Knowledge-based Systems and Artificial Intelligence

Ghent University

Neural Networks

1. Introduction

In this practicum we will use the 'Neural Network Toolbox' from Matlab 8.1 (available through Athena @ UGent) to get familiar with different theoretical and practical aspects of Neural Networks.

Read in preparation of the practice the following sections from the toolbox user's guide [1]:

- Getting Started
- Neuron Model and Network Architectures
- Backpropagation ("Train the network")

We will use feedforward networks and the most simple backprogation algorithm (gradient descent). The practice aims to investigate the following training aspects of artificial neural networks:

- Representativeness of the training set
- Use of the validation and test data
- Specialization versus generalization
- Influence of the network topology on the performance

2. Database

The data was obtained from the National Institute of Diabetes, Digestive and Kidney Diseases, USA [2]. The "Pima Indians Diabetes Database" contains data of 768 women in the area of Phoenix, Arizona, from which 268 result positive and 500 tested negative for diabetes.

The following attributes describe the population:

- Number of pregnancies
- Plasma glucose concentration after 2 hours of an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skinfold thickness (mm)
- 2-hour-insulin serum (mu U/ml)
- Body Mass Index (kg/m2)
- Predisposition to diabetes
- Age (years)
- Diabetes diagnosis (1: positive, 0: negative)

3. Matlab - Neural Network Toolbox

The following functions of the Neural Network Toolbox are useful:

- "mapstd": function that normalizes the input/output of a neural network.
- "dividevec": function that splits a dataset in training, validation, and test set.
- "newff": function to create a feedforward network.
- "init": function that initializes a neural network.
- "train": function that trains a neural network.

4. Training feedforward networks

4.1 Distribution of the training, validation and test set

Create an m-file and evaluate the following instructions:

a) Load the diabetes database and split the dataset in patterns *P* (input) and target values *T* (output):

```
diabetes = dlmread('pima-indians-diabetes_data.txt',',');% data
P = diabetes(:,1:end-1)';% patterns
T = diabetes(:,end)';% targets
```

b) Normalize the input P:

```
[PN, PS] = mapstd(P);% normalized patterns
```

c) Divide the normalized data into a training, validation and test set:

```
[TRN, VAL, TST] = dividevec (PN, T, 0.15,0.15);% split data
```

d) Create the following auxiliary variables:

```
P1 = TRN.P;% training set patterns
T1 = TRN.T;% training set targets
P2 = VAL.P;% validation set of patterns
T2 = VAL.T;% validation set targets
P3 = TST.P;% test set patterns
T3 = TST.T;% test set targets
```

e) Compare the histograms of T1, T2, T3.

```
figure
hist (T1)
xlabel('targets')
ylabel('Frequency')
title('Histogram of targets (Training set)')

figure
hist (T2)
xlabel('targets')
ylabel('Frequency')
title('Histogram of targets (Validation set)')

figure
hist (T3)
xlabel('targets')
ylabel('Frequency')
title('Histogram of targets (Test set)')
```

f) Also compare the histograms of the attributes of P1, P2, P3:

```
figure
[A1, bin1] = hist (P1(1, :));% histogram of attribute 1 in the training set
[A2, bin2] = hist (P2(1, :));% histogram of attribute 1 in the validation set
[A3, bin3] = hist (P3(1, :));% histogram of attribute 1 in the test set
plot (bin1, A1, bin2, A2, bin3, A3)
xlabel('Attribute 1')
ylabel('Frequency')
```

```
title('Distribution of attribute 1 with normalization')
legend('Training set','Validation set','Test set')
grid
```

Q1: Display the distribution of P1, P2, P3 without normalizing the input, and explain the differences when the inputs are normalized.

```
[TRN, VAL, TST] = dividevec (P, T, 0.15,0.15);% split data

P1 = TRN.P;% training set patterns
T1 = TRN.T;% training set targets
P2 = VAL.P;% validation set of patterns
T2 = VAL.T;% validation set targets
P3 = TST.P;% test set patterns
T3 = TST.T;% test set targets

[A1, bin1] = hist (P1(1, :));% histogram of attribute 1 in the training set
[A2, bin2] = hist (P2(1, :));% histogram of attribute 1 in the validation set
[A3, bin3] = hist (P3(1, :));% histogram of attribute 1 in the test set

figure
plot(bin1, A1, bin2, A2, bin3, A3)
xlabel('Attribute 1')
ylabel('Frequency')
title('Distribution of attribute 1 without normalization')
legend('Training set','Validation set','Test set')
grid
```

4.2 Configuration and training of a neural network

Open the file *ExerciseConfig.m* and verify the following configuration of the neural network:

Network Type: Feed-forward backpropagation
Input ranges: Get input from P1
Training function: TRAINGD
Adaptive learning function: LEARNGD
Performance function: MSE

- Number of layers: 2
- Transfer function (layer 1): LOGSIG
- Number of neurons (layer 1): 10
- Transfer function (layer 2): PURELIN

- epochs: 1000
- Goal: 0
- lr (learning rate): 0.005
- max-fails: 1000 (only applicable if a validation set is used)
- min-grad: 0
- show: 5

Run the script and follow the evolution of the performance in function of the number of epochs. Note that the quadratic error decreases as the training progresses. The training phase is completely determined by the number of 'epochs'. Also, it may be stopped when the gradient is very small, which means that the system converged to a local minimum.

Q2. Run the script without normalizing the input data. What did you determine? Explain and report the differences. Why the input normalization is recommended for any adaptive process?

Q3. Come back to the initial configuration of the network and run the script using different learning rates (e.g. 0.5, 0.05, 0.005, 0.0005). Describe the effects on the learning curve and report them.

4.3 Generalization versus specialization

In the file *ExerciseConfig.m* comment the following command:

```
[TRN,VAL,TST]=dividevec(PN,T,frac/2,frac/2);
```

Now, uncomment these lines:

```
load('TRN.mat')
load('VAL.mat')
load('TST.mat')
```

Q4. Run the script and check the evolution of the learning curve. Note that the error of the training and validation set initially decreases, but then the error of the validation curve starts increase while training error decreases. Is the training phase experiencing generalization or specialization? Explain your decision in your report.

Q5. The data used in the previous exercise were split as follows: TRN = 70%, VAL = 15% and TST = 15%. Now, run the neural network using the *dividevec* command using different data distributions (for example, with a relatively small amount of training data, say 5%). What are the effects on the performance of the training, validation and test set?

4.4 Influence of the number of neurons

Q6. Consider a neural network with 2 hidden layers. Evaluate the performance of this network by testing it with different number of neurons. Run the m-file named *feedforward* and check the evolution of the MSE as the number of neurons increases. Report your results and explain the behavior of the curve. What is the impact on the generalization and specialization of the network by increasing the number of neurons?

Q7. What would happen if the training, validation or test set are not representative? Run the m-file with different data distributions and report your results.

5. Assignment

Prepare a report of the practice including your results, graphs, explanations, conclusions, etc., for each task/question formulated in this document. Make sure to include the seed "rnd(mySeed)", which was used for the pseudo-random numbers.

Important remarks:

- The assignment is individual.
- The submitted files should follow this name convention:
  NeuralNetworks_LastName_FirstName.Extension
- Please indicate the number of the question when you answer it, and support it with a graph.
- It is the student's responsibility to check that the submitted documents are complete and not corrupted. Avoid any situation that prevents the correct evaluation of your report.
- Deadline: **December 19, 2014**

References:

[1] H. Demuth, M. Beale, M. Hagan, "Neural Network Toolbox User's Guide", The MathWorks, 1992-2006.

[2] V. Sigillito, "Pima Indians Diabetes Database", National Institute of Diabetes, Digestive and Kidney Diseases, 1990. (https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes)