Andrew Elysee (aelys2)
High-level discussion with Max Han (mhan36)
CS446

HW2

1.

a.

i. Since $P(x_d = 1) = q_d$, then the probability of getting a vector x given q, is

$$P(x|q) = \prod_{d=1}^{D} q_d^{x_d}(1 - q_d)^{1-x_d}$$

This makes sense because if $x_d = 1$, then the probability of that happening is just q_d, but if $x_d = 0$ then the probability of that is $1 - q_d$ so we take the product of all these probabilities to get $P(x|q)$.

ii. Since we know that each vector $x^{(i)}$ was drawn from one of our Bernoulli distributions with parameters $p^{(k)}$, the expression must be the sum of the products of the probabilities that the kth distribution was chosen and the probability of $x^{(i)}$ given $p^{(k)}$. So, we get

$$P\left(x^{(i)}\middle|p,\pi\right) = \sum_{k=1}^{K} \pi_k P(x^{(i)}|p^{(k)})$$

iii.

$$P(D|\pi,p) = \prod_{i=1}^{D} P\left(x^{(i)}\middle|p,\pi\right)$$

$$\log P(D|\pi,p) = \sum_{i=1}^{D} \log P\left(x^{(i)}\middle|p,\pi\right)$$

b.

i.

$$P\left(z^{(i)}\middle|\pi\right) = \prod_{k=1}^{K} \pi_k^{z_k^{(i)}}$$

ii.

$$P\left(x^{(i)}\middle|z^{(i)},p,\pi\right) = \prod_{k=1}^{K} P\left(x^{(i)}\middle|p^{(k)}\right)^{z_k^{(i)}}$$

iii.

$$P(Z,D|p,\pi) = \prod_{i=1}^{D} P\left(z^{(i)},x^{(i)}\middle|p,\pi\right) = \prod_{i=1}^{D} P\left(x^{(i)}\middle|z^{(i)},p,\pi\right) P\left(z^{(i)}\middle|\pi\right)$$

$$P(Z,D|p,\pi) = \prod_{i=1}^{D} \left[\prod_{k=1}^{K} P\left(x^{(i)}\middle|p^{(k)}\right)^{z_k^{(i)}}\right]\left[\prod_{k=1}^{K} \pi_k^{z_k^{(i)}}\right]$$

iv.

$$\eta\left(z_k^{(i)}\right) = E\left[z_k^{(i)}\middle|x^{(i)}, p, \pi\right]$$
$$= P\left(z_k^{(i)} = 1\middle|x^{(i)}, p, \pi\right)$$

Using Bayes rule,

$$= \frac{P\left(x^{(i)}\middle|z_k^{(i)}, p, \pi\right)P\left(z_k^{(i)} = 1\middle|p, \pi\right)}{\sum_{j=1}^{K} P\left(x^{(i)}\middle|z_j^{(i)}, p, \pi\right)P\left(z_j^{(i)} = 1\middle|p, \pi\right)}$$
$$= \frac{\pi_k \prod_{d=1}^{D}\left(p_d^{(k)}\right)^{x_d^{(i)}}\left(1 - p_d^{k}\right)^{1-x_d^{(i)}}}{\sum_{j=1}^{K}\pi_j \prod_{d=1}^{D}\left(p_d^{(j)}\right)^{x_d^{(i)}}\left(1 - p_d^{j}\right)^{1-x_d^{(i)}}}$$

v.

$$E[\log P(Z, D|\tilde{p}, \tilde{\pi}) \mid D, p, \pi] = \log P(Z, D|\tilde{p}, \tilde{\pi})$$
$$= \sum_{i=1}^{D}\left[\sum_{k=1}^{K} z_k^{(i)} \log P\left(x^{(i)}\middle|\tilde{p}^{(k)}\right)\right] + \left[\sum_{k=1}^{K} z_k^{(i)} \log\tilde{\pi}_k\right]$$
$$= \sum_{i=1}^{D}\left[\sum_{k=1}^{K} z_k^{(i)}\left(\log P\left(x^{(i)}\middle|\tilde{p}^{(k)}\right) + \log\tilde{\pi}_k\right)\right]$$
$$= \sum_{i=1}^{D}\sum_{k=1}^{K} z_k^{(i)}\left[\log\prod_{d=1}^{D}\left(\tilde{p}_d^{(k)}\right)^{x_d^{(i)}}\left(1 - \tilde{p}_d^{k}\right)^{1-x_d^{(i)}} + \log\tilde{\pi}_k\right]$$
$$= \sum_{i=1}^{D}\sum_{k=1}^{K} z_k^{(i)}\left[\sum_{d=1}^{D}\left(x_d^{(i)}\log(\tilde{p}_d^{(k)}) + \left(1 - x_d^{(i)}\right)\log(1 - \tilde{p}_d^{k})\right) + \log\tilde{\pi}_k\right]$$

Since there is an imaginary expected value sign around the whole expression, $z_k^{(i)} = E\left[z_k^{(i)}\right] = \eta\left(z_k^{(i)}\right)$, so we get:

$$= \sum_{i=1}^{D}\sum_{k=1}^{K}\eta\left(z_k^{(i)}\right)\left[\sum_{d=1}^{D}\left(x_d^{(i)}\log(\tilde{p}_d^{(k)}) + \left(1 - x_d^{(i)}\right)\log(1 - \tilde{p}_d^{k})\right) + \log\tilde{\pi}_k\right]$$

c.

i. To find the value of $\tilde{p}$ that maximizes the E step, we must set the derivative of the E step relative to $\tilde{p}$ to 0.

$$\frac{\partial}{\partial\tilde{p}_d^{(k)}} E[\log P(Z, D|\tilde{p}, \tilde{\pi}) \mid D, p, \pi] = 0$$
$$\sum_{i=1}^{D}\eta\left(z_k^{(i)}\right)\left[\frac{x_d^{(i)}}{\tilde{p}_d^{(k)}} + \frac{1 - x_d^{(i)}}{1 - \tilde{p}_d^{k}}\right] = 0$$
$$\sum_{i=1}^{D}\eta\left(z_k^{(i)}\right)\left[x_d^{(i)}\left(1 - \tilde{p}_d^{k}\right) + \tilde{p}_d^{(k)}\left(1 - x_d^{(i)}\right)\right] = 0 * \tilde{p}_d^{(k)}\left(1 - \tilde{p}_d^{k}\right) = 0$$

$$\sum_{i=1}^{D} \eta\left(z_k^{(i)}\right)\left[x_d^{(i)} - \tilde{p}_d^{(k)}\right] = \sum_{i=1}^{D} \eta\left(z_k^{(i)}\right) x_d^{(i)} - \sum_{i=1}^{D} \eta\left(z_k^{(i)}\right)\left(\tilde{p}_d^{(k)}\right) = 0$$

Rearranging to solve for $\tilde{p}_d^{(k)}$ we get:

$$\tilde{p}_d^{(k)} = \frac{\sum_{i=1}^{D} \eta\left(z_k^{(i)}\right) x_d^{(i)}}{\sum_{i=1}^{D} \eta\left(z_k^{(i)}\right)} = \frac{\sum_{i=1}^{D} \eta\left(z_k^{(i)}\right) x_d^{(i)}}{N_k}$$

ii. First, we will set up a Lagrangian, with λ to enforce the constraint that $\Sigma\pi_k = 1$. Then take the derivative, and set it equal to 0.

$$L(\tilde{\pi}, \lambda) = -\sum_{i=1}^{D}\sum_{k=1}^{K} \eta\left(z_k^{(i)}\right) \log \tilde{\pi}_k + \lambda\left(\sum_{k=1}^{K} \tilde{\pi}_k - 1\right)$$

$$\frac{\partial}{\partial \tilde{\pi}_k} L(\tilde{\pi}, \lambda) = -\sum_{i=1}^{D} \frac{\eta\left(z_k^{(i)}\right)}{\tilde{\pi}_k} + \lambda = 0$$

$$\sum_{i=1}^{D} \frac{\eta\left(z_k^{(i)}\right)}{\tilde{\pi}_k} = \lambda$$

$$\tilde{\pi}_k = \frac{\sum_{i=1}^{D} \eta\left(z_k^{(i)}\right)}{\lambda} = \frac{N_k}{\lambda}$$

First we see that our rearranged constraint is $\sum_{k=1}^{K} \tilde{\pi}_k - 1 = \sum_{k=1}^{K} \frac{N_k}{\lambda} - 1 = 0 \rightarrow \frac{1}{\lambda}\sum_{k=1}^{K} N_K = 1 \rightarrow \sum_{k=1}^{K} N_K = \lambda$ so, solving for λ and taking the derivative, we that

$$L(\lambda) = -\sum_{i=1}^{D}\sum_{k=1}^{K} \eta\left(z_k^{(i)}\right) \log \frac{N_k}{\lambda} + \lambda\left(\sum_{k=1}^{K} \frac{N_k}{\lambda} - 1\right)$$

$$= -\sum_{i=1}^{D}\sum_{k=1}^{K} \eta\left(z_k^{(i)}\right)(\log N_k - \log \lambda) + \lambda\left(\sum_{k=1}^{K} N_k - \lambda\right)$$

$$\frac{\partial}{\partial \lambda} L(\lambda) = \frac{1}{\lambda}\sum_{i=1}^{D}\sum_{k=1}^{K} \eta\left(z_k^{(i)}\right) - 1 = 0 \rightarrow \frac{1}{\lambda}\sum_{i=1}^{D}\sum_{k=1}^{K} \eta\left(z_k^{(i)}\right) = 1$$

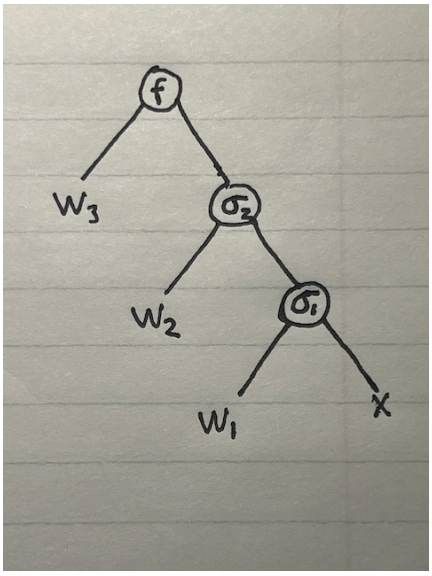
$$\lambda = \sum_{i=1}^{D}\sum_{k=1}^{K} \eta\left(z_k^{(i)}\right) = \sum_{k=1}^{K} N_k$$

This gives us the final piece for our update for $\tilde{\pi}_k$:

$$\tilde{\pi}_k = \frac{N_k}{\lambda} = \frac{N_k}{\sum_{k'=1}^{K} N_{k'}}$$

2.

a.



b. $\sigma_1(u) = \frac{1}{1+\exp(-u)} \rightarrow \frac{\partial \sigma_1}{\partial u} = -\frac{1}{(1+\exp(-u))^2} * -\exp(-u) = \frac{\exp(-u)}{(1+\exp(-u))^2}$

This is the partial derivative using only u, but with some manipulation we get:

$$\frac{\partial \sigma_1}{\partial u} = \frac{\exp(-u)}{(1+\exp(-u))^2} = \frac{1}{1+\exp(-u)} * \frac{\exp(-u)}{1+\exp(-u)} = \frac{1}{1+\exp(-u)} * \left(1 - \frac{1}{1+\exp(-u)}\right)$$

$$\frac{\partial \sigma_1}{\partial u} = \sigma_1(u)(1 - \sigma_1(u))$$

c. A forward pass refers to the calculation of the outputs from the input layers, while a backwards pass is the calculation of the change in weights using some gradient descent method.

d. $\frac{\partial f}{\partial w_3} = \sigma_2\big(w_2\sigma_1(w_1x)\big) = \sigma_2(x_2)$

To make the computation of this derivative easy, we should save the output of $\sigma_2(x_2)$.

e. $\frac{\partial f}{\partial w_2} = \frac{\partial f}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial w_2} = \sigma_1(w_1x)w_3\sigma_2\big(w_2\sigma_1(w_1x)\big)\Big(1 - \sigma_2\big(w_2\sigma_1(w_1x)\big)\Big) =$

$w_3\sigma_1(x_1)\sigma_2\big(w_2\sigma_1(x_1)\big)\Big(1 - \sigma_2\big(w_2\sigma_1(x_1)\big)\Big) = w_3\sigma_1(x_1)\sigma_2(x_2)\big(1 - \sigma_2(x_2)\big)$

To make this calculation easier, we should save $\sigma_1(x_1)$ and $\sigma_2(x_2)$, assuming that w_3 is implied to be saved.

f. $\frac{\partial f}{\partial w_1} = \frac{\partial f}{\partial \sigma_2}\frac{\partial \sigma_2}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial w_1} = w_3w_2x\sigma_1(w_1x)\big(1 - \sigma_1(w_1x)\big)\sigma_2\big(w_2\sigma_1(w_1x)\big)\Big(1 -$

$\sigma_2\big(w_2\sigma_1(w_1x)\big)\Big) = w_3w_2x\,\sigma_1(x_1)(1 - \sigma_1(x_1))\sigma_2(x_2)\big(1 - \sigma_2(x_2)\big) = \frac{\partial f}{\partial w_2} *$

$w_2x(1 - \sigma_1(x_1))$

We should save $\sigma_1(x_1)$ and $\sigma_2(x_2)$ from the forward pass. To obtain the result as early as possible and reuse as many results as possible we should do $\frac{\partial f}{\partial w_3}, \frac{\partial f}{\partial w_2}$, and finally $\frac{\partial f}{\partial w_1}$, because we can reuse the results of $\frac{\partial f}{\partial w_3}$ in $\frac{\partial f}{\partial w_2}$ and the results of $\frac{\partial f}{\partial w_2}$ in $\frac{\partial f}{\partial w_1}$ like I showed in the equation above. This is reverse of the forward pass.

g. After the first layer the output dimension is 20 X 24 X 24. After the max pooling layer, we get an output of dimension 20 X 12 X 12.
h. If we undo the max pooling on the 4x4 output, we see that the output of the second convolution layer is 8x8. From this, we can derive that our filters would be of size 5x5 with a stride of 1, because (input size) * (kernel size) + 1 = 12-5+1 = 8. The channel dimensions is 50.

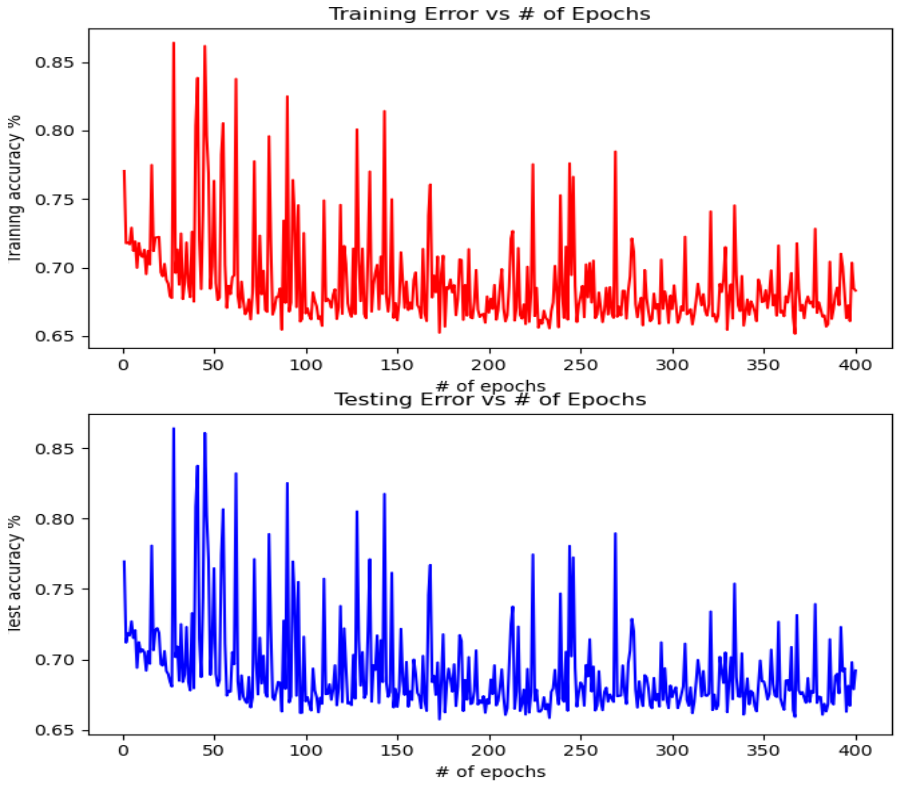
3.

a. Submitted on gradescope
b. Submitted on gradescope
c. In my graphs the Testing and training errors are nearly indistinguishable but looking closely I think that as C increases, the gap between the test and training error seems to increase, this is because the low complexity of the model helps to prevent overfitting.

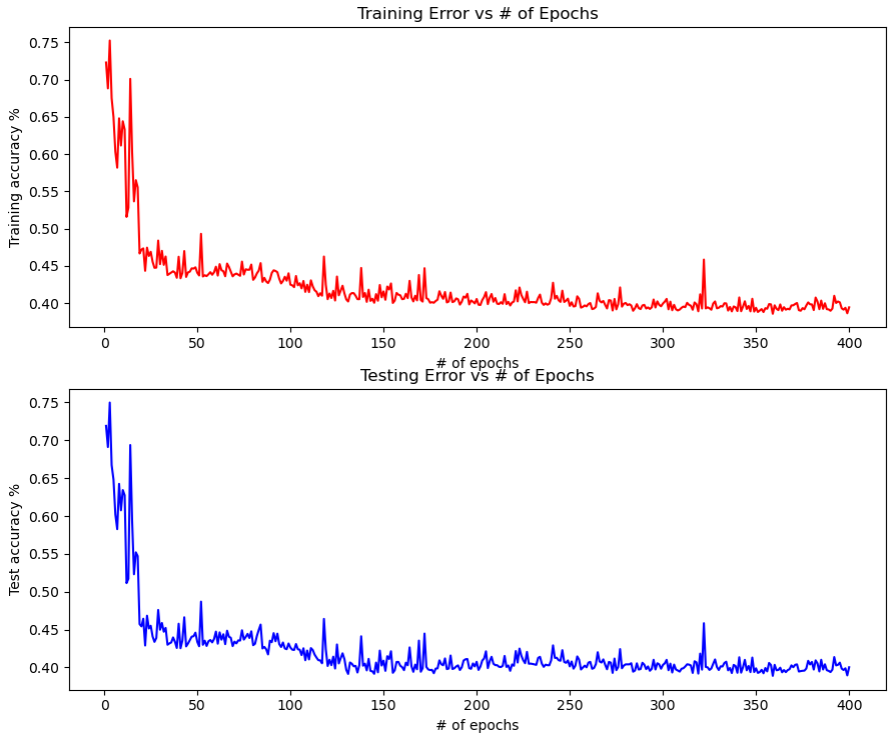
Plots are below.

Also note that the y-axis label is labeled ____ accuracy but should be ____ Error

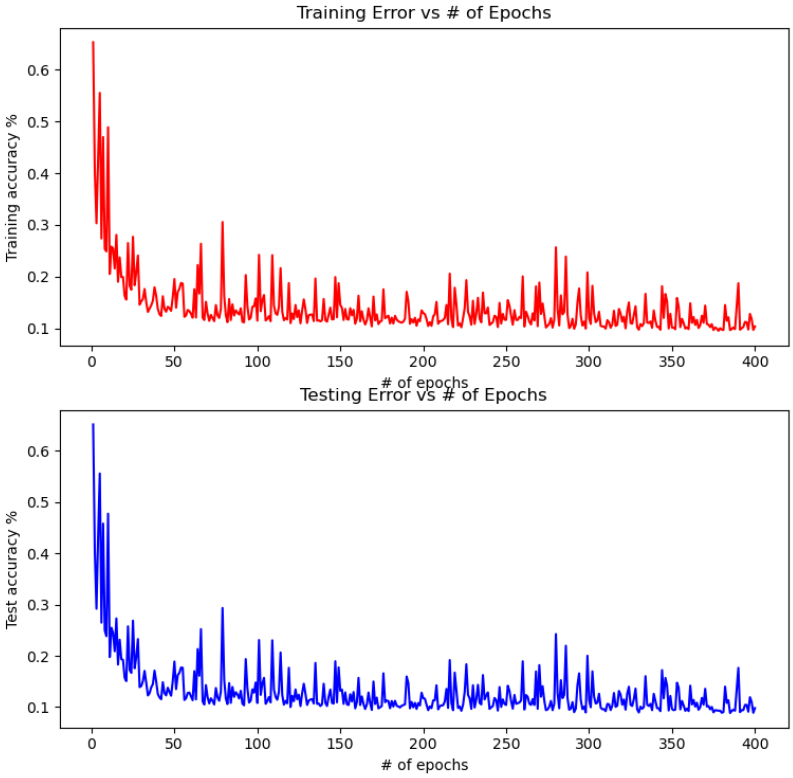
Plots for C = 1



Plots for C = 2



Plots for C = 4



d. For C=64, the testing and training errors curves are significantly smoother this also start off very accurate after the first epoch. Additionally, the gap between

the testing and training sets is much more noticeable as our training set gets a lot closer to 0 than our testing error, this could just be because of the scaling of the axis though. In hindsight, I should have overlayed the graphs.

Plots for C = 64