Implemented Stochastic Gradient Descent Method with $L2$ Regularization

pghw2_stochastic.py

1) Run 1

Initializations:

Stopping Criteria: 1000 iterations, **Batch Size:** 100, $\boldsymbol{\eta} = 0.0055$, $\boldsymbol{\lambda} = 0.02$,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 & \mathbf{W}_5 \\ W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & W_{5,0} \end{bmatrix}$$

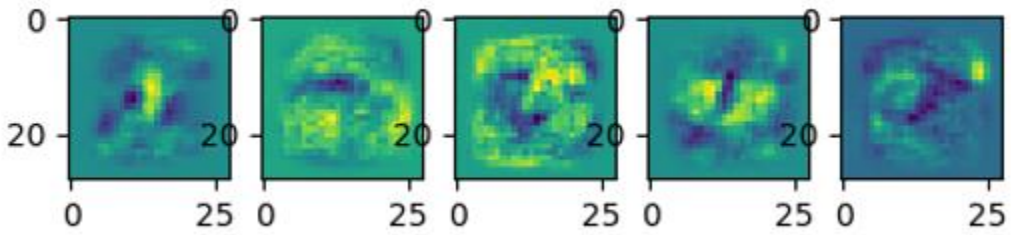
$\mathbf{W}^{t=0} = 0.1 * \text{tf.ones([K,imsize])}$ → **W** is initialized to all 0.1

Results:

(Accuracy Percentage, Wmatrix, Final Loss Value):

```
Final Loss Value:
20.2833
Wmatrix:
[[ 0.10300749  0.10961034  0.10721803  0.10465252  0.10378116]
 [ 0.10491639  0.10787065  0.11577661  0.10943156  0.0902743 ]
 [ 0.1020198   0.10927999  0.10653555  0.10794154  0.10249297]
 ...,
 [ 0.10565017  0.10664303  0.10509429  0.10569544  0.10518645]
 [ 0.10568698  0.10575463  0.10556185  0.10567396  0.10559209]
 [ 0.39836019 -0.14796783 -0.79332614 -0.15865548  1.22985995]]
Accuracy Percentage on Test Data Set:
95.0020074844
```

W also given in textfile as 'multiclass_parameters.txt' and 'multiclass_parameters_run1.txt'
W Visualized: (Color bar not shown because it was messing up the last image's resolution)



2) Run 2 (changed batch size)

Initializations:

Stopping Criteria: 1000 iterations, **Batch Size:** 50, $\boldsymbol{\eta} = 0.0055$, $\boldsymbol{\lambda} = 0.02$,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 & \mathbf{W}_5 \\ W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & W_{5,0} \end{bmatrix}$$

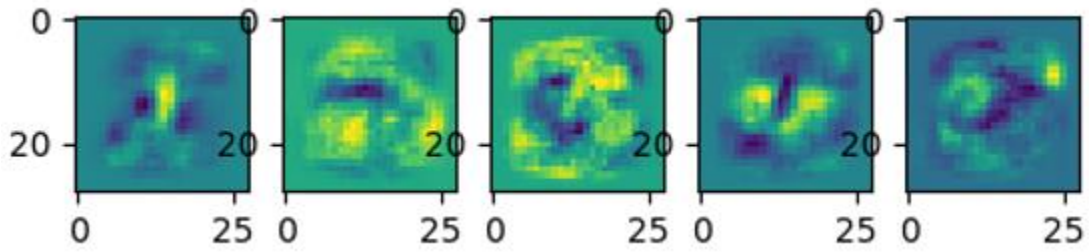
$\mathbf{W}^{t=0} = 0.1 * \text{tf.ones([K,imsize])}$ → **W** is initialized to all 0.1

Results:

(Accuracy Percentage, Wmatrix, Final Loss Value):

```
Final Loss Value:
15.4895
Wmatrix:
[[ 0.10330544  0.10902219  0.1058495   0.10479632  0.10529598]
 [ 0.10552111  0.10605799  0.10835398  0.10911086  0.09922549]
 [ 0.10249276  0.10778532  0.10947704  0.107303    0.10121168]
 ...,
 [ 0.10564003  0.10618648  0.10528947  0.10571675  0.10543676]
 [ 0.10569675  0.10575043  0.10559119  0.10565658  0.10557444]
 [ 0.36762959 -0.05180791 -0.56223512 -0.05322869  0.82791114]]
Accuracy Percentage on Test Data Set:
94.8213577271
```

W also given in textfile as 'multiclass_parameters.txt' and 'multiclass_parameters_run1.txt'
W Visualized: (Color bar not shown because it was messing up the last image's resolution)



3) Run 3 (changed learning rate)

Initializations:

Stopping Criteria: 1000 iterations, **Batch Size:** 100, $\boldsymbol{\eta} = 0.0001$, $\boldsymbol{\lambda} = 0.02$,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 & \mathbf{W}_5 \\ W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & W_{5,0} \end{bmatrix}$$

$\mathbf{W}^{t=0} = 0.1 * \text{tf.ones([K,imsize])}$ → **W** is initialized to all 0.1

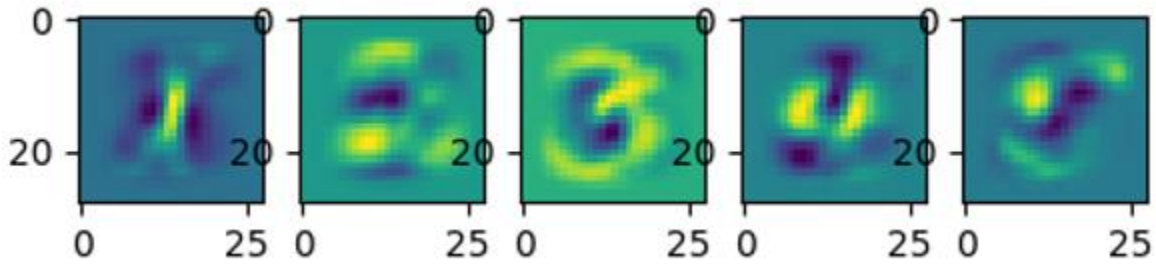
Results:

(Accuracy Percentage, Wmatrix, Final Loss Value):

```
Final Loss Value:
23.1322
Wmatrix:
[[ 0.09953406  0.100474    0.1008774   0.09994019  0.09967472]
 [ 0.09943537  0.1007536   0.10075753  0.10016283  0.09939089]
 [ 0.09932712  0.10089217  0.10077275  0.09998702  0.09952109]
 ...,
 [ 0.10009337  0.10012764  0.10007676  0.10009804  0.10008933]
 [ 0.10009737  0.10010888  0.10009015  0.10009531  0.10009322]
 [ 0.19060515  0.03731311  0.01205458  0.10432832  0.15619907]]
Accuracy Percentage on Test Data Set:
93.2557225227
```

W also given in textfile as 'multiclass_parameters.txt' and 'multiclass_parameters_run1.txt'

W Visualized: (Color bar not shown because it was messing up the last image's resolution)



4) Run 4 (Changed Number of Iterations)

Initializations:

Stopping Criteria: 100 iterations, **Batch Size:** 100, $\boldsymbol{\eta} = 0.0055$, $\boldsymbol{\lambda} = 0.02$,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 & \mathbf{W}_5 \\ W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & W_{5,0} \end{bmatrix}$$

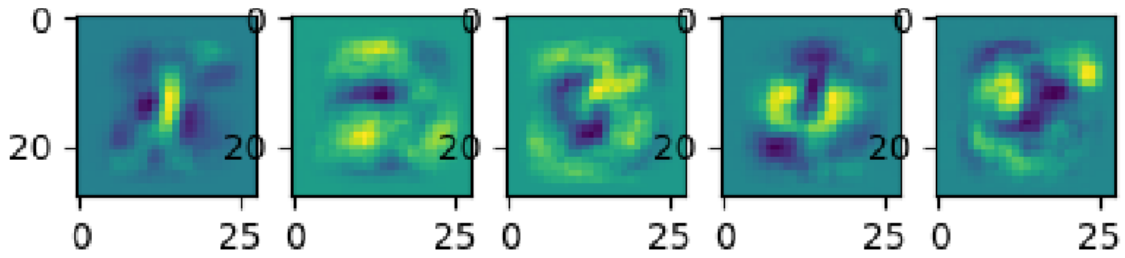
$\mathbf{W}^{t=0} = 0.1 * \text{tf.ones([K,imsize])}$ → **W** is initialized to all 0.1

Results:

(Accuracy Percentage, Wmatrix, Final Loss Value):

```
Final Loss Value:
34.1656
Wmatrix:
[[ 0.09920435  0.10154268  0.10277411  0.10016086  0.09907546]
 [ 0.09986544  0.10235468  0.1004577   0.10144468  0.09863498]
 [ 0.09879114  0.10348342  0.10311644  0.10021145  0.09715508]
 ...,
 [ 0.10055079  0.10056219  0.1005098   0.10065649  0.10047825]
 [ 0.10056786  0.10053948  0.10053393  0.10059662  0.10051961]
 [ 0.21072879 -0.0059609  -0.16023828  0.08018405  0.37804395]]
Accuracy Percentage on Test Data Set:
92.9947793484
```

W also given in textfile as 'multiclass_parameters.txt' and 'multiclass_parameters_run1.txt'
W Visualized: (Color bar not shown because it was messing up the last image's resolution)



5) Run 5 (Changed Regularization Parameter)

Initializations:

Stopping Criteria: 1000 iterations, **Batch Size:** 100, $\boldsymbol{\eta} = 0.0055$, $\boldsymbol{\lambda} = 0.2$,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 & \mathbf{W}_5 \\ W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & W_{5,0} \end{bmatrix}$$

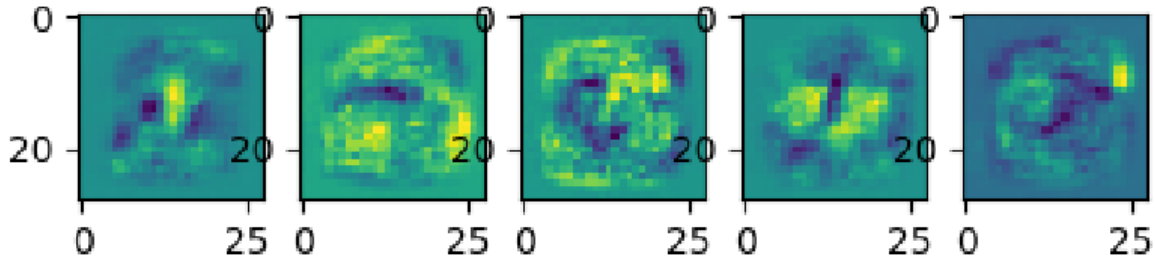
$\mathbf{W}^{t=0} = 0.1 * \text{tf.ones([K,imsize])}$ → **W** is initialized to all 0.1

Results:

(Accuracy Percentage, Wmatrix, Final Loss Value):

```
Final Loss Value:
26.1707
Wmatrix:
[[ 0.16872875  0.17553411  0.17708188  0.17266831  0.17248283]
 [ 0.17355974  0.17341419  0.18149704  0.17956778  0.1584571 ]
 [ 0.16755609  0.17562695  0.17844187  0.17981316  0.16505736]
 ...,
 [ 0.17332307  0.17416981  0.17243893  0.17350791  0.17305562]
 [ 0.17352779  0.17332351  0.17311385  0.17335697  0.17317376]
 [ 0.54348582 -0.13756201 -0.9529503  -0.08820105  1.50172532]]
Accuracy Percentage on Test Data Set:
94.7812139988
```

W also given in textfile as 'multiclass_parameters.txt' and 'multiclass_parameters_run1.txt'
W Visualized: (Color bar not shown because it was messing up the last image's resolution)



Comments:

We see that decreasing the batch size to 50 from a 100 reduced the accuracy of the learned parameters when we applied them for classification on the test data. But an increased batch size makes the algorithm slower to run. Also, by experimenting with different batch sizes, I have found that we need to find a balanced batch size because if we use too many images, the algorithm will be slow and if we use too little, we lose out on accuracy.

Next, we saw that an increased learning rate converges faster. But also, by experimenting with other learning rates, I saw that if it is too large, then the algorithm diverges and I got NaN (or infinity) for W values. Therefore, finding a balance here is also important.

Next we saw that increasing the regularization parameter didn't have much effect on the accuracy and both values gave good results for the test data. This may be due to the fact that our data set is really large and there is not much change to overfit the training data. However, a higher regularization parameter corresponds to less over-fitting.

Next, we saw that decreasing the number of iterations or stopping too early will give a very low accuracy. This means that the algorithm hasn't converged yet. To avoid this, I also have the loss term in my code. The change in the loss value can be used as a stopping criteria as well in a while loop. We can monitor the loss between 5 to 10 successive iterations and if the change is lower than some preset threshold, we can stop the algorithm.

I also Implemented Fixed-Step Gradient Descent (which was required for the 4000 level class only) It is given in pghw2.py