

# Neural Networks for Pattern Recognition:

## Homework 01

### I. MULTIPLE CHOICE / TRUE-FALSE

Select a choice **A** / **B** / **C**, etc., or indicate **TRUE** / **FALSE** for the following questions:

1. The #1 criterion for judging a supervised ML model is how well it explains/interprets what's going on in the Training data set
2. A pair of Learning Curves track a supervised ML model's performances on the Training and Test data sets
3. It is advisable for a supervised ML model to learn through repeated looks at data and gradual weight adjustments

### II. WRITTEN

Answer the following questions in < length of 3 tweets:

1. What does a cost function  $c(\dots)$  do? What goes into  $(\dots)$ ? What are  $c$ 's lower and upper numerical limits?
2. LOGISTIC REGRESSION: "signal"  $z_i = w_0 \cdot 1 + w_1 x_{1i} + w_2 x_{2i} + \dots + w_k x_{ki}$ , hypothetical prediction  $h_i = f(z_i)$ . Why is the **logistic function**  $h_i = \frac{1}{1 + \exp(-z_i)}$  a good function for Logistic Regression? Why is the **cross-entropy cost function**  $c_i = -y_i \ln h_i - (1 - y_i) \ln(1 - h_i)$  a good cost function?
3. What is a Big Risk when building a Big Machine that consumes Big Data containing many many variables? Name 2 ways to mitigate that problem.

### III. PROGRAMMING

Open Octave, navigate to the ***youTeach.machineLearn*** folder and enter ***start*** in the Command Window to activate the package.

We'll experiment with the Quality Control classification data set, with the following syntax to call the script:

**ffNN\_qc\_logReg**(weightPenaltyTerm\_\_\_\_ = **a real number**, bestStop\_\_\_\_ = **true or false**, numAdjustments\_perEpoch\_\_\_\_ = **an integer**, numEpochs\_\_\_\_ = **an integer**)

Call the script with the following input combinations, look at the **RESULTS** section and the plotted **Decision Boundaries**, and give the following 3 results for each case: **Training set actual classification accuracy %**, **Test set actual classification accuracy %**, and one of the following: **Good Fit** / **Overfit** / **Underfit**. Example: "85.4%, 78.6%, Good Fit".

1. 0.0, false, 10, 30
2. 0.0, true, 1, 100

3. 0.3, true, 1, 30

(the first run may take some time because it needs to warm the graphic system up - patience appreciated)