CPS 571 — HW 5

Shengxin Qian, sq16

1 Gradient Computation For Recursive Logistic Regression by Backpropagation

1.1 Gradient of 3-Layer Network

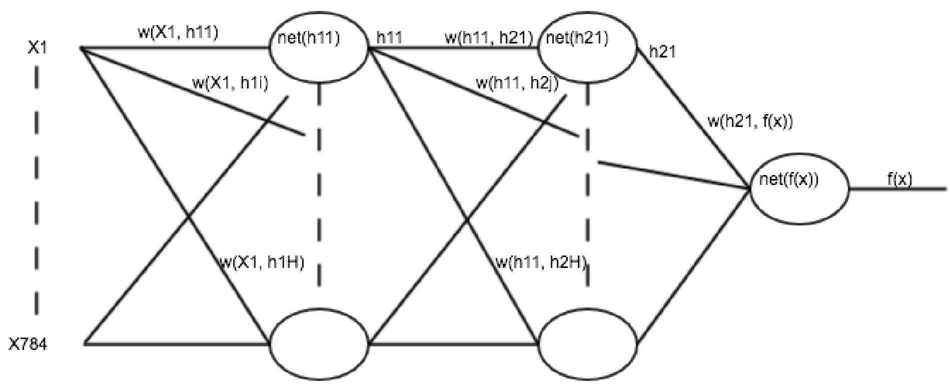

Figure 1: 3-layer neural network model

The 3-layer neural network required is shown as above. As one can see in Figure 1, the gradient $\frac{\partial L(\theta)}{\partial W_1}$ and $\frac{\partial L(\theta)}{\partial b_1}$ are derived as below.

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial W_{x_1,h_{11}}} &= \frac{\partial L(\theta)}{\partial h_{11}} \frac{\partial h_{11}}{\partial net_{h_{11}}} \frac{\partial net_{h_{11}}}{\partial W_{x_1,h_{11}}} \\
&= \frac{\partial L(\theta)}{\partial h_{11}}\ h_{11}(1-h_{11})x_1 \\
&= \sum_{i=1}^{H} \left[\frac{\partial L(\theta)}{\partial net_{h_{2i}}} w_{h11,h2i} \right] h_{11}(1-h_{11})x_1
\end{aligned} \tag{1}$$

$$
\begin{aligned}
\frac{\partial L(\theta)}{\partial net_{h_{2i}}} &= \frac{\partial L(\theta)}{\partial net_{f(x)}} \frac{\partial net_{f(x)}}{\partial h_{2i}} \frac{\partial h_{2i}}{\partial net_{h_{2i}}} \\
&= \frac{\partial L(\theta)}{\partial net_{f(x)}} w_{h_{2i},f(x)} h_{2i}(1-h_{2i}) \\
&= \frac{\partial L(\theta)}{\partial f(x)} \frac{\partial f(x)}{\partial net_{f(x)}} w_{h_{2i},f(x)} h_{2i}(1-h_{2i}) \\
&= \frac{\partial L(\theta)}{\partial f(x)} f(x)(1-f(x)) w_{h_{2i},f(x)} h_{2i}(1-h_{2i})
\end{aligned} \tag{2}
$$

$$
\frac{\partial L(\theta)}{\partial f(x)} = -\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1)\frac{1}{1-f(x)} \right] \tag{3}
$$

Overall,

$$
\begin{aligned}
\frac{\partial L(\theta)}{\partial W_{x_1,h_{11}}} =& \sum_{i=1}^{H} \left\{ -\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1)\frac{1}{1-f(x)} \right] \right\} \\
& f(x)(1-f(x)) w_{h_{2i},f(x)} h_{2i}(1-h_{2i}) w_{h11,h2i} h_{11}(1-h_{11}) x_1
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\frac{\partial L(\theta)}{\partial W_{x_1,h_{11}}} =& \sum_{i=1}^{H} \left\{ -\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1)\frac{1}{1-f(x)} \right] \right\} \\
& f(x)(1-f(x)) w_{h_{2i},f(x)} h_{2i}(1-h_{2i}) w_{h11,h2i} h_{11}(1-h_{11}) x_1
\end{aligned} \tag{5}
$$

Moreover,

$$
W_1 = \begin{bmatrix} W_{x_1,h_{11}} & \cdots & W_{x_1,h_{1H}} \\ \vdots & \ddots & \vdots \\ W_{x784,h_{11}} & \cdots & W_{x784,h_{1H}} \end{bmatrix} \tag{6}
$$

$$
\frac{\partial L(\theta)}{\partial W_1} = \begin{bmatrix} \frac{\partial L(\theta)}{\partial W_{x_1,h_{11}}} & \cdots & \frac{\partial L(\theta)}{\partial W_{x_1,h_{1H}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L(\theta)}{\partial W_{x_{784},h_{11}}} & \cdots & \frac{\partial L(\theta)}{\partial W_{x_{784},h_{1H}}} \end{bmatrix} \tag{7}
$$

Therefore, we can get the general equation of each element in the matrix,

$$
\begin{aligned}
\frac{\partial L(\theta)}{\partial W_{x_m,h_{1n}}} =& \sum_{i=1}^{H} \left\{ -\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1)\frac{1}{1-f(x)} \right] \right\} \\
& f(x)(1-f(x)) w_{h_{2i},f(x)} h_{2i}(1-h_{2i}) w_{h1n,h2i} h_{1n}(1-h_{1n}) x_m
\end{aligned} \tag{8}
$$

Similar to the derivation of $\frac{\partial L(\theta)}{\partial W_1}$, the first step of derivation of $\frac{\partial L(\theta)}{\partial b_1}$ is as below, the only difference is $\frac{\partial net_{h_{11}}}{\partial b_{h_{11}}} == 1$

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial b_{h_{11}}} &= \frac{\partial L(\theta)}{\partial h_{11}} \frac{\partial h_{11}}{\partial net_{h_{11}}} \frac{\partial net_{h_{11}}}{\partial b_{h_{11}}} \\
&= \frac{\partial L(\theta)}{\partial h_{11}} \; h_{11}(1 - h_{11}) \\
&= \sum_{i=1}^{H} \left[\frac{\partial L(\theta)}{\partial net_{h_{2i}}} w_{h11,h2i} \right] h_{11}(1 - h_{11})
\end{aligned} \tag{9}$$

Therefore, the general equation of each element in the vector $\frac{\partial L(\theta)}{\partial b_1}$ is,

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial b_{1n}} =& \sum_{i=1}^{H} \left\{ -\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1) \frac{1}{1 - f(x)} \right] \right\} \\
& f(x)(1 - f(x)) w_{h_{2i},f(x)} h_{2i}(1 - h_{2i}) w_{h1n,h2i} h_{1n}(1 - h_{1n})
\end{aligned} \tag{10}$$

1.2 Gradient of (L - 1)-Layer Network

As one can see in equation 1 and 2,

$$\frac{\partial L(\theta)}{\partial h_{1i_1}} = \sum_{i_2=1}^{H} \left[\frac{\partial L(\theta)}{\partial net_{h_{2i_2}}} w_{h1i_1,h2i_2} \right] \tag{11}$$

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial net_{h_{2i_2}}} &= \frac{\partial L(\theta)}{\partial h_{2i_2}} \frac{\partial h_{2i_2}}{\partial net_{h_{2i_2}}} \\
&= \sum_{i_3=1}^{H} \left[\frac{\partial L(\theta)}{\partial net_{h_{3i_3}}} w_{h2i_2,h3i_3} \right] h_{2i_2}(1 - h_{2i_2})
\end{aligned} \tag{12}$$

If we use the same rule when deriving the general formula of the entire $(L - 1)$ hidden layers network and name $h_{ti_t}(1 - h_{ti_t}) w_{h_{(t-1)i_{(t-1)}},h_{ti_t}}$ as δ_t (i_t is the index of summation at layer t), the general formula would be:

$$\begin{aligned}
&\frac{\partial L(\theta)}{\partial h_{1i_1}} = \sum_{i_2=1}^{H} \left[\sum_{i_3=1}^{H} \cdots \left[\sum_{i_{L-1}=1}^{H} \frac{\partial L(\theta)}{\partial f(x)} \delta_{f(x)} \delta_{L-1} \right] \delta_{L-2} \ldots \delta_2 \right] \\
&\delta_{f(x)} = f(x)(1 - f(x)) w_{h_{L-1} i_{L-1}, f(x)} \\
&\delta_t = h_{ti_t}(1 - h_{ti_t}) w_{h_{(t-1)} i_{(t-1)}, h_{ti_t}}
\end{aligned} \tag{13}$$

So, according to the equation 1 and equation 8, we can get the gradient of $\frac{\partial L(\theta)}{\partial W_{x_m,h_{1i_1}}}$,

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial W_{x_m,h_{1i_1}}} =& \sum_{i_2=1}^{H} \left[\sum_{i_3=1}^{H} \cdots \left[\sum_{i_{L-1}=1}^{H} \left[-\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1) \frac{1}{1 - f(x)} \right] \right] \delta_{f(x)} \delta_{L-1} \right] \delta_{L-2} \ldots \delta_2 \right] * \\
& h_{1i_1}(1 - h_{1i_1}) x_m
\end{aligned} \tag{14}$$

$$\frac{\partial L(\theta)}{\partial W_1} = \begin{bmatrix} \frac{\partial L(\theta)}{\partial W_{x_1,h_{11}}} & \cdots & \frac{\partial L(\theta)}{\partial W_{x_1,h_{1H}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L(\theta)}{\partial W_{x_{784},h_{11}}} & \cdots & \frac{\partial L(\theta)}{\partial W_{x_{784},h_{1H}}} \end{bmatrix} \tag{15}$$

Similar to $\frac{\partial L(\theta)}{\partial W_{x_m,h_{1i_1}}}$, the general formula of gradient of $\frac{\partial L(\theta)}{\partial b_{1i_1}}$(element i_1 of vector b_1) is,

$$\frac{\partial L(\theta)}{\partial b_{1i_1}} = \sum_{i_2=1}^{H} \left[\sum_{i_3=1}^{H} \cdots \left[\sum_{i_L=1}^{H} \left[-\sum_{j=1}^{N} \left[y_j \frac{1}{f(x)} + (y_j - 1) \frac{1}{1 - f(x)} \right] \right] \delta_{f(x)} \delta_L \right] \delta_{L-1} \ldots \delta_2 \right] * h_{1i_1}(1 - h_{1i_1}) \tag{16}$$

2 EM for Coin Toss

2.1 Estimation of θ

In this question, because we only know the number of the heads and tails of each sample, the distribution of the result of each sample x_i matches binomial distribution. Assuming x_i represent the number of heads in each sample. Therefore, $P(x_i \mid z = A, \theta) = P(x_i \mid z = A, \theta_A) = C_n^{x_i}\theta_A^{x_i}(1-\theta_A)^{(n-x_i)}$ and $P(x_i \mid z = B, \theta) = P(x_i \mid z = B, \theta_B) = C_n^{x_i}\theta_B^{x_i}(1-\theta_B)^{(n-x_i)}$. In addition to that, because we randomly choose the coins in each sample, $P(z = A \mid \theta) = P(z = B \mid \theta) = 1/2$. The EM algorithm used for coin toss is:

1. Estimation Step:

$$
\begin{aligned}
Q_i(z = A) &= P(z = A \mid x_i, \theta) \\
&= \frac{P(z = A, x_i \mid \theta)}{P(x_i \mid \theta)} \\
&= \frac{P(x_i \mid z = A, \theta) * P(z = A \mid \theta)}{P(x_i \mid \theta)} \\
&= \frac{P(x_i \mid z = A, \theta) * P(z = A \mid \theta)}{P(x_i \mid z = A, \theta) * P(z = A \mid \theta) + P(x_i \mid z = B, \theta) * P(z = B \mid \theta)} \\
&= \frac{P(x_i \mid z = A, \theta) * 1/2}{P(x_i \mid z = A, \theta) * 1/2 + P(x_i \mid z = B, \theta) * 1/2} = 1 - Q_i(z = B)
\end{aligned}
\tag{17}
$$

2. Maximization Step:

$$
\begin{aligned}
\frac{\partial L(\theta)}{\partial \theta_A} &= \left\{\sum_i \left[Q_i(z = A)\log\frac{P(x_i, z = A \mid \theta_A)}{Q_i(z = A)} + Q_i(z = B)\log\frac{P(x_i, z = B \mid \theta_B)}{Q_i(z = B)}\right]\right\}' \\
&= \left\{\sum_i Q_i(z = A)\log\frac{P(x_i \mid z = A, \theta_A) * 1/2}{Q_i(z = A)} + Q_i(z = B)\log\frac{P(x_i \mid z = B, \theta_B) * 1/2}{Q_i(z = B)}\right\}' \\
&= \sum_i Q_i(z = A)\frac{x_i(1 - \theta_A) - \theta_A(n - x_i)}{\theta_A(1 - \theta_A)} = 0
\end{aligned}
\tag{18}
$$

The maximum likelihood estimator of θ_A is

$$
\widehat{\theta_A} = \frac{\sum_i Q_i(z = A)x_i}{\sum_i Q_i(z = A)n} \tag{19}
$$

The equation of deriving $\widehat{\theta_B}$ is similar,

$$
\widehat{\theta_B} = \frac{\sum_i Q_i(z = B)x_i}{\sum_i Q_i(z = B)n} \tag{20}
$$

2.2 Implementation of EM algorithm for Coin Toss

One important thing is the initial θ estimation should not set as the same. In the simulation, I set the initial parameters as $\theta_A = 0.4$, $\theta_B = 0.2$. After 10 iterations, one result is [0.8, 0.34]. After 1000 iterations, one result is [0.8533, 0.3600].

3 K-means for Image Compression

Figure 2: 2-means recovered image

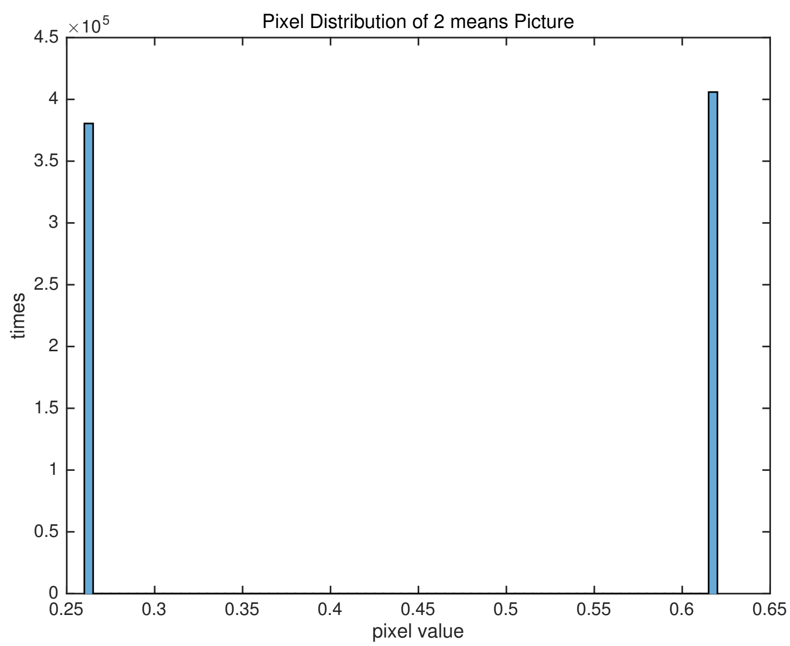

Figure 3: 2-means Pixel Value Distribution

Figure 4: 4-means recovered image

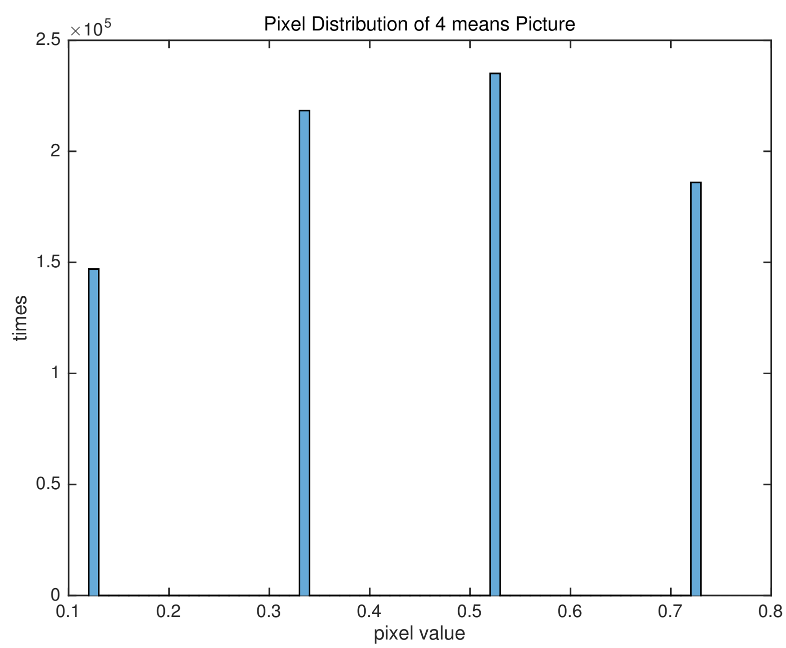

Figure 5: 4-means Pixel Value Distribution

Figure 6: 8-means recovered image

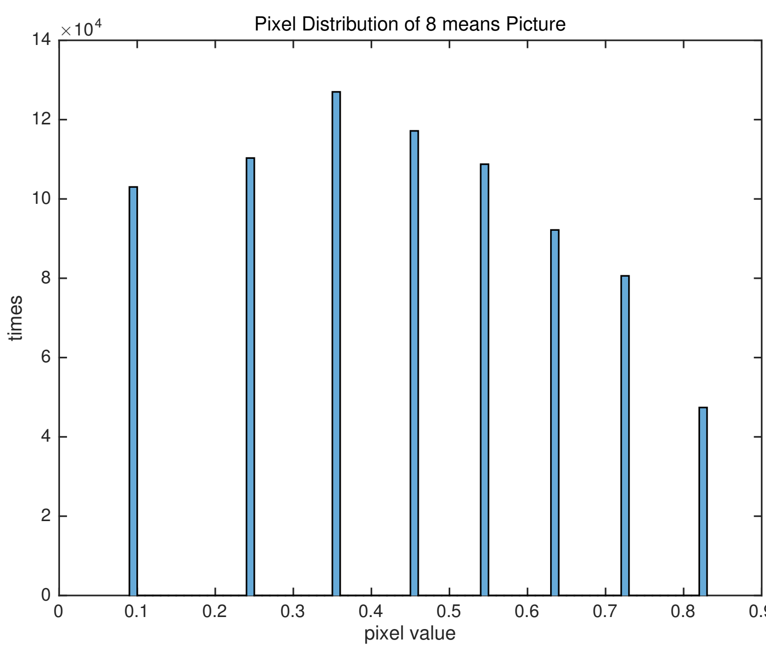

Figure 7: 8-means Pixel Value Distribution

Figure 8: original image

As we can see in the images above, after doing 10 iterations, the effect of compression is pretty good. Even with only 2-mean quantization, the contour is visible. The higher the number of means is, the more refine the graph is.