CS 541-Deep Learning Week3 Homework

Abhiroop Ajith [*], Youness Bani[†], Chinmaya Khamesra[‡]
Worcester Polytechnic Institute (WPI)
Dept. of Robotics Engineering
Worcester MA, USA
Email: [*]aajith@wpi.edu, [†]ybani@wpi.edu, [‡]ckhamesra@wpi.edu

***Index Terms*—Softmax Regression,Newtons Method,Deep-Learning**

I. Introduction

This document solves the problem sets given in the Week 3 Assignment for Dr. Jacob Whitehall's CS541 Deep Learning course for Spring 2022. The assignment focuses on using python and its libraries like numpy and scipy for the implementation of softmax regression from scratch without the use of off-the shelf software.It also deals with the mathematical derivations of Newton's Method,Derivation of Cross-Entropy as Negative Log-Likelihood and Softmax Weight Updates.

II. Methodology

A. Problem 1

The first problem deals with Newtons Method and to prove that it will converge to the optimal solution in 1 iteration no matter what the starting point w(0) of the search.
As given in the question, using [1] we get the equation of x^* as:

$$x^* = x^{(0)} - H(f)(x^{(0)})^{-1} \bigtriangledown_x f(x^0) \tag{1}$$

Therefore converting equation (1) to find the solution for $w^{(1)}$ we get:

$$w^{(1)} = w^{(0)} - H^{-1} \bigtriangledown_w f_{mse} \tag{2}$$

We know that the Hessian, that is a constant is equal to:

$$H = \frac{1}{n} X X^T$$

Substituting the value of H in equation (2):

$$w^{(1)} = w^{(0)} - (\frac{1}{n} X X^T)^{-1} \bigtriangledown_w f_{mse}$$

The cost function $\bigtriangledown_w f_{mse}$ is:

$$\bigtriangledown_w f_{mse} = \frac{1}{n} X(X^T w - y)$$

Substituing the value of the cost function we get the equation of $w^{(1)}$ in the terms of:

$$w^{(1)} = w^{(0)} - (\frac{1}{n} X X^T)^{-1} \frac{1}{n} X(X^T w - y)$$

$$w^{(1)} = w^{(0)} - n(X X^T)^{-1} \frac{1}{n} X(X^T w - y)$$

$$w^{(1)} = w^{(0)} - (X X^T)^{-1} X(X^T w - y)$$

$$w^{(1)} = w^{(0)} - (X X^T)^{-1} (X X^T w - Xy)$$

$$w^{(1)} = w^{(0)} - (X X^T)^{-1} (X X^T w - Xy)$$

$$w^{(1)} = w^{(0)} - ((X X^T)^{-1} X X^T w - (X X^T)^{-1} Xy)$$

$$w^{(1)} = w^{(0)} - Iw + (X X^T)^{-1} Xy$$

$$w^{(1)} = (X X^T)^{-1} Xy$$

Which is equal to the optimal solution w^*,Therefore:

$$w^{(1)} = w^* \tag{3}$$

Therefore from equation (3) we can conclude that Newton's method will converge to the optimal solution w^* in 1 iteration no matter what the starting point w(0) of the search is.

B. Problem 2

The second problem deals with the Derivation of softmax regression gradient updates.

1) For l=k:

$$\bigtriangledown_{w^{(l)}} \hat{y}_k^{(i)} = \bigtriangledown_{w^{(l)}} \left[\frac{expx^{(i)^T} w^{(l)}}{\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')}} \right] \tag{4}$$

Recalling the formula:

$$\left(\frac{f}{g}\right)' = \left(\frac{f'g - gf'}{g^2}\right)$$

Using the formula on equation (1) we get:

$$= \frac{x^{(i)} expx^{(i)^T} w^{(l)} \sum_{k'=1}^{c} expx^{(i)^T} w^{(k')} - x^{(i)} expx^{(i)^T} w^{(l)} expx^{(i)^T} w^{(l)}}{(\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')})^2}$$

$$= x^{(i)} \left[\frac{expx^{(i)^T} w^{(l)}}{\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')}} - \frac{(expx^{(i)^T} w^{(l)})^2}{(\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')})^2} \right]$$

$$= x^{(i)} \left[\hat{y}_l^{(i)} - (\hat{y}_l^{(i)})^2 \right]$$

$$\bigtriangledown_{w^{(l)}} \hat{y}_k^{(i)} = x^{(i)} \hat{y}_l^{(i)} \left[1 - \hat{y}_l^{(i)} \right] \tag{5}$$

Hence Proved.

2) for l≠k:

$$\nabla_{w^{(l)}} \hat{y}_k^{(i)} = \nabla_{w^{(l)}} \left[\frac{expx^{(i)^T} w^{(k)}}{\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')}} \right]$$

Differentiating we get:

$$= \frac{0 - x^{(i)^T} expx^{(i)^T} w^{(k)} expx^{(i)^T} w^{(l)}}{(\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')})^2}$$

$$= -x^{(i)} \frac{expx^{(i)^T} w^{(k)}}{\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')}} \frac{expx^{(i)^T} w^{(l)}}{\sum_{k'=1}^{c} expx^{(i)^T} w^{(k')}}$$

$$\nabla_{w^{(l)}} \hat{y}_k^{(i)} = -x^{(i)} \hat{y}_k^{(i)} \hat{y}_l^{(i)} \tag{6}$$

Hence proved.

3) $\nabla_{w^{(l)}} f_{CE}(W,b) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} y_k^{(i)} \nabla_{w^{(l)}} log\hat{y}_k^{(i)}$:

$$= -\frac{1}{n} \sum_{i=1}^{n} x^{(i)} \left[\frac{y_l^{(i)} \hat{y}_l^{(i)} (1 - \hat{y}_l^{(i)})}{y_l^{(i)}} - \sum_{k \neq l} \frac{y_k^{(i)} \hat{y}_k^{(i)} \hat{y}_l^{(i)}}{\hat{y}_k^{(i)}} \right]$$

$$= -\frac{1}{n} \sum_{i=1}^{n} x^{(i)} \left[y_l^{(i)} (1 - \hat{y}_l^{(i)}) - \sum_{k \neq l} y_k^{(i)} \hat{y}_l^{(i)} \right]$$

$$= -\frac{1}{n} \sum_{i=1}^{n} x^{(i)} \left[(y_l^{(i)} - y_l^{(i)} \hat{y}_l^{(i)}) + y_l^{(i)} \hat{y}_l^{(i)} - \sum_{k} y_k^{(i)} \hat{y}_l^{(i)} \right]$$

Simplifying we get:

$$= -\frac{1}{n} \sum_{i=1}^{n} x^{(i)} \left[(y_l^{(i)} - \hat{y}_l^{(i)}) \sum_{k} y_k^{(i)} \right]$$

Where we know that $\sum_k y_k^{(i)} = 1$, Therefore:

$$\nabla_{w^{(l)}} f_{CE}(W,b) = -\frac{1}{n} \sum_{i=1}^{n} x^{(i)} \left[y_l^{(i)} - \hat{y}_l^{(i)} \right] \tag{7}$$

4) $\nabla_b f_{CE}(W,b) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} y_k^{(i)} \nabla_b \, log\hat{y}_k^{(i)}$:

$$= -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} y_k^{(i)} \frac{1}{\hat{y}_k^{(i)}} \nabla_b \, \hat{y}_k^{(i)}$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \left[y_l^{(i)} \frac{1}{\hat{y}_l^{(i)}} \nabla_b \, \hat{y}_l^{(i)} + \sum_{k \neq l} y_k^{(i)} \frac{1}{\hat{y}_k^{(i)}} \hat{y}_l^{(i)} \nabla_b \, \hat{y}_k^{(i)} \right]$$

$$-\frac{1}{n} \sum_{i=1}^{n} \left[y_l^{(i)} \frac{1}{\hat{y}_l^{(i)}} \left(\hat{y}_l^{(i)} (1 - \hat{y}_l^{(i)}) \right) + \sum_{k \neq l} y_k^{(i)} \frac{1}{\hat{y}_k^{(i)}} (-\hat{y}_l^{(i)} \hat{y}_k^{(i)}) \right]$$

Cancelling the like terms we get:

$$= -\frac{1}{n} \sum_{i=1}^{n} \left[y_l^{(i)} (1 - \hat{y}_l^{(i)}) - \sum_{k} y_k^{(i)} \hat{y}_l^{(i)} \right]$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \left[y_l^{(i)} - y_l^{(i)} \hat{y}_l^{(i)} + y_l^{(i)} \hat{y}_l^{(i)} - \sum_{k} y_k^{(i)} \hat{y}_l^{(i)} \right]$$

Simplifying and eliminating more like terms:

$$-\frac{1}{n} \sum_{i=1}^{n} \left[y_l^{(i)} - \hat{y}_l^{(i)} \sum_{k} y_k^{(i)} \right]$$

Like the previous subdivision, we know that $\sum_k y_k^{(i)} = 1$, Therefore:

$$-\frac{1}{n} \sum_{i=1}^{n} \left[y_l^{(i)} - \hat{y}_l^{(i)} \right] \tag{8}$$

Hence Proved.

C. Problem 3

The likelihood can be given as:

$$P(D|W,b) = \sum_{i=1}^{n} \prod_{k=1}^{c} (\hat{y}_k^{(i)})^{y_k^{(i)}}$$

Taking log on both sides:

$$logP(D|W,b) = log \sum_{i=1}^{n} \prod_{k=1}^{c} (\hat{y}_k^{(i)})^{y_k^{(i)}}$$

$$-logP(D|W,b) = -\sum_{i=1}^{n} \sum_{k=1}^{c} y_k^{(i)} log(\hat{y}_k^{(i)})$$

$$-logP(D|W,b) = f_{ce}(D;W,b) \tag{9}$$

D. Problem 2

The second problem deals with the implementation of the softmax regression on the Fashion MNIST Dataset by Zalando's article images which consists of 28x28 images consisting of 10 classes. [2] The fine tuned hyper-parameters

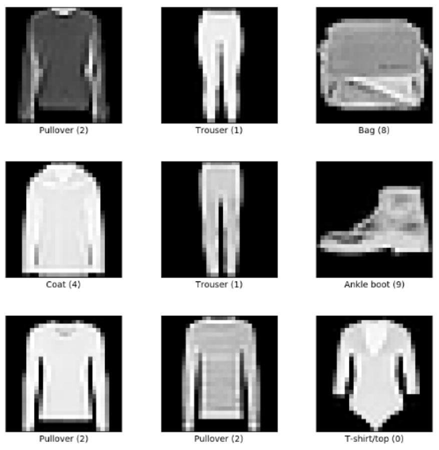

Fig. 1. Few Images from Fashion MNIST Dataset [2]

that were obtained are:

```
#The Obtained Parameters are:
"Learning Rate"=0.09
"Number of Epochs"=10
"Batch Size"=64
"Training Accuracy"=0.8598 or 85.98%
"Testing Accuracy"=0.8409 or 84.09%
```

Listing 1. Fined-Tuned Hypeparameters

The loss is plotted below and the average loss is around 0.5 as you can see in the figure below.

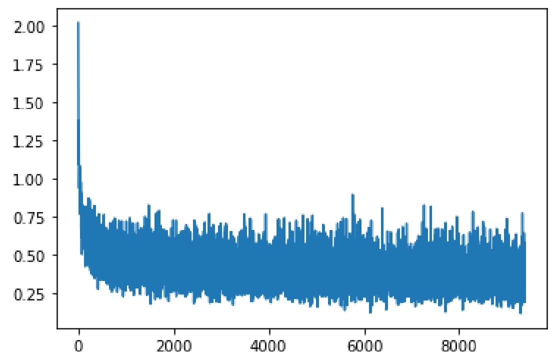

Fig. 2. The Unregularized cross entroy loss
[2]

```
#Last Epoch
Batch number: 934 | Training Loss (fCE):
    0.2999921564233597 | Accuracy: 0.890625
Batch number: 935 | Training Loss (fCE):
    0.2820675914319183 | Accuracy: 0.90625
Batch number: 936 | Training Loss (fCE):
    0.41484956992653343 | Accuracy: 0.859375
Batch number: 937 | Training Loss (fCE):
    0.17518961334315786 | Accuracy: 0.9375
Batch number: 938 | Training Loss (fCE):
    0.5163366776990248 | Accuracy: 0.875
```

Listing 2. Statistics for the last few batches in the last epoch

References

[1] https://www.deeplearningbook.org/
[2] https://www.tensorflow.org/datasets/catalog/fashion_mnist