Umut Akös
21202015
CS-464 HMW3

Q-1 -

Assuming we provided a two hidden layer neural network designed to solve a binary classification
problem. This architecture uses ReLU activation function between two hidden layers.

Final Activation function -:
For the final Layer we will use Sigmoid activation function as it gives the output value between 0 to 1. And for binary classification we use sigmoid for final layer and softmax function for the multiclass classification.

Error function -:

For this case we will use log loss function as the error function. It is a non-negative value, where the robustness of model increases along with the decrease of the value of loss function. Due to the non-convex curve arise in mean square error function we are not using it.

Variable used :

X : Input

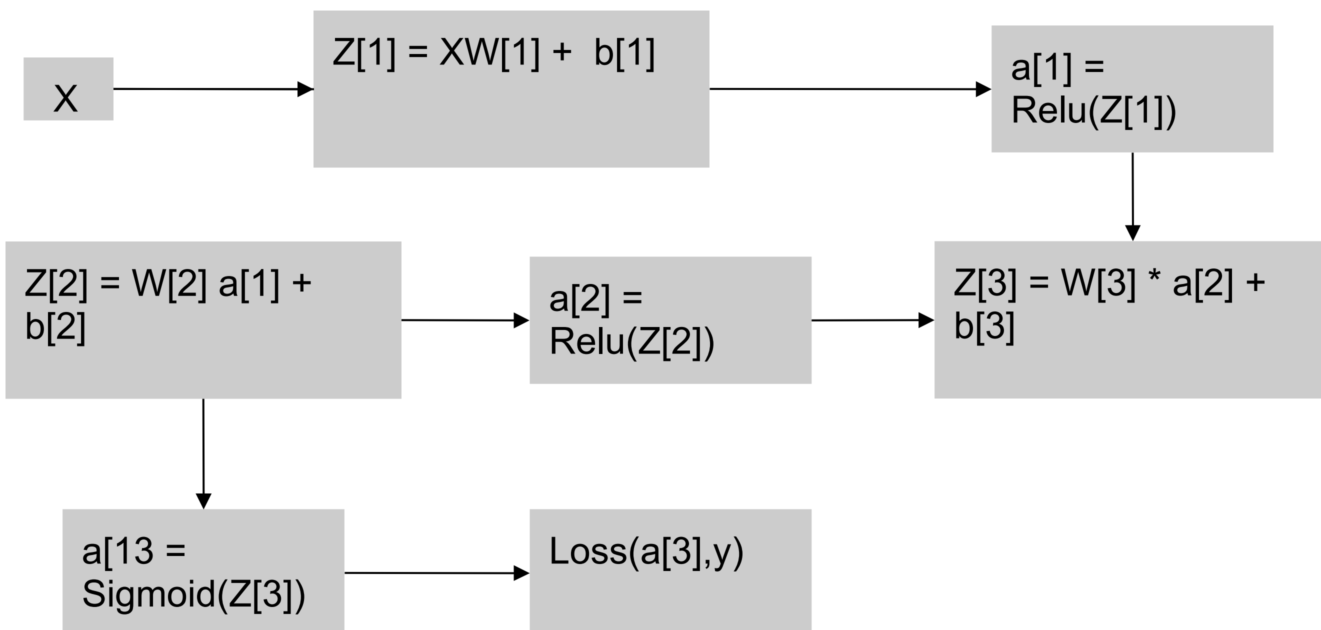
W[1] = Weights of first layer in Matrix form
b[1] = Bias of first layer in Matrix form

W[2] = Weights of 2nd layer in Matrix form
b[2] = Bias of 2nd layer in Matrix form

W[3] = Weights of 3rd layer in Matrix form
b[3] = Bias of 3rd layer in Matrix form

Relu = max(0,x)
Sigmoid = e^x / (1 + e^x)

Network Block with Forward propagation:



Back Propagation :-

Loss(a[3],y) = -y log(a[3]) - (1-y) * log(1 - a[3])

(dL/ da[3]) = (-y / a[3]) + ((1-y) / (1 - a[3]))

(dL/dZ[3]) = (dL/da[3]) * (da[3]/dZ[3])
= (dL/da[3]) * g’(z[3])
= a[3] - y

(dL/dW[3]) = (dL/dZ[3]) * a[2]
(dL/db[3]) = (dL/dZ[3])

(dL/dZ[2]) = (dL/da[3]) * (da[3]/dZ[3]) * (dZ[3]/da[2]) * (da[2]/dZ[2])

We know that

(dL/ dZ[3])   = (dL/da[3]) * (da[3]/dZ[3])
(dZ[3]/da[2]) = W[3]
 (da[2]/dZ[2]) = {1 if x > 0 otherwise 0}
(dL/dZ[3]) = a[3] - y

Putting in the equation we get

(dL/dZ[2]) = (a[3] - y) * W[3] * (0 or 1)

dL/dW[2] = (dL/dZ[2]) * a[1]
db[2] = dL/dZ[2]

Similarly we can find the result for
dL/dZ[1] = dL/dZ[2] * W[2] * {0 or 1}
dL/dZ[1] = (a[3] - y) * W[3] * W[2] * (0 or 1)

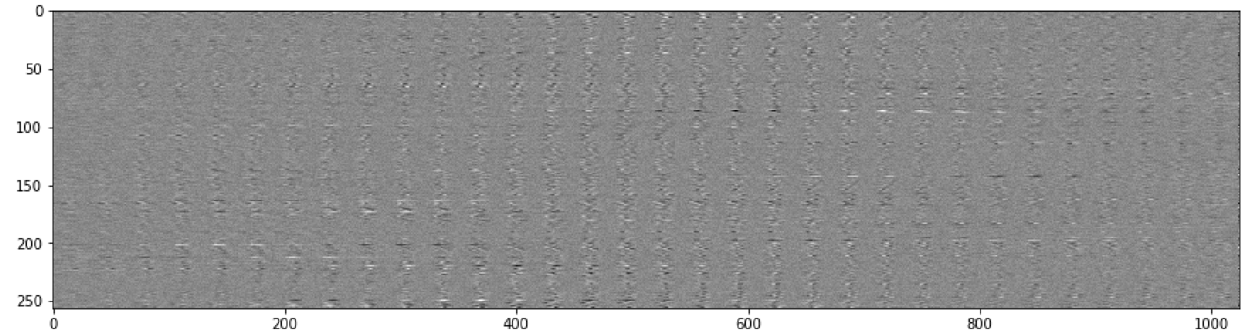
dL/dW[1] = (dL/dZ[1]) * X
db[1] = dL/dZ[1]

Here we find gradients of all the parameters.

Q-3

For this neural network we will use the softmax Activation function in pytorch.
Loss function again will be log loss function in the mlp.

Weight Visualisation For Neural Network



Weight Visualisation For Convolutional Neural Network