CS446 HW2

Shuzhen Zhang
Net ID: shuzhen2

1 Expectation Maximization

(a) Mxiture of Bernoullis
(i) As the information by given $P(x_d = 1) = q_d$

- Then $P(x_d = 1|p_d) = q_d$ and $P(x_d = 0|p_d) = 1 - q_d$
- $$P(x|p) = \Pi_d^D P(x_d = 1, x_d = 0|p_d) = \Pi_d^D q_d^{x_d}(1 - q_d)^{1-x_d} \tag{1}$$

(ii) Since the π donate the proportion of each index.

- Therefore, $P(x^{(i)}|P, \pi) = \sum_{k=1}^{K} \pi_k P(x^{(i)}|p^{(k)}), x^{(i)} \in D$

(iii) Since for different i, $x^{(i)}$ will be different. Therefore, the log-likelihood will be:

$$logP(D|\pi, p) = log\Pi_{i\in D}P(x^{(i)}|p, \pi) = \sum_{i\in D} logP(x^{(i)}|p, \pi) \tag{2}$$

(b) Expectation step
(i) Since $z^{(i)} = 1$ if $x^{(i)}$ is draw from a Bernoulli $p^{(k)}$, 0 otherwise

- Therefore, $P(z^{(i)}|\pi) = \Pi_{k=1}^{K} \pi_k^{z_k^{(i)}}$

(ii) $P(x^{(i)}|z^{(i)}, p, \pi) = \Pi_{k=1}^{K} P(x^{(i)}|p^{(k)})^{z_k^{(i)}}$
(iii)

$$P(Z, D|\pi, p) = \Pi_{i=1}^{n} P(z^{(i)}, x^{(i)}|\pi, p) = \Pi_{i=1}^{n} P(x^{(i)}|z^{(i)}, \pi, p) * P(z^{(i)}|\pi) \tag{3}$$
$$= \Pi_{i=1}^{n}\Pi_{k=1}^{K}\pi_k^{z_k^{(i)}} P(x^{(i)}|p^{(k)})^{z_k^{(i)}} \tag{4}$$

(iv)let $\eta(z_k^{(i)}) = \mathbf{E}[z_k^{(i)} = 1|x^{(i)}, \pi, p] = P(z_k^i = 1|x^{(i)}, \pi, p)$

which is a poister and it will be

$$P(z_k^i = 1|x^{(i)}, \pi, p) = \frac{P(z_k^{(i)} = 1|\pi, p)P(x^{(i)}|z_k^{(i)} = 1, p, \pi)}{\sum_{\hat{k}=1}^{K} P(z_{\hat{k}}^{(i)} = 1|\pi, p)P(x^{(i)}|z_{\hat{k}}^{(i)} = 1, p, \pi)} \tag{5}$$

- $P(z_k^{(i)} = 1|\pi, p)P(x^{(i)}|z_k^{(i)} = 1, p, \pi) = P(x^{(i)}|\pi, p)$

$$\Rightarrow \frac{P(x^{(i)}|\pi, p)}{\sum_{\hat{k}=1}^{K} P(x^{(i)}|\pi, p)} = \frac{\pi_k\Pi_d^D(p_d^{(k)})^{x_d}(1 - p_d^{(k)})^{1-x_d}}{\sum_{j=1}^{n} \pi_k\Pi_d^D(p_d^{(k)})^{x_d}(1 - p_d^{(k)})^{1-x_d}} \tag{6}$$

(V) To proof:

$$\mathbf{E}[logP(Z,D|\hat{p},\hat{\pi})|D,p,\pi] = \sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)})[log\hat{\pi}_k + \sum_{d=1}^{D}(x_d^{(i)}log\hat{p}_d^{(k)} + (1-x_d^{(i)})log(1-\hat{p}_d^{(k)}))] \tag{7}$$

- let first calculate $logP(Z,D|\hat{p},\hat{\pi})$

$$= log\Pi_{i=1}^{n}\Pi_{k=1}^{K}\hat{\pi}^{z_k^{(i)}}P(x^{(i)}|\hat{p}^{(k)})^{z_k^{(i)}} = \sum_{i=1}^{n}\sum_{k=1}^{K}z_k^{(i)}(log\hat{\pi}_k + logP(x^{(i)}|\hat{p}^{(k)})) \tag{8}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K}z_k^{(i)}(log\hat{\pi}_k + \sum_{d=1}^{D}log\hat{p}_d^{(k)x_d^{(i)}} * (1-\hat{p}_d^{(k)})^{(1-x_d^{(i)})}) \tag{9}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K}z_k^{(i)}(log\hat{\pi}_k + \sum_{d=1}^{D}x_d^{(i)}log\hat{p}_d^{(k)} + (1-x_d^{(i)})log(1-\hat{p}_d^{(k)})) \tag{10}$$

- since $\mathbf{E}(z_k^{(i)}|x^{(i)},p,\pi) = \eta(z_k^{(i)})$, and let's put that in to the equation:

$$\mathbf{E}[logP(Z,D|\hat{p},\hat{\pi})|D,p,\pi] = \sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)})[log\hat{\pi}_k + \sum_{d=1}^{D}(x_d^{(i)}log\hat{p}_d^{(k)} + (1-x_d^{(i)})log(1-\hat{p}_d^{(k)}))] \tag{11}$$

(c) Maximization step
(i)we can find the maximum by taking derivative when $\frac{\partial}{\partial\hat{p}^{(k)}}\mathbf{E}[logP(Z,D|\hat{p},\hat{\pi})|D,p,\pi] = 0$

$$\frac{\partial}{\partial\hat{p}^{(k)}}\mathbf{E}[logP(Z,D|\hat{p},\hat{\pi})|D,p,\pi] = \sum_{i=1}^{n}\eta(z_k^{(i)})\sum_{d=1}^{D}(\frac{x_d^{(i)}}{\hat{p}_d^{(k)}} - \frac{(1-x_d^{(i)})}{(1-\hat{p}_d^{(k)})}) = 0 \tag{12}$$

$$\Rightarrow \sum_{i=1}^{n}\eta(z_k^{(i)})(\frac{x^{(i)}}{\hat{p}^{(k)}} - \frac{(1-x^{(i)})}{(1-\hat{p}^{(k)})}) = 0 \Rightarrow \sum_{i=1}^{n}\eta(z_k^{(i)})(-p^{(i)} + x^{(i)}) = 0 \tag{13}$$

$$\Rightarrow p^{(i)} = \frac{\sum_{i=1}^{n}\eta(z_k^{(i)})x^{(i)}}{\sum_{i=1}^{n}\eta(z_k^{(i)})} \tag{14}$$

- since $N_k = \sum_{i=1}^{n}\eta(z_k^{(i)})$
- Therefore, $p^{(i)} = \frac{\sum_{i=1}^{n}\eta(z_k^{(i)})x^{(i)}}{N_k}$

(ii) let λ be the constrain and distribute $\sum_{k=1}^{k}\pi = 1$

- Then we obtain the equation that:

$$L(\pi,\lambda) = -\sum_{i=1}^{N}\sum_{k=1}^{K}\eta(z_k^{(i)})\log\pi_k + \lambda(\sum_{k=1}^{K}\pi_k - 1) \tag{15}$$

- By taking derivative:

$$\frac{d}{d\pi_k} L(\pi, \lambda) = -\sum_{i=1}^{N} \frac{\eta(z_k^{(i)})}{\pi_k} + \lambda = 0 \tag{16}$$

$$\Rightarrow \pi_k = \frac{\sum_{i=1}^{N} \eta(z_k^{(i)})}{\lambda} = \frac{N_k}{\lambda} \tag{17}$$

- then we can begin to solve λ:

$$L(\lambda) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)})(\log N_k - \log \lambda) + \sum_{k=1}^{K} N_k - \lambda \tag{18}$$

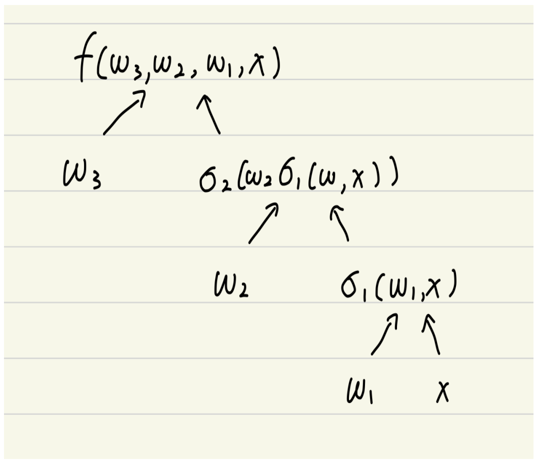
- by taking derivative of $\frac{\partial L(\lambda)}{\partial \lambda}$:

$$\frac{1}{\lambda} \sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) - 1 = 0 \Rightarrow \lambda = \sum_{i=1}^{N} \sum_{k=1}^{K} \eta(z_k^{(i)}) = \sum_{k=1}^{K} N_k \tag{19}$$

- Therefore, $\pi_k = \frac{N_k}{\sum_k^K N_k}$

2 Deep Net

(a) by given from the question, we can get the graph that:

- 

  (b) compute $\frac{\partial \sigma_1}{\partial u}$

- $\sigma_1(u) = (1 + \exp\{-u\})^{-1}$
- let $y = 1 + \exp\{-u\}$
- (1)
$$\frac{\partial \sigma_1}{\partial u} = \frac{\partial \sigma_1}{\partial y} * \frac{\partial y}{\partial u} = -1 * (1+\exp\{-u\})^{-2} * -\exp\{-u\} = (1+exp\{-u\})^{-2}\exp\{-u\} \quad (20)$$
- (2)
$$\sigma_1(u)(1 - \sigma_1(u)) \quad (21)$$

(c)

- **(1) Forward Pass:** It is a process that inputting data to neural network through layers and each layers has a weight which will generate a output. That output will become the input for next layers and finally to give the result
- **(2) Backward Pass:** It is a process that generate the error back through layers of neural network and adjust weights depends on the error for higher accuracy

(d) we should return $\frac{\partial f}{\partial w_3} = \sigma_2(w_2\sigma_1(w_1x))$
(e)calculate $\frac{\partial f}{\partial w_2}$: $f = w_3\sigma_2(w_2\sigma_1(w_1x))$

- let $y_1 = \sigma_2(w_2\sigma_1(w_1x))$
- $y_2 = w_1x$

- $$\frac{\partial f}{\partial w_2} = \frac{\partial f}{\partial y_1} * \frac{\partial y_1}{\partial w_2} = w_3 * \frac{\partial \sigma_2}{\partial y_1} * \frac{\partial y_1}{\partial w_2} = w_3 * \sigma_2(w_2\sigma_1(w_1x))(1-\sigma_2(w_2\sigma_1(w_1x))) * \sigma_1(w_1x) \tag{22}$$
- and that is what we should return

(f) compute $\frac{\partial f}{\partial w_1}$

$$\frac{\partial f}{\partial w_1} = w_3 * \frac{\partial f}{\partial y_1} * \frac{\partial y_1}{\partial y_2} * \frac{\partial y_2}{\partial w_1} = w_3 * \sigma_2(w_2\sigma_1(w_1x))(1-\sigma_2(w_2\sigma_1(w_1x))) * \frac{\partial \sigma_1}{\partial y_2} * \frac{\partial y_2}{\partial w_1} \tag{23}$$

$$= w_3 * w_2 * \sigma_2(w_2\sigma_1(w_1x))(1-\sigma_2(w_2\sigma_1(w_1x))) * \sigma_1(w_1x) * (1-\sigma_1(w_1x)) * x \tag{24}$$

- and that is what we should return that will make our computation easier.
- During the forward pass in a neural network, the output of each layer is obtained by computing the input of the current layer along with the learned parameters of that layer. This process involves a chain rule-like calculation, where the derivative of the output with respect to the input of each layer is computed step by step. This is analogous to computing the derivative of an inner function with respect to its input, by first computing the derivative of the outer function. The output of each layer serves as the input for the subsequent layer, making the overall computation more efficient and streamlined.

(g)

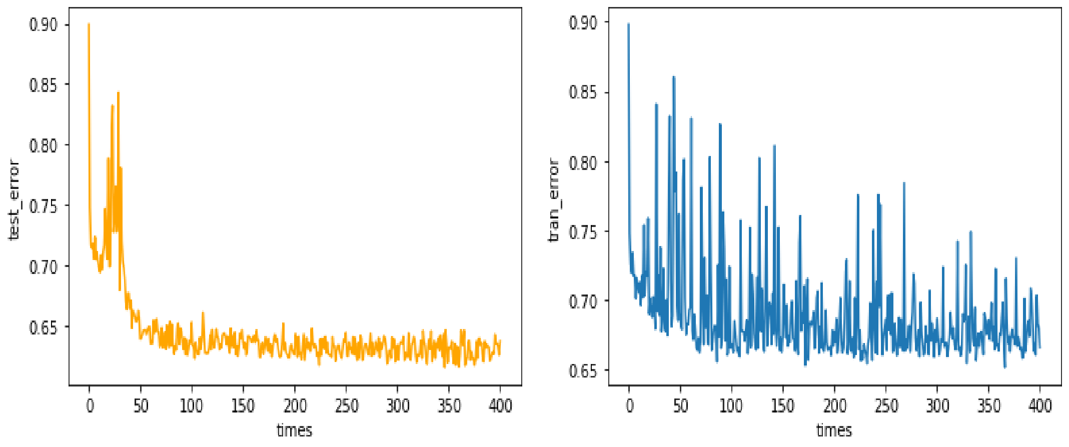
- image size 28x28
- 20 filter with size 5x5, stride 1, padding 0
- max-pooling size 2x2
- new image size after convolution will be $(28-2*2)$x$(28-2*2) = 24$x24
- since we have 20 filter, then the final size is 24x24x20
- after pooling, the image will shrink halve which is 12x12
- then the final size is 12x12x20

(h)

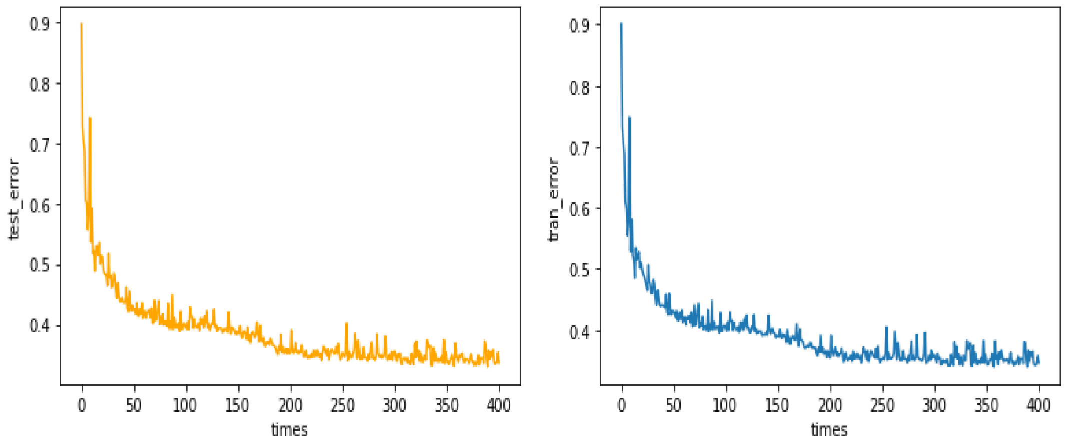
- since we want our final output is 50 channels
- we need $2.5 \approx 3$ filter in convolution
- after second convolution and pooling, our final size is 4x4
- then, our image size will be 8x8 before the pooling
- the filter size will be x: $8 = 12 - (x-1)$, which is 5x5

3 Res Net

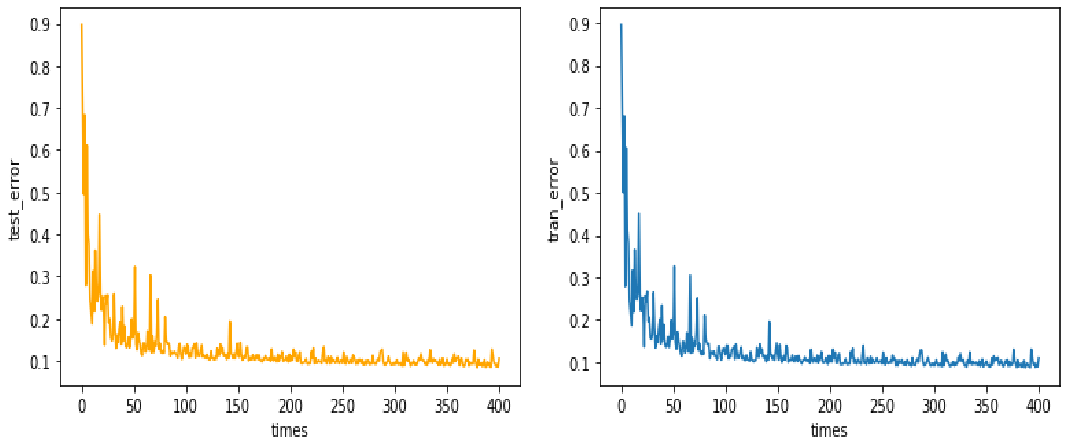
(c)the orange color is testLoader, the blue color is trainLoader



- **C=1**



- **C=2**

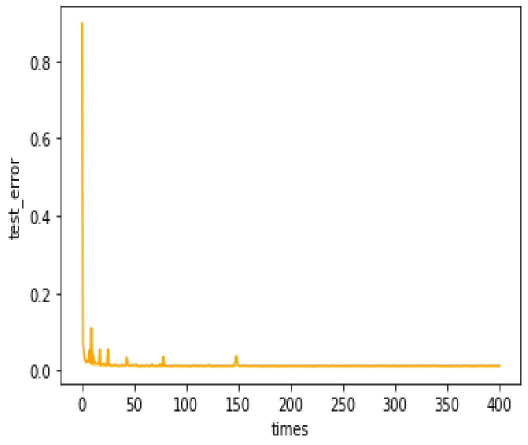


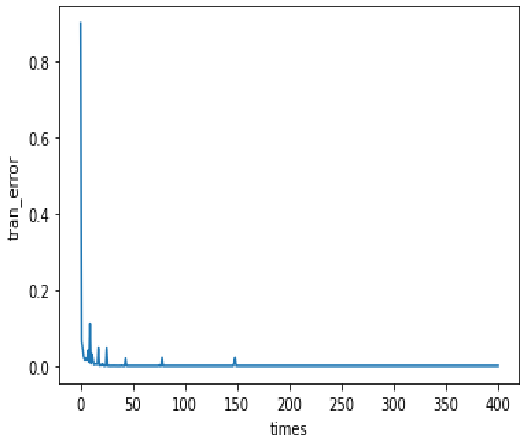
- **C=4**

- from graphs showing above, we can discover that when the number of channel(C) become larger, our training will be more accurate, and the graph will concave to 0 more aggressively.

(d)

- **C=64**





-

- In this part, we make the number of channel be 64, we can see the error is quickly near to zero in first 50 epoch, and the line is much smooth than previous graphs.