

Assignment 1: (Shallow) MLP neural network

Due date : 2016.04.05

Warning!! This assignment requires much time to train the neural network. We recommend you to get a start on it as soon as possible.

Important: Please submit your assignment to the povis LMS system.

Be as compact and to the point as possible in your description.

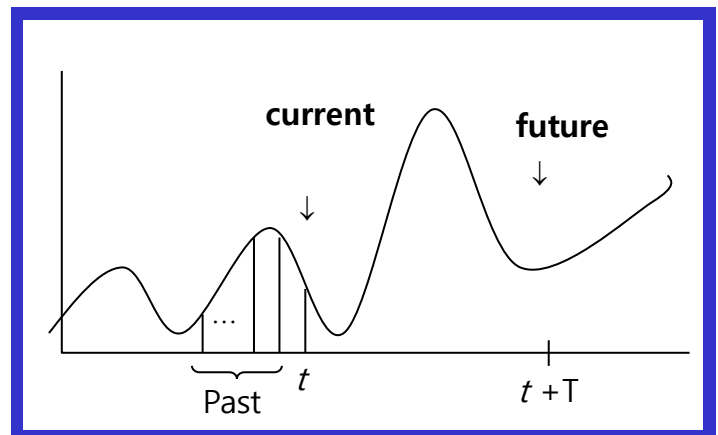
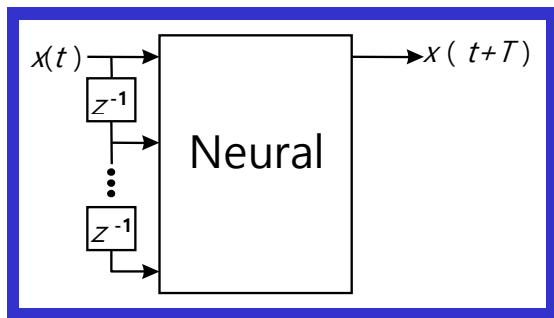
The results must be presented well and compactly to gain more credit.

In this assignment you will practice developing a simple classification algorithm, using a shallow neural network architecture.

- Understand the basic **neural network architecture** (neuron, transfer function, ...).
- Understand the **back-propagation** algorithm.
- Understand the train/validation/test data **splits** and the use of validation data for **hyper-parameter tuning**.
- Implement and apply a shallow neural network (having 1 or 2 hidden layer – choose the better one).
- Implement the **back-propagation** algorithm and train the neural network using this algorithm.
- Get a basic understanding of the **performance** and **limitations** of the shallow neural network.
- Reference: course textbook, Matlab - neural network toolbox.

Q1. Stock market prediction

- A stock market prediction is attempted here. The enclosed “Stock_Price.txt” file stores an annual listing of the stock price for a Korean company.



Given	Predict
$x(t - N + 1)$ $x(t - N + 2)$ \vdots $x(t)$	$x(t + T)$

- Design an MLP with 2 hidden layers (if needed, use jumping connections or even a general Feedforward Net) for
 - $T = 1$
 - $T = 10$
 - $T = 30$
 - $T = 180$
 Vary N and recommend its best value for each T .
- To assess the net's generalization capability, try the following 4 ways of training/testing.

Training (100 % samples)		Test on Training (100 % samples)	
Training (First 60 % samples)		Test (Last 40 % samples)	
Training (60 % random samples)		Test (Rest of the samples)	
Learning (50 % random)	Validation (10 % random)	Test (Rest 40 % of the samples)	

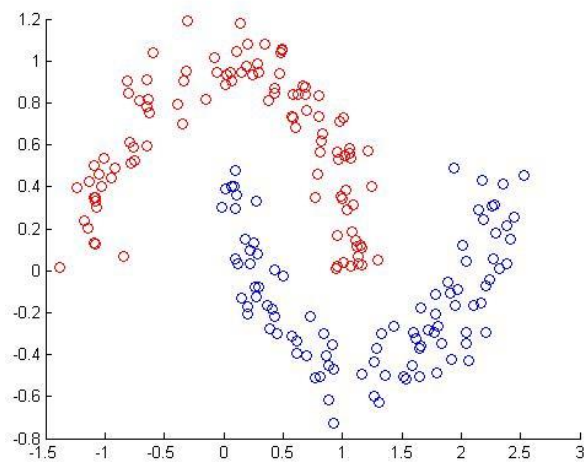
- To present your results of training:
 - Compute the learning curves – RMS Error vs. iteration - in one graph for the training, validation [only for Case 4 of training/testing], and the test data. The horizontal axis must be in # iterations. Also indicate the time to reach the best performance.
 - Plot the NN predicted data along with the ground truth in the same graph.
 - Extrapolate the data into the future: $x(248)$ to $x(277)$ and append this to the waveform plot above.
 - Show how the RMS error changes with $T = 1, 10, 30, 180$.
 - Repeat the above for a general feedforward network and compare.

Q2. Two moon dataset

- Detailed description: Design a shallow neural network to classify the two moon dataset shown below. Its data – two_moon.txt, two_moon.jpg – is provided.
- From the 200 total data points, take 50 %, 20 %, 30 % of them as the learning, validation, and test set.
- Show the training and generalization performance using the confusion matrix below:

Truth \ Recognized as	Recognized as		
	# 1	# 2	
Class # 1			} for training (using validation data)
# 2			
# 1			} for test data
# 2			

- DB: Two moon dataset

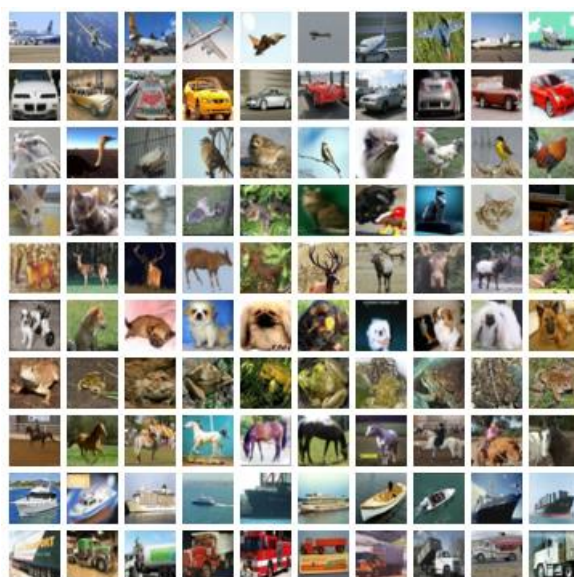


Q3. CIFAR-10

- Detailed description: Design a shallow neural network to classify the CIFAR-10 dataset. You can download the data from CIFAR-10 webpage (<http://www.cs.toronto.edu/~kriz/cifar.html>). It will be up to you to train the best network as much as you can. Show the training and generalization performance for each class data and total data using the table below

											(%)
Average precision	Airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	Mean
training											
test											

- DB : CIFAR-10



10 classes: Airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck

Submitting your work: Submit a report per team about Q1, Q2 and Q3 until the due date. The report should include details on (if not, you shall get a penalty):

- What you learned through this work.
- How to implement the neural network architecture and training algorithm. Attach some parts of your code, and explain the algorithm for the codes. If you use your own code, you will get extra credits.
- Hyper-parameters of the shallow neural network architecture and training method.
- Training, validation, test curves and the classification result.
- Experiment hardware specification and training time.
- The specific role of each member in your team, in great detail.

Contact TA

Name : 박병화 (Byung-Hwa Park)

Office : C5 247

Office phone : 054-279-2904

Mail : pbh0616@postech.ac.kr

Name : 윤여빈 (Yoon YuBin)

Office : LG302

Office phone : 054-279-2904

Mail : yubin87@postech.ac.kr