Introduction to Machine Learning, Spring 2025
Homework 4
(Due April 20, 2025 at 11:59pm (CST))

April 7, 2025

1. Please write your solutions in English.
2. Submit your solutions to the course Gradescope.
3. If you want to submit a handwritten version, scan it clearly.
4. Late homeworks submitted within 3 days of the due date will be marked down 25% each day cumulatively Homeworks submitted more than 3 days after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
5. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

1. [10 points] [Kernel Method]

Kernel regreesion is applying kernel methods into the regression problems. Consider the data $(X_1, Y_1), \ldots, (X_n, Y_n)$, where $X_i \in \mathbb{R}$ and $Y_i \in \mathbb{R}$. Inspired by the fact that

$$\mathbb{E}[Y|X = x] = \int_{-\infty}^{+\infty} yp(y|x)\mathrm{d}y = \int_{-\infty}^{+\infty} y\frac{p(x,y)}{p(x)}\mathrm{d}y$$

Define:

$$\hat{m}(x) = \frac{\int_{-\infty}^{+\infty} y\hat{p}(x,y)\mathrm{d}y}{\hat{p}(x)}$$

where

$$\hat{p}(x) = \frac{1}{n}\sum_i \frac{1}{h}K\left(\frac{X_i - x}{h}\right)$$

and

$$\hat{p}(x,y) = \frac{1}{n}\sum_i \frac{1}{h^2}K\left(\frac{X_i - x}{h}\right)K\left(\frac{Y_i - y}{h}\right).$$

Assume that the kernel functon has the properties

$$\int_{-\infty}^{+\infty} K(u)\mathrm{d}u = 1$$
$$\int_{-\infty}^{+\infty} uK(u)\mathrm{d}u = 0$$

Show that $\hat{m}(x)$ is exactly the kernel regression estimator

$$\hat{m}(x) = \frac{\sum K\left(\frac{X_i - x}{h}\right)Y_i}{\sum K\left(\frac{X_i - x}{h}\right)}$$

Solution

$$\begin{aligned}
\hat{m}(x) &= \frac{\int_{-\infty}^{+\infty} y\hat{p}(x,y)\mathrm{d}y}{\hat{p}(x)} \\
&= \frac{\int_{-\infty}^{+\infty} y\left(\frac{1}{n}\sum_i \frac{1}{h^2}K\left(\frac{X_i - x}{h}\right)K\left(\frac{Y_i - y}{h}\right)\right)\mathrm{d}y}{\frac{1}{n}\sum_i \frac{1}{h}K\left(\frac{X_i - x}{h}\right)} \\
&= \frac{\sum_i K\left(\frac{X_i - x}{h}\right)\int_{-\infty}^{+\infty} yK\left(\frac{Y_i - y}{h}\right)\mathrm{d}y}{h\sum_i K\left(\frac{X_i - x}{h}\right)}
\end{aligned}$$

Let $u = \dfrac{Y_i - y}{h}$, then $y = Y_i - hu$, and $\mathrm{d}y = -h\mathrm{d}u$. So we have:

$$\begin{aligned}
\int_{-\infty}^{+\infty} yK\left(\frac{Y_i - y}{h}\right)\mathrm{d}y &= \int_{+\infty}^{-\infty} (Y_i - hu)K(u)(-h)\mathrm{d}u \\
&= h\int_{-\infty}^{+\infty} (Y_i - hu)K(u)\mathrm{d}u \\
&= h\left(Y_i\int_{-\infty}^{+\infty} K(u)\mathrm{d}u - h\int_{-\infty}^{+\infty} uK(u)\mathrm{d}u\right) \\
&= h\left(Y_i - 0\right) \\
&= hY_i
\end{aligned}$$

Thus we have:

$$\begin{aligned}
\hat{m}(x) &= \frac{\sum_i K\left(\frac{X_i - x}{h}\right)hY_i}{h\sum_i K\left(\frac{X_i - x}{h}\right)} \\
&= \frac{\sum_i K\left(\frac{X_i - x}{h}\right)Y_i}{\sum_i K\left(\frac{X_i - x}{h}\right)}
\end{aligned}$$

2. [15 points] [Maximum Likelihood Estimation (MLE)]

Consider real-valued variables X and Y, in which Y is generated conditional on X according to

$$Y = aX + b + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Here ϵ is an independent variable, called a noise term, which is drawn from a Gaussian distribution with mean 0, and variance σ^2. This is a single variable linear regression model, where a is the only weight parameter and b denotes the intercept. The conditional probability of Y has a distribution $p(Y|X, a, b) \sim \mathcal{N}(aX + b, \sigma^2)$, so it can be written as:

$$p(Y|X, a, b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y - aX - b)^2\right)$$

(a) Assume we have a training dataset of n i.i.d. pairs $(x_i, y_i), i = 1, 2, \ldots, n$, and the likelihood function is defined by $\mathcal{L}(a, b) = \prod_{i=1}^{n} p(y_i|x_i, a, b)$. Please write the Maximum Likelihood Estimation (MLE) problem for estimating a and b. [5 points]

(b) Estimate the optimal solution of a and b by solving the MLE problem in (a). [5 points]

(c) Based on the result in (b), argue that the learned linear model $f(X) = aX + b$, always passes through the point $(\bar{x}, \bar{y})$, where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ denote the sample means. [5 points]

Solution

(a) the MLE of a and b is:

$$\hat{a}, \hat{b} = \underset{a,b}{\text{argmax}}\, \mathcal{L}(a, b) = \underset{a,b}{\text{argmax}} \prod_{i=1}^{n} p(y_i|x_i, a, b) = \underset{a,b}{\text{argmax}} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_i - ax_i - b)^2\right)$$

Take log to the likelihood function, we could get:

$$\hat{a}, \hat{b} = \underset{a,b}{\text{argmax}} \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_i - ax_i - b)^2\right)\right)$$

Since $\frac{1}{\sqrt{2\pi}\sigma}$ has nothing with a, b, and σ is just the variance of the noise term, so $\frac{1}{\sqrt{2\pi}\sigma}, -\frac{1}{2\sigma^2}$ are just constants. So

$$\hat{a}, \hat{b} = \underset{a,b}{\text{argmax}} \sum_{i=1}^{n} -(y_i - ax_i - b)^2 = \underset{a,b}{\text{argmin}} \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

So above all, the MLE problem for estimating a and b is:

$$\hat{a}, \hat{b} = \underset{a,b}{\text{argmin}} \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

(b) Since $\sum_{i=1}^{n} (y_i - ax_i - b)^2$ is a convex function both for a and b, so we just need to set the derivative of $\sum_{i=1}^{n} (y_i - ax_i - b)^2$ to 0 to get the optimal solution.

Let $f(a, b) = \sum_{i=1}^{n} (y_i - ax_i - b)^2, \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, thus

$$\frac{\partial f}{\partial b} = \sum_{i=1}^{n} -2(y_i - ax_i - b) = 2nb - 2\sum_{i=1}^{n} (y_i - ax_i)$$

$$\frac{\partial f}{\partial b} = 0 \Rightarrow 2nb = 2\sum_{i=1}^{n} (y_i - ax_i)$$

$$\Rightarrow b = \frac{1}{n}\sum_{i=1}^{n} y_i - \frac{1}{n} a \sum_{i=1}^{n} x_i$$

$$\Rightarrow b = \bar{y} - a\bar{x}$$

Similarly

$$\frac{\partial f}{\partial a} = \sum_{i=1}^{n} -2x_i(y_i - ax_i - b) = (-2)\sum_{i=1}^{n} x_i y_i - (-2)\sum_{i=1}^{n} ax_i^2 - (-2)\sum_{i=1}^{n} bx_i$$

$$\frac{\partial f}{\partial a} = 0 \Rightarrow \sum_{i=1}^{n} x_i y_i - a\sum_{i=1}^{n} x_i^2 - b\sum_{i=1}^{n} x_i = 0$$

put $b = \bar{y} - a\bar{x}$ into the above equation, we could get:

$$\sum_{i=1}^{n} x_i y_i - a\sum_{i=1}^{n} x_i^2 - (\bar{y} - a\bar{x})\sum_{i=1}^{n} x_i = 0 \Rightarrow a = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n(\bar{x})^2}$$

And put $a = \dfrac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n(\bar{x})^2}$ into $b = \bar{y} - a\bar{x}$, we could get:

$$b = \bar{y} - \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n(\bar{x})^2}\bar{x} = \frac{\sum_{i=1}^{n} x_i^2\bar{y} - \sum_{i=1}^{n} x_i y_i\bar{x}}{\sum_{i=1}^{n} x_i^2 - n(\bar{x})^2}$$

So above all, the optimal solution of a and b is:

$$a = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n(\bar{x})^2}$$

$$b = \bar{y} - a\bar{x} = \frac{\sum_{i=1}^{n} x_i^2\bar{y} - \sum_{i=1}^{n} x_i y_i\bar{x}}{\sum_{i=1}^{n} x_i^2 - n(\bar{x})^2}$$

(c) From the analysis in (b), we could get that:

$$b = \bar{y} - a\bar{x}$$

Put $(\bar{x}, \bar{y})$ into the linear model $f(X) = aX + b$, we could get:

$$f(\bar{x}) = a\bar{x} + b = a\bar{x} + \bar{y} - a\bar{x} = \bar{y}$$

So above all, the learned linear model $f(X) = aX + b$ always passes through the point $(\bar{x}, \bar{y})$.

3. [10 points] [Linear Regression and Classification]

(a) When we talk about linear regression, what does 'linear' regard to? [2 points]

(b) Assume that there are n given training examples $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where each input data point x_i has m real valued features. When $m > n$, the linear regression model is equivalent to solving an under-determined system of linear equations $\mathbf{y} = \mathbf{X}\beta$. One popular way to estimate β is to consider the so-called ridge regression:

$$\underset{\beta}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda||\beta||_2^2$$

for some $\lambda > 0$. This is also known as Tikhonov regularization. Show that the optimal solution β^* to the above optimization problem is given by

$$\beta^* = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

Hint: You need to prove that given $\lambda > 0$, $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ is invertible. [5 points]

(c) Is the given data set linear separable? If yes, construct a linear hypothesis function to separate the given data set. If no, explain the reason. [3 points]

Data	(1,3)	(4,4)	(3,-6)	(-2,1)	(-3,5)	(-6,-4)
Label	+1	-1	-1	+1	-1	-1

Solution

(a) Linear is to the for all parameters of the regression variable β.

(b) As we have learned in linear algebra, we know that tha matrix $X^\top X$ is symmetric, which must be diagonalizable. i.e. there must exist a matrix P and a diagonal matrix Λ such that $X^\top X = P\Lambda P^{-1}$.
Also $\forall x \in \mathbb{R}^n$, we have

$$x^\top(\mathbf{X}^\top\mathbf{X})x = (\mathbf{X}x)^\top(\mathbf{X}x) = \|\mathbf{X}x\|_2^2 \geq 0$$

So $X^\top X$ is positive semi-definite, which means all eigenvalues of $X^\top X$ are non-negative, i.e. the diagonal matrix Λ's elements are all positive.
And since $\lambda > 0$, so λI's all elements are all also non-negative, and λI is also a diagonal matrix. Since

$$X^\top X + \lambda I = P\Lambda P^{-1} + \lambda PIP^{-1} = P(\Lambda + \lambda I)P^{-1}$$

And $\Lambda, \lambda I$ are all diagonal matrix, so $\Lambda + \lambda I$ is also a diagonal matrix. And all elements in $\Lambda + \lambda I$ are all positive, so $\Lambda + \lambda I$ is positive defined. Since $X^\top X + \lambda I = P(\Lambda + \lambda I)P^{-1}$, from the knowledge of similarity and diagonalizable, we could know that $X^\top X + \lambda I$ is also positive defined. So $X^\top X + \lambda I$ is invertible.

And let

$$f(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_2^2 = (\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^\top\beta = \mathbf{y}^\top\mathbf{y} - \beta^\top\mathbf{X}^\top\mathbf{y} - \mathbf{y}^\top\mathbf{X}\beta + \beta^\top\mathbf{X}^\top\mathbf{X}\beta + \lambda\beta^\top\beta$$

Since $f(\beta)$ is convex(prove below), so we just need to set the derivative of $f(\beta)$ to 0 to get the optimal solution.

$$\frac{\partial f(\beta)}{\partial \beta} = 2(-\mathbf{X}^\top\mathbf{y} + \mathbf{X}^\top\mathbf{X}\beta + \lambda\beta)$$

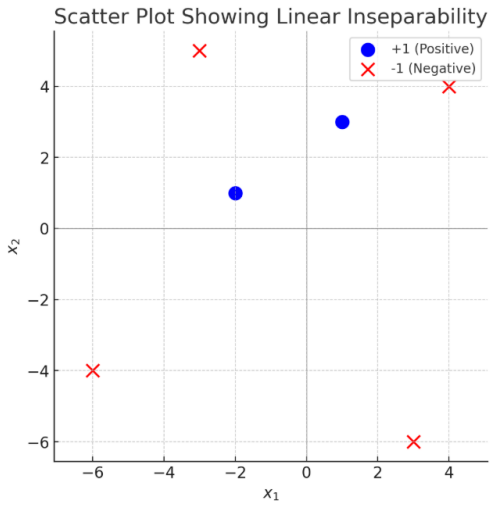
$$\frac{\partial f(\beta)}{\partial \beta} = 0 \Rightarrow (\mathbf{X}^{\mathbf{X}} + \lambda I)\beta = \mathbf{X}^\top\mathbf{y} \Rightarrow \beta = (\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\top\mathbf{y}$$

$$\nabla_\beta^2 f(\beta) = 2(2\mathbf{X}^\top\mathbf{X} + \lambda I) \succ 0$$

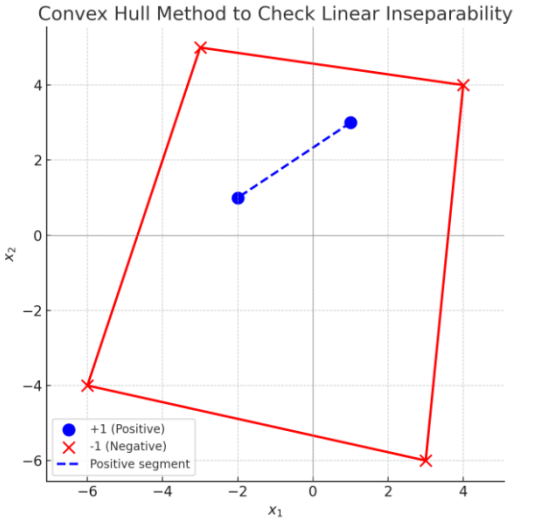
Since we have proved that $X^\top X + \lambda I$ is invertible, so $(\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}$ exists, so

$$\beta^* = (\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\top\mathbf{y}$$

(c) Actually this question is not very reasonable, sorry for causing you confusions. It is better to ask are the data linear separable in the origin feature space. Then the answer is no.
Solution1: only consider in the origin feature space, the data is not linear separable.
To prove this, you may plot a figure as follows. And briefly, intuitively say that it is not seperable.



Or a more theoretical way:



According to the Convex Set Separation Theorem, if two sets of data points are linearly separable, the convex hulls formed by each set must not intersect, and there must exist at least one hyperplane strictly separating these convex hulls.

Conversely, if the convex hulls of the two sets of data points intersect, it implies that there is no linear hyperplane capable of strictly separating these two sets, indicating that the data is linearly inseparable in the given space.

Solution2: if we consider another feature space:
Let the data be formed as $\mathbf{X}_i = (x_1, x_2)$, and let the hypothesis function be $f(\mathbf{X}) = b_1 x_1^2 + b_2 x_2^2 + b_3$. Let $b_1 = 1, b_2 = 1, b_3 = -25$, so the regression function is $f(\mathbf{X}) = x_1^2 + x_2^2 - 25$, and its a linear regression. Make the separate line be $f(\mathbf{X}) = 0$, so the separate line is $x_1^2 + x_2^2 - 25 = 0$. If $f(\mathbf{X}) \leq 0$, set the label to be $+1$, else set the label to be -1. And we could get the result as below:

Data $\mathbf{X}$	(1,3)	(4,4)	(3,-6)	(-2,1)	(-3,5)	(-6,-4)
Function value $f(\mathbf{X})$	-15	7	20	-20	9	27
Label	+1	-1	-1	+1	-1	-1

So above all, we could find that we can construct a linear hypothesis function to separate the given data set.

4. [20 points] [Multi-layer neural network and back-propagation]

Figure 1 below shows an example of a multi-layer neural network, with 1 input layer (2 input units x_1, x_2), h_3 and h_4 are hidden units and 1 output unit h_5. w_{ij} means the weight from unit i to unit j, w_{0j} means the bias, and a_i is the activation function.

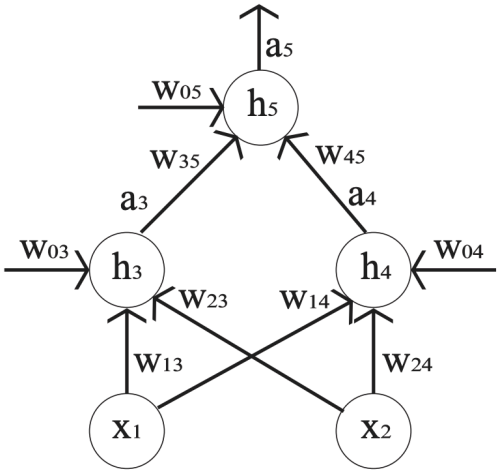


Figure 1: Multi-layer neural network architecture

When an input is passed into a neural network, the signals are passed along the network by following the connections between the units, starting at the input layer and ending in the output layer. This process is referred to as **forward propagation**. In order for the neural network to learn, it must update its weights across its multiple layers. The common approach for neural network learning is called **back-propagation**, which relies on the gradient descent method.

(a) Imagine you are using backprop to do weight updates for an example for which $x_1 = 1$ and $x_2 = -1$. Use Δw_{ij} to refer to the amount weight w_{ij} changes as the result of the update. If $\Delta w_{03} = d_3$ and $\Delta w_{04} = d_4$, what are Δw_{13}, Δw_{23}, Δw_{14}, and Δw_{24}? [5 points]

(b) This question will have you simulate the process of forward propagation and back-propagation on an example for which $x_1 = 1$ and $x_2 = -1$ and whose desired output is $y = 1$. The values for the various weights are as follows:

- Weight and bias for h_3: $w_{03} = -1$, $w_{13} = 1$, $w_{23} = -1$
- Weight and bias for h_4: $w_{04} = 2, w_{14} = -1, w_{24} = 1$
- Weight and bias for h_5: $w_{05} = -2, w_{35} = 1, w_{45} = 1$

For the rest of this question you'll be using the algorithm for forward and back-propagation given in class. Note: **assume the learning rate α is 0.5 and Error function for the output is $(\hat{y}-y)^2$.**

What is the output and error after the forward propagation? And what is the gradient and updated value after the back-propagation for each weight with the logistic activation function $g(z) = \frac{1}{1+e^{-z}}$? Recall that $g'(z) = g(z)(1 - g(z))$. Please also give the network's new output and error after the update, and keep your answer to **three** significant digits after the decimal. [15 points]

Solution

(a) $\Delta w_{13} = d_3, \Delta w_{23} = -d_3, \Delta w_{14} = d_4, \Delta w_{24} = -d_4$

(b)

- output and error: $a_5 = 0.317, \text{Error}(W) = 0.467$
- gradient and updated value for w_{03} : $-0.058, -0.971$
- gradient and updated value for w_{13} : $-0.058, 1.029$
- gradient and updated value for w_{23} : $0.058, -1.029$
- gradient and updated value for w_{04} : $-0.074, 2.037$

- gradient and updated value for w_{14} : $-0.074, -0.963$
- gradient and updated value for w_{24} : $0.074, 0.963$
- gradient and updated value for w_{05} : $-0.296, -1.852$
- gradient and updated value for w_{35} : $-0.216, 1.108$
- gradient and updated value for w_{45} : $-0.148, 1.074$
- new output and error after the update: $a_5 = 0.388, \text{Error}(W) = 0.375$

5. [30 points] [Feed Forward and Backpropagation]

Network Overview Consider the neural network with one hidden layer shown in Figure 2. The input layer consists of 6 features $x = [x_1, ..., x_6]^\top$, the hidden layer has 4 nodes $z = [z_1, ..., z_4]^\top$, and the output layer is a probability distribution $y = [y_1, y_2, y_3]^\top$ over 3 classes. We also add a bias to the input, $x_0 = 1$ and the hidden layer $z_0 = 1$, both of which are fixed to 1.

$\boldsymbol{\alpha}$ is the matrix of weights from the inputs to the hidden layer and $\boldsymbol{\beta}$ is the matrix of weights from the hidden layer to the output layer. $\alpha_{j,i}$ represents the weight going *to* the node z_j in the hidden layer *from* the node x_i in the input layer (e.g. $\alpha_{1,2}$ is the weight from x_2 to z_1), and $\boldsymbol{\beta}$ is defined similarly. We will use a sigmoid activation function for the hidden layer and a softmax for the output layer.

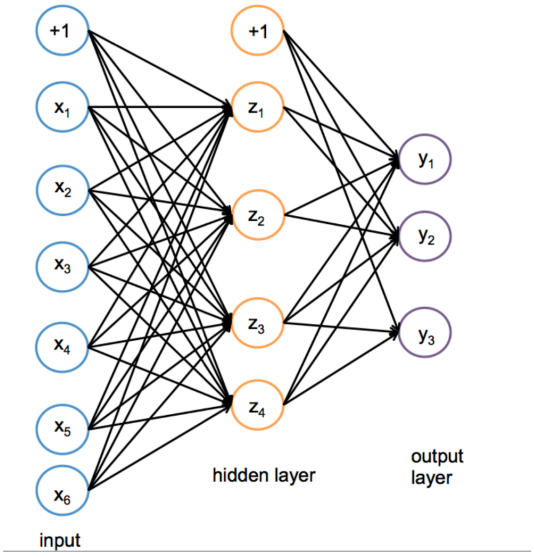


Figure 2: A One Hidden Layer Neural Network

Network Details Equivalently, we define each of the following.
The input:

$$x = [x_1, x_2, x_3, x_4, x_5, x_6]^\top$$

Linear combination at first (hidden) layer:

$$a_j = \alpha_{j,0} + \sum_{i=1}^{6} \alpha_{j,i} * x_i, \ \forall j \in \{1, \ldots, 4\}$$

Activation at first (hidden) layer:

$$z_j = \sigma(a_j) = \frac{1}{1 + \exp(-a_j)}, \ \forall j \in \{1, \ldots, 4\}$$

Linear combination at second (output) layer:

$$b_k = \beta_{k,0} + \sum_{j=1}^{4} \beta_{k,j} * z_j, \ \forall k \in \{1, \ldots, 3\}$$

Activation at second (output) layer:

$$\hat{y}_k = \frac{\exp(b_k)}{\sum_{l=1}^{3} \exp(b_l)}, \ \forall k \in \{1, \ldots, 3\}$$

Note that the linear combination equations can be written equivalently as the product of the transpose of the weight matrix with the input vector. We can even fold in the bias term $\alpha_{j,0}$ by thinking of $x_0 = 1$, and fold in $\beta_{j,0}$ by thinking of $z_0 = 1$.

Loss We will use cross-entropy loss, $\ell(\hat{y}, y)$. If y represents our target output, which will be a one-hot vector representing the correct class, and $\hat{y}$ represents the output of the network, the loss is calculated by:

$$\ell(\hat{y}, y) = -\sum_{i=1}^{3} y_i \ln(\hat{y}_i)$$

Prediction When doing prediction, we will predict the argmax of the output layer. For example, if $\hat{y}_1 = 0.3, \hat{y}_2 = 0.2, \hat{y}_3 = 0.5$ we would predict class 3. If the true class from the training data was 2 we would have a one-hot vector y with values $y_1 = 0$, $y_2 = 1$, $y_3 = 0$.

(a) [8 points] We initialize the weights as:

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 & 2 & -3 & 0 & 1 & -3 \\ 3 & 1 & 2 & 1 & 0 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 0 & 2 & 1 & -2 & 2 \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} 1 & 2 & -2 & 1 \\ 1 & -1 & 1 & 2 \\ 3 & 1 & -1 & 1 \end{bmatrix}$$

And weights on the bias terms ($\alpha_{j,0}$ and $\beta_{j,0}$) are initialized to 1.

You are given a training example $x^{(1)} = [1, 1, 0, 0, 1, 1]^\top$ with label class 2, so $y^{(1)} = [0, 1, 0]^\top$. Using the initial weights, run the feed forward of the network over this example (without rounding during the calculation) and then answer the following questions, round to 4 decimal places.

(1) What is a_1?

2

(2) What is z_1?

0.8808

(3) What is a_3?

8

(4) What is z_3?

0.9997

(5) What is b_2?

3.6430

(6) What is $\hat{y}_2$?

0.2615

(7) Which class would we predict on this example?

3

(8) What is the total loss on this example?

1.3412

(b) [10 points] Now use the results of the previous question to run backpropagation over the network and update the weights. Use learning rate $\eta = 1$.

Do your backpropagation calculations without rounding then answer the following questions, then in your responses, round to 4 decimal places.

(1) What is the updated value of $\beta_{2,1}$?

1.6505

(2) What is the updated weight of the hidden layer bias term applied to y_1 (i.e. $\beta_{1,0}$)?

0.8917

(3) What is the updated value of $\alpha_{3,4}$?

2.0000

(4) What is the updated weight of the input layer bias term applied to z_2 (i.e. $\alpha_{2,0}$)?

0.9986

(5) If we ran backpropagation on this example for a large number of iterations and then ran feed forward over the same example again, which class would we predict?

2

(c) [12 points] Let us now introduce regularization into our neural network. For this question, we will incorporate L2 regularization into our loss function $\ell(\hat{y}, y)$, with the parameter λ controlling the weight given to the regularization term. Round your answer to 4 decimal places

(1) Write the expression for the regularized loss function of our network after adding L2 regularization (**Hint:** Remember that bias terms should not be regularized!)

$$\ell(\hat{y}, y) = -\sum_{i=1}^{3} y_i \ln(\hat{y}_i) + \lambda \left(\sum_{j=1}^{4} \sum_{i=1}^{6} \alpha_{j,i}^2 + \sum_{j=1}^{3} \sum_{i=1}^{4} \beta_{j,i}^2 \right)$$

or

$$\ell(\hat{y}, y) = -\sum_{i=1}^{3} y_i \ln(\hat{y}_i) + \frac{\lambda}{2} \left(\sum_{j=1}^{4} \sum_{i=1}^{6} \alpha_{j,i}^2 + \sum_{j=1}^{3} \sum_{i=1}^{4} \beta_{j,i}^2 \right)$$

Or any other suitable equivalent form, for example $\| \cdot \|_F^2$. But $\| \cdot \|_2^2$ is not allowed. The spectrum norm is not the matrix's L2 norm, but Frobenius norm is.

(2) Compute the regularized loss for training example $x^{(1)}$ (assume $\lambda = 0.01$ and use the weights before backpropagation)

2.4112
or
1.8763

Suppose the weight initialization for α is changed to the following:

$$\boldsymbol{\alpha} = \begin{bmatrix} 10 & 20 & -30 & 0 & 10 & -30 \\ 30 & 10 & 20 & 10 & 0 & 20 \\ 20 & 20 & 20 & 20 & 20 & 10 \\ 10 & 0 & 20 & 10 & -20 & 20 \end{bmatrix}$$

β and bias terms are not changed.

(3) Report the non-regularized loss for the network on training example $x^{(1)}$

1.4076

(4) Report the regularized loss for the network on training example $x^{(1)}$ ($\lambda = 0.01$)

79.6976
or
40.5526

(5) For a network which uses the regularized loss function, write the gradient update equation for $\alpha_{j,i}$. You may use $\frac{\partial \ell(\hat{y},y)}{\partial \alpha_{j,i}}$ to denote the gradient update w.r.t non-regularized loss and η to denote the learning rate.

$$\alpha_{j,i} \leftarrow \alpha_{j,i} - \eta \left(\frac{\partial \ell(\hat{y}, y)}{\partial \alpha_{j,i}} + 2\lambda \alpha_{j,i} \right) \text{ for } i \neq 0$$
$$\alpha_{j,0} \leftarrow \alpha_{j,0} - \eta \cdot \frac{\partial \ell(\hat{y}, y)}{\partial \alpha_{j,0}}$$

or

$$\alpha_{j,i} \leftarrow \alpha_{j,i} - \eta \left(\frac{\partial \ell(\hat{y}, y)}{\partial \alpha_{j,i}} + \lambda \alpha_{j,i} \right) \text{ for } i \neq 0$$
$$\alpha_{j,0} \leftarrow \alpha_{j,0} - \eta \cdot \frac{\partial \ell(\hat{y}, y)}{\partial \alpha_{j,0}}$$

(6) Based on your observations from previous questions, **select all statements which are true**:

☐ The non-regularized loss is always higher than the regularized loss

■ As weights become larger, the regularized loss increases faster than non-regularized loss

☐ On adding regularization to the loss function, gradient updates for the network become larger

■ When using large initial weights, weight values decrease more rapidly for a network which uses regularized loss

☐ None of the above