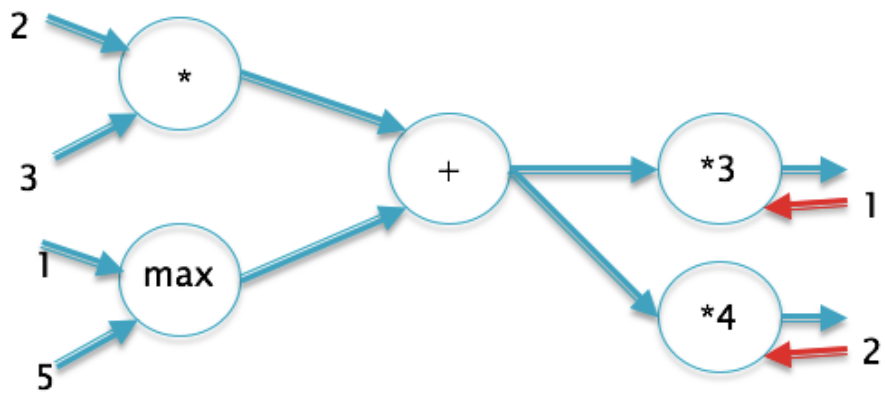
Deep Learning
Assignment 2

1. Consider the following computational graph:



Use the Gradient Flow Rules to compute the gradients for all the arcs in the graph.

(2 points)

2. Consider a Deep Feed Forward Network, of the type shown on Page 24 of Lecture 4 Slides, composed of an Input Layer with 10 nodes, followed by a Hidden Layer with 50 nodes and finally an Output Layer with 3 nodes. Assume the activation function f is given by ReLU, and the output function h is given by Softmax.
    a. Compute the number of parameters (weights and biases) required to describe the network.
    b. Write a Python program to do a forward pass through the network. Assume that the input training sample X = (0.5, 0.6, 0.1, 0.25, 0.33, 0.9, 0.88, 0.76, 0.69, 0.95). Also assume that all the weights and biases are initialized according to a Uniform distribution between [0,0.1]
    c. Assuming that the output Label corresponding to X, is T = (1, 0, 0). Using the results of part (b), compute the Cross Entropy Loss Function.
    In order to verify the forward pass, the Loss Function should compute to log(3), explain why.
    d. Write a Python program to do a backward pass through the network, and compute all the delta values.
    e. Finally extend the Python program to compute the new weight parameters using stochastic gradient descent.
    In order to verify that the algorithm has been coded correctly, you can do the following: Put the forward and backward passes in a loop (with the single training sample given in Part (b) and the corresponding Label in Part (c)). After several

iterations you should see the Loss Function go to zero (explain why).

(15 points)

3. With reference to the Keras program “MNIST Dataset” (distributed as part of the Week 4 modules): Modify the Neural Network to add 2 hidden layers consisting of 50 nodes each. Investigate the effects of the following Optimization Algorithms on the rate of convergence, by plotting the Validation Accuracy (on a per Epoch basis) vs Number of Epochs. For each case, note down the number of epochs at which the Validation Accuracy stops increasing.
    a. Regular Gradient Descent
       optimizer = tf.train.GradientDescentOptimizer(learning_rate = 0.001)
    b. Momentum
       optimizer = tf.train.MomentumOptimizer(learning_rate = 0.001, momentum = 0.9)
    c. Adam
       optimizer = tf.train.AdamOptimizer(learning_rate = 0.001)

(5 Points)

4. The Keras program “Fasion Dataset (Linear Model)” (in Week 4 Modules) is a Linear Model in which the input dataset corresponds to grayscale coded images of various clothing items. Just as in MNIST, there are 10 classes and each image is a 28*28 matrix (see https://www.tensorflow.org/tutorials/keras/basic_classification for more information about this dataset). Answer the following:
    a. What is the best classification accuracy that you can achieve with the Linear Model on this dataset?
    b. Replace the Linear model with a Fully Connected Feed Forward Model with one or more Hidden Layers. What is the best classification accuracy that you can achieve with this model? In order to get better performance , you can vary the number of hidden layers, the number of nodes per hidden layer, the gradient descent algorithm, add regularization etc.

(5 Points)