CSCI 635 Homework 2

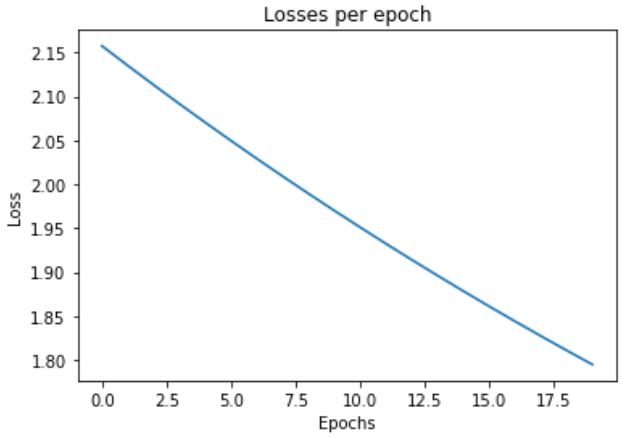
Name: Hitesh Vaidya (hv8322@rit.edu)

Q1a) Tackling the XOR problem

- The hyperparameters, Weights and Bias are initialized based on a normal distribution. This is done using numpy syntax,

  self.W = np.random.normal(scale=1/D**0.5, size=(D,k))
  self.b = np.random.normal(scale=1/k**0.5, size=k)

  where, D = number of features
  k = number of classes
- The value of lambda for regularization in loss function is 0.9 and 0.8 in gradient function for calculating dj/dw and dj/db.
- The learning rate used is 0.01 while the epsilon value is 10e-3 in secant approximation.
- These values were selected with trial and error method. I tried bunch of values and found that these valued worked relatively better.
- No matter which values are selected as hyperparameters, the model does not correctly predict all labels for XOR dataset.
- Loss per epoch

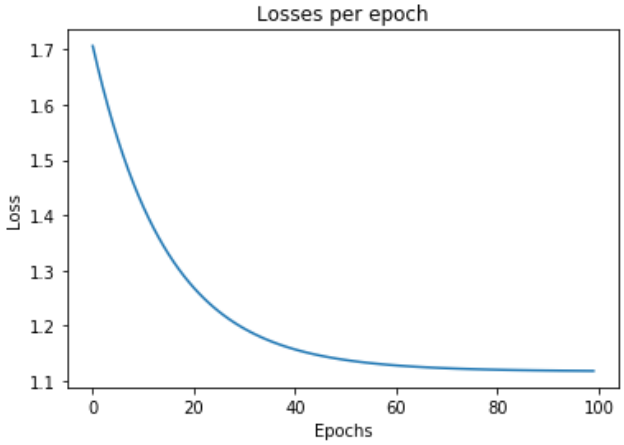


- `Final accuracy = 50.0`
- Values of hyperparameters to obtain good results:

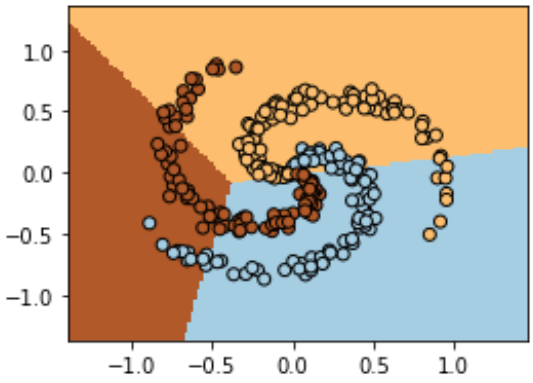
```
W: [[-1.58480352 -0.17151065]
 [-0.44616976 -0.29651584]]
b: [-0.397066    0.02003822]
```

Q1b) Softmax Regression and Spiral Problem

- Loss per epoch:



- Decision Boundary:



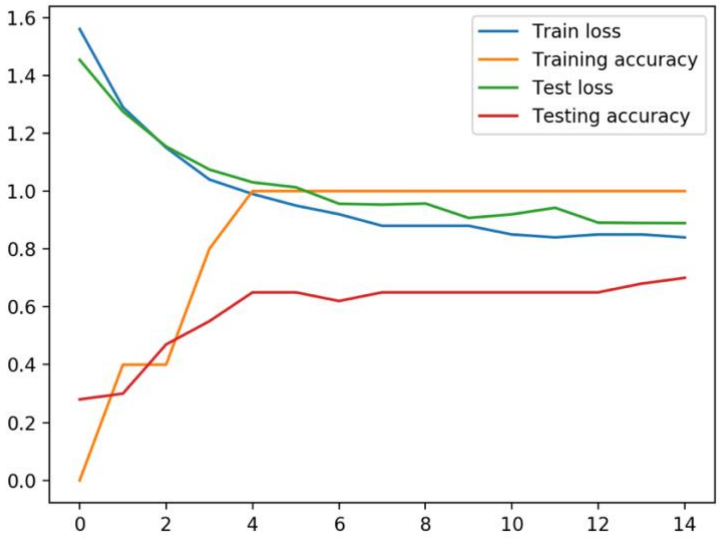
- `accuracy = 52.33333333333333`
- The model should give one of the best values of accuracy for the following hyperparamerters

```
W: [[ 1.13395259 -0.18132901 -0.88227882]
    [ 0.51842919 -0.11516643 -0.43056516]]
b: [ 0.08036394  0.07812435 -0.17722656]
```

- The model gives different results for different values of hyper parameters. After a bunch of trials , the above values gave better results.

Q1c) The IRIS Maximum Entropy Model

- Loss and Accuracy values



- Hyperparameters that gives one of the best results

```
W: [[-0.00311582  0.27692917  0.38580981]
 [ 0.7022309  -0.06115604 -0.0741226 ]
 [ 0.106968    1.24999431  0.03054709]
 [ 0.22779927 -0.52283921  0.09578072]]
b: [-1.42871464 -0.63090459  0.43391054]
```

- `Final Test loss: 0.8892035780429228`
- `Final Test accuracy: 0.7`
- As we go on running this model for more number of epochs, the values in graph go on getting saturated i.e. they hardly show any significant change in values. They flatten to almost one value in further epochs.

Q2) Learning a Naïve Bayes Classifier

- Part a) Model Parameters:

Here, for every attribute in given dataset, I store a confusion matrix as follows:

	Is_Spam: True	Is_Spam: False
Attribute: True	TT	TF
Attribute: False	FT	FF

If the attribute contains numerical values,

Is_Spam: True	mean	variance
Is_Spam: False	Mean	variance

```
Part a) Parameters:
True and False probability: 0.172 0.828
For each attribute:
in html :
 [[0.75581395 0.58695652]
 [0.24418605 0.41304348]]
 has emoji :
 [[0.19767442 0.147343  ]
 [0.80232558 0.852657  ]]
 sent to list :
 [[0.06976744 0.3115942 ]
 [0.93023256 0.6884058 ]]
 from .com :
 [[0.74418605 0.27536232]
 [0.25581395 0.72463768]]
 has my name :
 [[0.34883721 0.60144928]
 [0.65116279 0.39855072]]
 has sig :
 [[0.6627907 0.3236715]
 [0.3372093 0.6763285]]
 # sentences :
 [[3.97674419 3.7203894 ]
 [6.19082126 6.40078532]]
 # words :
 [[ 68.8372093   79.34559221]
 [ 70.7705314  912.76618474]]
```

- Part b) values:

```
Test Accuracy = 0.43
Loss = 0.5700000000000001
```

- Part c)
  - I declared an array of random choices of attributes. All the model parameters were recalculated based on these attribute values. The accuracy result obtained on test dataset for each of the attribute combination were recorded and the one giving maximum accuracy was selected.
  - The best accuracy and its corresponding combination of attributes is as follows:

```
Maximum accuracy, 0.81 for attributes: [0, 1, 2, 8]
```