CSE 493 G1/ 599 G1
Deep Learning
Winter 2024 Quiz 1

Jan 19, 2024

Full Name: _____

UW Net ID: _____

Question		Score
True/False	(4 pts)	
Multiple Choice	(8 pts)	
Short Answer	(8 pts)	
Total	(20 pts)	

Welcome to the CSE 493 G1 Quiz 1!

- The exam is 20 min and is **double-sided**.
- No electronic devices are allowed.

I understand and agree to uphold the University of Washington Honor Code during this exam.

Signature: _____ Date: _____

Good luck!

This page is left blank for scratch work only. DO NOT write your answers here.

1 True / False (4 points) - Recommended 4 Minutes

Fill in the circle next to True or False, or fill in neither. Fill it in completely like this: ●. No explanations are required.

Scoring: Correct answer is worth 1 points.

1.1 KNN parameters only gets updated after the test phase.

◯ True

◯ False

SOLUTION:
False, KNN doesn't have parameters

1.2 In a typical train-test-validation split, the training set is used to train the model, the validation set is used for hyperparameter tuning, and the test set is reserved for evaluating the model's performance on unseen data.

◯ True

◯ False

SOLUTION:
True

1.3 The softmax function is not differentiable, and we need some adjustments to make it suitable for gradient-based optimization algorithms.

◯ True

◯ False

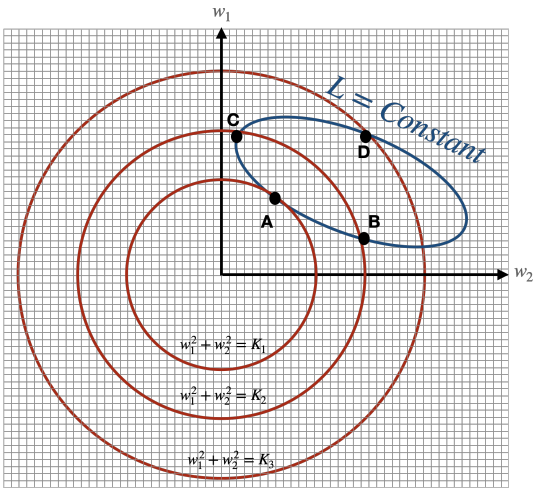
SOLUTION:
False. The softmax function is differentiable

1.4 Let's consider the Error function as follows: $\mathbf{E} = \mathbf{L} + \mathcal{R}(w)$, where $\mathcal{R}(w)$ represents the L2 regularization term, $\mathbf{L} = \sum_i^N \mathbf{L}_i$ is the data loss, and w is a 2D vector $[w_1, w_2]$. In the provided figure, L is a constant value across the ellipse (We know $\mathbf{L}$ is a function of w). Point D exhibits the lowest Error ($\mathbf{E}$) compared to points A, B, and C.

◯ True

◯ False

SOLUTION:
False, A has the lowest error because $w_1^2 + w_2^2$ is the lowest there



2 Multiple Choices (8 points) - Recommended 8 Minutes

Fill in the circle next to the letter(s) of your choice (like this: ●). No explanations are required. **Choose ALL options that apply.**

Each question is worth 4 points and the answer may contain one or more options. Selecting all of the correct options and none of the incorrect options will get full credits. For questions with multiple correct options, each incorrect or missing selection gets a 4-point deduction (up to 4 points).

2.1 Mark the correct statement(s) about loss functions.

$$SVM\ (Hinge)\ loss = \frac{1}{N}\sum_{i}^{N}\sum_{j\neq y_i}\max\left(0, score_j - score_{y_i} + 1\right)$$

$$Cross\ Entropy\ (Softmax)\ loss = -\frac{1}{N}\sum_{i}^{N}\log\left(\frac{e^{score_{y_i}}}{\sum_j e^{score_j}}\right),$$

$$[z_1, z_2, ..., z_d] \rightarrow softmax \rightarrow [\frac{e^{z_1}}{\sum_{j=1}^{d} e^{z_j}}, \frac{e^{z_2}}{\sum_{j=1}^{d} e^{z_j}}, ..., \frac{e^{z_d}}{\sum_{j=1}^{d} e^{z_j}}]$$

- ◯ A: Softmax loss is an extension of hinge loss for handling multiple classes.
- ◯ B: if we add a fixed constant value to all softmax inputs, the softmax outputs or the probabilities do not change.
- ◯ C: Cross-entropy loss is suitable for 2 and more classes classification problems.
- ◯ D: We add regularization term to the loss functions to improve the performance on the train and test data.
- ◯ E: Let assume the output scores for sample i is $[20, -10, -10, -20]$ and the index of its ground truth label is 0. Both the softmax loss and hinge loss for this sample is 0.

SOLUTION:

- ◯ A: False. Softmax loss is not related to the hinge loss.

◯ B: True.

◯ C: True.

$$[z_1+\alpha, z_2+\alpha, ..., z_d+\alpha] \rightarrow softmax \rightarrow [\frac{e^{z_1+\alpha}}{\sum_{j=1}^{d} e^{z_j+\alpha}}, \frac{e^{z_2+\alpha}}{\sum_{j=1}^{d} e^{z_j+\alpha}}, ..., \frac{e^{z_d+\alpha}}{\sum_{j=1}^{d} e^{z_j+\alpha}}] =$$

$$[\frac{e^{z_1} \times e^{\alpha}}{e^{\alpha} \times \sum_{j=1}^{d} e^{z_j}}, \frac{e^{z_2} \times e^{\alpha}}{e^{\alpha} \times \sum_{j=1}^{d} e^{z_j}}, ..., \frac{e^{z_d} \times e^{\alpha}}{e^{\alpha} \times \sum_{j=1}^{d} e^{z_j}}] = [\frac{e^{z_1}}{\sum_{j=1}^{d} e^{z_j}}, \frac{e^{z_2}}{\sum_{j=1}^{d} e^{z_j}}, ..., \frac{e^{z_d}}{\sum_{j=1}^{d} e^{z_j}}]$$

◯ D: False.Regularization improves the generalization and performance on the test set but decreases the performance on the train set because of the introduction of a new bias to the model.

◯ E: False. Softmax loss is not zero. $log\frac{e^{20}}{e^{20}+e^{-10}+e^{-10}+e^{-20}} \neq 0$

2.2 Mark the true statement(s) about optimization.

◯ A: Given the loss function $L(w) = w^2$ (w is 1D), and initializing w with $w = 1$, and step size of 1, the optimization process will converge, bringing w to the global minimum of the function.

◯ B: Given the loss function $L(w) = w^2$ (w is 1D), and initializing w with $w = 100$, and step size of 0.9, the optimization process will converge, bringing w to the global minimum of the function

◯ C: Gradient descent is always better than stochastic gradient descent.

◯ D: Gradient descent only works when neural networks have non-linear layers.

SOLUTION:
update rule : $w_{new} = w_{old} - step\_size * gradient$

◯ A: False, it fluctuates between 1 and -1 and doesnt converge. $w_0 = 1 \rightarrow w_1 = 1 - 1 * 2 = -1 \rightarrow w_2 = -1 - 1 * (-2) = 1, w_3 = -1, \ldots$

◯ B: True. $w_0 = 100 \rightarrow w_1 = 100 - 0.9 * 200 = -80 \rightarrow w_2 = -80 - 0.9 * (-160) = 64 \cdots \rightarrow w_n = 0$

◯ C: False. Gradient descent is **not** always better than stochastic gradient descent. The randomness in SGD can help SGD escape saddle points and local minimas more effectively than deterministic gradient descent.

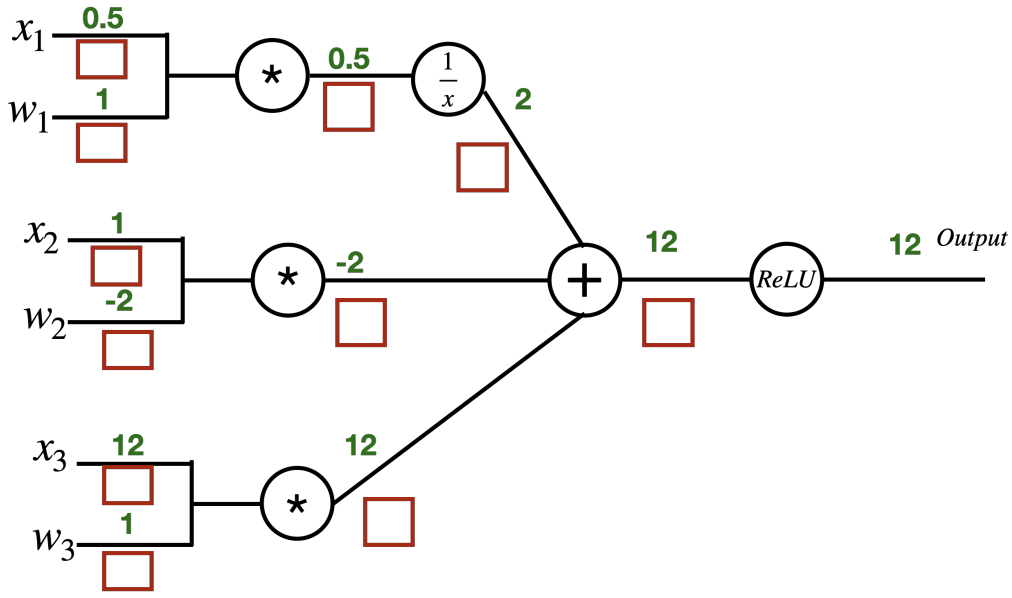
◯ D: False Gradient descent works with non-linear layers too.

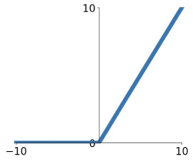
3 Short Answers (8 points) - Recommended 8 Minutes

Please make sure to write your answer only in the provided space.

3.1 Back Propagation (8 points)

1. (4 points) In the following computational graph, the activations for the forward pass have been filled out (numbers above the connections). Please fill out the gradients up to inputs (x_is) and weights (w_is).





ReLU

SOLUTION:

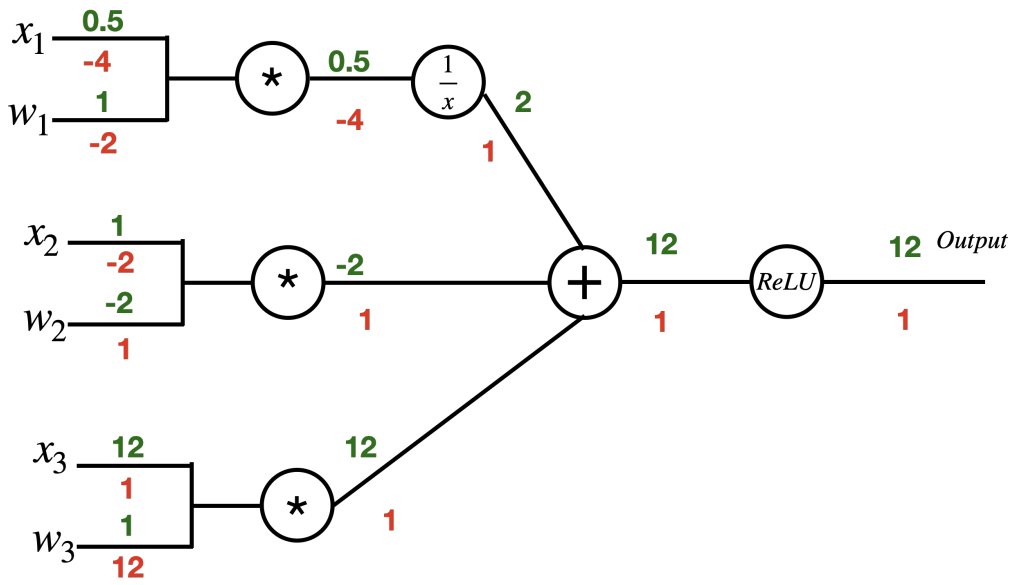
2. (4 points) After computing gradients in 3.1, use them to update w_1, w_2, w_3 with step size (learning rate) of 0.1. Then with the new weights, do the forward pass again and calculate the *output*. (** DONT change x_1, x_2, x_3)

   (a) $w1 =$

   (b) $w2 =$

   (c) $w3 =$

   (d) $output =$



SOLUTION:
w1 = 1.2, w2 = -2.1, w3 = -0.2, output = relu(10/6+ -2.1 -2.4) =0