ECE C147/C247, Winter 2022 **Midterm Review**
Department of Electrical and Computer Engineering Prof. J.C. Kao
University of California, Los Angeles TAs: T. Monsoor, T. Wang, P. Lu, Y. Li

1. **Noisy linear regression (Tonmoy)**

A real estate company have assigned us the task of building a model to predict the house prices in Westwood. For this task, the company has provided us with a dataset $\mathcal{D}$:

$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$$

where $x^{(i)} \in \mathbb{R}^d$ is a feature vector of the i^{th} house and $y^{(i)} \in \mathbb{R}$ is the price of the i^{th} house. Since we just learned about linear regression , so we have decided to use a linear regression model for this task. Additionally, the IT manager of the real estate company have requested us to design a model with small weights. In order to accommodate his request, we will design a linear regression model with parameter regularization. In this problem, we will navigate through the process of achieving regularization by adding noise to the feature vectors. Recall, that we define the cost function in a linear regression problem as:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - (x^{(i)})^T\theta)^2$$

where $\theta \in \mathbb{R}^d$ is the parameter vector. As mentioned earlier, we will be adding noise to the feature vectors in the dataset. Specifically, we will be adding zero-mean gaussian noise of known variance σ^2 from the distribution

$$\mathcal{N}(0, \sigma^2 I)$$

where $I \in \mathbb{R}^{d\times d}$ and $\sigma \in \mathbb{R}$. With the addition of gaussian noise the modified cost function is given by,

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - (x^{(i)} + \delta^{(i)})^T\theta)^2$$

where $\delta^{(i)}$ are i.i.d noise vectors with $\delta^{(i)} \sim \mathcal{N}(0, \sigma^2 I)$.

(a) Express the expectation of the modified loss over the gaussian noise, $\mathbb{E}_{\delta\sim\mathcal{N}}[\tilde{\mathcal{L}}(\theta)]$, in terms of the original loss plus a term independent of the data $\mathcal{D}$. To be precise, your answer should be of the form:

$$\mathbb{E}_{\delta\sim\mathcal{N}}[\tilde{\mathcal{L}}(\theta)] = \mathcal{L}(\theta) + R$$

where R is not a function of $\mathcal{D}$. For answering this part, you might find the following result useful:

$$\mathbb{E}_{\delta\sim\mathcal{N}}[\delta\delta^T] = \sigma^2 I$$

.

(b) Based on your answer to (a), under expectation what regularization effect would the addition of the noise have on the model?

(c) Suppose $\sigma \longrightarrow 0$, what effect would this have on the model?

(d) Suppose $\sigma \longrightarrow \infty$, what effect would this have on the model?

2. **Machine learning basics (Yang)**

(a) Please select all that apply about k-NN in the following options. Assume a point can be its own neighbor.

Select all that apply:

A k-NN works great with a small amount of data, but struggles when the amount of data becomes large.

B k-NN is sensitive to outliers; therefore, in general we decrease k to avoid overfitting.

C k-NN can only be applied to classification problems, but it cannot be used to solve regression problems.

D We can always achieve zero training error (perfect classification) with k-NN, but it may not generalize well in testing.

(b) Imagine you are using a k-Nearest Neighbor classifier on a data set with lots of noise. You want your classifier to be less sensitive to the noise. Which is more likely to help and with what side-effect? **Select one:**

A. Increase the value of k $\Rightarrow$ Increase in prediction time

B. Decrease the value of k $\Rightarrow$ Increase in prediction time

C. Increase the value of k $\Rightarrow$ Decrease in prediction time

D. Decrease the value of k $\Rightarrow$ Decrease in prediction time

(c) Which of the following is true about the regularization parameter λ (the parameter that controls the extent of regularization): **Select one:**

A. Larger values of λ can overfit the data.

B. Larger λ does not affect the performance of your hypothesis

C. Adding a regularization term to a classifier, (λ != 0), may cause some training examples to be classified incorrectly.

3. **Training neural networks (Pan)**

Covered topics

(a) Gradient problems

(b) Weight initialization

(c) Batch normalization

(d) L2 and L1 regularization

(e) Dataset augmentation

(f) Multitask and transfer learning

(g) Ensemble methods

(h) Dropout

(i) Other techniques discussed in the lectures

Review materials

(a) Lecture slides (formal notes)

(b) Homework

(c) Discussion handouts

(d) Textbook (optional)

Problem examples

(a) Which of the following techniques can be used to reduce overfitting? **Select all that apply**.
A. Batch normalization
B. Data augmentation
C. Dropout
D. Use less training data
E. None of the above

(b) In homework 3, assume you are implementing a fully connected neural network using the sigmoid activation function. Is this a good idea to initialize the weights with large positive numbers? Explain why or why not.

4. **Backpropagation (Tonmoy)**

Let $\mathbf{x} \in \mathbb{R}^n$, $\hat{\mathbf{x}} \in \mathbb{R}^n$, $y \in \mathbb{R}$ and $\mathbf{W} \in \mathbb{R}^{n \times n}$. We define the following loss function

$$\mathcal{L} = y \cdot \frac{(\mathbf{W}\mathbf{x})^T(\mathbf{W}\hat{\mathbf{x}})}{\|\mathbf{W}\mathbf{x}\|_2 \|\mathbf{W}\hat{\mathbf{x}}\|_2}$$

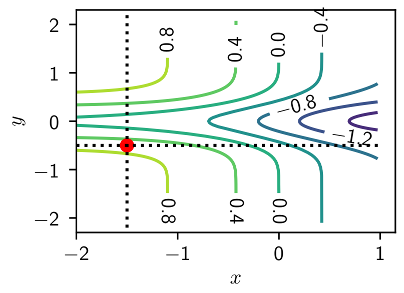
Draw the computational graph and use backpropagation to compute $\nabla_{\mathbf{W}}\mathcal{L}$.

5. **Optimization Methods (Tianyi)**

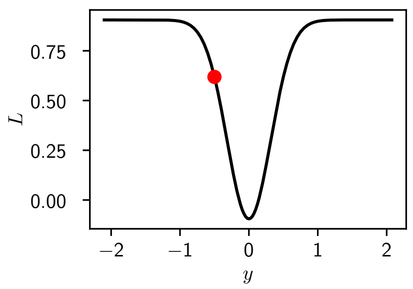
Figure 1 shows the landscape of a bivariate loss function. From the contour plot in fig. 1a, we can tell that the shown region contains a "valley", aligned parallel to the x-axis.

The valley shape is more clear if one considers the cross-section plot (fig. 1b) of the loss surface along the vertical dotted line in fig. 1a. The "valley" goes lower in the loss value as x increase, as shown in another cross-section plot fig. 1c.

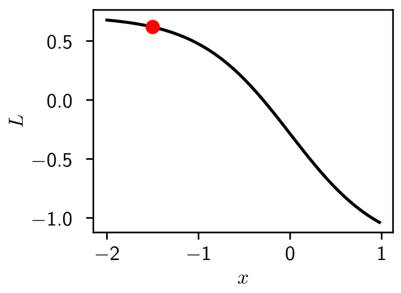
Now carefully observe the plots, and answer the following questions.



(a) Loss landscape contour plot

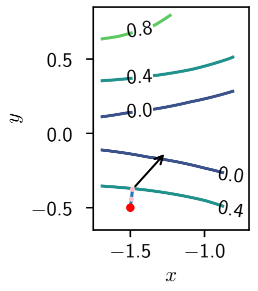


(b) Loss surface cross-section curve along the **vertical** dotted line in fig. 1a

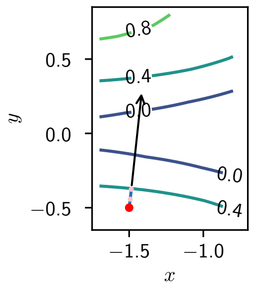


(c) Loss surface cross-section curve along the **horizontal** dotted line in fig. 1a

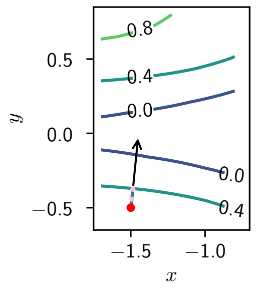
Figure 1: Loss landscape of $L(x, y)$



(a)



(b)



(c)

Figure 2: Update step given by different optimization algorithms. The same 2-step-long history is shared for the three cases, where the red point is the starting point, the pink points show the historical locations, and the blue line connecting the pink points show the history trajectory.

(a) **Optimization step**. We have learned several gradient-based optimization algorithms. Different optimization algorithm will give different update step, even given the same history trajectory in a zoomed-in region of the loss landscape. In fig. 2, update step given by three different algorithms based on the same history trajectory are plotted. Please match the plots with the following algorithms and briefly explain your reasons.

- ☐ Gradient descent

- ☐ Gradient descent with momentum

- ☐ Adagrad

(b) **Escaping the valley**. Based on your analysis in question 5a and considering the loss landscape overview in fig. 1a, which of the three optimization algorithms can "escape" from the valley the most efficiently? What problem will the other algorithms suffer from?