Deep Learning – Dr. Fatemizadeh

Parnian Taheri – 99106352

HW 2

Fall 2024

Repository Link:

Question 1:

The cost function E(w,b) is a cross-entropy loss function for a single neuron with a sigmoid activation. Since the sigmoid function is differentiable and the cross-entropy loss is convex with respect to w and b, this guarantees that the cost function E(w,b) has a unique minimum.

$$\frac{\partial E}{\partial b, w} = \frac{\partial E}{\partial \hat{y}(x_n)} \frac{\partial \hat{y}(x_n)}{\partial b, w} = \sum_n \frac{y_n}{\hat{y}(x_n)} - \frac{1 - y_n}{1 - \hat{y}(x_n)} = 0 \to \hat{y}(x) = \sum_n y_n \, ,$$

$$\hat{y}(x) = \sigma(wx + b), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Update Rule:

$$\frac{\partial \mathrm{E}}{\partial \mathrm{w}} = \sum_{n=1:N} (\hat{y}(x_n) - y_n) x_n$$

$$\frac{\partial \mathrm{E}}{\partial \mathrm{b}} = \sum_{n=1:N} (\hat{y}(x_n) - y_n)$$

$$\to w_{n+1} = w_n - \alpha \frac{\partial \mathrm{E}}{\partial \mathrm{w}_n}, \qquad b_{n+1} = b_n - \alpha \frac{\partial \mathrm{E}}{\partial \mathrm{b}_n},$$

Question 2:

a)

Covariate Shift in neural networks refers to changes in the distribution of input data that a model's layers receive as training progresses. As each layer's inputs change due to the updates of previous layers, the network may experience instability, requiring longer training times to converge.

Batch Normalization addresses this by normalizing the inputs of each layer within each mini-batch. Specifically, BN normalizes each feature so it has a mean of zero and a variance of one within the mini-batch, followed by a learnable scaling and shifting. This normalization reduces the impact of covariate shift, as the layers have a stable distribution of inputs during training, allowing the network to train faster and more reliably.

b)

Batch Normalization aids in **generalization** by adding a regularizing effect. Since BN computes mean and variance within a mini-batch, each mini-batch has slight variations in normalization parameters. This randomization acts like noise, which makes the model more robust to input changes, similar to other regularization techniques like dropout. As a result, the model avoids overfitting on training data and generalizes better to unseen data.

c)

$$\frac{\partial \mathrm{L}}{\partial \mathrm{x}_i} = \sum_{j=1}^{n} \frac{\partial \mathrm{L}}{\partial \mathrm{y}_j} \frac{\partial \mathrm{y}_j}{\partial \mathrm{x}_i}, \qquad y_i = \gamma \left(x_i - \frac{1}{n} \sum_{j=1}^{n} x_j \right) + \beta$$

$$\rightarrow \frac{\partial \mathrm{L}}{\partial \mathrm{x}_i} = f(x) = \begin{cases} \gamma \left(1 - \frac{1}{n}\right), & i = j \\ \gamma \left(-\frac{1}{n}\right), & i \neq j \end{cases}$$

d)

If n=1:

$$\frac{\partial \mathrm{L}}{\partial \mathrm{x}_i} = f(x) = \begin{cases} \gamma \left(1 - \frac{1}{n}\right) = 0, & i = j \\ \gamma \left(-\frac{1}{n}\right) = -\gamma, & i \neq j \end{cases}$$

If n = ∞:

$$\frac{\partial \mathrm{L}}{\partial \mathrm{x}_i} = f(x) = \begin{cases} \gamma \left(1 - \frac{1}{n}\right) = \gamma, & i = j \\ \gamma \left(-\frac{1}{n}\right) = 0, & i \neq j \end{cases}$$

Therefore:

With only one input, normalization does not alter x_1, therefore, $y_1 = \beta$ and does not depend on x_1, thus, $\frac{\partial \mathrm{L}}{\partial \mathrm{x}_1} = 0$.

As $n \rightarrow \infty$, the influence of each x_i on the batch mean μ becomes negligible, consequently, $\frac{\partial \mathrm{L}}{\partial \mathrm{x}_i} \approx \frac{\partial \mathrm{L}}{\partial \mathrm{y}_i}$

Question 3:

a)

$$\frac{\partial \hat{\mathrm{y}}_k}{\partial \mathrm{z}_i^{(2)}} = ?$$

$$\hat{\mathrm{y}}_k = \frac{e^{z_i^{(2)}}}{\sum_{j=1}^{K} e^{z_j^{(2)}}}$$

If $i = k$:

$$\frac{\partial \hat{\mathrm{y}}_k}{\partial \mathrm{z}_i^{(2)}} = \frac{e^{z_i^{(2)}} \sum_{j=1}^{K} e^{z_j^{(2)}} - e^{z_i^{(2)}} . e^{z_i^{(2)}}}{\left(\sum_{j=1}^{K} e^{z_j^{(2)}}\right)^2} = \hat{\mathrm{y}}_k(1 - \hat{\mathrm{y}}_k)$$

If $i \neq k$:

$$\frac{\partial \hat{\mathrm{y}}_k}{\partial \mathrm{z}_i^{(2)}} = \frac{-e^{z_i^{(2)}} . e^{z_k^{(2)}}}{\left(\sum_{j=1}^{K} e^{z_j^{(2)}}\right)^2} = -\hat{\mathrm{y}}_k \hat{\mathrm{y}}_i$$

$$\rightarrow \boldsymbol{f(x)} = \begin{cases} \boldsymbol{\hat{\mathrm{y}}_k(1 - \hat{\mathrm{y}}_k),} & \boldsymbol{i = k} \\ \boldsymbol{-\hat{\mathrm{y}}_k y_i,} & \boldsymbol{i \neq k} \end{cases}$$

b) Assume that the k-th element of y is 1 (i.e. $y_k = 1$) and all other elements are zero:

$$L = -y_k \log \hat{y}_k = -\log \hat{y}_k, \qquad y_i = \frac{e^{z_i^{(2)}}}{\sum_{j=1}^{K} e^{z_j^{(2)}}}$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{z}_i^{(2)}} = \frac{\partial \mathrm{L}}{\partial \hat{\mathrm{y}}_i} \frac{\partial \hat{\mathrm{y}}_i}{\partial \mathrm{z}_j^{(2)}}$$

$$\frac{\partial \mathrm{L}}{\partial \hat{\mathrm{y}}_i} = \begin{cases} -\frac{1}{\hat{y}_k}, & i = k \\ 0, & i \neq k \end{cases}$$

$$\rightarrow \frac{\boldsymbol{\partial L}}{\boldsymbol{\partial z_i^{(2)}}} = \boldsymbol{\hat{y}_k - 1 = \hat{y}_i - y_i}$$

c)

$$\frac{\partial \mathrm{L}}{\partial \mathrm{W}^{(1)}} = ?$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{W}^{(1)}} = \frac{\partial \mathrm{L}}{\partial \mathrm{z}^{(2)}} * \frac{\partial \mathrm{z}^{(2)}}{\partial \mathrm{a}^{(1)}} * \frac{\partial \mathrm{a}^{(1)}}{\partial \mathrm{z}^{(1)}} * \frac{\partial \mathrm{z}^{(1)}}{\partial \mathrm{W}^{(1)}}$$

$$\frac{\partial \mathrm{z}^{(2)}}{\partial \mathrm{a}^{(1)}} = W^{(2)}$$

$$g(z(1)) = \begin{cases} \mathrm{z}^{(1)}, & \mathrm{z}^{(1)} > 0 \\ 0.01\mathrm{z}^{(1)}, & \mathrm{z}^{(1)} \le 0 \end{cases} \rightarrow \frac{\partial \mathrm{a}^{(1)}}{\partial \mathrm{z}^{(1)}} = g'(z(1))$$

$$\frac{\partial \mathrm{z}^{(1)}}{\partial \mathrm{W}^{(1)}} = x$$

$$\rightarrow \frac{\boldsymbol{\partial L}}{\boldsymbol{\partial W^{(1)}}} = \left(\left(\boldsymbol{W}^{(2)}\right)^{\boldsymbol{T}} (\boldsymbol{\hat{y}_i - y_i}) \odot \boldsymbol{g'(z(1))}\right) . \boldsymbol{x^T}$$

Which $\odot$ is the element-wise (Hadamard) product.

Question 4:

The gradient vector ∇y is a vector of first derivatives. The Jacobian of a vector function is the matrix of all its first-order partial derivatives. When we compute the second-order partial derivatives (i.e., the Hessian matrix), we are essentially finding the Jacobian of the gradient.

The gradient of y(u, v, z):

$$\nabla y = \left(\frac{\partial y}{\partial u}, \frac{\partial y}{\partial v}, \frac{\partial y}{\partial z}\right)$$

The jacobian of the gradient:

$$H = \begin{pmatrix} \frac{\partial^2 y}{\partial u^2}, \frac{\partial^2 y}{\partial u \partial v}, \frac{\partial^2 y}{\partial u \partial z} \\ \frac{\partial^2 y}{\partial v \partial u}, \frac{\partial^2 y}{\partial v^2}, \frac{\partial^2 y}{\partial v \partial z} \\ \frac{\partial^2 y}{\partial z \partial u}, \frac{\partial^2 y}{\partial z \partial v}, \frac{\partial^2 y}{\partial z^2} \end{pmatrix}$$

Which is the Hessian matrix.

Question 5:

$$J_1 = 0.5\left(y_d - \sum_{k=1}^{n} \delta_k W_k x_k\right)^2$$

$$E\left[\frac{\partial J_1}{\partial W_i}\right] = ?$$

$$\frac{\partial J_1}{\partial W_i} = -\delta_i x_i \left(y_d - \sum_{k=1}^{n} \delta_k W_k x_k\right)$$

$$\rightarrow E\left[\frac{\partial J_1}{\partial W_i}\right] = -x_i \left(E\left[\delta_i \left(y_d - \sum_{k=1}^{n} \delta_k W_k x_k\right)\right]\right) = -x_i \left(E[\delta_i] y_d - \sum_{k=1}^{n} E[\delta_i \delta_k] W_k x_k\right)$$

$E[\delta_i] = 1$

$E[\delta_i \delta_k] = 1 + \sigma^2$ when i = k (since Var(δ_k)= σ^2

$E[\delta_i \delta_k] = 1$ when i ≠ k (since the variables are independent and both have mean 1)

$$\rightarrow E\left[\frac{\partial J_1}{\partial W_i}\right] = -x_i \left(y_d - W_i x_i (1 + \sigma^2) - \sum_{k \neq \mathrm{i}} W_k x_k\right)$$

Yes, we can interpret this form of Gaussian multiplicative dropout as a form of regularization. Because by applying Gaussian dropout, we're adding stochastic noise to each weight W_k during training, which has an effect similar to regularization.

Non-Regularized objective function:

$$J_{Non-Reg} = 0.5\left(y_d - \sum_{k=1}^{n} W_k x_k\right)^2$$

Question 6:

Newton's method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$f(x) = f(x^*) + f'(x^*)(x - x^*) + \cdots , \qquad f(x^*) = 0$$

$$\rightarrow f(x) \approx f'(x^*)(x - x^*)$$

$$\rightarrow x_{n+1} = x_n - \frac{f'(x^*)(x_n - x^*)}{f'(x^*)} = x_n - (x_n - x^*) = x^*$$

This shows that x_{n+1} converges to x^*.

Since f(x)=g′(x), finding the root of f(x)=0 using Newton's method effectively finds the critical point x^* of g(x), which is the optimal point of g(x). Newton's method is therefore efficient (quadratic convergence) in finding x^*, as it leverages both f(x) and f′(x) to achieve rapid convergence.

Question 7:

a)

Assume $y_k = 1$ and the rest of $y_i s$ are zero:

$$\frac{\partial L}{\partial z_i} = \frac{\partial L}{\partial \hat{y}_k}\frac{\partial \hat{y}_k}{\partial z_i} = -\frac{y_k}{\hat{y}_k}\frac{\partial \hat{y}_k}{\partial z_i}$$

If k = i:

$$\frac{\partial \hat{y}_k}{\partial z_i} = \hat{y}_k(1 - \hat{y}_k)$$

$$\rightarrow \frac{\partial L}{\partial z} = \hat{y} - y$$

b)

(1) Hessian:

$$\frac{\partial^2 L}{\partial z_i \partial z_j} = \frac{\partial \hat{y}_i}{\partial z_j} - \frac{\partial y_i}{\partial z_j} = \frac{\partial \hat{y}_i}{\partial z_j}$$

$$\rightarrow H_{ij} = \begin{cases} \hat{y}_i(1-\hat{y}_i), & i = j \\ -\hat{y}_i\hat{y}_j, & i \neq j \end{cases}$$

$$\rightarrow H = diag(\hat{y}) - \hat{y}\hat{y}^T$$

(2) Positive semi-definite:

$$v^\top H v = v^\top diag(\hat{y}) v - v^\top \hat{y}\hat{y}^\top v$$

$$= \sum_{i=1}^{k} v_i^2 \hat{y}_i - \left(\sum_{i=1}^{k} v_i \hat{y}_i\right)^2$$

since $0 \leq \hat{y}_i \leq 1$, both terms are positive and the first term is larger than the second term.

c)

Since the Hessian matrix of the cross-entropy loss function L(z,y) is positive semi-definite, we can conclude that the cross-entropy loss function is **convex** with respect to z.