

Problem 1 – Newtons Method

According to the lecture slides the Hessian is a constant: $H = 1/n XX^T$ therefore when updating the weight vectors (with w being the initial weight vector) the optimal solution w^* is:

$$\begin{aligned} w^* &= w - H^{-1} \nabla_w F_{mse} = w - \left(\frac{1}{n} XX^T \right)^{-1} \nabla_w F_{mse} = w - n(XX^T)^{-1} \nabla_w F_{mse} \\ &= w - n(XX^T)^{-1} \frac{1}{n} X(X^T w - y) = w - (XX^T)^{-1} XX^T w + (XX^T)^{-1} Xy \\ &= w - Iw + (XX^T)^{-1} Xy = (XX^T)^{-1} Xy = w^* \end{aligned}$$

Problem 2 – Derivation of SoftMax Regression Gradient Updates

a) For $l = k$

$$\begin{aligned} \nabla_{w^{(i)}} \hat{y}_k^{(i)} &= \nabla_{w^{(i)}} \left[\frac{\exp X^{(i)T} w^{(i)}}{\sum_{k'=1}^c \exp X^{(i)T} w^{(k')}} \right] \\ &= X^{(i)} \frac{\exp X^{(i)T} w^{(i)}}{\sum_{k'=1}^c \exp X^{(i)T} w^{(k')}} - X^{(i)} \frac{\exp X^{(i)T} w^{(i)}}{\left(\sum_{k'=1}^c \exp X^{(i)T} w^{(k')} \right)^2} \exp(X^{(i)T} w^{(i)}) \\ &= X^{(i)} \left[\frac{\exp X^{(i)T} w^{(i)}}{\sum_{k'=1}^c \exp X^{(i)T} w^{(k')}} - \frac{(\exp X^{(i)T} w^{(i)})^2}{\left(\sum_{k'=1}^c \exp X^{(i)T} w^{(k')} \right)^2} \right] \\ &= X^{(i)} \left[\hat{y}_1^{(i)} - (\hat{y}_1^{(i)})^2 \right] \\ &= X^{(i)} \hat{y}_1^{(i)} (1 - \hat{y}_1^{(i)}) \end{aligned}$$

b) For $l \neq k$

$$\begin{aligned} \nabla_{w^{(i)}} \hat{y}_k^{(i)} &= \nabla_{w^{(i)}} \left[\frac{\exp X^{(i)T} w^{(k)}}{\sum_{k'=1}^c \exp X^{(i)T} w^{(k')}} \right] \\ &= -X^{(i)} \frac{\exp X^{(i)T} w^{(k)}}{\left(\sum_{k'=1}^c \exp X^{(i)T} w^{(k')} \right)^2} \exp X^{(i)T} w^{(i)} \\ &= -X^{(i)} \frac{(\exp X^{(i)T} w^{(k)}) (\exp X^{(i)T} w^{(i)})}{\left(\sum_{k'=1}^c \exp X^{(i)T} w^{(k')} \right)^2} \\ &= -X^{(i)} \hat{y}_k^{(i)} \hat{y}_1^{(i)} \end{aligned}$$

$$\begin{aligned}
c) \quad \nabla_w^{(i)} \mathcal{L}_{CE}(W, b) &= - \sum_{i=1}^m \sum_{k=1}^n y_k^{(i)} \nabla_w^{(i)} \log \hat{y}_k^{(i)} \\
&= - \sum_{i=1}^m x^{(i)} \left[\frac{y_1^{(i)} \hat{y}_1^{(i)} (1 - \hat{y}_1^{(i)})}{\hat{y}_1^{(i)}} - \sum_{k \neq 1} \frac{y_k^{(i)} \hat{y}_k^{(i)} \hat{y}_1^{(i)}}{\hat{y}_k^{(i)}} \right] \\
&= - \sum_{i=1}^m x^{(i)} \left[y_1^{(i)} (1 - \hat{y}_1^{(i)}) - \sum_{k \neq 1} y_k^{(i)} \hat{y}_1^{(i)} \right] \\
&= - \sum_{i=1}^m x^{(i)} \left[y_1^{(i)} (1 - \hat{y}_1^{(i)}) + y_1^{(i)} \hat{y}_1^{(i)} - \sum_k y_k^{(i)} \hat{y}_1^{(i)} \right] \\
&= - \sum_{i=1}^m x^{(i)} \left[y_1^{(i)} - y_1^{(i)} \hat{y}_1^{(i)} + y_1^{(i)} \hat{y}_1^{(i)} - \hat{y}_1^{(i)} \sum_k y_k^{(i)} \right] \\
&= - \sum_{i=1}^m x^{(i)} \left[y_1^{(i)} - \hat{y}_1^{(i)} \right]
\end{aligned}$$

Combine into a single vector with the knowledge that b is derived the same way as ∇_w except with no $x^{(i)}$ term

$$\nabla_b \mathcal{L}_{CE}(W, b) = - \frac{1}{n} \sum_{i=1}^n [y^{(i)} - \hat{y}^{(i)}]$$

Problem 3 – Derivation of Cross-Entropy as Negative Log-Likelihood

Question ③ Given $P(y|x, w, b) =$

$$= \frac{e}{\prod_{k=1}^K y_k^{(i)}}$$

For entire Dataset

$$-\log P(D|w, b) = \sum_{i=1}^n \sum_{k=1}^K \frac{e}{y_k^{(i)}}$$

$$= -\log \left(\prod_{i=1}^n \prod_{k=1}^K \frac{e}{y_k^{(i)}} \right) \quad \left[\because \log(A \cdot B) = \log A + \log B \right]$$

~~$\therefore \log(A \cdot B) = \log A + \log B$~~

$$= \sum_{i=1}^n \sum_{k=1}^K \log \left(\frac{e}{y_k^{(i)}} \right)$$

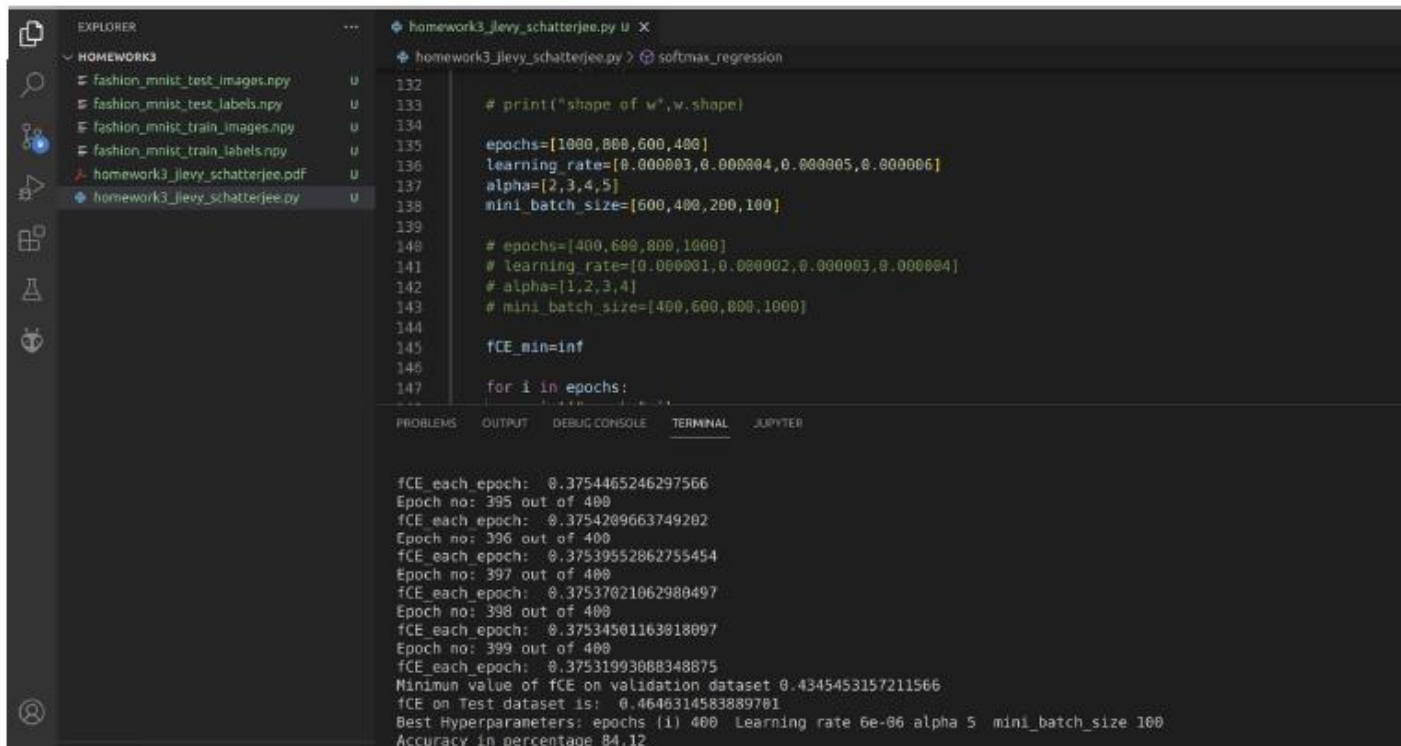
$$= \sum_{i=1}^n \sum_{k=1}^K y_k^{(i)} \log y_k^{(i)}$$

Problem 4 – Implementation of SoftMax Regression

Question 4)

Hyperparameters:

```
epochs=[1000,800,600,400]
learning_rate=[0.000003,0.000004,0.000005,0.000006]
alpha=[2,3,4,5]
mini_batch_size=[600,400,200,100]
```



```
homework3_levy_schatterjee.py X
homework3_levy_schatterjee.py > softmax_regression

# print("shape of w",w.shape)

epochs=[1000,800,600,400]
learning_rate=[0.000003,0.000004,0.000005,0.000006]
alpha=[2,3,4,5]
mini_batch_size=[600,400,200,100]

# epochs=[400,600,800,1000]
# learning_rate=[0.000001,0.000002,0.000003,0.000004]
# alpha=[1,2,3,4]
# mini_batch_size=[400,600,800,1000]

fCE_min=inf

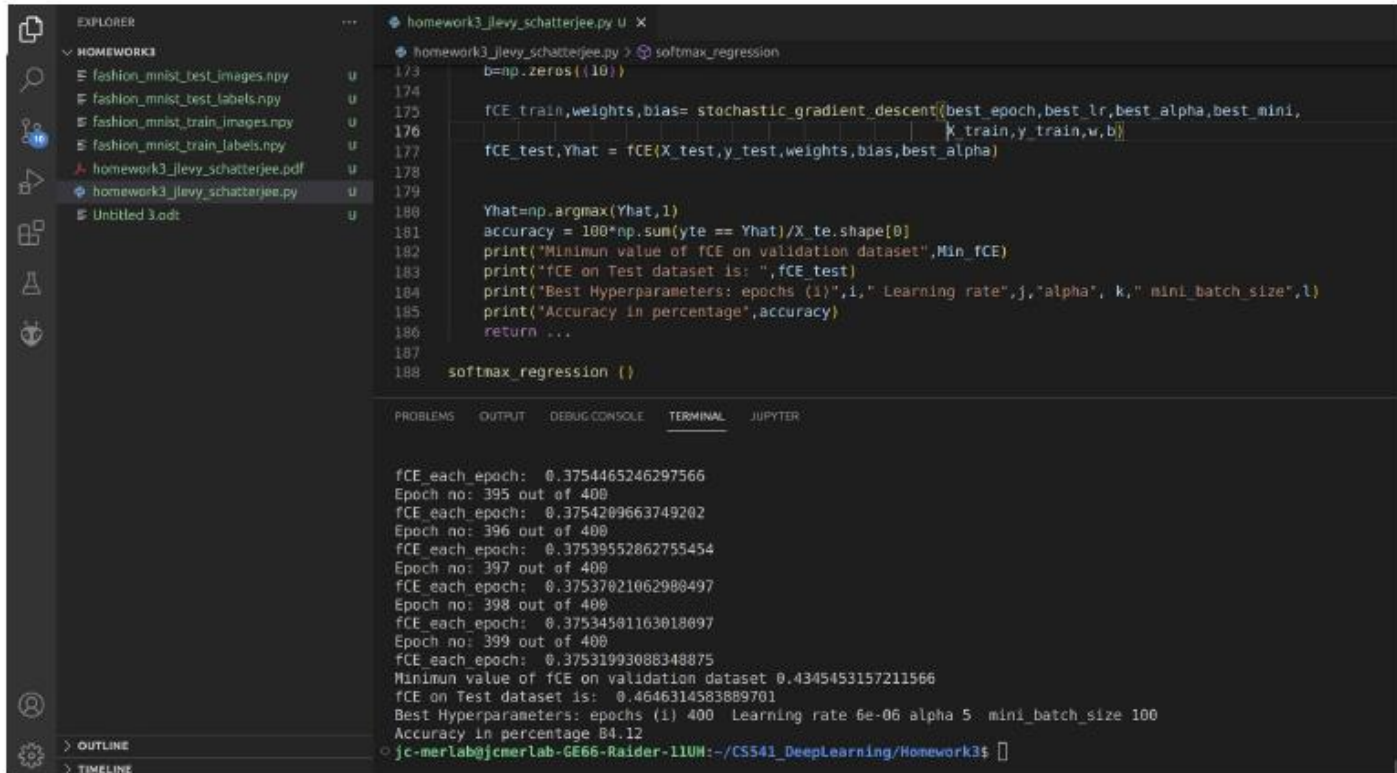
for i in epochs:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
fCE each epoch: 0.3754465246297566
Epoch no: 395 out of 400
fCE each epoch: 0.3754209663749202
Epoch no: 396 out of 400
fCE each epoch: 0.37539552862755454
Epoch no: 397 out of 400
fCE each epoch: 0.37537021062980497
Epoch no: 398 out of 400
fCE each epoch: 0.37534501163018097
Epoch no: 399 out of 400
fCE each epoch: 0.37531993088348875
Minimum value of fCE on validation dataset 0.4345453157211566
fCE on Test dataset is: 0.4646314583889701
Best Hyperparameters: epochs (1) 400 Learning rate 6e-06 alpha 5 mini_batch_size 100
Accuracy in percentage 84.12
```


Results:

```
Minimun value of fCE on validation dataset 0.4345453157211566
fCE on Test dataset is: 0.4646314583889701
Best Hyperparameters: epochs (i) 400 Learning rate 6e-06 alpha 5 mini_batch_size 100
Accuracy in percentage 84.12
```



The screenshot shows a Jupyter Notebook environment. On the left is the Explorer sidebar with a file tree containing files like `fashion_mnist_test_images.npy`, `fashion_mnist_test_labels.npy`, `fashion_mnist_train_images.npy`, `fashion_mnist_train_labels.npy`, `homework3_levy_schatterjee.pdf`, `homework3_levy_schatterjee.py`, and `Untitled3.odt`. The main area displays the code for `homework3_levy_schatterjee.py`, which includes a `softmax_regression` function. The code uses `stochastic_gradient_descent` to train a model and prints the minimum value of fCE on the validation dataset, the fCE on the test dataset, the best hyperparameters, and the accuracy in percentage. The bottom panel shows the terminal output, which matches the results displayed above the screenshot.

```
173 b=np.zeros((10))
174
175 fCE_train,weights,bias= stochastic_gradient_descent(best_epoch,best_lr,best_alpha,best_mini,
176                                                    k_train,y_train,w,b)
177 fCE_test,Yhat = fCE(X_test,y_test,weights,bias,best_alpha)
178
179 Yhat=np.argmax(Yhat,1)
180 accuracy = 100*np.sum(yte == Yhat)/X_te.shape[0]
181 print("Minimun value of fCE on validation dataset",Min_fCE)
182 print("fCE on Test dataset is: ",fCE_test)
183 print("Best Hyperparameters: epochs (i)",i," Learning rate",j,"alpha", k," mini_batch_size",l)
184 print("Accuracy in percentage",accuracy)
185 return ...
186
187
188 softmax_regression ()
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```
fCE_each_epoch: 0.3754465246297566
Epoch no: 395 out of 400
fCE_each_epoch: 0.3754209663749202
Epoch no: 396 out of 400
fCE_each_epoch: 0.37539552862755454
Epoch no: 397 out of 400
fCE_each_epoch: 0.37537021062908497
Epoch no: 398 out of 400
fCE_each_epoch: 0.37534501163018097
Epoch no: 399 out of 400
fCE_each_epoch: 0.37531993088348875
Minimun value of fCE on validation dataset 0.4345453157211566
fCE on Test dataset is: 0.4646314583889701
Best Hyperparameters: epochs (i) 400 Learning rate 6e-06 alpha 5 mini_batch_size 100
Accuracy in percentage 84.12
jc-merlab@jcmerlab-GE66-Raider-11UM:~/CS541_DeepLearning/Homework3$
```

fCE on Test data: 0.4345
Optimized Hyperparameter values -
Epochs: 400
Learning rate: 0.000006
alpha: 5
Minibatch size: 100
Final Accuracy : 84.12