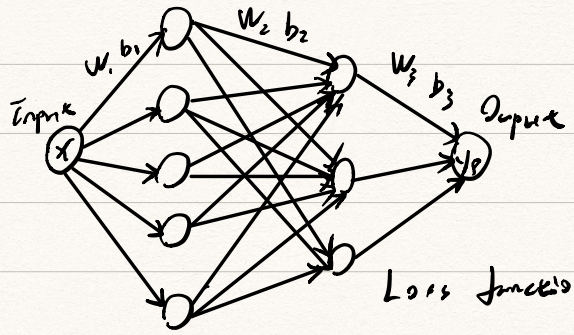
1. Backpropagation Algorithm for Neural Networks

(a)



Loss function: $\sum_{i=1}^{N} L(y_i, f(x_i | w, b))$

$= \sum_{i=1}^{N} (y_i - f(x_i | w, b))^2$

(b) $MSE(\hat{y}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$$\frac{\partial MSE(\hat{y})}{\partial \hat{y}} = \frac{1}{2n} \times 2(y_i - \hat{y}_i) \times (-1)$$

$$= \frac{(\hat{y}_i - y_i)}{n}$$

The code is included in the appendix.

(c) Linear function

$$\sigma_{linear}(z) = z$$

$$\frac{\partial \sigma_{linear}(z)}{\partial z} = 1$$

ReLU function

$$\sigma_{ReLU}(z) = \begin{cases} 0 & z < 0 \end{cases}$$

$$| z \quad \text{otherwise}$$

$$\frac{\partial \sigma_{ReLU}(z)}{\partial z} = \begin{cases} 0 & z \le 0 \\ 1 & \text{otherwise} \end{cases}$$

tanh function

$$\sigma_{tanh}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$\frac{\partial \sigma_{tanh}(z)}{\partial z} = \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})(e^z - e^{-z})}{(e^z + e^{-z})^2}$$

$$= 1 - \sigma_{tanh}^2(z)$$

The code is included in the appendix.

(d) The code is included in the appendix

$$\frac{\partial MSE}{\partial y_p} \times \frac{\partial y_p}{\partial z_i} \times \frac{\partial z_i}{\partial a_i} \times \frac{\partial a_i}{\partial z_{i-1}} \times \frac{\partial z_{i-1}}{\partial a_{i-1}}$$

(e) With the same batch size, the training error decreases as the increase of epoch. This is obvious because the number of iterations match my expec increases. However, with the same epoch, the training error increases with the increase of bach size. This maybe due to as the increase of batch not match size, large batch tend to converge sharp minimizers. my expec

The performance of three activation function is: tanh > ReLU > linear

The code is included in appendix.

(f) With 2 different activation function : tanh, ReLU. The MSE both decrease first and then increase as the increase of network width. The best width is around 8. This matches my expectation. This is the trade off between width and MSE.

2. Regularized and Kernel k-means

(a) 0

(b)

$$\lambda \mu_i + \sum_{x_j \in C_i} 2(\mu_i - x_j) = 0.$$

$$\lambda \mu_i + |C_i| \mu_i - \sum_{x_j \in C_i} x_j = 0$$

$$\mu_i = \frac{1}{|C_i| + \lambda} \sum_{x_j \in C_i} x_j$$

(c)

$$\min_{C_1 C_2 \cdots C_k} \sum_{i=1}^{k} \left(\|\mu_i\|_2 + \sum_{x_j \in C_i} \|x_j - \mu_i\|_2 \right)$$

(d)

$$\text{Class}(j) = \text{argmin}_k \| x_j - \mu_k \|^2$$

$$= \text{argmin}_k \left\| x_j - \frac{\sum_{l \in S_k} x_l}{|S_k|} \right\|^2$$

$$= \text{argmin}_k \; k(x_j, x_j) - \frac{2 \sum_{l \in S_k} k(x_j, x_l)}{|S_k|} + \frac{\sum_{l \in S_k} \sum_{l' \in S_k} k(x_l, x_{l'})}{|S_k|^2}$$

(e) I see there are multiple $k(x_1, x_2)$ through out the algorithm. In order to save

time, all kernels should not be calculated twice or more. So the idea is to precompute $K(x_i, x_j)$ for all possible combinations of i and j and store them for use in the algorithm. Where $i, j \in \{1, 2, \ldots, n\}$ and i can be equal to j.

3. Expectation Maximization (EM) Algorithm: in action!

(a) The initial guesses for 3 different algorithms are very different because the distance between different clusters are small.

After looking at the results of 3 algorithms. The EM performs the best, next is Kmeans, the worse is KNN

(b) After the increase of factor from 1 to 10, the distance between different clusters are much larger (The clusters are much obvious). Now the initial guesses are much closer to the truth. And the results show that the performances of the 3 algorithms are very close. They all perform very well.

4. One dimensional Mixture of Two Gaussians

(a) $$p_Z(z) = \begin{cases} \beta & z = 1 \\ 1-\beta & z = 2 \end{cases}$$

$$(X \mid Z = k) \sim N(\mu_k, \sigma_k^2)$$

$$\therefore \quad P_\theta(X=x_i, Z_i=1)$$
$$= P_\theta(X=x_i \mid Z_i=1)\, P_\theta(Z_i=1)$$
$$= \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} \times \beta = \frac{\beta}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}}$$

$$P_\theta(X=x_i, Z_i=2)$$
$$= P_\theta(X=x_i \mid Z_i=2)\, P_\theta(Z_i=2)$$
$$= \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}} \times (1-\beta) = \frac{1-\beta}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}}$$

$$\therefore \quad P_\theta(X=x_i) = \frac{\beta}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} + \frac{1-\beta}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}}$$

(b) $$\ell_\theta(x) = \sum_i \log \sum_{k=1}^{2} N(x_i \mid \mu_k, \sigma_k)\, P(z=k)$$

$$= \sum_i \log\left(\frac{\beta}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} + \frac{1-\beta}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}} \right)$$

It is obvious that this log likelihood function is very hard to take derivative. So it is very hard to set the derivative to 0 and get the closed form solution of MLE.

(c) $$\ell_\theta(x_i) = \log\left(\frac{\beta}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} + \frac{1-\beta}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}} \right)$$

According to Jensen's Inequality

$$\log P(x_i \mid \theta) = \log \sum P(x_i, z_i \mid \theta)$$

$$l_\theta(x_i) = \log p(x_i|\theta) = \log \sum_z p(x_i, z_i|\theta)$$

$$= \log \sum_z \frac{q(z_i|x_i,\theta)\, p(x_i, z_i|\theta)}{q(z_i|x_i,\theta)}$$

$$= \log E_q\left(\frac{p(x_i, z_i|\theta)}{q(z_i|x_i,\theta)}\right) \quad \text{(log is concave function)}$$

$$\therefore \geq E_q\left[\log\left(\frac{p(x_i, z_i|\theta)}{q(z_i|x_i,\theta)}\right)\right]$$

$$\text{when } q = p_\theta \Rightarrow l_\theta(x_i) \geq E_q\left[\log \frac{p_\theta(x = x_i, z = k)}{q_\theta(z_i = k | x = x_i)}\right]$$

(d) lower bound: $E_q\left[\log \frac{p_\theta(x = x_i, z_i = k)}{q_\theta(z_i = k | x = x_i)}\right]$

$$= -\sum_z q_\theta(z_i = k | x = x_i) \log\left[q_\theta(z_i = k | x = x_i)\right]$$

$$+ \sum_z q_\theta(z_i = k | x = x_i) \log\left[p_\theta(x = x_i, z_i = k)\right]$$

$$= H\left(q_\theta(z_i = k | x = x_i)\right) + E_q\left[L_c(x_i, z_i|\theta)\right]$$

$$\equiv F_i(q, \theta)$$

$\therefore$ In E-step: lower bound: $F_i(q, \theta^t)$

In M-step: lower bound: $F_i(q^{t+1}, \theta)$

$\therefore$ In E-step: $q^{t+1} = \arg\max_q F(q, \theta^t)$

In M-step: $\theta^{t+1} = \arg\max_\theta F(q^{t+1}, \theta^t)$

$\therefore$ Alternating between E-step and and M-step until

both try to maximize $F(q, \theta)$

(e) $p(z_i = k \mid x_i, \theta^t) = \frac{p(z_i, x_i \mid \theta^t)}{p(x_i \mid \theta^t)}$

$= \frac{p(z_i, x_i \mid \theta^t)}{\sum_{z_i} p(x_i, z_i \mid \theta^t)} = \frac{p(x_i \mid z_i, \theta^t)\, p(z_i \mid \theta^t)}{\sum_{z_i} p(x_i \mid z_i, \theta^t)\, p(z_i \mid \theta^t)}$

$= \frac{\alpha_k^t \, \mathcal{N}(x_i \mid \mu_k^t, \sigma_k^t)}{\sum_k \alpha_k^t \, \mathcal{N}(x_i \mid \mu_k^t, \sigma_k^t)}$

$\therefore q_{i,1}^{t+1} = \frac{\frac{\beta^t}{\sqrt{2\pi}\sigma_1^t} e^{-\frac{(x_i - \mu_1^t)^2}{2\sigma_1^{t2}}}}{\frac{\beta^t}{\sqrt{2\pi}\sigma_1^t} e^{-\frac{(x_i - \mu_1^t)^2}{2\sigma_1^{t2}}} + \frac{1-\beta^t}{\sqrt{2\pi}\sigma_2^t} e^{-\frac{(x_i - \mu_2^t)^2}{2\sigma_2^{t2}}}}$

$q_{i,2}^{t+1} = \frac{\frac{1-\beta^t}{\sqrt{2\pi}\sigma_2^t} e^{-\frac{(x_i - \mu_2^t)^2}{2\sigma_2^{t2}}}}{\frac{\beta^t}{\sqrt{2\pi}\sigma_1^t} e^{-\frac{(x_i - \mu_1^t)^2}{2\sigma_1^{t2}}} + \frac{1-\beta^t}{\sqrt{2\pi}\sigma_2^t} e^{-\frac{(x_i - \mu_2^t)^2}{2\sigma_2^{t2}}}}$

(f) $\mu_1^{t+1} = \frac{\sum_i q_{i,1}^{t+1} x_i}{\sum_i q_{i,1}^{t+1}}$

(g) MOG fits very well with EM. But k-means and soft k-means does not fit well with EM.

- k-means gives more flexibility than k-means.

MoG allows non-spherical clusters And different scales of covariance.

MoG makes explicit assumptions in the form of statistical distributions thus easier to generalize.