Report for exercise 3

Part2 :

Pseudo code for part 2 as required :

1. Init W = 0 (W is vector of zeros), T = number of iteration of the Algorithm
2. $S = \{X_i, Y_i\}_{i=1}^{m}$ (data)
3. For t= 1 to T

3.1 random = get random number between 1 to size X

3.2 $prod =< W_{t-1}, X_{\_random} >$

3.3 maxFetureValue = argmax(prod)

3.4 SG = empty matrix with data size

3.5 $update\ SG: SG(Y_{rand}) - X_{rand}$

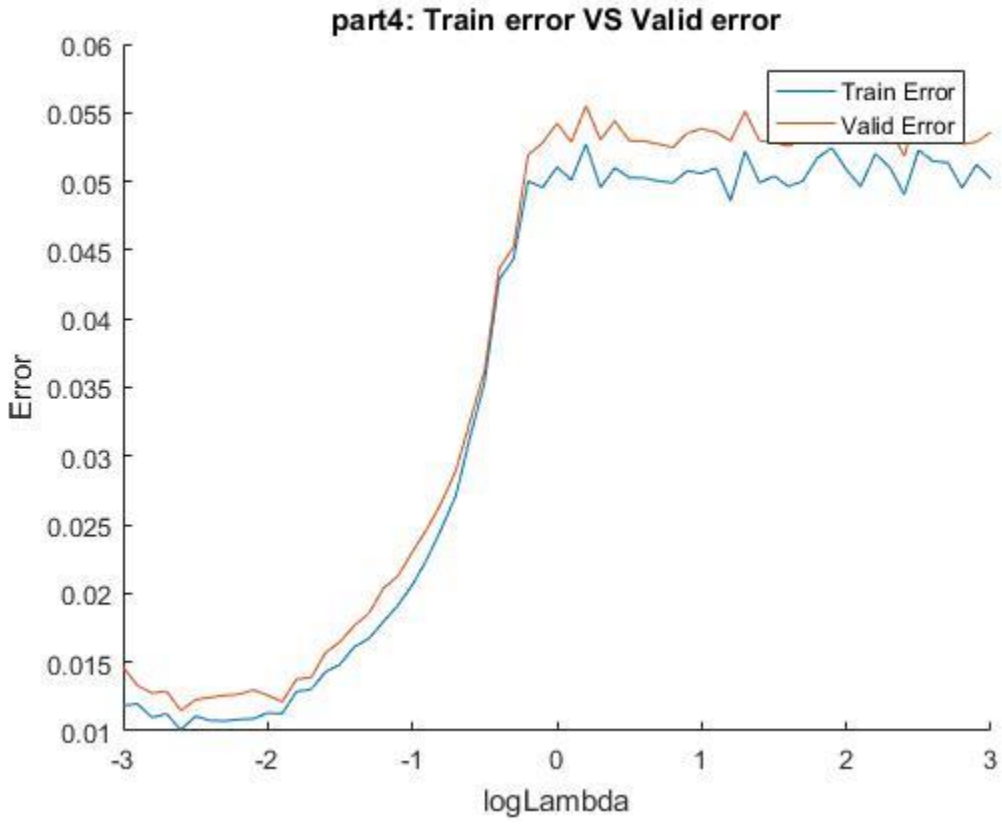
3.6 $update\ SG: SG(argmaxFeture) = SG(argmaxFeture) + X_{rand}$

3.7 $W_t = W_t + T^{-1} * W_{t-1}$

3.8 $W_{t-1} = (1 - 2 * t^{-1}) * W_{t-1} - (\lambda * t)^{-1} * \mathrm{SG}$

Part4:

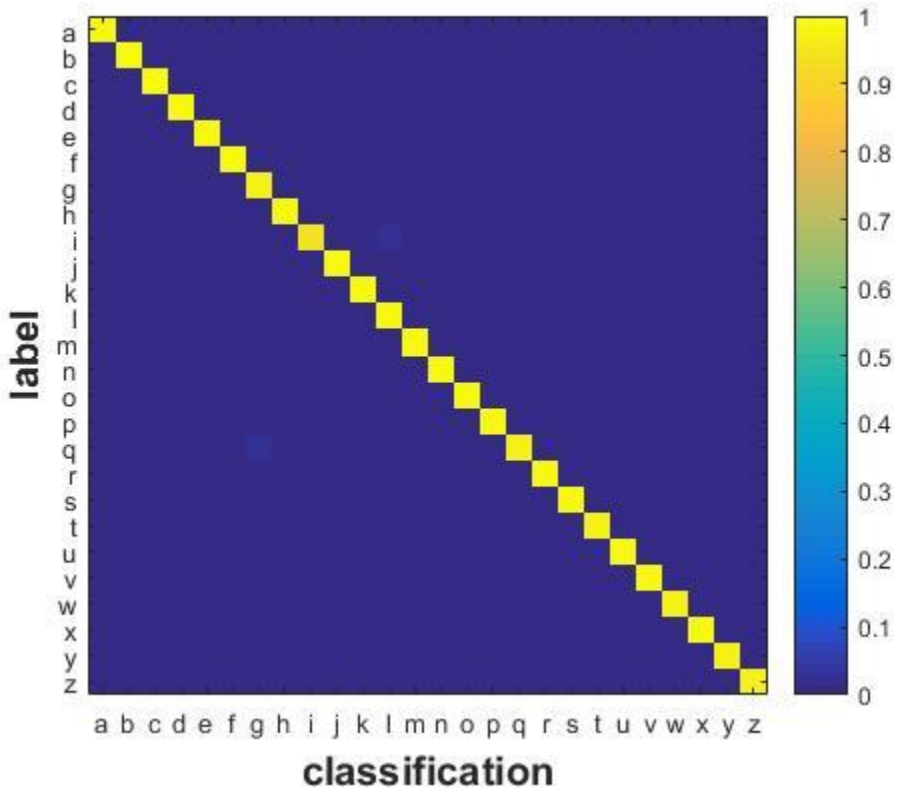
Graph for part 4:



I choose range of lambda to be between $[10^{-3}, 10^{3}]$ and the ideal lambda that led to minimize the validation error was $10^{-2.2}$ and the minimum validation error for this lambda is: 0.0113872 , and the training error was 0.0106962.

The algorithm run for T = 75,000 iteration.

I use the ideal lambda to plot the `confusion matrix function attached to the exercise and get this results:`



Notice that the classification is so far correct ,and that the prediction is Ok , but some wrong between the letter (i) and the letter (l) , and it is not so problematic because this two letters are similar to each other.

Part6:

Number of errors for the different strategies:

1)strategy1: **276** strategy1 error: **0.1355**

2)strategy2: **114** strategy2 error: **0.0570**

3)strategy3: **251** strategy3 error: **0.1240**

4)strategy4: **91** strategy4 error:**0.0455**

The letter_henge_SGD run by the ideal lambda from part 4 and also for T = 75,000 iterations.

As we see that Strategy 1 less better than strategy2 according to the strategies errors, but notice that there are words that classified better by strategy1.

In strategy 2 we have classified by W_wrd (normal distribution of letters) and it is more effective than classifier by ltr(letter).

Examles for words that classified in strategy 1 better than strategy 2: grouchy, rifleproof

Examles for words that classified in strategy 2 better than strategy 1: carbonado, irreferable

As we see that Strategy 3 less better than strategy4 according to the strategies errors, it makes sense to get this result because in strategy4 the words is trained but in strategy3 letter is trained, so training words surely is more effective .

But notice that there are words that classified better by strategy3.

Examles for words that classified in strategy 3 better than strategy 4: opinionativeness, supermedial.

Examles for words that classified in strategy 4 better than strategy 3: fackeltanz, picumnus.

Also comparison between strategy1 and 4 shows that strategy4 is better than strategy1 , because training words are more effective than training letters.

But notice that there are words that classified better by strategy1.

Examles for words that classified in strategy 1 better than strategy 4: germanness, sorrowfully.

Examles for words that classified in strategy 4 better than strategy 1: agyieus, despecificate.