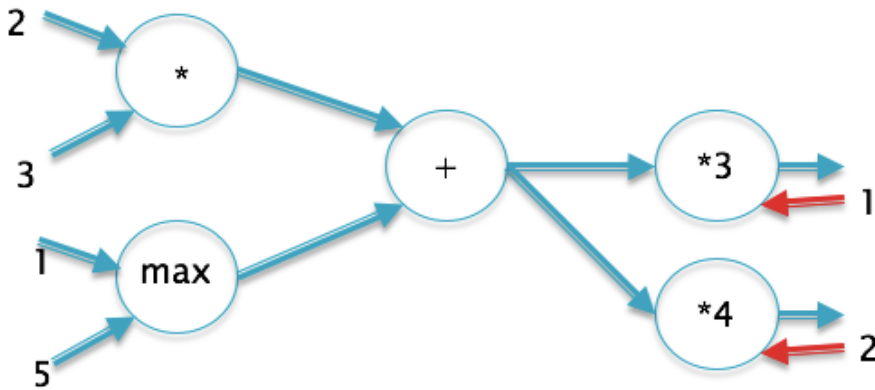


Deep Learning

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 16 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. The MNIST_Fashion dataset comes with Keras, and can be loaded using the commands:

```
fashion_mnist = keras.datasets.fashion_mnist
```

```
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

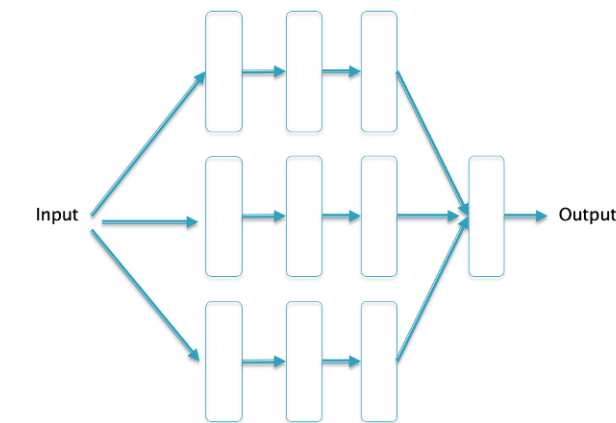
This dataset corresponds to grayscale coded images of various clothing items. Just as in MNIST, there are 10 classes and each image is a 28*28 matrix (see

https://www.tensorflow.org/tutorials/keras/basic_classification for more information about this dataset). Answer the following:

- What is the best classification accuracy that you can achieve with the Linear Model on this dataset?
- 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 and the number of nodes per hidden layer

(8 Points)

4.



The figure above shows an example of Ensemble Processing (see Lecture 4). Start with the best model that you have after doing Problem 3, and replicate that model 3 times in a parallel configuration as shown in the figure (use the Keras Functional API for doing this). Take the final logit nodes from each branch and average it to create the final logit layer for the model (see https://keras.io/api/layers/merging_layers/average/ on how to do this). Train the model again on the fashion_mnist dataset, and check for an improvement in

prediction accuracy (if any).

(5 Points)