Multilayer Perceptrons (MLPs) and Backpropagation

Using MLPs with 1 hidden layer, create an ANN model of the following 3 problems and train them. Do it in Jupiter Notebook and document your training with program output and charts. The number of neurons in hidden layer and the training constant will need to be varied to find out what sized network models the data.

Capture the work you do on these problems in a Jupyter Notebook.

Strictly speaking we should call them sigmoid neurons rather than perceptrons as the sigmoid activation function is used rather than the step or threshold function.
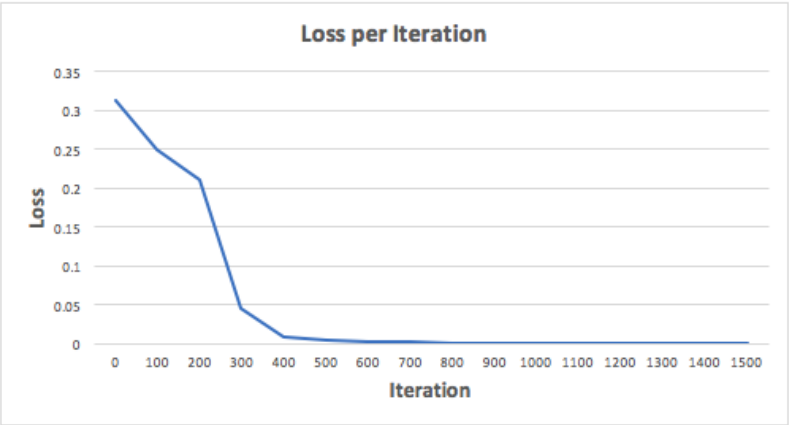
Problem 1

Here we consider a MLP with a hidden layer of 4 sigmoid neurons and the following is the data set.

X1	X2	X3	Y
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

Weights 1   Weights 2

Input Layer   Hidden Layer   Output Layer
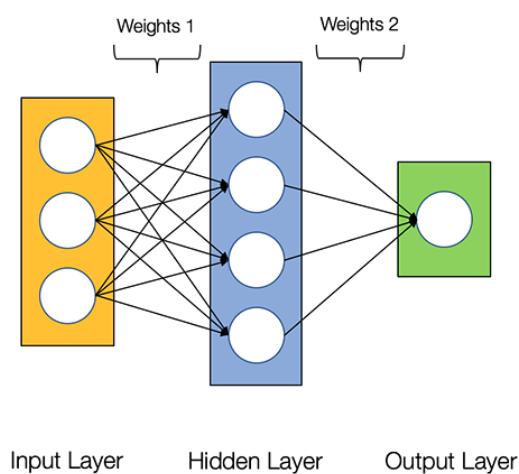
Start with sigmoid neurons and train the network with 2000 iterations. Try different learning rates, see which learning rate make the learning unstable.

Loss per Iteration

Let's look at the final prediction (output) from the Neural Network after 1500 iterations.

Prediction	Y (Actual)
0.023	0
0.979	1
0.975	1
0.025	0

Predictions after 1500 training iterations

Problem 2 - Neural Network with 3 input and 2 output

Try a MLP on the following training data. Vary the number of neurons in the hidden layer, experiment with learning rates and epochs. Analyse the training.

```
X_training=[[ 1,  1,  0],
            [ 1, -1, -1],
            [-1,  1,  1],
            [-1, -1,  1],
            [ 0,  1, -1],
            [ 0, -1, -1],
            [ 1,  1,  1]
           ]
y_training=[[1, 0],
            [0, 1],
            [1, 1],
            [1, 0],
            [1, 0],
            [1, 1],
            [1, 1]
           ]
```

Problem 3 - Transportation Mode Choice

Suppose we have the following 10 rows of training data.

The training data is supposed to be part of a transportation study regarding the mode choice to select bus, car or train among commuters along a major route in a city, gathered through a questionnaire study. For simplicity and clarity, we selected only 4 attributes. Attribute 'gender' is a binary type, while 'car ownership' is a quantitative integer. 'Travel cost/km' is a quantitative of ratio type but here it was converted into an ordinal type. 'Income level' is also an ordinal type. Train a neural network to predict the transport mode of a person, given the four attributes: gender, car ownership, travel cost, and income level.

Training Data

Gender	Car ownership	Travel Cost	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

After training the neural network using the data above, try to predict the Transportation Mode Choice of the following instance of data: *Female without car ownership, willing to pay expensive travel cost and having medium income level.*

Using Pandas, see if you can copy the data to a dataframe and save it to a comma separated text file, **transport.csv**.

Problem 4 - Iris flower classification

See problem dataset elsewhere.

```
1. Title: Iris Plants Database
      Updated Sept 21 by C.Blake - Added discrepency information

2. Sources:
     (a) Creator: R.A. Fisher
     (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
     (c) Date: July, 1988

3. Past Usage:
    - Publications: too many to mention!!!  Here are a few.
    1. Fisher,R.A. "The use of multiple measurements in taxonomic problems"
       Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions
       to Mathematical Statistics" (John Wiley, NY, 1950).
    2. Duda,R.O., & Hart,P.E. (1973) Pattern Classification and Scene Analysis.
       (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.
    3. Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System
       Structure and Classification Rule for Recognition in Partially Exposed
       Environments".  IEEE Transactions on Pattern Analysis and Machine
       Intelligence, Vol. PAMI-2, No. 1, 67-71.
       -- Results:
          -- very low misclassification rates (0% for the setosa class)
    4. Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE
       Transactions on Information Theory, May 1972, 431-433.
       -- Results:
          -- very low misclassification rates again
    5. See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al's AUTOCLASS II
       conceptual clustering system finds 3 classes in the data.

4. Relevant Information:
    --- This is perhaps the best known database to be found in the pattern
        recognition literature.  Fisher's paper is a classic in the field
```

and is referenced frequently to this day. (See Duda & Hart, for
example.) The data set contains 3 classes of 50 instances each,
where each class refers to a type of iris plant. One class is
linearly separable from the other 2; the latter are NOT linearly
separable from each other.
--- Predicted attribute: class of iris plant.
--- This is an exceedingly simple domain.
--- This data differs from the data presented in Fishers article
(identified by Steve Chadwick, spchadwick@espeedaz.net)
The 35th sample should be: 4.9,3.1,1.5,0.2,"Iris-setosa"
where the error is in the fourth feature.
The 38th sample: 4.9,3.6,1.4,0.1,"Iris-setosa"
where the errors are in the second and third features.

5. Number of Instances: 150 (50 in each of three classes)

6. Number of Attributes: 4 numeric, predictive attributes and the class

7. Attribute Information:
   1. sepal length in cm
   2. sepal width in cm
   3. petal length in cm
   4. petal width in cm
   5. class:
      -- Iris Setosa
      -- Iris Versicolour
      -- Iris Virginica

8. Missing Attribute Values: None

Summary Statistics:
            Min Max   Mean   SD   Class Correlation
   sepal length: 4.3 7.9   5.84 0.83   0.7826
    sepal width: 2.0 4.4   3.05 0.43   -0.4194
   petal length: 1.0 6.9   3.76 1.76   0.9490 (high!)
    petal width: 0.1 2.5   1.20 0.76   0.9565 (high!)

9. Class Distribution: 33.3% for each of 3 classes.