CSC520 Fall 2021 Assignment 6

Due November 29^{th} at 11:59pm

This assignment consists of two questions which involve written answers and code. In order to complete the assignment you must submit a written report in pdf form detailing your answers to the questions as well as your code. As discussed in class all work *must* be your own. You may not use third party libraries or example code to complete the assignment with the exception of csv file loaders. All reports must be clear and well written. All code must be clear, readable, and well-commented. Upload your report as a file called `<unityid>-Assign6.pdf` and your sourcecode in a zip called `<unityid>-Assign6.zip`.

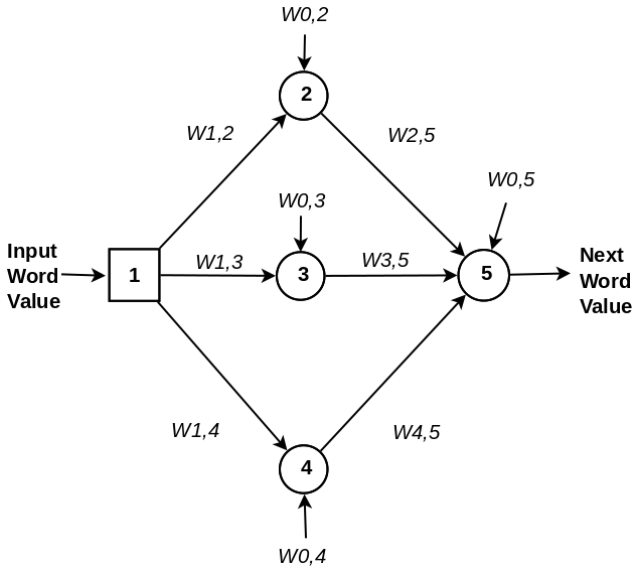
Question 1 Basic Next Word Prediction (60 points)

For this assignment you will implement a basic neural network with a single hidden layer containing 3 nodes. The purpose of this network will be to perform *next-word-prediction*. You have been given two files. Representing training and testing sets for the mapping function. Each entry in the files is a comma-separated pair of numbers which represent word indices. Your job is to build a simple single layer NN that predicts these pairings.

On execution your code must:

1. Iterate over the word pairs in the training file and apply the *simple* hill-climbing algorithm described below to update the weights.
2. Use the supplied network to predict values on the test set and record your performance and print the results to the user.

For the update function you will use the simple gradient descent algorithm shown in Lecture 22, with one key change, we will not worry about convergence, only a single pass through the training set. *NOTE: There is no expectation that these models will be the best performing. The function of this assignment is to implement the structure, not to implement the best possible NLP model.*



Your code should be called as follows:

```
NeuralModel.py <ALPHA> <TRAIN> <TEST>
```

Where the arguments specify the learning rate (α) and the necessary files

On execution your code should print out the the training and testing events as follows:

```
Train:  <Predicted Value> <Actual Value>

Test:  <Predicted Value> <Actual Value>
```

And it should print out the final performance as a sum of squared errors $(prediction - actual)^2$

Question 2 Code Analysis (20 points extra credit)

In order to complete this assignment you will need to produce two versions of your code above, the existing one with 3 internal nodes, and a second with 10. You will then compare the performance of the two on the test set with different learning rates. Along with your extra code submit a report answering the following questions. You may use figures or statistics as appropriate.

1. How did the two models compare on the training and testing sets?
2. Was there a clear relationship between the alpha values and the performance on the data?