Homework 3 **Daniel Wivagg**
RBE 549 Due Date: 2/8/19
Prof. Jacob Whitehill

Problem 1

For this problem, the two-layer neural network is represented by:

$$f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$$

And the cost function is:

$$J(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} (\hat{y}^{(i)} - y^{(i)})^2 = \frac{1}{2n} (\mathbf{X}^\top \mathbf{w} - \mathbf{y})^\top (\mathbf{X}^\top \mathbf{w} - \mathbf{y})$$

Using Newton's method, the optimal weights are given by:

$$\mathbf{w}^* = \mathbf{w}^{(0)} - \mathbf{H}\left[J(\mathbf{w}^{(0)})\right]^{-1} \nabla_{\mathbf{w}} J(\mathbf{w}^{(0)})$$

Here, $\mathbf{H}$ is the Hessian matrix and $\nabla_{\mathbf{w}}$ is the gradient with respect to $\mathbf{w}$. Plugging in the cost function will allow us to solve for $\mathbf{w}^*$ for any arbitrary starting value of $\mathbf{w}^{(0)}$.

$$\mathbf{w}^* = \mathbf{w}^{(0)} - \mathbf{H}\left[\frac{1}{2n} (\mathbf{X}^\top \mathbf{w} - \mathbf{y})^\top (\mathbf{X}^\top \mathbf{w} - \mathbf{y})\right]^{-1} \nabla_{\mathbf{w}} \frac{1}{2n} (\mathbf{X}^\top \mathbf{w} - \mathbf{y})^\top (\mathbf{X}^\top \mathbf{w} - \mathbf{y})$$

The gradient of the cost function in matrix form is:

$$\begin{aligned}
\nabla_{\mathbf{w}} \left[\frac{1}{2n} (\mathbf{X}^\top \mathbf{w} - \mathbf{y})^\top (\mathbf{X}^\top \mathbf{w} - \mathbf{y})\right] &= \frac{1}{n} \mathbf{X}(\mathbf{X}^\top \mathbf{w} - \mathbf{y}) \\
&= \frac{1}{n} (\mathbf{X}\mathbf{X}^\top \mathbf{w} - \mathbf{X}\mathbf{y})
\end{aligned}$$

The Hessian matrix is computed by taking the Jacobian of the gradient from the previous step.

$$\begin{aligned}
\mathbf{H}\left[J(\mathbf{w}^{(0)})\right] &= \mathbf{J}(\nabla_{\mathbf{w}} J(\mathbf{w}^{(0)})) \\
&= \mathbf{J}\left[\frac{1}{n} (\mathbf{X}\mathbf{X}^\top \mathbf{w} - \mathbf{X}\mathbf{y})\right] \\
&= \frac{1}{n} \mathbf{X}\mathbf{X}^\top
\end{aligned}$$

Finally, putting it all together:

$$\begin{aligned}
\mathbf{w}^* &= \mathbf{w}^{(0)} - \left[\frac{1}{n} \mathbf{X}\mathbf{X}^\top\right]^{-1} \left[\frac{1}{n} (\mathbf{X}\mathbf{X}^\top \mathbf{w}^{(0)} - \mathbf{X}\mathbf{y})\right] \\
&= \mathbf{w}^{(0)} - \left[n(\mathbf{X}\mathbf{X}^\top)^{-1}\right] \left[\frac{1}{n} (\mathbf{X}\mathbf{X}^\top \mathbf{w}^{(0)} - \mathbf{X}\mathbf{y})\right] \\
&= \mathbf{w}^{(0)} - (\mathbf{X}\mathbf{X}^\top)^{-1} (\mathbf{X}\mathbf{X}^\top \mathbf{w}^{(0)} - \mathbf{X}\mathbf{y}) \\
&= \mathbf{w}^{(0)} - \mathbf{w}^{(0)} + (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}\mathbf{y} \\
&= (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}\mathbf{y}
\end{aligned}$$

This is the final desired form. Thus, Newton's method converges in one iteration to the optimal solution as found by gradient descent.

Problem 2

The completion of the derivations is shown below. Note that the activation is given by:

$$\hat{y}_k^{(i)} = \frac{\exp z_k}{\sum_{k'=1}^{c} \exp z_{k'}}$$

Where the pre-activation function is:

$$z_k = \mathbf{x}^\top \mathbf{w}_k$$

Part A Derivation of $\nabla_{\mathbf{w}_l} \hat{y}_k^{(i)}$ for $l = k$

The gradient computation requires the use of the chain rule and quotient rule.

$$\begin{aligned}
\nabla_{\mathbf{w}_l} \hat{y}_k^{(i)} &= \nabla_{\mathbf{w}_l} \frac{\exp(\mathbf{x}^{(i)}\mathbf{w}_l)}{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})} \\
&= \frac{\mathbf{x}^{(i)} \exp(\mathbf{x}^{(i)}\mathbf{w}_l) \sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'}) - \mathbf{x}^{(i)} \exp(\mathbf{x}^{(i)}\mathbf{w}_l) \exp(\mathbf{x}^{(i)}\mathbf{w}_l)}{\left[\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})\right]^2} \\
&= \mathbf{x}^{(i)} \frac{\exp(\mathbf{x}^{(i)}\mathbf{w}_l)}{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})} \frac{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'}) - \exp(\mathbf{x}^{(i)}\mathbf{w}_l)}{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})} \\
&= \mathbf{x}^{(i)} \hat{y}_l^{(i)} (1 - \hat{y}_l^{(i)})
\end{aligned}$$

It is important to note that the derivative of the summation is equal to the derivative of $\exp z_l$ since all the other terms are eliminated where $k' \neq l$.

Part B Derivation of $\nabla_{\mathbf{w}_l} \hat{y}_k^{(i)}$ for $l \neq k$

Once, again, the quotient rule and chain rule are needed.

$$\begin{aligned}
\nabla_{\mathbf{w}_l} \hat{y}_k^{(i)} &= \nabla_{\mathbf{w}_l} \frac{\exp(\mathbf{x}^{(i)}\mathbf{w}_k)}{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})} \\
&= \frac{0 - \mathbf{x}^{(i)} \exp(\mathbf{x}^{(i)}\mathbf{w}_k) \exp(\mathbf{x}^{(i)}\mathbf{w}_l)}{\left[\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})\right]^2} \\
&= -\mathbf{x}^{(i)} \frac{\exp(\mathbf{x}^{(i)}\mathbf{w}_k)}{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})} \frac{\exp(\mathbf{x}^{(i)}\mathbf{w}_l)}{\sum_{k'=1}^{c} \exp(\mathbf{x}^{(i)}\mathbf{w}_{k'})} \\
&= -\mathbf{x}^{(i)} \hat{y}_k^{(i)} \hat{y}_l^{(i)}
\end{aligned}$$

Once again, the derivative of the summation is simply equal to $\exp z_l$ since the gradient is only taken with respect to w_l.

Part C Computation of the total gradient, $\nabla_{\mathbf{w}_l} f_{CE}(\mathbf{W})$

The final derivation is completed below using the gradients from Parts A and B.

$$
\begin{aligned}
\nabla_{\mathbf{w}_l} f_{CE}(\mathbf{W}) &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} y_k^{(i)} \nabla_{\mathbf{w}_l} \log \hat{y}_k^{(i)} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} y_k^{(i)} \left(\frac{\nabla_{\mathbf{w}_l} \hat{y}_k^{(i)}}{\hat{y}_k^{(i)}}\right) \\
&= -\frac{1}{n}\sum_{i=1}^{n}\left[y_l^{(i)} \frac{\mathbf{x}^{(i)}\hat{y}_l^{(i)}(1-\hat{y}_l^{(i)})}{\hat{y}_l^{(i)}} - \sum_{k\neq l} y_k^{(i)} \frac{\mathbf{x}^{(i)}\hat{y}_k^{(i)}\hat{y}_l^{(i)}}{\hat{y}_k^{(i)}}\right] \\
&= -\frac{1}{n}\sum_{i=1}^{n}\left[y_l^{(i)}\mathbf{x}^{(i)}(1-\hat{y}_l^{(i)}) - \sum_{k\neq l} y_k^{(i)}\mathbf{x}^{(i)}\hat{y}_l^{(i)}\right] \\
&= -\frac{1}{n}\sum_{i=1}^{n}\left[y_l^{(i)}\mathbf{x}^{(i)}(1-\hat{y}_l^{(i)}) - \mathbf{x}^{(i)}\hat{y}_l^{(i)}\sum_{k\neq l} y_k^{(i)}\right] \\
&= -\frac{1}{n}\sum_{i=1}^{n} y_l^{(i)}\mathbf{x}^{(i)}(1-\hat{y}_l^{(i)}) - \mathbf{x}^{(i)}\hat{y}_l^{(i)}(1-y_l^{(i)}) \\
&= -\frac{1}{n}\sum_{i=1}^{n} y_l^{(i)}\mathbf{x}^{(i)} - y_l^{(i)}\mathbf{x}^{(i)}\hat{y}_l^{(i)} - \mathbf{x}^{(i)}\hat{y}_l^{(i)} + \mathbf{x}^{(i)}\hat{y}_l^{(i)}y_l^{(i)} \\
&= -\frac{1}{n}\sum_{i=1}^{n} y_l^{(i)}\mathbf{x}^{(i)} - \mathbf{x}^{(i)}\hat{y}_l^{(i)} \\
&= -\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}^{(i)}(y_l^{(i)} - \hat{y}_l^{(i)})
\end{aligned}
$$

This is the final desired form.

It is important to note that for the label vector $\mathbf{y}$, only one value (corresponding to the true class label) is equal to 1. Thus, for the summation $\sum_{k\neq l} y_k$: if y_l is the true class label the summation will equal 0, otherwise it will equal 1. Therefore, it is replaced in the steps above by $1 - y_l$.

Problem 3

The final performance achieved by the network on handwritten digits was as follows:

- Test set cross-entropy loss (unregularized): 0.17
- Test set accuracy (percent correct): 90.8 %

This can be verified by running the attached Python file. The execution time is very slow (around 20 minutes) because of the limitations of *np.dot*.