Student Name: Mingqian Liao
Student ID: 3038745426

Deep Neural Networks(EECS182)

Homework0

---

1. Surveys

   I have submitted the surveys, my SID is 3038745426.

2. Course Policies

   (a) Yes. (b) No. (c) No. (d) Yes. (e) Yes.

3. Gradient Descent Doesn't Go Nuts with Ill-Conditioning

   The gradient-descent update for $t > 0$ is:

$$\begin{aligned} w_t &= w_{t-1} - \eta(F^T(Fw_{t-1} - y)) \\ &= (I - \eta F^T F)w_{t-1} + \eta F^T y \end{aligned}$$

   When the gradient descent cannot diverge, the eigenvalues of $(I - \eta F^T F)$ are smaller than 1.

   Combining with the Absolute value inequalities, we can derive that

$$\|w_t\|_2 \leq \|(I - \eta F^T F)\| \|w_{t-1}\|_2 + \|\eta F^T\| \|y\|_2$$

   And since when $n = d$, the singular value of feature matrix $F \in \mathbb{R}^{n \times n}$ is not greater than α, $\|\eta F^T\| \leq \eta\alpha$

   Since $I - \eta F^T F$ is a square matrix, it can be decomposed to $U\Sigma V^*$, where U and V^* are both orthogonal matrices. So that $\|I - \eta F^T F w_{t-1}\|_2 = \|U\Sigma V^* x w_{t-1}\| = \|\Sigma w_{t-1}\| \leq \|w_{t-1}\|$.

   Therefore,

$$\|w_t\|_2 \leq \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$$

4. Regularization from the Augmentation Perspective

   We can derive that,

$$\hat{X} = \begin{bmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_n^T \\ \gamma_1^T \\ \gamma_2^T \\ \cdots \\ \gamma_d^T \end{bmatrix} \in \mathbb{R}^{(n+d)\times d}, \hat{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \in \mathbb{R}^{n+d}$$

$$\hat{X}^T y = \begin{bmatrix} X^T & \Gamma^T \end{bmatrix} \begin{bmatrix} X \\ 0_d \end{bmatrix} = X^T X$$

$$\hat{X}^T \hat{X} = \begin{bmatrix} X^T & \Gamma^T \end{bmatrix} \begin{bmatrix} X \\ \Gamma \end{bmatrix}$$

Since the X and Γ are both square matrix, the result of $\begin{bmatrix} X^T & \Gamma^T \end{bmatrix} \begin{bmatrix} X \\ \Gamma \end{bmatrix}$ is $X^T X + \Gamma^T \Gamma = X^T X + \Sigma^{-1}$

To find the $\hat{w}$ to minimize the $\|\hat{y} - \hat{X}w\|_2^2$, it is known from the OLS solution that the following formula holds

$$\hat{w} = (\hat{X}^T X)^{-1} X^T y = (X^T X + \Sigma^{-1})^{-1} X^T y$$

which is the same as (2)

5. Vector Calculus Review

According to the fully differential equations, we know that for a scalar f and a $m * n$ matrix X, and since in the question, the vector derivatives of a scalar are expressed as a row vector, we have $df = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial f}{\partial X_{ij}} dX_{ij} = tr(\frac{\partial f}{\partial x}) dX$.

(a)

Let $f = x^T c$, so that

$$df = d(x^T c) = dx^T c = tr(dx^T c) = tr(c^T dx^T)$$

Therefore, $\frac{\partial f}{\partial x} = c^T$

(b)

Let $f = \|x\|_2^2 = x^T x$, $df = d(x^T x) = dx^T x + x^T dx = tr(dx^T x + x^T dx) = tr(dx^T x) + tr(x^T dx) = tr(x^T dx) + tr(x^T dx) = tr(2x^T dx)$, therefore, $\frac{\partial f}{\partial x} = 2x^T$

(c)

Let $f = Ax$, $df = d(Ax) = Adx = tr(Adx)$, therefore, $\frac{\partial f}{\partial x} = A$

(d)

Let $f = x^T Ax$, $df = dx^T Ax + x^T Adx = tr(dx^T Ax + x^T Adx) = tr(dx^T Ax) + tr(x^T Adx) = tr((Ax)^T dx) + tr(x^T Adx) = tr(x^T (A + A^T))$, therefore, $\frac{\partial f}{\partial x} = x^T (A + A^T)$

(e)

When $A = A^T$, the previous derivative equal to $2x^T A$

6. ReLU Elbow Update under SGD

(a)

(i)

The location of elbow is the point that make $wx + b < 0$ change to $wx + b > 0$,which is $-\frac{b}{w}$

(ii)

$$l = \frac{1}{2}(\phi(x) - y)^T (\phi(x) - y)$$

$$dl = \frac{1}{2}[d(\phi(x) - y)^T (\phi(x) - y) + (\phi(x) - y)^T d(\phi(x) - y)]$$

$$= \frac{1}{2}[tr(d(\phi(x) - y)^T (\phi(x) - y)) + tr(\phi(x) - y)^T d(\phi(x))]$$

$$= \frac{1}{2}[tr((\phi(x) - y)^T d(\phi(x) - y)) + tr(\phi(x) - y)^T d(\phi(x) - y)]$$

$$= (\phi(x) - y)^T d(\phi(x) - y)$$

so that

$$\frac{dl}{d\phi} = \begin{cases} (\phi(x) - y)^T & wx + b > 0 \\ 0 & else \end{cases}$$

(iii)

From (ii), we know that

$$dl = (\phi(x) - y)^T d\phi(x) = (\phi(x) - y)^T d(wx + b) = (\phi(x) - y)^T x dw$$

Therefore

$$\frac{\partial l}{\partial w} = \begin{cases} x^T(\phi(x) - y) & wx + b > 0 \\ 0 & else \end{cases}$$

(iv)

From (ii), we know that

$$dl = (\phi(x) - y)^T d\phi(x) = (\phi(x) - y)^T d(wx + b) = (\phi(x) - y)^T db$$

Therefore

$$\frac{\partial l}{\partial b} = \begin{cases} \phi(x) - y & wx + b > 0 \\ 0 & else \end{cases}$$

(b)

The gradient descent update formula of w and b is as below.

$$w_{t+1} \longleftarrow w_t - \lambda \frac{\partial l}{\partial w}$$

$$b_{t+1} \longleftarrow b_t - \lambda \frac{\partial l}{\partial b}$$

where the λ is the step size.

(i)

When $\phi(x) = 0$, $\frac{\partial l}{\partial w} = 0$, $\frac{\partial l}{\partial b} = 0$, $w_{t+1} \longleftarrow w_t$, $b_{t+1} \longleftarrow b_t$. The elbow and the slope will not change. The image is shown as figure1.

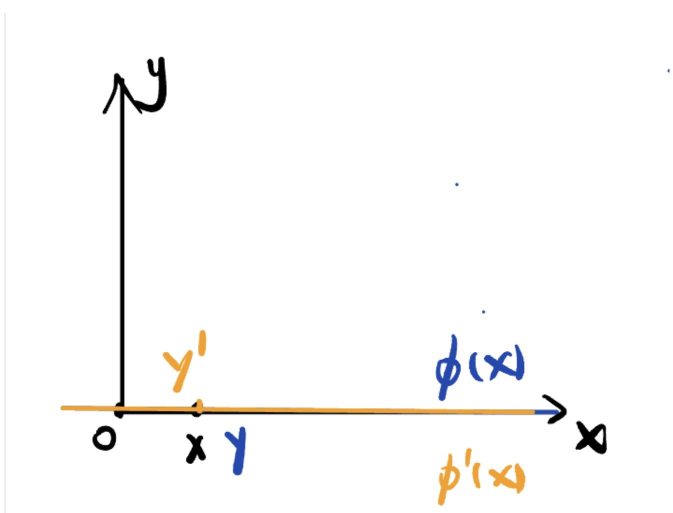

Figure 1: The elbow and slope will not change during updating.

(ii)

When $w > 0, x > 0, \phi(x) > 0$, $\frac{\partial l}{\partial w} = x^T(\phi(x) - y)$, $\frac{\partial l}{\partial b} = \phi(x) - y$, $w_{t+1} \longleftarrow w_t - \lambda x^T(\phi(x) - y) = w_t - \lambda x^T$, $\quad b_{t+1} \longleftarrow b_t - \lambda(\phi(x) - y) = b_t - \lambda$.

The slope will decrease, and the b will decrease, too. In figure 2, we can see that the elbow moves left during update. The corresponding y decreases.

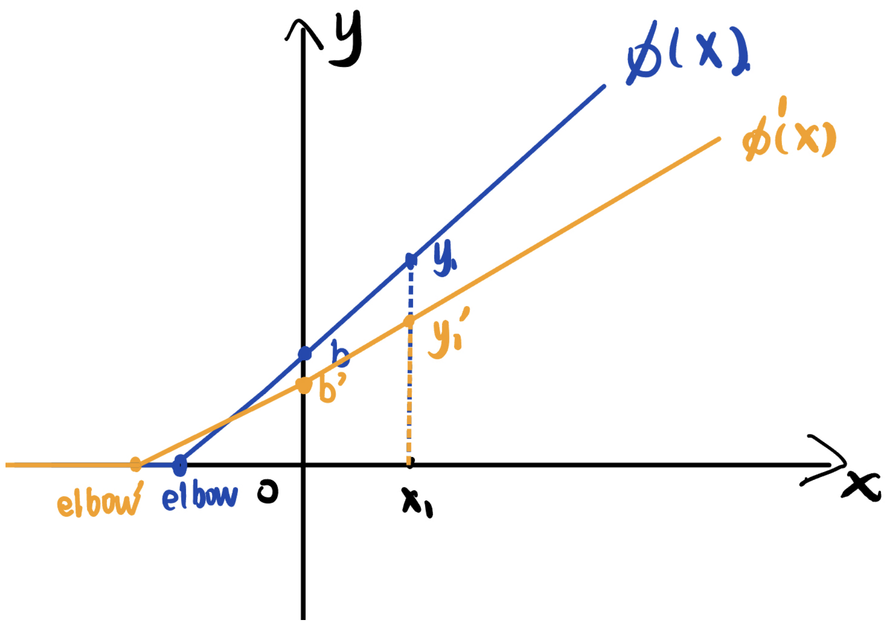

Figure 2: The elbow will move left and slope will decrease during updating.

(iii)

When $w > 0, x < 0, \phi(x) > 0$, $\frac{\partial l}{\partial w} = x^T(\phi(x) - y)$, $\frac{\partial l}{\partial b} = \phi(x) - y$, $w_{t+1} \longleftarrow w_t - \lambda x^T(\phi(x) - y) = w_t - \lambda x^T$, $\quad b_{t+1} \longleftarrow b_t - \lambda(\phi(x) - y) = b_t - \lambda$.

The slope will increase while the b will decrease. In figure 3, we can see that the elbow moves right during update. The corresponding y decreases.

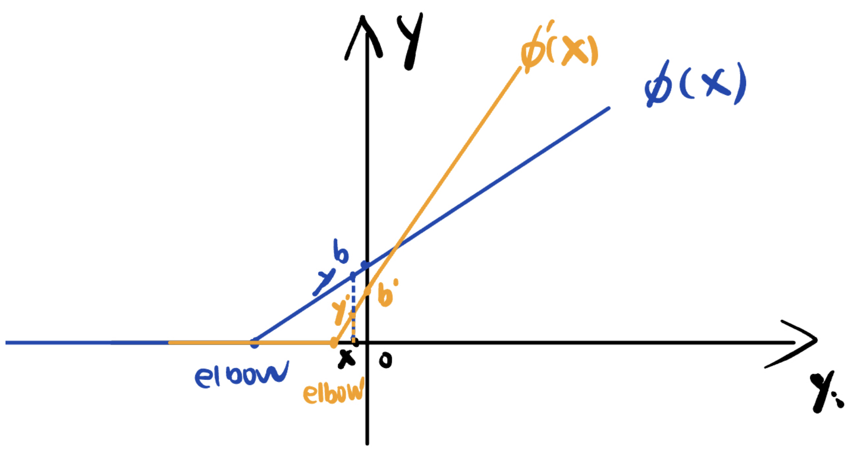

Figure 3: The elbow will move right and slope will decrease during updating.

(iv)

When $w < 0, x > 0, \phi(x) > 0$, $\frac{\partial l}{\partial w} = x^T(\phi(x) - y)$, $\frac{\partial l}{\partial b} = \phi(x) - y$, $w_{t+1} \longleftarrow w_t - \lambda x^T(\phi(x) - y) = w_t - \lambda x^T$, $\quad b_{t+1} \longleftarrow b_t - \lambda(\phi(x) - y) = b_t - \lambda$.

The slope the b will both decrease. In figure 4, we can see that the elbow moves left during update. The corresponding y decreases.

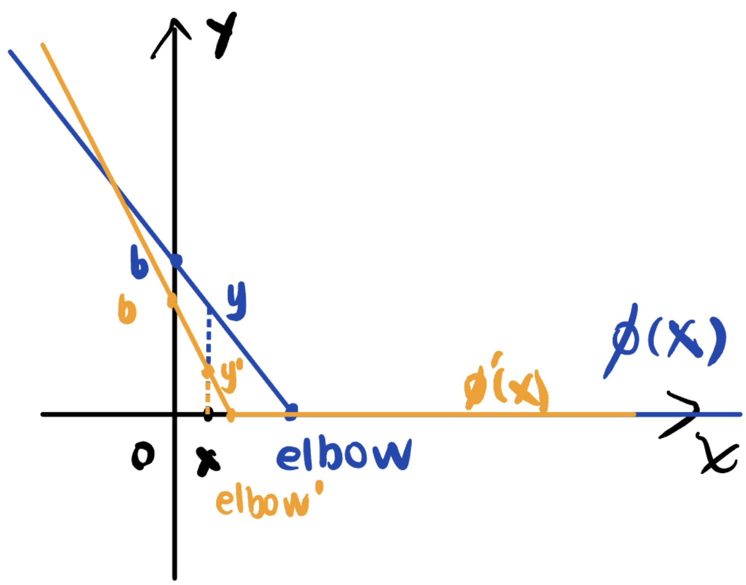

Figure 4: The elbow will move left and slope will decrease during updating.

(c)

$$e_i = -\frac{b^i}{w_i^{(1)}}$$

(d)When $W^{(1)}x + b = \leq 0$, elbow will not change, $e_i^{(')} = e_i = -\frac{b^i}{w_i^{(1)}}$, otherwise, $\frac{\partial(W^{(1)}x+b)}{\partial W_i^1} = x$, $\frac{\partial(W^{(1)}x+b)}{\partial b_i} = 1$, after one gradient update, $w_i^{(i)'} = w_i^{(1)} - \lambda x$, $b^{i'} = b^i - \lambda$. Therefore, the new elbow $e_i^{'} = \frac{w_i^{(1)} - \lambda x}{b^i - \lambda}$

7. Using PyTorch to Learn the Color Organ

   (a) The resistor value is 200 such that the predicted and desired transfer functions match.

   (b) The resistor value is 200 and the corresponding cutoff frequency is 829Hz.

   (c) Yes, we can learn the resistor value by means of neural network.

   The circuit take 4 minutes and 28 seconds to converge, and the final value of R is 200, which is the same as the value I found in the previous part.

   When the value of lr is 20000000, it cause the training to diverge.

   When the value of lr is 200000, it converged in a flash.

   (d) The learned resistor value is 320.

(e) I used the cross entropy to be the loss function, which is $loss\_fn = lambdax, y : (torch.exp(x) - torch.exp(y)) * *2$, and the predicted value is 243, which is close to the real value.

(f) The learned resistor value is 24.

(g)

(h) Yes, it does. Yes.

(i) Under the same learning rate, the larger the initial resistor's value is, the longer training time it takes.

8. Homework Process and Study Group

(a) Pytorch Tutorial

(b)

Name: Chuan Chen

SID: 3038743333

(c) 15 hours.