Armaan Lalani 1005023225

Assignment 2: Neural Networks

1.1 Helper Functions

1. ReLU

```
def relu(x):
    return np.maximum(x,0)
```

2. Softmax

```
def softmax(x):
    x = x - np.max(x)
    exp_x = np.exp(x)
    return exp_x/(np.sum(exp_x,axis=1,keepdims=True)+0.00001)
```

3. Compute

```
def compute(X, W, b):
    return np.matmul(W,X) + b
```

4. averageCE

```
def averageCE(target, prediction):
    return -np.mean(target*np.log(prediction+0.00001))
```

5. gradCE

```
def gradCE(target, prediction):
    return prediction-target
```

$$\frac{dL}{do} = \frac{dL}{dp}\frac{dp}{do}, \quad \text{where } p = \text{softmax}(o) \text{ and } o = W_o h + b$$

$$\frac{dL}{dp} = \frac{d}{dp}\left(-\sum_{k=1}^{K} y_k \log(p_k)\right)$$
$$= -y^T \cdot \frac{d}{dp}\log(p)$$
$$= -y^T\left(\frac{1}{p}\right)$$

$$\frac{dp}{do} = \frac{d}{do}\left(\frac{\exp(o)}{\sum_k \exp(o_k)}\right)$$
$$= \frac{\exp(o)\sum_k \exp(o_k) - \exp(o)^T\exp(o)}{\left(\sum_k \exp(o_k)\right)^2}$$
$$= \frac{\exp(o)\left(\sum_k \exp(o_k) - \exp(o)\right)}{\left(\sum_k \exp(o_k)\right)^2}$$
$$= p(1-p)$$

$$\therefore \frac{dL}{do} = -y^T \cdot \frac{1}{p} \cdot p(1-p)$$
$$= p - y$$

1.2 : Backpropagation Derivation

1) Analytical Expression of $\frac{dL}{dW_o}$

$$\frac{dL}{dW_o} = \frac{dL}{dp}\frac{dp}{do}\frac{do}{dW_o}$$
$$= (p-y)\frac{do}{dW_o}$$
$$= h^T \cdot (p-y)$$

2) Analytical Expression of $\frac{dL}{db_o}$

$$\frac{dL}{db_o} = \frac{dL}{dp}\frac{dp}{do}\frac{do}{db_o}$$
$$= (p-y)\frac{do}{db_o}$$
$$= 1^T \cdot (p-y)$$ sum of the elements of $(p-y)$

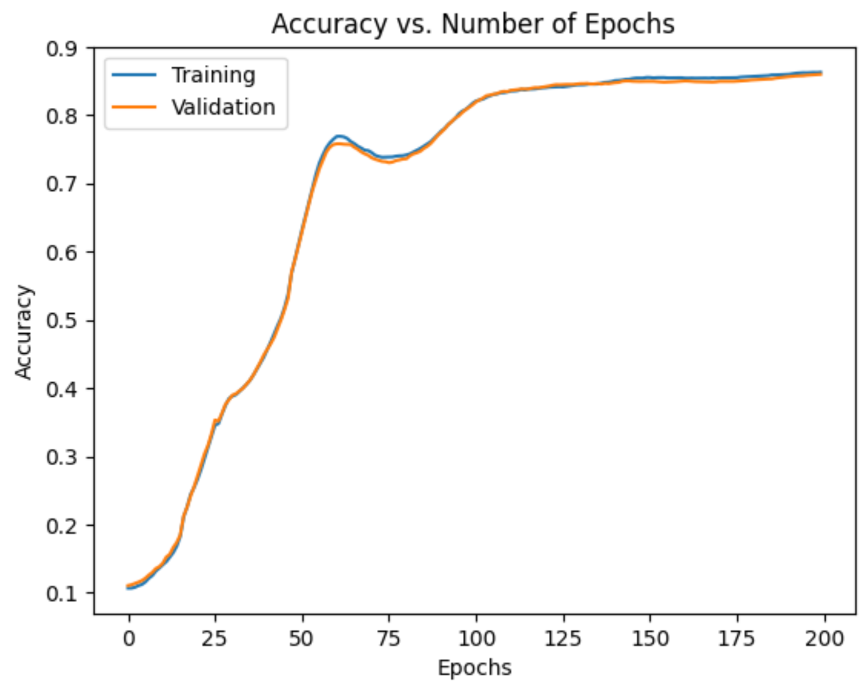
3) Analytical Expression of $\frac{dL}{dW_h}$

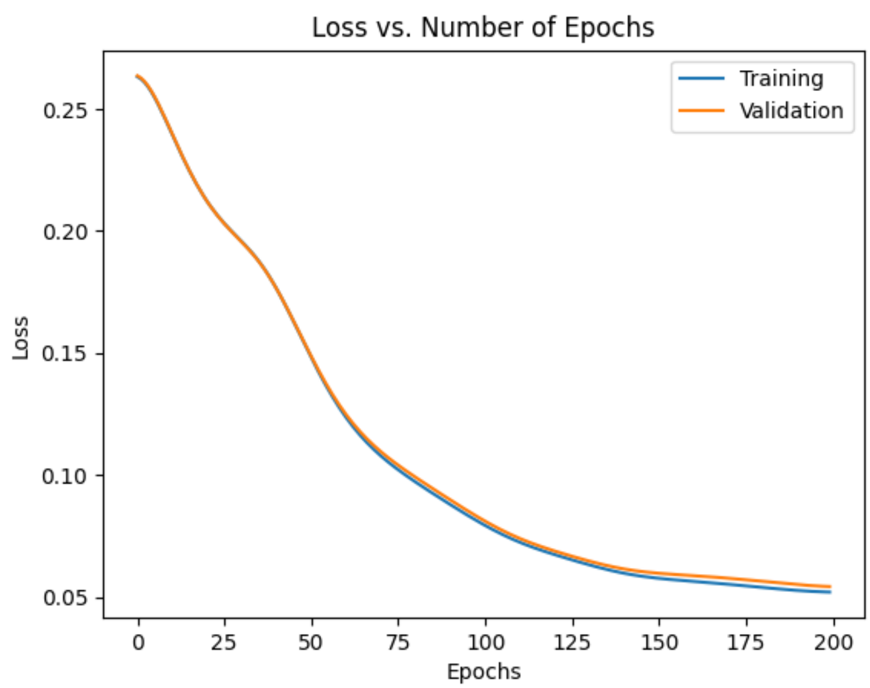
$$\frac{dL}{dW_h} = \frac{dL}{do}\frac{do}{dh}\frac{dh}{dz_h}\frac{dz_h}{dW_h}$$ where $z_h = W_h x + b_h$
$$= (p-y)\, W_o\, x \cdot \frac{dh}{dz_h}$$
$$= \begin{cases} x^T \otimes (p-y)W_o^T & \text{if } z_h > 0 \\ 0 & \text{otherwise} \end{cases}$$ for each element individually

4) Analytical Expression of $\frac{dL}{db_h}$

$$\frac{dL}{db_h} = \frac{dL}{do}\frac{do}{dh}\frac{dh}{dz_h}\frac{dz_h}{db_h}$$
$$= (p-y)\, W_o\, 1 \cdot \frac{dh}{dz_h}$$
$$= \begin{cases} 1^T \otimes (p-y)W_o^T & \text{if } z_h > 0 \\ 0 & \text{otherwise} \end{cases}$$ for each element individually

1.3: Learning





Test Accuracy: 0.8682
Test Loss: 0.0518