HW0

Name: Mengying Lin.

SID: 3038737132.

2. Course Policies

Go to the course website and read the course policies carefully. Leave a followup in the Homework 0, Question 2 thread on Ed if you have any questions. Are the following situations violations of course policy? Write "Yes" or "No", and a short explanation for each.

(a) Alice and Bob work on a problem in a study group. They write up a solution together and submit it, noting on their submissions that they wrote up their homework answers together.

(b) Carol goes to a homework party and listens to Dan describe his approach to a problem on the board, taking notes in the process. She writes up her homework submission from her notes, crediting Dan.

(c) Erin gets frustrated by the fact that a homework problem given seems to have nothing in the lecture, notes, or discussion that is parallel to it. So, she starts searching for the problem online. She finds a solution to the homework problem on a website. She reads it and then, after she has understood it, writes her own solution using the same approach. She submits the homework with a citation to the website.

(d) Frank is having trouble with his homework and asks Grace for help. Grace lets Frank look at her written solution. Frank copies it onto his notebook and uses the copy to write and submit his homework, crediting Grace.

(e) Heidi has completed her homework. Her friend Irene has been working on a homework problem for hours, and asks Heidi for help. Heidi sends Irene her photos of her solution through an instant messaging service, and Irene uses it to write her own solution with a citation to Heidi.

a) Yes. Each students should write up their solutions individually.

b) No. She has credited external resources.

c) ~~No.~~ Yes. She finally made the information clear to herself and wrote up the homework on her own.

d) Yes. He shouldn't directly copy others' solutions.

e) ~~No.~~ Yes. It is OK to work in group but Heidi shouldn't send her homework to others directly.

Yes. Erin shouldn't look at at a soln to hw on a website directly

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

Consider a linear regression problem with n training points and d features. When $n = d$, the feature matrix $F \in \mathbb{R}^{n \times n}$ has some maximum singular value α and an extremely tiny minimum singular value. We have noisy observations $\mathbf{y} = F\mathbf{w}^* + \epsilon$. If we compute $\hat{\mathbf{w}}_{inv} = F^{-1}\mathbf{y}$, then due to the tiny singular value of F and the presence of noise we observe that $\|\hat{\mathbf{w}}_{inv} - \mathbf{w}^*\|_2 = 10^{10}$.

Suppose instead of inverting the matrix we decide to use gradient descent instead. We run k iterations of gradient descent to minimize the loss $\ell(w) = \frac{1}{2}\|\mathbf{y} - F\mathbf{w}\|^2$ starting from $\mathbf{w}_0 = \mathbf{0}$. We use a learning rate η which is *small enough* that gradient descent cannot possibly diverge for the given problem. (**This is important. You will need to use this.**)

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

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta\left(F^\top(F\mathbf{w}_{t-1} - \mathbf{y})\right).$$

We are interested in the error $\|\mathbf{w}_k - \mathbf{w}^*\|_2^2$. We want to show that in the worst case, this error can grow at most linearly with iterations k and in particular $\|\mathbf{w}_k - \mathbf{w}^*\|_2 \leq k\eta\alpha\|\mathbf{y}\|_2 + \|\mathbf{w}^*\|_2$.

i.e. The error cannot go "nuts," at least not very fast.

For the purposes of the homework, you only have to prove the key idea, since the rest follows by applying induction and the triangle inequality.

Show that for $t > 0$, $\|\mathbf{w}_t\|_2 \leq \|\mathbf{w}_{t-1}\|_2 + \eta\alpha\|\mathbf{y}\|_2$.

(HINT: What do you know about $(I - \eta F^\top F)$ if gradient descent cannot diverge? What are its eigenvalues like? Use this fact.)

Proof: $\|w_t\|_2 = \|w_{t-1} - \eta(F^T(Fw_{t-1} - y)\|_2 = \|w_{t-1} - \eta F^T F w_{t-1} + \eta F^T y\|_2$

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

Now that α is the max singular value for F.

we have $\|F^T y\|_2 \leq \alpha \|y\|_2$. ... ②

In addition, $\|(I - \eta F^T F) w_{t-1}\|_2 \leq \|I - \eta F^T F\|_2 \|w_{t-1}\|_2$

$= |\sigma_{max}(I - \eta F^T F)| \|w_{t-1}\|_2$. ... ③

Assume η is small enough, we have → η is chosen so that gradient descent does not diverge,

$\sigma_{max}(I - \eta F^T F) = 1 - \eta \sigma_{max}(F^T F) > 0$ so $|\lambda_{max}(I - F^T F)| < 1$.

Thus ③ $\leq \|w_{t-1}\|_2$. ... ④

According to ①, ②, ④, we have $\|w_t\|_2 \leq \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$.

4. Regularization from the Augmentation Perspective

Assume $\mathbf{w}$ is a d-dimensional Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and Σ is symmetric positive-definite. Our model for how the $\{y_i\}$ training data is generated is

$$y = \mathbf{w}^\top \mathbf{x} + Z, \quad Z \sim \mathcal{N}(0, 1), \tag{1}$$

where the noise variables Z are independent of $\mathbf{w}$ and iid across training samples. Notice that all the training $\{y_i\}$ and the parameters $\mathbf{w}$ are jointly normal/Gaussian random variables conditioned on the training inputs $\{\mathbf{x}_i\}$. Let us define the standard data matrix and measurement vector:

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

In this model, the MAP estimate of $\mathbf{w}$ is given by the Tikhonov regularization counterpart of ridge regression:

$$\widehat{\mathbf{w}} = (X^\top X + \Sigma^{-1})^{-1} X^\top \mathbf{y}, \tag{2}$$

In this question, we explore Tikhonov regularization from the data augmentation perspective.

Define the matrix Γ as a $d \times d$ matrix that satisfies $\Gamma^\top \Gamma = \Sigma^{-1}$. Consider the following augmented design matrix (data) $\hat{X}$ and augmented measurement vector $\hat{y}$:

$$\hat{X} = \begin{bmatrix} X \\ \Gamma \end{bmatrix} \in \mathbb{R}^{(n+d)\times d}, \quad \text{and} \quad \hat{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_d \end{bmatrix} \in \mathbb{R}^{n+d},$$

where $\mathbf{0}_d$ is the zero vector in $\mathbb{R}^d$. **Show that the ordinary least squares problem**

$$\operatorname*{argmin}_{\mathbf{w}} \|\hat{\mathbf{y}} - \hat{X}\mathbf{w}\|_2^2$$

has the same solution as (2).

(HINT: Feel free to just use the formula you know for the OLS solution. You don't have to rederive that. This problem is not intended to be hard or time consuming.)

Define $t(w) = \|\hat{y} - \hat{X}w\|_2^2$

$$\sum_{i=1}^{n+d} (\hat{y}_i - \hat{x}_i w)^2 = \sum_{i=1}^{n} (\hat{y}_i - \hat{x}_i w)^2 + \sum_{i=n+1}^{n+d} (\hat{y}_i - \hat{x}_i w)^2$$

$$= (Xw - y)^T (Xw - y) + (\Gamma w)^T (\Gamma w).$$

$$= w^T X^T X w - y^T X w - w^T X^T y + y^T y + w^T \Gamma^T \Gamma w.$$

$$\frac{\partial t}{\partial w} = 2X^T X w - X^T y - X^T y + 2\Gamma^T \Gamma w$$

Let $\frac{\partial t}{\partial w} = 0$. $\quad w = (X^T X + \Gamma^T \Gamma)^{-1} X^T y$

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

So the solution is same as (2).

$$(\hat{X}^T \hat{X})^{-1} \hat{X}^T \hat{y}.$$

$$= \left(\begin{bmatrix} X^T & \Gamma^T \end{bmatrix} \begin{bmatrix} X \\ \Gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} X^T & \Gamma^T \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix}$$

$$= (X^T X + \Gamma^T \Gamma)^{-1} (X^T y)$$

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

5. Vector Calculus Review

Let $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. For the following parts, before taking any derivatives, identify what the derivative looks like (is it a scalar, vector, or matrix?) and how we calculate each term in the derivative. Then carefully solve for an arbitrary entry of the derivative, then stack/arrange all of them to get the final result. Note that the convention we will use going forward is that vector derivatives of a scalar (with respect to a column vector) are expressed as a row vector, i.e. $\frac{\partial f}{\partial \mathbf{x}} = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n}]$ since a row acting on a column gives a scalar. You may have seen alternative conventions before, but the important thing is that you need to understand the types of objects and how they map to the shapes of the multidimensional arrays we use to represent those types.

(a) Show $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T\mathbf{c}) = \mathbf{c}^T$

(b) Show $\frac{\partial}{\partial \mathbf{x}}||\mathbf{x}||_2^2 = 2\mathbf{x}^T$

(c) Show $\frac{\partial}{\partial \mathbf{x}}(A\mathbf{x}) = A$

(d) Show $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A\mathbf{x}) = \mathbf{x}^T(A + A^T)$

(e) Under what condition is the previous derivative equal to $2\mathbf{x}^T A$?

a) $\frac{\partial}{\partial x}(x^Tc) = \left(\frac{\partial(x^Tc)}{\partial x_1}, \cdots \frac{\partial(x^Tc)}{\partial x_n}\right)$

Note $x^Tc = \sum_{i=1}^{n} x_i c_i$,

Therefore $\frac{\partial(x^Tc)}{\partial x_i} = c_i$. $\frac{\partial}{\partial x}(x^Tc) = (c_1, \cdots c_n)$

$= c^T$.

b) $\frac{\partial \|x\|_2^2}{\partial x} = \frac{\partial(\sum x_i^2)}{\partial x} = (2x_1, 2x_2 \cdots, 2x_n) = 2x^T$.

c) $\frac{\partial}{\partial x}(Ax) = \frac{\partial}{\partial x}\begin{pmatrix}\sum a_{1i}x_i \\ \vdots \\ \sum a_{ni}x_i\end{pmatrix} = \begin{pmatrix}\frac{\partial}{\partial x}\sum a_{1i}x_i \\ \vdots \\ \frac{\partial}{\partial x}\sum a_{ni}x_i\end{pmatrix} = \begin{pmatrix}a_{11} \cdots a_{1n} \\ \vdots \quad \vdots \\ a_{n1} \cdots a_{nn}\end{pmatrix}$

d) $\frac{\partial(x^TAx)}{\partial x} = \frac{\partial(x^T(Ax))}{\partial x} = (Ax)^T + x^T\frac{\partial(Ax)}{\partial x}$

$= x^T(A^T + A)$.

e) Let $x^T(A + A^T) = 2x^TA$.

$x^TA + x^TA^T = 2x^TA$.

$x^TA^T = x^TA$.

$x^T(A^T - A) = 0_n$. for all $x \in \mathbb{R}^n$.

Therefore when $A^T = A$, i.e. A is symmetric,

the previous derivative equals to $2x^TA$.

6. ReLU Elbow Update under SGD

In this question we will explore the behavior of the ReLU nonlinearity with Stochastic Gradient Descent (SGD) updates. The hope is that this problem should help you build a more intuitive understanding for how SGD works and how it iteratively adjusts the learned function.

We want to model a 1D function $y = f(x)$ using a 1-hidden layer network with ReLU activations and no biases in the linear output layer. Mathematically, our network is

$$\hat{f}(x) = \mathbf{W}^{(2)}\Phi\left(\mathbf{W}^{(1)}x + \mathbf{b}\right)$$

where $x, y \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^d$, $\mathbf{W}^{(1)} \in \mathbb{R}^{d\times 1}$, and $\mathbf{W}^{(2)} \in \mathbb{R}^{1\times d}$. We define our loss function to be the squared error,

$$\ell\left(x, y, \mathbf{W}^{(1)}, \mathbf{b}, \mathbf{W}^{(2)}\right) = \frac{1}{2}\left\|\hat{f}(x) - y\right\|_2^2.$$

For the purposes of this problem, we define the gradient of a ReLU at 0 to be 0.

(a) Let's start by examining the behavior of a single ReLU with a linear function of x as the input,

$$\phi(x) = \begin{cases} wx + b, & wx + b > 0 \\ 0, & \text{else} \end{cases}.$$

Notice that the slope of $\phi(x)$ is w in the non-zero domain.
We define a loss function $\ell(x, y, \phi) = \frac{1}{2}\|\phi(x) - y\|_2^2$. **Find the following:**

(i) The location of the 'elbow' e of the function, where it transitions from 0 to something else.

(ii) The derivative of the loss w.r.t. $\phi(x)$, namely $\frac{d\ell}{d\phi}$

(iii) The partial derivative of the loss w.r.t. w, namely $\frac{\partial \ell}{\partial w}$

(iv) The partial derivative of the loss w.r.t. b, namely $\frac{\partial \ell}{\partial b}$

a) i). let $\phi(x) = 0$.

$x = -\frac{b}{w}$. The elbow is $(-\frac{b}{w}, 0)$.

ii). $\frac{\partial \ell}{\partial \phi} = \frac{\partial\left[\frac{1}{2}\|\phi - y\|_2^2\right]}{\partial \phi} = \frac{1}{2}\frac{\partial(\phi^2 - 2\phi y + y^2)}{\partial \phi} = \phi - y.$

iii) $\ell = \begin{cases} \frac{1}{2}\|wx+b-y\|_2^2, & wx+b>0 \\ 0, & \text{else} \end{cases}$

$\frac{\partial \ell}{\partial w} = \frac{\partial \ell}{\partial \phi}\cdot\frac{\partial \phi}{\partial w} = \begin{cases} (wx+b-y)\cdot x, & wx+b>0 \\ 0 & \text{else} \end{cases}$

iv) $\frac{\partial \ell}{\partial b} = \frac{\partial \ell}{\partial \phi}\cdot\frac{\partial \phi}{\partial b} = \begin{cases} (wx+b-y), & b > -wx \\ 0 & \text{else} \end{cases}$

(b) Now suppose we have some training point (x, y) such that $\phi(x) - y = 1$. In other words, the prediction $\phi(x)$ is 1 unit above the target y — we are too high and are trying to pull the function downward.

Describe what happpens to the slope and elbow of $\phi(x)$ when we perform gradient descent in the following cases:

(i) $\phi(x) = 0$.

(ii) $w > 0$, $x > 0$, **and** $\phi(x) > 0$. **It is fine to check the behavior of the elbow numerically in this case.**

(iii) $w > 0$, $x < 0$, **and** $\phi(x) > 0$.

(iv) $w < 0$, $x > 0$, **and** $\phi(x) > 0$. **It is fine to check the behavior of the elbow numerically in this case.**

Additionally, draw and label $\phi(x)$, the elbow, and the qualitative changes to the slope and elbow after a gradient update to w and b. You should label the elbow location and a candidate (x, y) pair. Remember that the update for some parameter vector $\mathbf{p}$ and loss ℓ under SGD is

$$\mathbf{p}' = \mathbf{p} - \lambda \nabla_{\mathbf{p}}(\ell),\ \lambda > 0.$$

b) Gradient descent:

$$\begin{cases} w \leftarrow w - \eta \frac{\partial L}{\partial w} \\ b \leftarrow b - \eta \frac{\partial L}{\partial b} \end{cases}$$

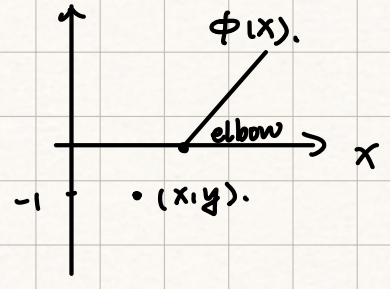
Denote the updated w and b as w', b'.

$$\begin{cases} \begin{cases} w' = w - \eta (wx+b-y)x = w - \eta x \\ b' = b - \eta (w+b-y) = b - \eta \end{cases} & ① \\ \begin{cases} w' = w \\ b' = b \end{cases} \quad \phi \le 0. & ② \end{cases}$$

i). We have $wx+b=0$.

Note in this case $\frac{\partial L}{\partial b}, \frac{\partial L}{\partial w} = 0$. so w and b remain unchanged.

The slope and the elbow do not change.

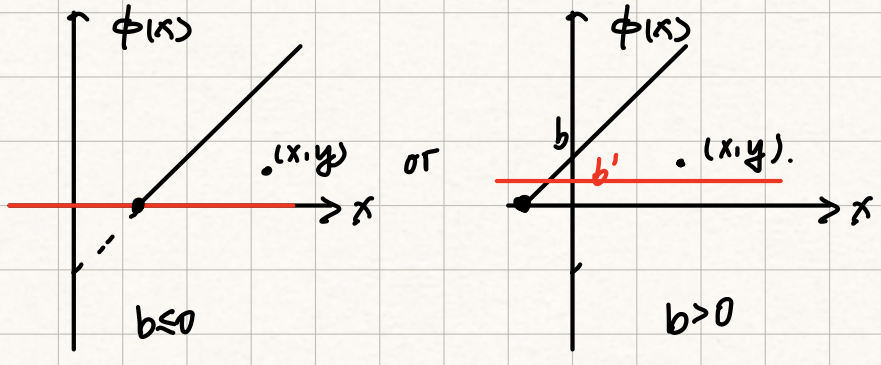


* Red line indicates the changed $\phi(x)$.

ii). As indicated in ①, $w' = w - \eta x$ and $x > 0$, so the slope is dropping.

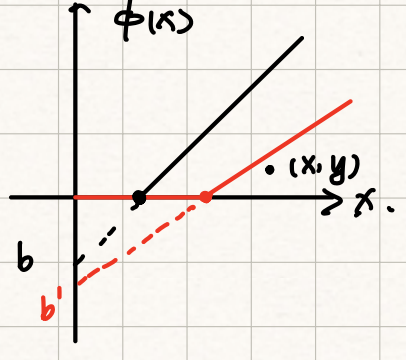
① If $w' = 0$. $\phi = \begin{cases} b' & (b' > 0) \\ 0 & (\text{else}) \end{cases}$. the elbow disappears.

Corresponding graphs:



② if $w' \neq 0$. Let $\phi = w'x + b' = 0$. $x = -\frac{b'}{w'} = -\frac{b-\eta}{w-\eta x}$.

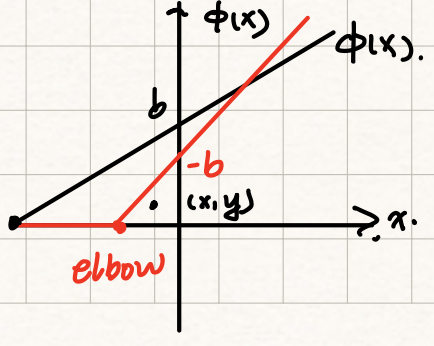
The elbow is moving towards right.



iii). $w' = w - \eta x > w > 0$, so the slope is increasing.

$b' = b - \eta$.

Let $\phi = w'x + b' = 0$, $x = -\frac{b-\eta}{w-\eta x}$.



The elbow is moving towards right.

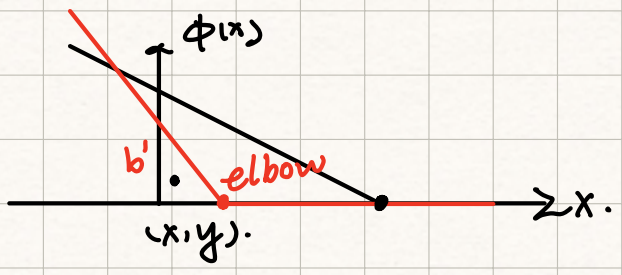
= gets steeper ($w<0$).

iv). $w' = w - \eta x < w$, so the slope is decreasing numerically.

$b' = b - \eta$

Let $\phi = w'x + b' = 0$. $x = -\frac{b-\eta}{w-\eta x}$.

The elbow is moving towards left.



(c) Now we return to the full network function $\hat{f}(x)$. **Derive the location e_i of the elbow of the i'th elementwise ReLU activation.**

(d) **Derive the new elbow location e_i' of the i'th elementwise ReLU activation after one stochastic gradient update with learning rate λ.**

c). Denote $W^{(1)} = \begin{pmatrix} w_1^{(1)} \\ \vdots \\ w_d^{(1)} \end{pmatrix}$ $b = \begin{pmatrix} b_1 \\ \vdots \\ b_d \end{pmatrix}$.

$$\ell\left(x, y, \mathbf{W}^{(1)}, \mathbf{b}, \mathbf{W}^{(2)}\right) = \frac{1}{2}\left\|\hat{f}(x) - y\right\|_2^2.$$

$$\hat{f}(x) = \mathbf{W}^{(2)}\Phi\left(\mathbf{W}^{(1)}x + \mathbf{b}\right)$$

Let $w_i^{(1)} e_i + b_i = 0$.

If $w_i^{(1)} \neq 0$, $e_i = -\frac{b_i}{w_i^{(1)}}$

d). $\frac{\partial \ell}{\partial w_i^{(1)}} = \frac{\partial \ell}{\partial \hat{f}} \cdot \frac{\partial \hat{f}}{\partial \phi} \cdot \frac{\partial \phi}{\partial w_i} \cdot \frac{\partial w^{(1)}}{\partial w_i^{(1)}}$

$= \begin{cases} (\hat{f}(x) - y)\, w_i^{(2)} x & (w_i^{(1)} x + b_i > 0) \\ 0 & (\text{else}). \end{cases}$

$\frac{\partial \ell}{\partial b} = \frac{\partial \ell}{\partial \hat{f}} \cdot \frac{\partial \hat{f}}{\partial \phi} \cdot \frac{\partial \phi}{\partial b}$

$= \begin{cases} (\hat{f}(x) - y)\, w_i^{(2)} & (w_i^{(1)} x + b_i > 0) \\ 0 & (\text{else}). \end{cases}$

Hence if $w_i^{(1)} x + b_i > 0$

$\begin{cases} w_i^{(1)} \leftarrow w_i^{(1)} - \lambda (\hat{f}(x) - y)\, w_i^{(2)} x. \\ b_i \leftarrow b_i - \lambda (\hat{f}(x) - y)\, w_i^{(2)} \end{cases}$

let $w_i^{(1)} x + b_i = 0$.

$$x = -\frac{b_i - \lambda(\hat{f}(x) - y)\, w_i^{(2)}}{w_i^{(1)} - \lambda(\hat{f}(x) - y)\, w_i^{(2)} x}$$

Else if $w_i^{(1)} x + b_i \leq 0$.

$\begin{cases} w_i^{(1)} \leftarrow w_i^{(1)} \\ b_i \leftarrow b_i. \end{cases}$ $x = -\frac{b_i}{w_i^{(1)}}$.

7. Using PyTorch to Learn the Color Organ

a). The value I found : 200

b). The value I found : 200.

c) ① Yes

②

R_init(Ω)	Iterations	Final R (Ω).
0	Fail to train	
100	20321	197
200	0	200
300	27335	200
500	43670	200
1000	77121	200

③ It does not always coverge to the same value.

④ If lr is too small, (e.g. lr = 20, R_init = 1000), it will fail to converge within 1×10^5 iterations. If lr is too large (e.g. $lr = 2 \times 10^7$), it diverges.

⑤ Generally, within a moderate interval (e.g. $lr \in [200, 2 \times 10^5]$). the larger the learning rate is, the faster it converges.

I test on $lr = 2 \times 10^5$, and it converges in 87 iterations.

d) learned value: R = 339 Ω

e) Yes, The loss I find is:

$$l(\hat{y}, y) = (1-y) \times \max(\hat{y} - cutoff\_mag, 0) + y \times \max(cutoff - mag, 0)$$

This loss has 2 term.

① Let us first look at $y \times \max(cutoff - mag, 0)$

For positive sample ($y = 1$), we only need the predicted value to be greater than threshold. If it does, we force the loss to equal to zero.

② $(1-y) \times \max(\hat{y} - cutoff\_mag, 0)$ aims at negative sample,

and the key idea is similar to ①.

f). The learned value is 32 Ω.

g). ① The learned value:

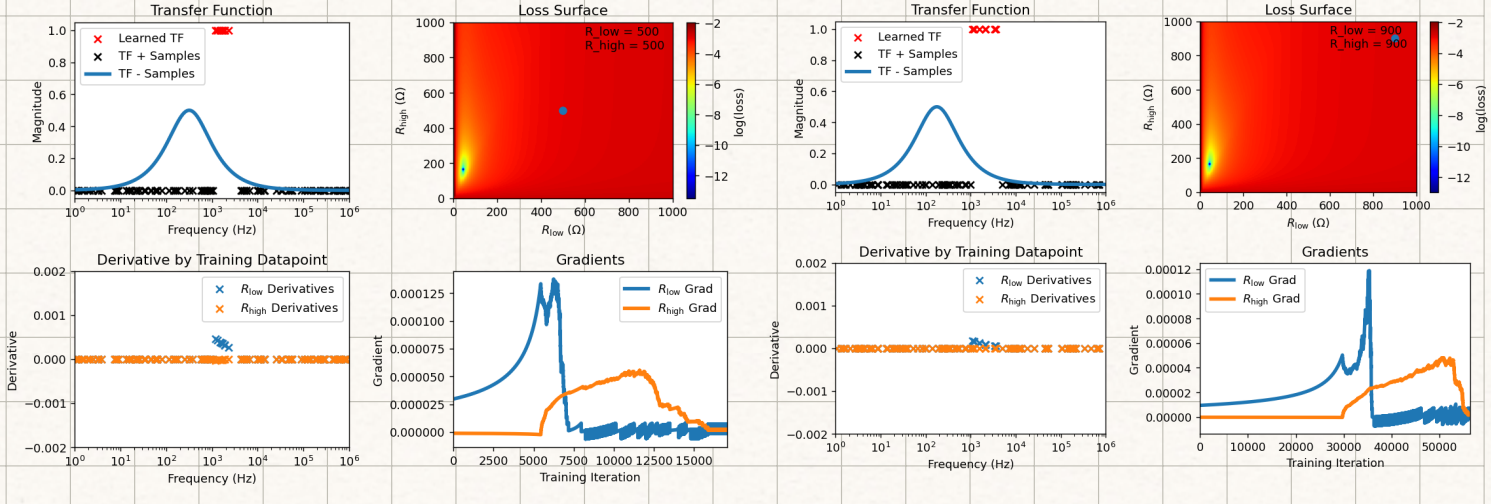
$$\begin{cases} R\text{-}low = 40\,\Omega \\ R\text{-}high = 160\,\Omega \end{cases}$$

② If the initial values are too far from the solution, the circuit may not converge.

When the initial values are both set to 900 Ω, the circuit converges with $R_{low} = 40\,\Omega$, $R_{high} = 161\,\Omega$.

③ When initialized to 500Ω, it only takes 17090 iterations, and <9873> iterations when set to 900Ω.

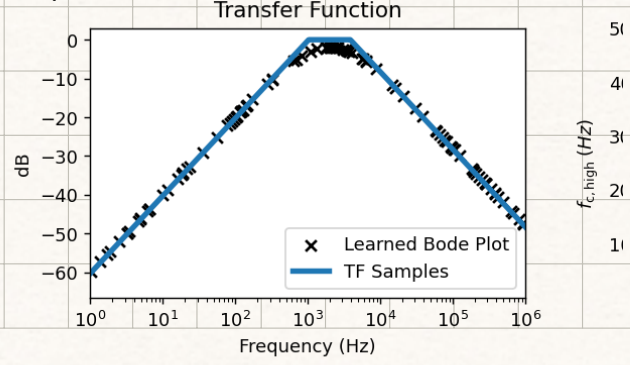
④ In this case loss is relatively high and the start point is far from low-loss region as shown in loss surface



Initialized with 500Ω.

Initialized with 900Ω.

h) ① The bode plot match the expected curve.



② The learned cutoff low pass frequency is 3864 Hz. and the high pass frequency is 1029 Hz, which are slightly different from the given 4000 Hz and 1000 Hz.

i) It takes similar amount of iterations because the parameters will all be updated in a singe interation, but if evaluated with computing time. it'll definitely takes longer.

8. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

(c) **Roughly how many total hours did you work on this homework? Write it down here where you'll need to remember it for the self-grade form.**

a) https://eecs16b.org/notes/fa21/note6.pdf

b)

Names	Xiang Fei
SID	3038733024.

c) 15 h