Machine Learning Assignment 3 : Neural Network for classification
Bhawna Bhoria (M22MA003)

The model architecture satisfies the following conditions:
1. Number of neurons in the output layer: Equal to the number of classes - Number of output classes - 10 . The given data is transposed to convert the label column into a row and then fetch the label values with a single index of the array.

2. Activation function: Sigmoid (except in the output layer); Softmax in the output layer

Sigmoid Activation function value

output = 1 / (1 + e^(-activation))

$$\begin{aligned} S(x) &= \frac{1}{1+e^{-x}} \\ &= \frac{e^x}{e^x+1} \end{aligned}$$

Softmax Activation function value

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

3. Weight initialization: Random : Using random.rand function

```
      W1 = np.random.rand(hidden_nodes, input_nodes) - 0.5
```

4. Use Forward propagation to find the pre-activation and activation values.

```
 Z1 = np.matmul(W1, X) + b1
 A1 = sigmoid_act(Z1)
 Z2 = W2.dot(A1) + b2
 A2 = softmax(Z2)
```

5. Find the derivative/gradient of parameters with back propagation to update parameters with learning rate.

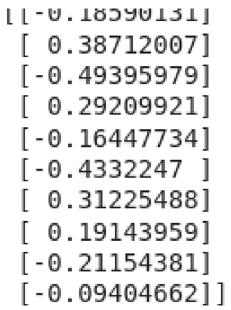
```
dW1, db1, dW2, db2,one_hot_encoded_Y = back_prop(Z1,A1,Z2,A2,W2,X,Y)
W1 = W1 - alpha*dW1
 b1 = b1 - alpha*db1
 W2 = W2 - alpha*dW2
 b2 = b2 - alpha*db2
```

6. Calculated Loss using Cross-entropy function and evaluated Accuracy.

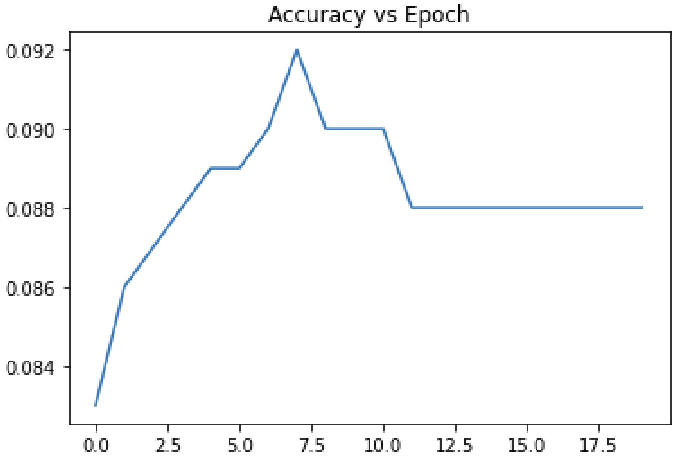
Colab File Link : [**https://colab.research.google.com/drive/1q9uy_wz383vPncL7as2t99sdoPdzK6BR?usp=share_link**](https://colab.research.google.com/drive/1q9uy_wz383vPncL7as2t99sdoPdzK6BR?usp=share_link)

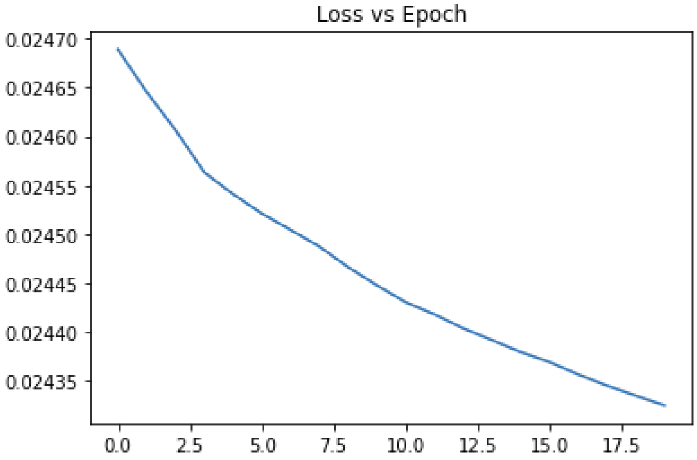
Given Configuration Sets

1. N = size of the input; H = 10; η = 0.01; E = 20

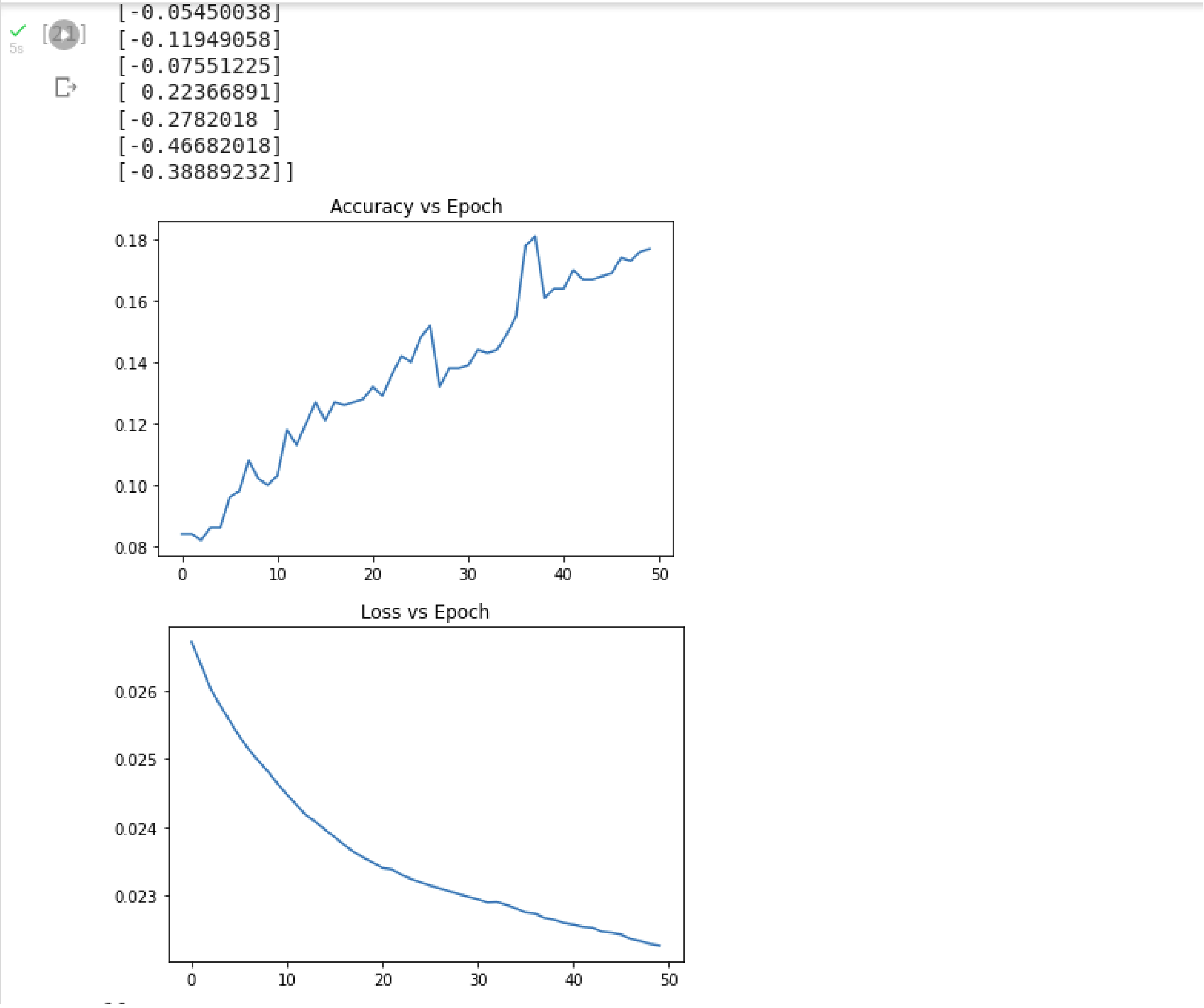


```
[[-0.18590131]
 [ 0.38712007]
 [-0.49395979]
 [ 0.29209921]
 [-0.16447734]
 [-0.4332247 ]
 [ 0.31225488]
 [ 0.19143959]
 [-0.21154381]
 [-0.09404662]]
```



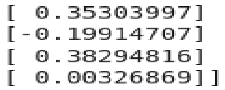


2. N = size of the input; H = 10; η= 0.1; E = 50

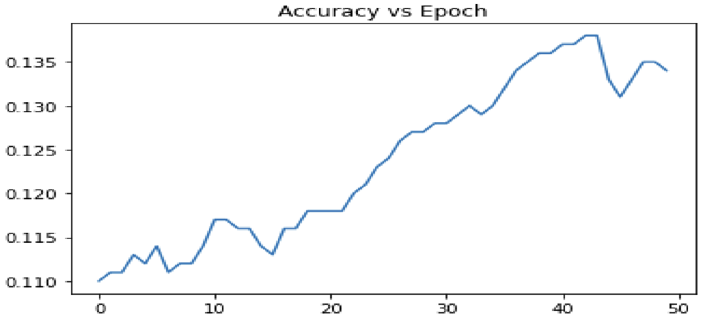


```
 [-0.05450038]
 [-0.11949058]
 [-0.07551225]
 [ 0.22366891]
 [-0.2782018 ]
 [-0.46682018]
 [-0.38889232]]
```

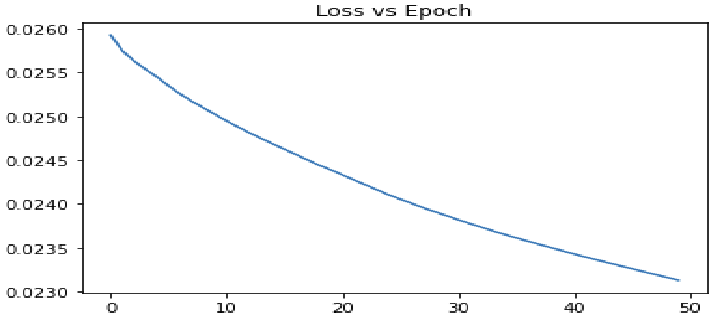
3. N = size of the input; H = 20; η = 0.02; E = 50



```
[ 0.35303997]
[-0.19914707]
[ 0.38294816]
[ 0.00326869]]
```

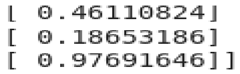


Accuracy vs Epoch

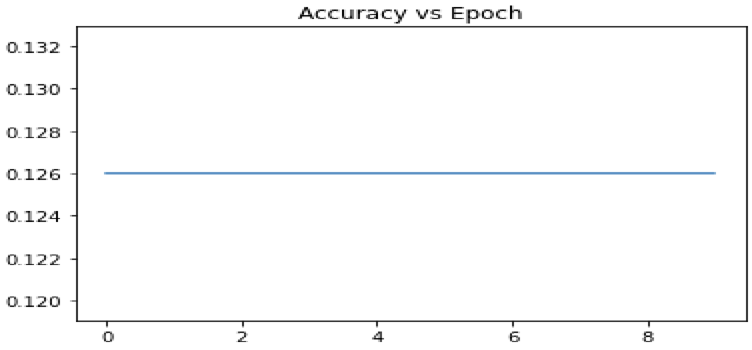


Loss vs Epoch

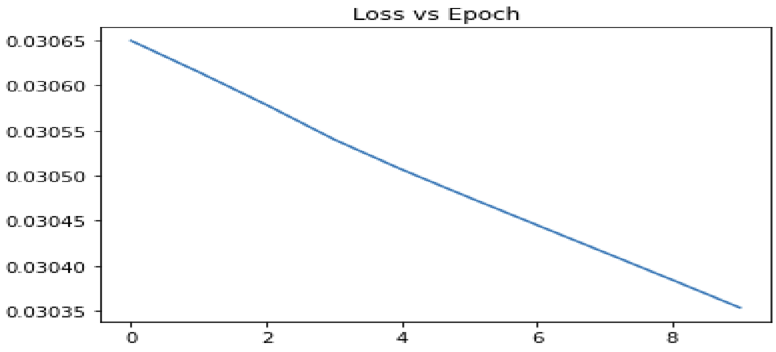
4. N = size of the input; H = 2; η = 0.01; E = 10



```
[ 0.46110824]
[ 0.18653186]
[ 0.97691646]]
```

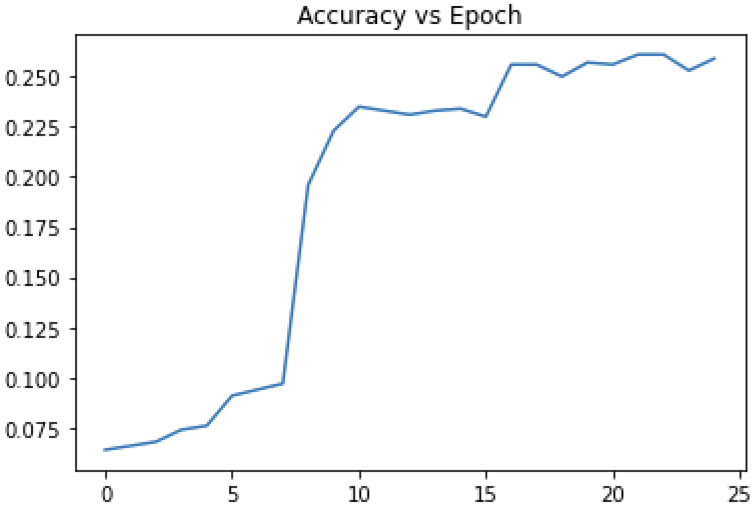


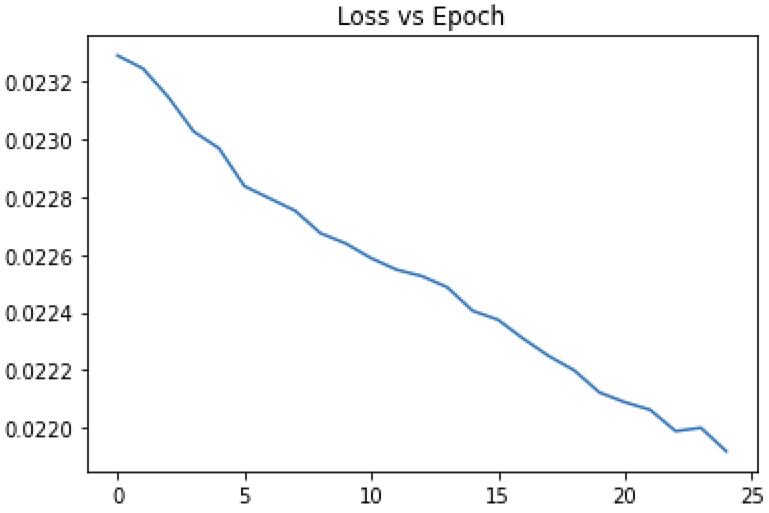
Accuracy vs Epoch



Loss vs Epoch

5. N = size of the input; H = 4; η= 0.5; E = 25





References:
https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/
https://www.youtube.com/watch?v=w8yWXqWQYmU
https://gist.github.com/Atlas7/22372a4f6b0846cfc3797766d7b529e8