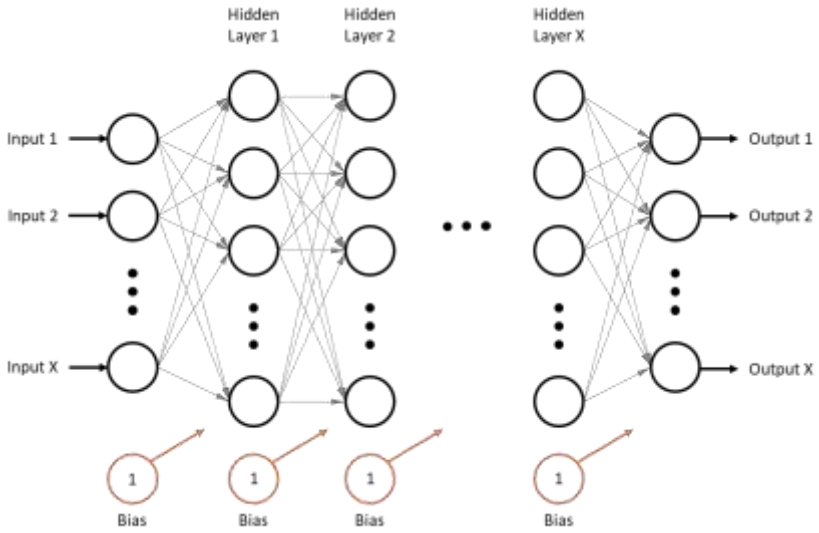
Job Summary & Deliverables

1) Create a simple Feed Forward Multilayer Perceptron Neural Network as a Python3 script that can be customized easily by taking user inputs through a Config.txt file (number of inputs/outputs/hidden layers/nodes in hidden layers/inclusion of Bias nodes/learning rate/epochs/mini-batches/activation functions). A CSV file will provide training data. The general topology of the network is shown below.



2) The code should have comments to identify what each section is doing. Please provide specific command line instructions to use the script, including any needed dependencies.

3) At the conclusion of training, the script should automatically output a CSV file containing the trained weights, one CSV file for each set of weights.

4) At the conclusion of training, the script should automatically output a file that contains the complete network as a mathematical equation or model. I hope this will not be challenging as a Feed Forward Perceptron network is essentially just a series of matrix multiplication, summation and applying an activation function for each layer. My goal is to be able to import the equation and weight CSV matrices into another application, like Microsoft Excel, and feed sample inputs through it outside of the Python environment. I am open to suggestions on the best way to do this.

Config File

The Config file should contain the following fields, editable by the user:

Input_Filepath –Location and filename of the input data.
Example: C:/folder1/folder2/inputs.csv

Input_Headers – Is the first row in the CSV file headers? True or False.
Example: True

Input_Columns – The CSV file may contain both input and output fields for training in the same file. Must be able to specify which fields are Inputs.
Example: 1-60
Suppose the CSV file has 65 fields. In this example, the user specifies that fields 1-60 are Inputs.

Matthew Tonneson, 18-Jun-2018

Output_Filepath –Location and filename of the onput data. This may be the same location and file as the Input_Filepath if both Input and Output data are contained in a single file.

Example: C:/folder1/folder2/output.csv

Output_Headers – Is the first row in the CSV file headers? True or False.

Example: True

Output_Columns – The CSV file may contain both input and output fields for training in the same file. Must be able to specify which fields are Outputs.

Example: 61-65

Suppose the CSV file has 65 fields. In this example, the user specifies that fields 61-65 are Outputs.

The Hidden_Layers field below is my own idea for specifying the number of hidden layers and nodes in each hidden layer. I am open to alternate ways to specify this information.

Hidden_Layers – Specifies the number of hidden layers in the network, along with the number of nodes to be included in each layer.

Example: 75,50,25

The example specifies there will be three hidden layers, the first with 75 nodes, the second with 50 nodes a third layer with 25 nodes. The number of Input nodes is determined from the Input_Columns value, and the number of Output nodes is determined from the Output_Columns value.

Bias Nodes – Will Bias nodes (value 1) be included as inputs to each layer of the network? True or False.

Example: True

For the Learning_Rate field below, I would like to be able to specify either my own learning rate, or the ADADELTA method (link). I am open to alternate ways to specify this information.

Learning_Rate – Specify gradient decent.

Example: 0.01

Example: ADADELTA

Epochs – Specify number of training epochs.

Example: 30

Mini_Batches – Size of the mini batches to use when sampling.

Example: 10

For the Activation_Function field below, I would like to be able to specify Sigmoid, TanH, or ReLU functions. I am open to alternate ways to specify this information.

Activation_Function

Example: Sigmoid ← $f(x) = \frac{1}{1+e^{-x}}$

Example: TanH ← $tanh(x) = \frac{2}{1+e^{-2x}} - 1$

Example: ReLU ← $f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$

Matthew Tonneson, 18-Jun-2018