CSE 493 G1/ 599 G1
Deep Learning
Spring 2023 Midterm Exam

May 04, 2023

Full Name: ______________________________

UW Net ID: ______________________________

Question	Score
True/False (20 pts)	
Multiple Choice (40 pts)	
Short Answer (40 pts)	
Total (100 pts)	

Welcome to the CSE 493 G1 Midterm Exam!

- The exam is 1 hour and 20 min and is **double-sided**.
- No electronic devices are allowed.
- One handwritten double sided cheat sheet is allowed.
- There is a potential for 12 points of extra credit.

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 (20 points) - Recommended 10 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 2 points. To discourage guessing, incorrect answers are worth -1 points. Leaving a question blank will give 0 points.

1.1 If Model A has a lower test loss on a dataset than Model B, then Model A must have a higher accuracy on the test dataset than Model B.

○ True

○ False

SOLUTION:
False. Consider a dataset of 2 images, image 1 has ground truth value of Cat and and Image 2 has ground truth value of Dog. Let Model A and Model B have SVM loss with margin 1. Model A says that the first image score is [Dog: 1.1, Cat: 1.0] and the second image score is [Dog: 1.0, Cat: 1.1]. It therefore has accuracy of 0, but loss of 2.2. Model B says that the first image score is [Dog: 1.0, Cat: 1.1] and the second image score is [Dog: 1.0, Cat: 200.0]. This model has accuracy of 50%, but loss 200.9

1.2 High train loss is a sign of overfitting.

○ True

○ False

SOLUTION:
False. High test loss may be a sign of overfitting, but high train loss is a sign of underfitting

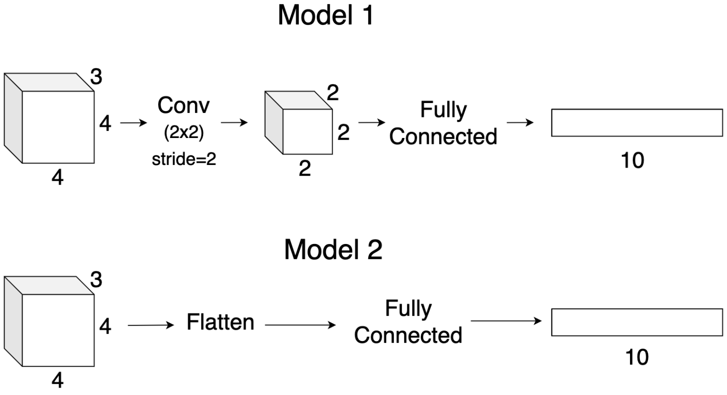
1.3 Consider a fully connected layer $\mathbf{W}$ just before a ReLU function in a network. If an element $w_{i,j}$ of the weight matrix $\mathbf{W}$ has a negative value then the gradient of the loss with respect to this weight is guaranteed to be zero.

○ True

○ False

SOLUTION:
False. ReLU acts on the activations, not the weights.

1.4 Consider 2 models which take as input an image of dimensions $4 \times 4 \times 3$ and output scores over 10 classes. Model 1 is made up of 1 convolution and 1 fully connected layer. Model 2 is made up of 1 fully connected layer. All input and output dimensions are noted in the figure above. Assert the following statement: Model 1 has more parameters than Model 2.



○ True
○ False

SOLUTION:
False. Model 1 has 2*2*3*2 + 2*2*2*10 = 104 params. Model 2 has 4*4*3*10 = 480 params.

1.5 LSTMs can be computationally expensive because their runtime scales quadratically with the length of the sequence.

○ True
○ False

SOLUTION:
False. It scales linearly with the length of the sequence.

1.6 You design a network that takes 2 images from the same ImageNet category and attempts to maximize the cosine similarity of their image embeddings. This is an example of self-supervised learning.

○ True
○ False

SOLUTION:
False. It is not self-supervised because it uses the ground truth category of the images.

1.7 When learning does converge, the initial point can determine how quickly learning converges and whether it converges to a point with high or low cost

SOLUTION:
True

○ True
○ False

1.8 In general, gradient descent with early stopping is theoretically equivalent to weight decay.

SOLUTION:
False

○ True

○ False

1.9 We may set the bias of a ReLU hidden unit to 0.1 rather than 0 to avoid saturating the ReLU at initialization.

SOLUTION:
True

○ True

○ False

1.10 You train one model with L2 regularization (model A) and one without (model B). The weights of model A will most likely be smaller in magnitude than those of model B.

SOLUTION:
True

○ True

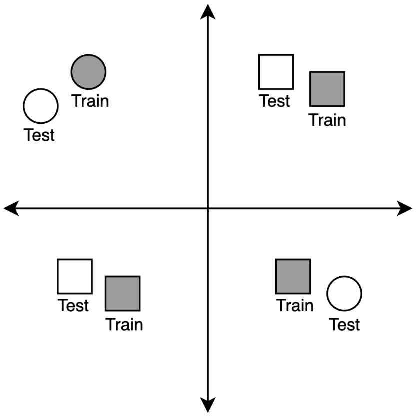
○ False

2 Multiple Choices (40 points) - Recommended 25 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 2-point deduction (up to 4 points).

2.1 Consider the dataset pictured below. The features of each datapoint are given by its position. So the datapoint (0,1) appears at position (0,1). The ground truth label of the datapoint is given by its shape, either circle or square. You have a test set of datapoints, shown with no fill, and a train set of data, shown with a grey fill. Which of the following statements are true about classifying this data?



○ A: It is possible for a linear SVM to have 100% train accuracy

○ B: It is possible for a linear SVM to have 100% test accuracy

○ C: KNN with K=1 has higher test accuracy than with K=4

○ D: KNN with K=1 has higher train accuracy than with K=4

○ E: None of the above

SOLUTION:
A, C, D

A is True because we are using a linear SVM and it is possible to linearly separate the train data.

B is False because it is not possible to linearly seperate the text data.

C is True because test accuracy is 75% when K=1, but 50% when K=4.

D is True because train accuracy is 100% when K=1, but 75% when K=4.

E is False

2.2 Why might you decide to train a neural network with softmax instead of using a KNN classifier on raw data?

○ A: You believe your data is not linearly separable.

○ B: Your train set is large, and you want a quick train time.

○ C: Your train set is large, and you want a quick evaluation time.

○ D: Your train set is large, and you want low memory costs at evaluation.

○ E: You do not believe you have hand-constructed features that are useful for classification.

SOLUTION:
C, D, E

A is False because KNN is not a linear classifier, so you stil might use a KNN even if you believe your data is not linearly separable.

B is False because neural networks generally need to train for a long time, while KNN do not.

C is True because neural networks have the same evaluation time regardless of train set size, but the evaluation time of KNNs scale with the train set size because you need to calculate the distance of a point to each train set and then sort them.

D is True because neural networks have the same memory cost regardless of train set size, but the memory cost time of KNNs scale with the train set size because you need to store the points to calculate the distances.

E is True because KNNs operate on some hand constructed features, but neural networks are able to learn features from the data.

2.3 You are training a neural network with SGD using the cross-entropy loss. You calculate the train loss, take the gradient, and update your network. However, you notice your train cross-entropy loss goes up. What could be a possible explanation for this?

○ A: You are overfitting to the training dataset.

○ B: You are evaluating on a different set of data then the previous step.

○ C: Your previous set of weights were as close to the global minimum as your learning rate will allow.

○ D: You are getting vanishing gradients.

○ E: Your regularizer is causing the cross-entropy loss to increase.

SOLUTION:
B, C, E

A is False because overfitting on the training set would not lead to train loss going up.

B is True because SGD evaluates and takes the gradient with respect to a random selection of data at each step, so the current step may have lead to a higher loss than the previous step.

C is True because if your learning rate is too high, then it only allows the loss to get to a certain level and then after that the next step will lead the loss to increase.

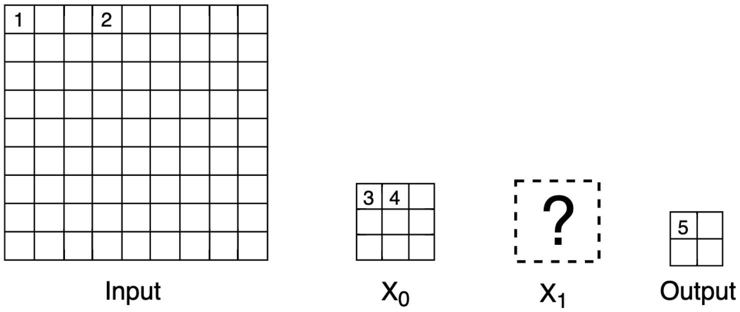
D is False as vanashing gradients explain slow/no loss change, while this network is having a loss that goes up.

E is True as a network optimizes a combination of the cross-entropy loss and the regularization term, so optimizing the regularization term may cause the cross-entropy loss to increase

2.4 Consider the following network, where $conv_1$ is a 3×3 convolution and $conv_2$ is a 2×2 convolution, both with no padding.

$$x_0 = conv_1(\text{input})$$
$$x_1 = \text{ReLU}(x_0)$$
$$\text{output} = conv_2(x_0)$$

We picture the WxH dimension of a number of tensors from the above algorithm.



Which of the following statements are true about the function and tensors in this algorithm?

○ A: x_1 is the same dimension as x_0.

○ B: The stride length of $conv_1$ is 1.

○ C: The stride length of $conv_2$ is 1.

○ D: Cell ‘1’ is in the receptive field of Cell ‘4’.

○ E: Cell ‘1’ is in the receptive field of Cell ‘5’.

○ F: Cell ‘2’ is in the receptive field of Cell ‘5’.

SOLUTION:
A, C, E, F

A is True because the ReLU function does not change the size of a tensor.

B is False because the stride length is 3, based on the kernel size and the input and output dimensions.

C is True because the stride length is 1, based on the kernel size and the input and output dimensions.

D is False

E is True

F is True

2.5 Let x be some image with ground truth label y for a classification problem between K classes. Let s_j denote a network's score for for class j. Consider the modified version of cross-entropy loss where we have variables a, b and c.

$$\mathcal{L}(x) = -\log\left(\frac{a \cdot e^{b \cdot s_y}}{\sum_{j=1}^{K} e^{s_j}}\right) + c$$

Now let $\frac{\partial \mathcal{L}}{\partial w}$ denote the gradient of the loss with respect to the final layer in the network. Which of the following is true about the relationship between $\frac{\partial \mathcal{L}}{\partial w}$, a, b, and c.

○ A: Increasing a could increase $\frac{\partial \mathcal{L}}{\partial w}$.

○ B: Decreasing a could increase $\frac{\partial \mathcal{L}}{\partial w}$.

○ C: Increasing b could increase $\frac{\partial \mathcal{L}}{\partial w}$.

○ D: Decreasing b could increase $\frac{\partial \mathcal{L}}{\partial w}$.

○ E: Increasing c could increase $\frac{\partial \mathcal{L}}{\partial w}$.

○ F: Decreasing c could increase $\frac{\partial \mathcal{L}}{\partial w}$.

SOLUTION:
C, D

Let us rewrite the expression as follows

$$\mathcal{L}(x) = -\log\left(\frac{a \cdot e^{b \cdot s_y}}{\sum_{j=1}^{K} e^{s_j}}\right) + c$$

$$= -\log\left(a \cdot \frac{e^{b \cdot s_y}}{\sum_{j=1}^{K} e^{s_j}}\right) + c$$

$$= -\left(\log(a) + \log\left(\frac{e^{b \cdot s_y}}{\sum_{j=1}^{K} e^{s_j}}\right)\right) + c$$

$$= -\log(a) - \log\left(\frac{e^{b \cdot s_y}}{\sum_{j=1}^{K} e^{s_j}}\right) + c$$

We can now see that a and c cannot affect $\frac{\partial \mathcal{L}}{\partial w}$, so A, B, E and F are False. However, changing b will affect $\frac{\partial \mathcal{L}}{\partial w}$ differently depending on if the current score is positive or negative and so therefore increasing or decreasing b could increase $\frac{\partial \mathcal{L}}{\partial w}$, so C and D are True.

2.6 A librarian/ deep learning engineer wants to create a model that classifies a sentence based on who wrote it between k different authors. She has two ideas for how to do this:

Network 1 - Supervised Model: Select a sentence x written by author y. Process the sentence with an RNN. Then apply a linear layer on the final state and apply softmax to produce a distribution over k possible authors. During training, use cross entropy loss. During inference, classify the sentence as the author with the highest probability.

Network 2 - Self-Supervised Model: Select a sentence x from book B_0. Then select another sentence x' from the same book. Next select n other sentences from n different books. Process all sentences independently with an RNN and then apply a linear layer to obtain an embedding of each sentence. During training use a contrastive loss to train the network to keep the embedding of x close to the embedding of x' and far from the other n negative examples. During inference, embed a pre-determinded sentence from each of the k authors. Then to classify a sentence, find its embedding using the RNN and match it to the closest of the k sentences.

Which of the following are true about these two networks?

○ A: Consider the case where the vast majority of the books in the training set do not have a known author. This is a reason to choose the Self-Supervised Model over the Supervised Model.

○ B: The librarian/ deep learning engineer realizes she might want to classify a sentence between $k+1$ authors instead of k authors sometime in the future after she is done training the network. This is a reason to choose the Supervised Model over the Self-Supervised Model.

○ C: Consider the case where the vast majority of the books in the training set are written by a single author. This is a reason to choose the Supervised Model over the Self-Supervised Model.

○ D: After she is done training the network, our librarian/ deep learning engineer wants to classify 10 sentences between 20 authors. She has unlimited storage/memory but only enough compute to run 1 forward pass at a time with batch size 1. For the Self-Supervised Model, she must run a minimum of 210 forward passes, but for the Supervised Model she must run a minimum of 10 forward passes.

SOLUTION:
A, C

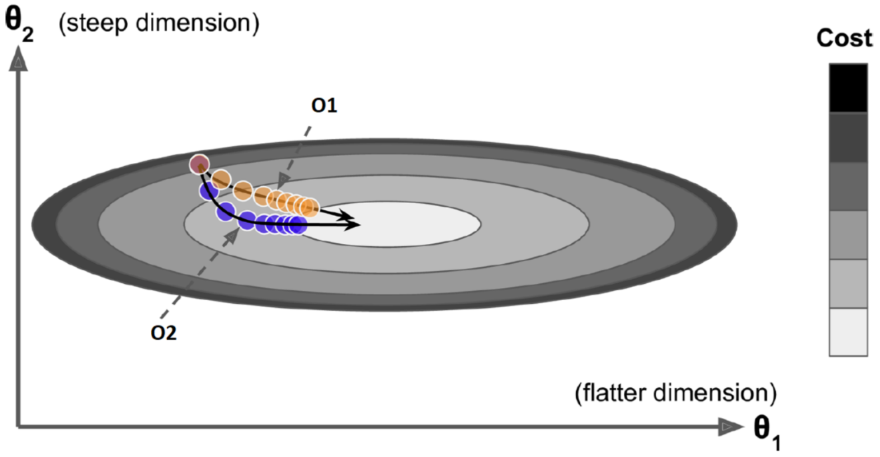
A is True because if many of the books do not have a known author, we would not be able to use them at all in our Supervised Model, but could still take advantage of this data in our Self-Supervised Model

B is False. The Self-Supervised Model would not have to be retrained to accommodate another author, but the Supervised Model would need to be (at least in the final classification layer, to make it predict between $k+1$ authors)

C is True. If all the books are written by a single author this would mess up the training scheme in the Self-Supervised Model because when we contrast two different sentences from two different books, these sentences are actually likely by the same author. In other words, both our positive and negative examples are likely by the same author.

D is False because we are able to store the representations for our comparison sentences from each of the 20 authors so we do not need to recompute them each time.

2.7 The figure below compares AdaGrad and Gradient Descent(without momentum) optimization. Which of the following is/are true?



○ A: O1 corresponds to AdaGrad while O2 corresponds to Gradient Descent

○ B: O1 corresponds to Gradient Descent while O2 corresponds to AdaGrad

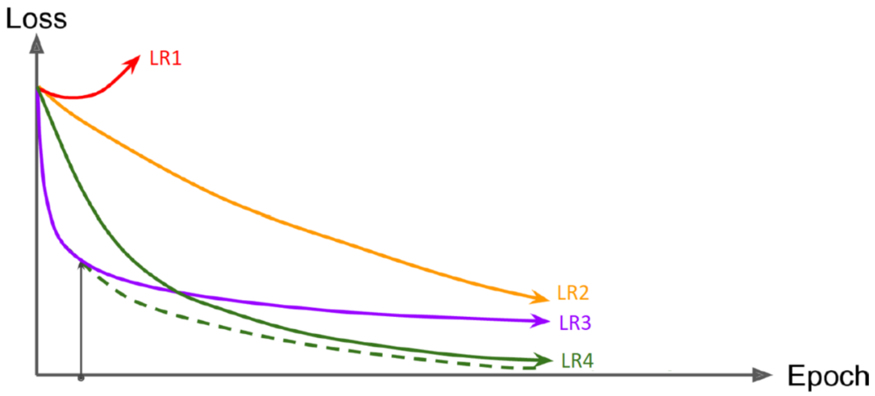
○ C: O1 helps point the resulting updates more directly toward the global optimum compared to O2

○ D: O2 helps point the resulting updates more directly toward the global optimum compared to O1

SOLUTION:
A, C

AdaGrad has a smoother trajectory than GD without momentum and generally leads to a more direct path towards the global optimum.

2.8 The below figure shows learning curves for various learning rates. What is the correct order of learning rates?



○ A: $LR1 > LR2 > LR3 > LR4$

○ B: $LR2 > LR3 > LR1 > LR4$

○ C: $LR1 > LR3 > LR4 > LR2$

○ D: $LR3 > LR1 > LR2 > LR4$

SOLUTION:
C

L1 is the largest learning rate as we see it diverges away from a low loss. Additionally, we know that LR3 > LR2 as LR3 drops quickly but then plateaus while LR2 continues to drop in loss, but very slowly. Therefore the correct answer is C.

2.9 Your notice your vanilla RNN has a vanishing gradient problem. Which one(s) of the following methods can help?

○ A: Use gradient clipping

○ B: Add more RNN layers.

○ C: Add more training data.

○ D: Replace vanilla RNN with LSTM or GRU.

SOLUTION:
D

A if False because gradient clipping can help exploding gradients, not vanishing gradients

B is False because adding more layers can only increase the vanishing gradient problem.

C is False because the amount of training data has not affect on vanishing gradients.

D is True because LSTMs and GRUs have a cell state that acts as a "gradient highway" to prevent vanishing gradients.

2.10 You are tasked with training a model to accurately predict whether it will rain in the University District in the month of May on a day-to-day basis (good luck!). During your hyperparameter search, you consider two options: try to babysit one model ("Panda" strategy) or train a lot of models in parallel ("Caviar"). Your choice of strategy is largely determined by:

○ A: Whether you use batch or mini-batch optimization

○ B: The presence of local minima (and saddle points) in your neural network

○ C: The amount of computational power you can access

○ D: The number of hyperparameters you have to tune

SOLUTION:
C

The only thing that affect this is the amount of compute you have.

3 Short Answers (40 points) - Recommended 45 Minutes

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

3.1 Accuracy

We use accuracy as a metric for measuring the strength of our machine learning classifiers. Let us play with a couple of them to understand the metric better.

3.1.1 Multiclass Classification (2 points)

You evaluate a neural network on the CIFAR-10 testset and get 0% accuracy. Answer the following questions in 1-2 sentences: (a) Is this network likely randomly initialized and untrained? (b) Give a strategy to improve accuracy on this network by approximately 10%.

SOLUTION:
(a) No, it is not because the random accuracy for a 10-class classification on a balanced test set like CIFAR will be around 10%

(b) Predict the same class for every input image.

3.1.2 Binary Classification (2 points)

You evaluate a binary classifier on a balanced testset and get 20% accuracy. Answer the following questions in 1-2 sentences: (a) Give a strategy to improve accuracy on this network by approximately 30%. (b) Give a strategy to improve accuracy on this network by approximately 60%.

SOLUTION:
(a) Predict the same label for all the data

(b) Flip the sign of the prediction ie., for a prediction that is class 1, label is class 2 and vice versa

3.2 Distances (3 points)

There are two unit vectors $x, y \in \mathcal{R}^n$. What is the Euclidean distance (D_E) in terms of cosine similarity (D_C) between these two vectors (x, y)?

Euclidean distance $= \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ Cosine similarity $= \frac{x \cdot y}{\|x\|\|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

SOLUTION:
$D_E = \sqrt{(2 - 2 * D_C)}$

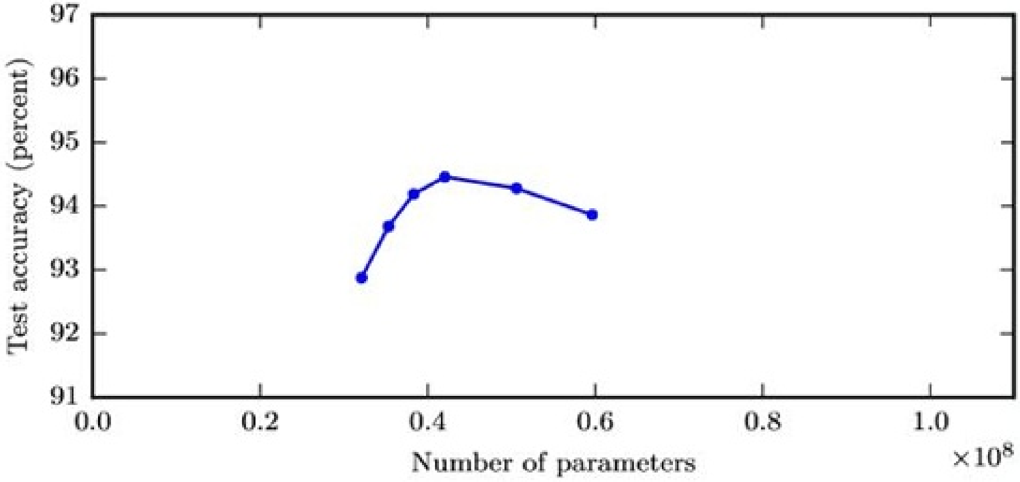
3.3 Batch Norm (2 points)

Is there a problem with using batch normalization on a batch size of 1 during **training** and **testing**? If so, what? (1-2 sentences for each setting independently)

SOLUTION:
Yes for training because of no batch-wide statistics and No for testing.

3.4 CNN Width (4 points)

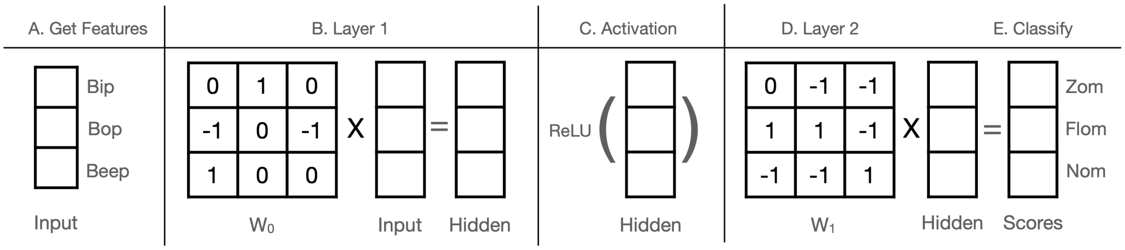
The below graph shows the accuracy of a trained 3-layer convolutional neural network vs the number of parameters (i.e. number of feature kernels). The trend suggests that as you increase the width of a neural network, the accuracy increases till a certain threshold value, and then starts decreasing. What could be the possible reason for this decrease?



SOLUTION:
Increasing the number of filter kernels is a way to increase model capacity/complexity. Too many filter kernels may overfit the training distribution and lead to bad generalizability on the test set.

3.5 Two layer network

You have a series of datapoints which can be classified as one of 3 classes: Zom, Flom, or Nom. Each data point is composed of 3 features: Bip value, Bop value or Beep value. To train a 2 layer network (with no bias term) to predict if something is a Zom, Flom or a Nom based on its Bip, Bop and Beep values. After training your two layer network to high accuracy, you get the following network. Answer the following question about this network. Note: Layer 1 and Layer 2 are matrix-vector multiplications as in any neural network layer.



3.5.1 Running prediction (6 points)

1. (3 points) Consider a data point with Bip = 0, Bop = 1 and Beep = 1. How will the model classify this point?

   SOLUTION:

   Flom

2. (3 points) Consider a data point with Bip = -1, Bop = 0 and Beep = 1. How will the model classify this point?

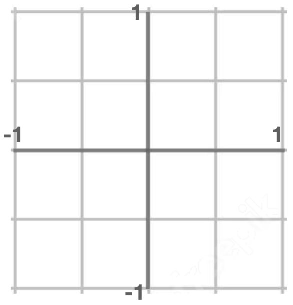
   SOLUTION:

   Tie for all classes

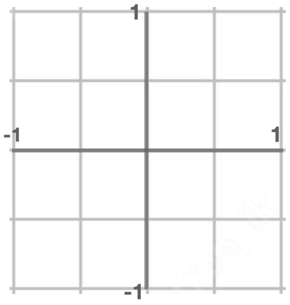
3.5.2 Visualizing + Interpretting the model (15 points)

3. (9 points) Lets start to understand how changing different values affects the classification score of Zom. In other words, we will figure out what the model believes are the attribute of a Zom. On the plots below, plot how the score of a Zom changes as you change each of the values. For each plot assume that that the two values that you are not considering are set at 0. So if you are plotting Bip Value vs Zom score, you can assume Bop and Beep value are set at 0. Your Bip/Bop/Beep value is represented by the x-axis of the plot, while the Zom score is the y-axis of the plot.

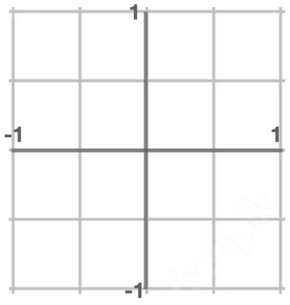
Zom Score vs Bip Value



Zom Score vs Bop Value

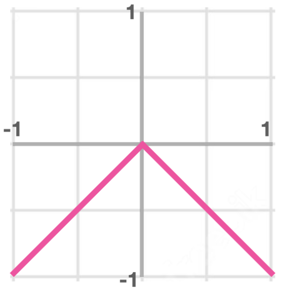


Zom Score vs Beep Value

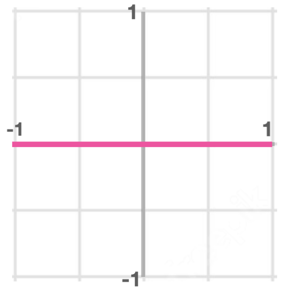


SOLUTION:

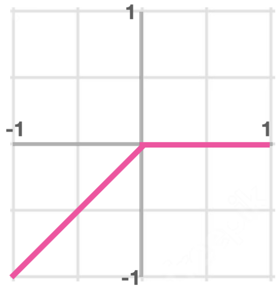
Zom Score vs Bip Value



Zom Score vs Bop Value



Zom Score vs Beep Value



4. (6 points) Now use your plots in order to describe the relationship between these two values. For example, if your plot just looks like a line with a positive constant slope your interpretation might be "A large positive value indicates that something is a Zom, and a large negative value indicates that something is not a Zom". Write a similar sentence for each of the 3 values:

Bip Value Interpretation:

SOLUTION:
A large negative or positive score indicates that something is not a Zom

Bop Value Interpretation:

SOLUTION:
Bop value has no affect on Zom score

Beep Value Interpretation:

SOLUTION:
A large negative value indicates that something is not a Zom

3.5.3 Getting Gradients (Extra Credit - 6 points)

You now decide to calculate the gradients of the model with respect to a single input. The input has Bip = -1, Bop = 1 and Beep = 0, and has a ground truth label of 'Flom'. Fill in the **sign** of the gradient for each element of W_1 (either + or - or 0, to indicate positive gradient, negative gradient or zero gradient) for the following loss functions.

5. (3 points) Loss function = SVM loss function with margin of 1:

0	-1	-1
1	1	-1
-1	-1	1

W_1

(Fill me in!)

$\text{sign}(\frac{dL}{dW_1})$

SOLUTION:
Loss is 0 because score of correct answer is more than 1 greater than all incorrect answers.

0	-1	-1
1	1	-1
-1	-1	1

W_1

(Fill me in!)

0	0	0
0	0	0
0	0	0

$\text{sign}(\frac{dL}{dW_1})$

6. (3 points) Loss function = Cross entropy loss:

0	-1	-1
1	1	-1
-1	-1	1

W_1

(Fill me in!)

$\text{sign}(\frac{dL}{dW_1})$

SOLUTION:

0	-1	-1
1	1	-1
-1	-1	1

W_1

(Fill me in!)

+	+	0
-	-	0
+	+	0

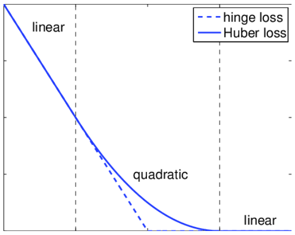
sign($\frac{dL}{dW_1}$)

3.6 Hinge Loss Variants

In class, we learned that one characteristic of multi-class hinge loss was that it was not differentiable at the hinge point; in this question we consider two variations that address this problem.

3.6.1 Huber Loss (6 points)

One variant of the Hinge loss is the Huber classification loss; the following plot shows the comparison between the two.



Recall that given input vector x_i, true label y_i, weight matrix W, the class score is calculated as $s = x_i^T W = (s_1, s_2, \dots s_C)$, and the hinge loss of class $j \neq y_i$ for some given margin Δ is

$$L_j(x_i) = \max(s_j - s_{y_i} + \Delta, 0)$$

Now the Huber classification loss looks like this for some given parameters a, b and Δ:

$$L_j(x_i) = \begin{cases} a \cdot \max(s_j - s_{y_i} + \Delta, 0)^2, & \text{if } s_j \leq s_{y_i} \\ b \cdot (s_j - s_{y_i} + 0.5\Delta), & \text{otherwise} \end{cases}$$

1. (3 points) Same as the multiclass hinge loss, the Huber loss incurred by one point is the sum of the losses over each incorrect class, i.e. $L(x_i) = \sum_{j \neq y_i} L_j(x_i)$. Derive the partial derivative of the single-datapoint Huber loss $\frac{\partial L}{\partial s_j}$ for $j \neq y_i$ (No regularization term needed.)

SOLUTION:

$$\frac{\partial}{\partial s_j} L(x_i) = \begin{cases} b & \text{if } s_j - s_{y_i} > 0 \\ 2a(s_j - s_{y_i} + \Delta), & \text{if } 0 \geq s_j - s_{y_i} > -\Delta \\ 0 & \text{otherwise} \end{cases}$$

2. (3 points) For this loss function to be differentiable everywhere, the function value and the first order partial derivative have to be consistent at the hinges, i.e. the function values and gradients obtained by approaching from $s_j \leq s_{y_i}$ and from $s_j > s_{y_i}$ should be the same. Given a fixed Δ, what constraint(s) should the values of a and b satisfy?

SOLUTION:

$a = \frac{b}{2\Delta}$.

3.6.2 Smooth Hinge Loss (Extra Credit - 6 points)

Now let's consider the following form:

$$L = \max(\max_{j \neq y_i}\{s_j\} - s_{y_i}, 0)$$

This is slightly different than what we've seen in class, where instead of summing over all the non-ground-truth classes, we take the maximum of them.

1. (4 points) Please rewrite the above expression into a smooth (i.e. differentiable) version of hinge loss using the following continuous relaxation:

$$\max(x, y) = \lim_{k \to \infty} \frac{1}{k} \ln(e^{kx} + e^{ky}) \approx \log(e^x + e^y) \tag{1}$$

$$\max(x_1, \ldots, x_n) = \lim_{k \to \infty} \frac{1}{k} \ln(\sum_{i=1}^{n} e^{kx_i}) \approx \log(\sum_{i=1}^{n} e^{x_i}) \tag{2}$$

Your answer should resemble a loss function that we've seen in class.

SOLUTION:

$$
\begin{aligned}
\mathcal{L} &= \max(\max_{i\neq y}\{z_i\} - z_j, 0) \\
&= \max\left(\log(\sum_{i=1,i\neq y}^{C} e^{z_i}) - z_y, 0\right) \\
&\approx \log\left(1 + e^{\log\left(\sum_{i=1,i\neq y}^{C} e^{z_i}\right) - z_y}\right) \\
&= \log\left(1 + \frac{\sum_{i=1,i\neq y}^{C} e^{z_i}}{e^{z_y}}\right) \\
&= \log\frac{\sum_{i=1}^{C} e^{z_i}}{e^{z_y}} \\
&= -\log\frac{e^{z_y}}{\sum_{i=1}^{C} e^{z_i}} \qquad (3)
\end{aligned}
$$

Wow, we get the softmax function here.

2. (2 points) A margin Δ is usually introduced in hinge loss to encourage large intra-class distances. This takes the form

$$L = \max(\max_{j\neq y_i}\{s_j\} - s_{y_i} + \Delta, 0)$$

Using your answer above, please write down the smooth hinge loss when margin Δ is used.

SOLUTION:

$$\mathcal{L} = -\log\frac{e^{s_{y_i}-\Delta}}{e^{s_{y_i}-\Delta} + \sum_{j\neq y_i} e^{s_j}}$$

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

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