0	6	94

Confusion matrix (2)

- Two classes

		prediction outcome		
		p	*n*	total
actual value	*p'*	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
total		P	N	

  - *True positives (TP)* - the number of elements correctly classified as positive by the test;
  - *True negatives (TN)* - the number of elements correctly classified as negative by the test;
  - *False positive (FP)* - also known as type I error, is the number of elements classified as positive by the test, but they are not;
  - *False positive (FN)* - also known as type II error, is the number of elements classified as negative by the test, but they are not.

Exercise 3

- Identify the sex of crabs from physical dimensions of the crab.
  - Six physical characterstics of a crab are considered: species, frontallip, rearwidth, length, width and depth.
- Problem:
  - 2 classes
  - 6 features
- The goal is to use the confusion matrix in a practical example
- MATLAB DEMO
  - http://www.mathworks.it/products/neural-network/demos.html?file=/products/demos/shipping/nnet/classify_crab_demo.html

Exercise 4

- A neural network that can classify wines from three wineries by thirteen attributes:
  - Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline

- The goal is to use the confusion matrix with more than two classes

- MATLAB DEMO
  - http://www.mathworks.it/products/neural-network/demos.html?file=/products/demos/shipping/nnet/classify_wine_demo.html

Cross Validation

- Cross-validation, sometimes called rotation estimation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set.
- Cross-validation is important in guarding against testing hypotheses suggested by the data, especially where further samples are hazardous, costly or impossible to collect.

k-Fold Cross Validation

- Algorithm
  - In *k*-fold cross-validation, the original sample is randomly partitioned into *k* equal size subsamples.
  - Of the *k* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining $k - 1$ subsamples are used as training data.
  - The cross-validation process is then repeated *k* times (the *folds*), with each of the *k* subsamples used exactly once as the validation data.
  - The *k* results from the folds then can be averaged (or otherwise combined) to produce a single estimation.
- The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.
- 10-fold cross-validation is the most commonly used.

k-Fold Cross Validation: Matlab script

```
TC = []; % T computed (results of NN)
TR = []; % T real(targets)

% k-fold validation is not automatic...
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio =  1;
net.divideParam.valRatio =  0;
net.divideParam.testRatio = 0;
```

```
% NEURAL NETWORK TRAINING AND K-FOLD VALIDATION
indices = crossvalind('Kfold', T ,k);

for i = 1: k
  % indexes of the testing and training elements
   test1 = (indices == i);
   test = find(test1 > 0);
   traini1 = ~test1;
   traini = find(traini1 > 0);
    % train a neural network
   [net,tr,Y,E] = train(net,P(:, traini), T(traini));
   % test
   testResultK = net(P(:,test));
   ind0 = find(testResultK < 0.5);
   testResultK(ind0) = 0;
   ind1 = find(testResultK >= 0.5);
   testResultK(ind1) = 1;
   % add elements to the global vectors TR and TC
   TR = [TR, T(test)];
   TC = [TC, testResultK];
end
```

k-Fold Cross Validation: Exercise

- Exercise 3 with 10-Fold Cross Validation

- Compare the previously obtained results and the ones obtained using 10-Fold Cross Validation