HW02-DL

عادل موحدیان ۴۰۰۱۰۲۰۷۴

سوال 1

$\sigma(x) = \dfrac{1}{1+e^{-x}}$

$E_{w,b} = -\sum_{n} [y_n \ln \hat{y}(x_n) + (1-y_n)\ln(1-\hat{y}(x_n))]$

$E_{w,b} \geq 0 \longrightarrow$ has a min because both $\ln < 0$

$y_i \in \{0,1\}$ true label

$\hat{y}_i \in [0\ 1]$ predicted

$\hat{y}(x_n) = \sigma(w^T x_n + b) = \dfrac{1}{1+e^{-(w^T x_n + b)}} = \sigma(z)$

$\dfrac{\partial E}{\partial w}$

$\longrightarrow \dfrac{\partial e_n}{\partial w} = \dfrac{\partial e_n}{\partial \hat{y}(x_n)} \cdot \dfrac{\partial \hat{y}(x_n)}{\partial (z_n)} \cdot \dfrac{\partial z_n}{\partial w}$

$\dfrac{\partial e_n}{\partial \hat{y}(x_n)} = \dfrac{-y_n}{\hat{y}(x_n)} + \dfrac{1-y_n}{1-\hat{y}(x_n)} = \dfrac{-y_n + \hat{y}(x_n)}{\hat{y}(x_n)(1-\hat{y}(x_n))}$

$\dfrac{\partial \hat{y}(x_n)}{\partial (z_n)} = \dfrac{\partial \sigma(z_n)}{\partial (z_n)} = \sigma(z_n)(1-\sigma(z_n)) = \hat{y}(x_n)(1-\hat{y}(x_n))$

$\dfrac{\partial z_n}{\partial w} = x_n$

$\longrightarrow \dfrac{\partial e_n}{\partial w} = \dfrac{\hat{y}(x_n) - y_n}{\hat{y}(x_n)(1-\hat{y}(x_n))} \times \hat{y}(x_n)(1-\hat{y}(x_n)) \times x_n$

$\longrightarrow \dfrac{\partial E_{w,b}}{\partial w} = \sum_{n} (\hat{y}(x_n) - y_n) x_n$

$\dfrac{\partial E}{\partial b} = \sum_{n} (\hat{y}(x_n) - y_n)$

we can prove that the $E_{w,b}$ function has a minimum because it has a lower bound and the loss function is convex : with GD we can find the local (global) minimum.

Using gradiant descent we can say:

$$w = w + \eta \sum_{n} (y_n - \hat{y}(x_n)) x_n \qquad \eta \text{ is learning rate}$$

$$b = b + \eta \sum_{n} (y_n - \hat{y}(x_n))$$

we compute $\hat{y}(x_n)$ for each sample using the current w and b.

then using GD we update w and b.

We repeat the process until the cost function $E_{w,b}$ converges to a minimum.

basicly we wouldn't see any differance between updated w,b and previous ones, so we can say we have a minimum point at this situation.

to prove we have a minimum we use Hessian matrix $\rightarrow$ $H = \begin{bmatrix} \frac{\partial^2 E}{\partial w^2} & \frac{\partial^2 E}{\partial w \partial b} \\ \frac{\partial^2 E}{\partial b \partial w} & \frac{\partial^2 E}{\partial b^2} \end{bmatrix}$ $|H| > 0$ ✓

each term in H is non-negative $\rightarrow$ H is PSD $\xrightarrow{\text{E has a min}}$ $\boxed{\text{E is convex}}$

$$\frac{\partial^2 E}{\partial w^2} = \sum_n \frac{\partial}{\partial w}((\hat{y}(x_n) - y_n)x_n) = \sum_n \hat{y}(x_n)(1-\hat{y}(x_n))x_n^2 \geq 0$$

$$\frac{\partial^2 E}{\partial b^2} = \sum_n \hat{y}(x_n)(1-\hat{y}(x_n)) \geq 0 \qquad \frac{\partial^2 E}{\partial w \partial b} = \sum_n \hat{y}(x_n)(1-\hat{y}(x_n))x_n$$

$\nabla E(w,b) = 0 \rightsquigarrow$ minimum

Covariate Shift: سوال ②

آ)

Covariate shift refers to the change in the input distribution to a neural network layer during training. In neural network, as the input propagates through each layer, the distribution of the activations (input values to each layer) can shift due to weight updates. This shift can slow down training, as each layer has to constantly adjust to the new distribution of inputs, which affects the learning process.

How BN solves covariate shift:

BN helps mitigate covariate shift by normalizing the input of each layer to have a mean of zero and a standard deviaton of one (or some other desired values controlled by learnable params). By doing this, BN stabilizes the distribution of layer inputs, making the training faster and more stable. Essentially, BN ensures that each layer receives inputs with a consistant distribution, reducing the impact of covariate shift.

How BN helps generalization:

ب)

BN adds a small amount of noise to the activations due to the random sampling of batches. this noise acts as a regularizer, similar to dropout, which helps prevent overfitting and improves the network's generalization ability. since each mini-batch is normalized independently BN introduces variations in the input seen by the model during training. this added noise can encourage the model to learn more robust features that generalize better to unseen data.

پ)

since $\hat{x}_i = x_i - \mu$ $\left(\mu = \frac{1}{n}\sum_{k=1}^{n} x_k\right)$ $\longrightarrow \frac{\partial \hat{x}_i}{\partial x_j} = \delta_{ij} - \frac{1}{n}$

for calculating $\frac{\partial L}{\partial x_j} \Rightarrow \frac{\partial L}{\partial x_j} = \sum_{i=1}^{n} \frac{\partial L}{\partial y_i} \cdot \frac{\partial y_i}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial x_j}$ $\hookrightarrow \begin{cases} \text{if } i=j \rightarrow \delta_{ij}=1 \\ \text{if } i\neq j \rightarrow \delta_{ij}=0 \end{cases}$

$\rightsquigarrow$ given $y_i = \gamma \hat{x}_i + \beta$, we have $\frac{\partial y_i}{\partial \hat{x}_i} = \gamma$. so $\boxed{\frac{\partial L}{\partial x_j} = \sum_{i=1}^{n} \frac{\partial L}{\partial y_i} \cdot \gamma \cdot \left(\delta_{ij} - \frac{1}{n}\right)}$

ت)

$n=1$ $\mu = x_1$ $\hat{x}_1 = x_1 - x_1 = 0$ the derivative $\frac{\partial L}{\partial x_1}$ would simply depend on the behavior of y_1, since there is no variation in a single sample case.

$n \rightarrow \infty$, the effect of subtracting the mean stabilizes across a large batch, so the normalization becomes more consistent.

سوال ③

آ)

$$\frac{\partial \hat{y}_k}{\partial z_i^{(2)}} \longrightarrow \hat{y}_i = \frac{e^{z_i^{(2)}}}{\sum_{j=1}^{k} e^{z_j^{(2)}}} \rightarrow \frac{\partial \hat{y}_k}{\partial z_i^{(2)}} = \begin{cases} i=k \rightarrow \boxed{\hat{y}_k(1-\hat{y}_k)} \\ i\neq k \rightarrow \boxed{-\hat{y}_k\hat{y}_i} \end{cases}$$

from chian rule

ب)

$$\frac{\partial L}{\partial z_i^{(2)}} \longrightarrow$$ y is one_hot, $y_k=1$, $y_i=0$ $k\neq i$

$y_k=1$

$y_i=0$

$i\neq k$

$$L = \sum_{i=1}^{K} y_i \log(\hat{y}_i) \xrightarrow{y_k=1} L = -\log(\hat{y}_k)$$

$$\longrightarrow \frac{\partial L}{\partial z_i^{(2)}} = \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_i^{(2)}}$$

$\longrightarrow$ for $i=k$ (the correct class): $\frac{\partial L}{\partial z_k^{(2)}} = -\frac{1}{\hat{y}_k} \cdot \hat{y}_k(1-\hat{y}_k) = \hat{y}_k - 1$

for $i\neq k$: $\frac{\partial L}{\partial z_i^{(2)}} = -\frac{1}{\hat{y}_k} \cdot (-\hat{y}_k\hat{y}_i) = \hat{y}_i$

So we have :

$$\frac{\partial L}{\partial z_i^{(2)}} = \begin{cases} \hat{y}_k - 1 & \text{if } i=k \\ \hat{y}_i & \text{if } i\neq k \end{cases}$$

پ)

$$\frac{\partial L}{\partial Z^{(2)}} = \hat{y} - y \quad , \frac{\partial L}{\partial a^{(1)}} = \frac{\partial L}{\partial z^{(2)}} \cdot \left(W^{(2)}\right)^T = (\hat{y} - y) \cdot \left(w^{(2)}\right)^T \qquad \frac{\partial L}{\partial w^{(1)}} ? \quad ($$ ب

dropout = binary mask $\rightarrow$ $M = \begin{cases} 1 & 1-P = .8 \\ 0 & P = .2 \end{cases}$ $\rightarrow \frac{\partial L}{\partial \hat{a}^{(1)}} = (\hat{y} - y) \cdot \left(w^{(2)}\right)^T \otimes M$

element wise

$$\frac{\partial \hat{a}_i^{(1)}}{\partial z_i^{(1)}} = \begin{cases} 1 & z_i^{(1)} > 0 \\ 0.1 & z_i^{(1)} < 0 \end{cases} \qquad \mathbb{I} \rightarrow \mathbb{I}(z^{(1)} > 0) z^{(1)} + 0.1\, \mathbb{I}(z^{(1)} \leq 0) z^{(1)}$$

we writedown leaky relu

A calculation: $\frac{\partial L}{\partial Z^{(1)}} = \frac{\partial L}{\partial \hat{a}^{(1)}} \otimes \text{Leaky relu}'(z^{(1)})$ chain rule

$$\longrightarrow \frac{\partial L}{\partial z^{(1)}} = \left(\left((\hat{y} - y) \cdot \left(w^{(2)}\right)^T\right) \otimes M\right) \otimes \text{leaky relu}'(z^{(1)})$$

$$\begin{cases} 1 & z_i^{(1)} > 0 \\ 0.1 & z_i^{(1)} \leq 0 \end{cases}$$

calculate gradiant with respect to $w^{(1)}$:

$$\frac{\partial L}{\partial w^{(1)}} = \frac{\partial L}{\partial z^{(1)}} \cdot x^T \longrightarrow \frac{\partial L}{\partial w^{(1)}} = \left(\left(\hat{y} - y\right) \cdot \left(w^{(2)}\right)^T\right) \otimes M \otimes \text{leakyrelu}'(z^{(1)})\big) \cdot x^T$$

سوال 4

$J_y = \left[\frac{\partial y}{\partial u} \quad \frac{\partial y}{\partial v} \quad \frac{\partial y}{\partial z} \right]$ → each component represents the rate of change of y with respect to variables u, v, and z is essentially the gradient vector:

$$J_y = \left[\frac{\partial y}{\partial u} \quad \frac{\partial y}{\partial v} \quad \frac{\partial y}{\partial z} \right]$$

each component represents the rate of change of y with respect to one of the variables. the jacobian captures this first-order information.

The Hessian matrix H_y is the square matrix of second-order partial derivatives of y.

$$H_y = \begin{bmatrix} \frac{\partial^2 y}{\partial u^2} & \frac{\partial^2 y}{\partial u \partial v} & \frac{\partial^2 y}{\partial u \partial z} \\ \frac{\partial^2 y}{\partial v \partial u} & \frac{\partial^2 y}{\partial v^2} & \frac{\partial^2 y}{\partial v \partial z} \\ \frac{\partial^2 y}{\partial z \partial u} & \frac{\partial^2 y}{\partial z \partial v} & \frac{\partial^2 y}{\partial z^2} \end{bmatrix}$$

each element H_{ij} represents the rate of change of the partial derivative with respect to one variable when another variable changes, which gives us information about the concavity or convexity of y in that direction.

the jacobian vector (or gradient) J_y represents the direction and rate of the changes in y with respect to the variables u, v and z. if we are looking at how the function $\psi(u,v,z) = y(u,v,z)$ changes locally. the jacobian provides the first-order approximation.

The Hessian matrix H_y provides the second-order information, capturing how the rate of changes itself varies with each pair of variables. Allows us to understand the curvature.

سوال 5

$$\frac{\partial j_1}{\partial w_i} = -\delta_i x_i \left(y_d - \sum_{k=1}^{n} \delta_k w_k x_k \right)$$

$$E\left[\frac{\partial j_1}{\partial w_i}\right] = E\left[-\delta_i x_i \left(y_d - \sum_{k=1}^{n} \delta_k w_k x_k\right)\right] = -x_i\left(\underset{1}{\underbrace{E[\delta_i]}} y_d - \sum_{k=1}^{n} w_k x_k E[\delta_i \delta_k]\right)$$

$$E[\delta_i \delta_k] = \sigma^2 + 1 \quad (i=k) \xrightarrow[E[\delta^2] = var(\delta_i) + E[\delta_i]^2]{} \text{ if } i \neq k \rightarrow E[\delta_i \delta_k] = \underset{1}{E[\delta_i]} \times \underset{1}{E[\delta_k]} = 1$$

$$\text{therefore} \rightarrow E\left[\frac{\partial j_1}{\partial w_i}\right] = -x_i\left(y_d - w_i x_i(\sigma^2+1) - \sum_{\substack{k=1 \\ k \neq i}}^{n} w_k x_k\right)$$

without regularization $j_1 = 0.5\left(y_d - \sum_{k=1}^{n} w_k x_k\right)^2$

$$\delta_k \sim \text{Normal}(1, \sigma^2) \xrightarrow[\text{gaussian DO applied}]{} j_{1,regul} = 0.5 E_\delta\left[\left(y_d - \sum_{k=1}^{n} \delta_k w_k x_k\right)^2\right]$$

$$= \underbrace{0.5\left(y_d - \sum_{k=1}^{n} w_k x_k\right)^2}_{\text{non regulized}} + \underbrace{0.5\sigma^2 \sum_{k=1}^{n} w_k^2 x_k^2}_{\text{regularization term}}$$

non-regularized target

$$j_1 = 0.5\left(y_d - \sum_{k=1}^{n} w_k x_k\right)^2$$

regularized target (gausian dropout):

$$j_{1,regul} = 0.5\left(y_d - \sum_{k=1}^{n} w_k x_k\right)^2 + 0.5\sigma^2 \sum_{k=1}^{n} w_k^2 x_k^2$$

$0.5\sigma^2 \sum_{k=1}^{n} w_k^2 x_k^2$: Penalty function for weights → decrease overfitting

→ smoothness during training because of random noise

:6 سوال

Newton : $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$

$$f(x) \approx \underbrace{f(\hat{x})}_{\to 0} + f'(\hat{x})(x-\hat{x}) + \frac{f''(\hat{x})}{2}(x-\hat{x})^2$$

taylor

error in k'th eteration : $e_k = x_k - \hat{x}$ $\longrightarrow f(x) \approx f'(\hat{x})\, e_k + \dfrac{f''(\hat{x})}{2} e_k^2$

$\xrightarrow{\text{calculation}}$ $x_{k+1} = x_k - e_k - \dfrac{f''(\hat{x})}{2f'(\hat{x})} e_k^2$

$$e_{k+1} = x_k - \hat{x} - e_k - \frac{f''(\hat{x})}{2f'(\hat{x})} e_k^2 = -\frac{f''(\hat{x})}{2f'(\hat{x})} e_k^2 \implies |e_{k+1}| \approx \overset{\left|\frac{f''(\hat{x})}{2f'(\hat{x})}\right|}{\overset{\uparrow}{c}}\, |e_k|^2$$

e_k decreases quadratically $\to$ newton method converges to x^*, witch is the optimal point of function.

سوال 7:

a) $\hat{y}_k = \dfrac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \rightarrow L(z, y) = -\sum_{k=1}^{K} y_k \log\left(\dfrac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}\right) = -\sum_{k=1}^{K} y_k \left(z_k - \log \sum_{j=1}^{K} e^{z_j}\right)$

$$\frac{\partial}{\partial z_i}\left(\sum_{k=1}^{K} y_k \log\left(\sum_{j=1}^{K} e^{z_j}\right)\right) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \underbrace{\sum_{k=1}^{K} y_k}_{y = \text{onehot } v} = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} = \hat{y}_i$$

$$\rightarrow \frac{\partial L}{\partial z_i} = -y_i + \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} = -y_i + \hat{y}_i \rightarrow \boxed{\nabla_z L = \hat{y} - y}$$

b) 1) $H_{ij} = \dfrac{\partial^2 L}{\partial z_i \partial z_j} = \dfrac{\partial}{\partial z_j}(\hat{y}_i - y_i) = \dfrac{\partial}{\partial z_j}\left(\dfrac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}\right) = \dfrac{e^{z_i} \cdot \delta_{ij} \sum_{k=1}^{K} e^{z_k} - e^{z_i} \cdot e^{z_j}}{\left(\sum_{k=1}^{K} e^{z_k}\right)^2}$

where δ_{ij} is 1 if $i=j$ else is 0 (as previously said). so $H_{ij} = \hat{y}_i (\delta_{ij} - \hat{y}_j)$

$H = \text{diagonal}(\hat{y}) - \hat{y}\hat{y}^T$

2) $x \in \mathbb{R}^K \rightarrow x^T H x = x^T D(\hat{y}) x - x^T \hat{y} \hat{y}^T x = \underbrace{\sum_{i=1}^{K} \hat{y}_i x_i^2}_{\text{non-negative}} - \underbrace{\left(\sum_{i=1}^{K} \hat{y}_i x_i\right)^2}_{(x^T \hat{y})^2}$

So we just need to say $(x^T \hat{y})^2 \leq \left(\sum_{i=1}^{K} \hat{y}_i\right)\left(\sum_{i=1}^{K} x_i^2 \hat{y}_i\right) = \sum_{i=1}^{K} \hat{y}_i x_i^2$

کوشی شوارتز

$\longrightarrow x^T H x \geq 0$ for all $x \in \mathbb{R}^k \rightarrow \boxed{H \text{ is PSD}}$

c) As we concluded H is PSD and therefore its function is convex with respect to z and has an optimal point.

CamScanner