

CS230: Deep Learning

Winter Quarter 2019

Stanford University

Midterm Examination

180 minutes

	Problem	Full Points	Your Score
1	Multiple Choice Questions	13	
2	Short Answer Questions	23	
3	Convolutional Neural Networks	28	
4	Adversarial Attacks	10	
5	Loss comparisons	23	
6	The Optimizer	20	
Total		117	

The exam contains 29 pages including this cover page.

- This exam is **closed book i.e. no laptops, notes, textbooks, etc. during the exam**. However, you may use one A4 sheet (front and back) of notes as reference.
- In all cases, and especially if you're stuck or unsure of your answers, **explain your work, including showing your calculations and derivations!** We'll give partial credit for good explanations of what you were trying to do.

Name: _____

SUNETID: _____@stanford.edu

The Stanford University Honor Code:

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.

Signature: _____

Question 1 (Multiple Choice Questions, 13 points)

For each of the following questions, circle the letter of your choice. There is only ONE correct choice unless explicitly mentioned. No explanation is required. There is no penalty for a wrong answer.

- (a) **(1 point)** Consider a Generative Adversarial Network (GAN) which successfully produces images of apples. Which of the following propositions is **false**?
- (i) The generator aims to learn the distribution of apple images.
 - (ii) The discriminator can be used to classify images as apple vs. non-apple.
 - (iii) After training the GAN, the discriminator loss eventually reaches a constant value.
 - (iv) The generator can produce unseen images of apples.

Solution: (ii)

- (b) **(1 point)** Which of the following activation functions can lead to vanishing gradients?
- (i) ReLU
 - (ii) Tanh
 - (iii) Leaky ReLU
 - (iv) None of the above

Solution: (ii)

- (c) **(1 point)** Consider a univariate regression $\hat{y} = wx$ where $w \in \mathbb{R}$, and $x \in \mathbb{R}^{1 \times m}$. The cost function is the squared-error cost $\mathcal{J} = \frac{1}{m} \| \hat{y} - y \|^2$. Which of the following equations is **true**?
- (i) $\frac{\partial \mathcal{J}}{\partial w} = \frac{1}{m} (\hat{y} - y) x^T$
 - (ii) $\frac{\partial \mathcal{J}}{\partial w} = \frac{1}{m} (\hat{y} - y) x$
 - (iii) $\frac{\partial \mathcal{J}}{\partial w} = \frac{2}{m} (\hat{y} - y) x^T$
 - (iv) $\frac{\partial \mathcal{J}}{\partial w} = \frac{2}{m} (\hat{y} - y) x$

Solution: (iii)

- (d) **(1 point)** Which of the following costs is the non-saturating generator cost for GANs (G is the generator and D is the discriminator)?

- (i) $J^{(G)} = \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^{(i)})))$
- (ii) $J^{(G)} = -\frac{1}{m} \sum_{i=1}^m \log(D(G(z^{(i)})))$
- (iii) $J^{(G)} = \frac{1}{m} \sum_{i=1}^m \log(1 - G(D(z^{(i)})))$
- (iv) $J^{(G)} = -\frac{1}{m} \sum_{i=1}^m \log(G(D(z^{(i)})))$

Solution: (ii)

- (e) **(1 point)** After training a neural network, you observe a large gap between the training accuracy (100%) and the test accuracy (42%). Which of the following methods is commonly used to reduce this gap?

- (i) Generative Adversarial Networks
- (ii) Dropout
- (iii) Sigmoid activation
- (iv) RMSprop optimizer

Solution: (ii)

- (f) **(1 point)** Which of the following is **true** about Batchnorm?

- (i) Batchnorm is another way of performing dropout.
- (ii) Batchnorm makes training faster.
- (iii) In Batchnorm, the mean is computed over the features.
- (iv) Batchnorm is a non-linear transformation to center the dataset around the origin

Solution: (ii)

- (g) **(1 point)** Which of the following statements is **true** about Xavier Initialization?

- (i) It is only used in fully connected neural networks.
- (ii) It applies a scaling factor to the mean of the random weights.
- (iii) It is commonly used in logistic regression.
- (iv) The assumptions made are only valid at the beginning of training.

Solution: (iv)

- (h) **(1 point)** When should multi-task learning be used?
- (i) When your problem involves more than one class label.
 - (ii) When two tasks have the same dataset.
 - (iii) When you have a small amount of data for a particular task that would benefit from the large dataset of another task.
 - (iv) When the tasks have datasets of different formats (text and images).

Solution: (iii)

- (i) **(1 point)** Which of the following is an advantage of end-to-end learning? (**Check all that apply.**)
- (i) It usually requires less data.
 - (ii) It doesn't need hand crafted features.
 - (iii) It generally leads to lower bias.
 - (iv) None of the above.

Solution: (ii), (iii) We accepted any combination of the above two answers as correct.

- (j) **(2 points)** Which of the following propositions are true about a CONV layer? (**Check all that apply.**)
- (i) The number of weights depends on the depth of the input volume.
 - (ii) The number of biases is equal to the number of filters.
 - (iii) The total number of parameters depends on the stride.
 - (iv) The total number of parameters depends on the padding.

Solution: (i), (ii)

- (k) **(1 point)** What is Error Analysis?
- (i) The process of analyzing the performance of a model through metrics such as precision, recall or F1-score.
 - (ii) The process of scanning mis-classified examples to identify weaknesses of a model.
 - (iii) The process of tuning hyperparameters to reduce the loss function during training.
 - (iv) The process of identifying which parts of your model contributed to the error.

Solution: (ii)

- (1) (1 point) Which of the following is a non-iterative method to generate adversarial examples?
- (i) Non-Saturating Cost Method
 - (ii) Input Optimization Method
 - (iii) Adversarial Training
 - (iv) Logit Pairing
 - (v) Fast Gradient Sign Method
 - (vi) Real-time Cryptographic Dropout Method

Solution: (v)

Question 2 (Short Answer Questions, 23 points)

Please write concise answers.

- (a) **(2 points)** How does splitting a dataset into train, dev and test sets help identify overfitting?

Solution:

- Overfitting: the model fits the training set so much that it does not generalize well.
- Low training error and high dev error can be used to identify this
- Must ensure that the distribution of train and dev is the same/similar!

- (b) **(2 points)** Which regularization method leads to weight sparsity? Explain why.

Solution: L1 regularization leads to weight sparsity. This comes from the shape of the L1 loss. Since even small weights are penalised the same amount as large weights, more weight values will tend closer to 0.

L2 on the other hand penalizes smaller weights less, which leads to smaller weights but does not ensure sparsity.

- (c) **(2 points)** You are designing a deep learning system to detect driver fatigue in cars. It is crucial that that your model detects fatigue, to prevent any accidents. Which of the following is the most appropriate evaluation metric: Accuracy, Precision, Recall, Loss Value. Explain your choice.

Solution: Recall.

It is important that we do not miss any cases where the driver is tired.

- (d) **(4 points)** You have a single hidden-layer neural network for a binary classification task. The input is $X \in \mathbb{R}^{n \times m}$, output $\hat{y} \in \mathbb{R}^{1 \times m}$ and true label $y \in \mathbb{R}^{1 \times m}$. The forward propagation equations are:

$$z^{[1]} = W^{[1]}X + b^{[1]}$$

$$a^{[1]} = \sigma(z^{[1]})$$

$$\hat{y} = a^{[1]}$$

$$\mathcal{J} = - \sum_{i=1}^m y^{(i)} \log(\hat{y}^{[i]}) + (1 - y^{(i)}) \log(1 - \hat{y}^{[i]})$$

Write the expression for $\frac{\partial \mathcal{J}}{\partial W^{[1]}}$ as a matrix product of two terms.

Solution:

$$\frac{\partial \mathcal{J}}{\partial \hat{W}^{[1]}} = (\hat{y} - y)X^T$$

- (e) **(3 points)** You want to solve a classification task. You first train your network on 20 samples. Training converges, but the training loss is very high. You then decide to train this network on 10,000 examples. Is your approach to fixing the problem correct? If yes, explain the most likely results of training with 10,000 examples. If not, give a solution to this problem.

Solution: The model is suffering from a bias problem.

Increasing the amount of data reduces the variance, and is not likely to solve the problem.

A better approach would be to decrease the bias of the model by maybe adding more layers/ learnable parameters. It is possible that training converged to a local optimum. Training longer/using a better optimizer/ restarting from a different initialization could also work.

- (f) **(2 points)** Give two benefits of using convolutional layers instead of fully connected ones for visual tasks.

Solution:

- Uses spatial context (by only assigning weights to nearby pixels)
- Translation invariance
- Have lot less parameters, since CNN's share weights

- (g) **(2 points)** You have a dataset \mathcal{D}_1 with 1 million labelled training examples for classification, and dataset \mathcal{D}_2 with 100 labelled training examples. Your friend trains a model from scratch on dataset \mathcal{D}_2 . You decide to train on \mathcal{D}_1 , and then apply transfer learning to train on \mathcal{D}_2 . State one problem your friend is likely to find with his approach. How does your approach address this problem?

Solution: Friend is likely to see overfitting. Model is not going to generalise well to unseen data. By using transfer learning and freezing the weights in the earlier layers, you reduce the number of learnable parameters, while using the weights which have been pretrained on a much larger dataset.

- (h) **(2 points)** You are solving the binary classification task of classifying images as cat vs. non-cat. You design a CNN with a single output neuron. Let the output of this neuron be z . The final output of your network, \hat{y} is given by:

$$\hat{y} = \sigma(\text{ReLU}(z))$$

You classify all inputs with a final value $\hat{y} \geq 0.5$ as cat images. What problem are you going to encounter?

Solution: Using ReLU then sigmoid will cause all predictions to be positive ($\sigma(\text{ReLU}(z)) \geq 0.5 \quad \forall z$).

- (i) **(2 points)** You are given a content image X_C and a style image, X_S . You would like to apply neural style transfer to obtain an output image Y , with the content of X_C and the style of X_S , as discussed in section. You are told that you need a pretrained VGG-16 network to do this. What is the function of this pretrained network?

Solution: The pretrained network is used to extract the content and the style from the two images. Intermediate features are used to extract the content, and the gram matrix is used to extract the style.

- (j) **(2 points)** You are given the following piece of code for forward propagation through a single hidden layer in a neural network. This layer uses the sigmoid activation. Identify and correct the error.


```
import numpy as np

def forward_prop(W, a_prev, b):
    z = W*a_prev + b
    a = 1/(1+np.exp(-z)) #sigmoid
    return a
```

Solution: $z = \text{np.matmul}(W, a_prev) + b$ OR $z = \text{np.dot}(W, a_prev) + b$

Question 3 (Convolutional Neural Networks, 28 points)

Two historians approach you for your deep learning expertise. They want to classify images of historical objects into 3 classes depending on the time they were created:

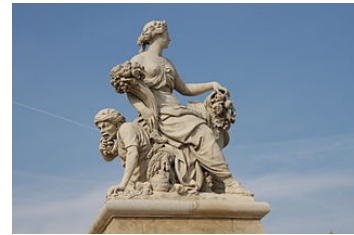
- Antiquity ($y = 0$)
- Middle Ages ($y = 1$)
- Modern Era ($y = 2$)



(A) Class: Antiquity



(B) Class: Middle Ages



(C) Class: Modern Era

Figure 1: Example of images found in the dataset along with their classes

- (a) Over the last few years, the historians have collected nearly 5,000 hand-labelled RGB images.
- (i) **(2 points)** Before training your model, you want to decide the image resolution to be used. Why is the choice of image resolution important?

Solution: Trade-off between accuracy and model complexity.

- (ii) **(1 point)** If you had 1 hour to choose the resolution to be used, what would you do?

Solution: (See lecture of Prof. Katanforoosh)
Print pictures of images with different resolutions and ask friends if they can properly recognize images.

- (b) You have now figured out a good image resolution to use.

- (i) **(2 points)** How would you partition your dataset? Formulate your answer in percentages.

Solution: Several ratios possible. One way of doing it: split the initial dataset into 64% training/16% dev/20% testing set. Training on the training set and tuning the hyperparameters after looking at the performance on the dev set.

- (ii) **(3 points)** After visually inspecting the dataset, you realize that the training set only contains pictures taken during the day, whereas the dev set only has pictures taken at night. Explain what is the issue **and** how you would correct it.

Solution:

- It can cause a domain mismatch.
- The difference in the distribution of the images between training and dev might lead to faulty hyperparameter tuning on the dev set, resulting in poor performance on unseen data.
- Solution: randomly mix pictures taken at day and at night in the two sets and then resplit the data.

- (c) **(3 points)** As you train your model, you realize that you do not have enough data. Cite 3 data augmentation techniques that can be used to overcome the shortage of data.

Solution: A lot of answers can be accepted, including Rotation, Cropping, Flipping, Luminosity/Contrast Changes

- (d) **(8 points)** You come up with a CNN classifier. For each layer, calculate the number of weights, number of biases and the size of the associated feature maps.

The notation follows the convention:

- CONV- K - N denotes a convolutional layer with N filters, each them of size $K \times K$, Padding and stride parameters are always 0 and 1 respectively.
- POOL- K indicates a $K \times K$ pooling layer with stride K and padding 0.
- FC- N stands for a fully-connected layer with N neurons.

Layer	Activation map dimensions	Number of weights	Number of biases
INPUT	$128 \times 128 \times 3$	0	0
CONV-9-32			
POOL-2			
CONV-5-64			
POOL-2			
CONV-5-64			
POOL-2			
FC-3			

Solution: Successively:

- $120 \times 120 \times 32$ and $32 \times (9 \times 9 \times 3 + 1)$
- $60 \times 60 \times 32$ and 0
- $56 \times 56 \times 64$ and $64 \times (5 \times 5 \times 32 + 1)$
- $28 \times 28 \times 64$ and 0
- $24 \times 24 \times 64$ and $64 \times (5 \times 5 \times 64 + 1)$
- $12 \times 12 \times 64$ and 0
- 3 and $3 \times (12 \times 12 \times 64 + 1)$

- (e) **(2 points)** Why is it important to place non-linearities between the layers of neural networks?

Solution: Non-linearity introduces more degrees of freedom to the model. It lets it capture more complex representations which can be used towards the task at hand. A deep neural network without non-linearities is essentially a linear regression.

- (f) **(3 points)** Following the last FC-3 layer of your network, what activation must be applied? Given a vector $a = [0.3, 0.3, 0.3]$, what is the result of using your activation on this vector?

Solution: Softmax is the one that is used as it can output class probabilities. Output is $[0.33, 0.33, 0.33]$. (You don't need a calculator!)

- (g) You find online that the exact same network has already been trained on 1,000,000 historical objects from a slightly different time period.

- (i) (1 point) What is the name of the method that could reuse these pretrained weights for the task at hand?

Solution: Transfer learning

(ii) (3 points) What are the new hyperparameters to tune for this method?

Solution: How many layers of the pretrained model to freeze, how many extra layers to add, and how many of the weights in the new network we want to initialize with the pretrained weights.

Question 4 (Adversarial Attacks, 10 points)

Alice and Bob work on a self-driving car project. They want to classify various traffic signs among 10 different classes. Bob has trained a deep convolutional neural network (CNN), f , on a dataset with 100,000 samples. Given an input image \mathbf{x} , his model predicts $\hat{y} = f(\mathbf{x})$. Overall, it achieves 95.6% test accuracy.

- (a) **(2 points)** Alice has recently heard about adversarial attacks and is worried about the problems they could cause. To show Bob the potential dangers of adversarial attacks, she decides to design an input \mathbf{x} which is classified as a "STOP" sign by Bob's CNN. Propose a loss function for this task, and explicitly state the parameter(s) being optimized. You are not allowed to use any images other than \mathbf{x} for this optimization.

Solution:

$$\frac{1}{2} \|f(x) - y_{\text{STOP}}\|^2$$

and x is being optimized

- (b) **(2 points)** You run the optimization in part (a). Will the generated image look like a real image? Explain why.

Solution: No, Alice's optimized \mathbf{x} will likely look like noise image since the space of inputs classified as STOP sign is much bigger than the space of images looking real to the human eye.

- (c) **(3 points)** Alice looks for better evidence to convince Bob that his trained CNN is not a robust classifier. She decides to take the image $\mathbf{x}_{\text{no park}}$, which is a real image of a "No Parking" sign, and finds an input \mathbf{x} such that:
- \mathbf{x} looks similar to $\mathbf{x}_{\text{no park}}$
 - \mathbf{x} is classified by Bob's network as a "STOP" sign. i.e. $f(\mathbf{x}) = \hat{y}_{\text{STOP}}$

Give the cost function for an iterative method which will achieve the above two objectives.

Solution:

$$\frac{1}{2} \|f(\mathbf{x}) - \hat{y}_{\text{STOP}}\|^2 + \lambda \|\mathbf{x} - \mathbf{x}_{\text{no park}}\|^2$$



Figure 2: Input $\mathbf{x}_{\text{no park}}$

- (d) **(3 points)** After seeing the results of Alice's experiments, Bob decides to retrain the deep convolutional network in a way that the trained classifier would be robust to adversarial attacks. Suggest two different solutions for improving the robustness of his CNN classifier.

Solution:

- Creating a safety-net
- Training using adversarial examples
- Adversarial training

Question 5 (Loss comparisons, 23 points)

Part I. You want to perform a regression task with the following dataset: $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \mathbb{R}$, $i = 1, \dots, m$ are the the i th example and output in the dataset, respectively. Denote the prediction for example i by $f(x^{(i)})$. Remember that for a given loss \mathcal{L} we minimize the following cost function

$$\mathcal{J} = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(f(x^{(i)}), y^{(i)}).$$

In this part we are deciding between using loss 1 and loss 2, given by:

$$\begin{aligned} \mathcal{L}_1(f(x^{(i)}), y^{(i)}) &= |y^{(i)} - f(x^{(i)})|, \\ \mathcal{L}_2(f(x^{(i)}), y^{(i)}) &= (y^{(i)} - f(x^{(i)}))^2. \end{aligned}$$

- (a) **(4 points)** Draw $\mathcal{L}_1(x, 0)$ and $\mathcal{L}_2(x, 0)$ versus $x \in \mathbb{R}$ on the same plot.

Solution:

- (b) **(2 points)** An outlier is a datapoint which is very different from other datapoints of the same class. Based on your plots, which method do you think works better when there is a large number of outliers in your dataset? Hint: Contributions of outliers to gradient calculations should be as small as possible.

Solution: L_1 . The reason is it penalizes less for outliers. We would like to ignore outliers if possible.

When it isn't possible, using a loss function which penalises outliers less, is more robust.

- (c) **(3 points)** "Using \mathcal{L}_1 loss enforces sparsity on the weights of the network." Do you agree with this statement? Why/Why not?

Solution: This is false because in this case, the residual will be forced to be sparse not the weights.

- (d) **(3 points)** "Using \mathcal{L}_2 loss forces the weights of the network to end up small." Do you agree with this statement? Why/Why not?

Solution: This is false because in this case the residuals will be forced to be small not the weights.

Part II. You want to perform a classification task. You are hesitant between two choices: Approach A and Approach B. The only difference between these two approaches is the loss function that is minimized. Assume that $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \{+1, -1\}$, $i = 1, \dots, m$ are the i th example and output label in the dataset, respectively. $f(x^{(i)})$ denotes the output of the classifier for the i th example. Recall that for a given loss \mathcal{L} you minimize the following cost function:

$$\mathcal{J} = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(f(x^{(i)}), y^{(i)}).$$

As we mentioned, the only difference between approach A and approach B is the choice of the loss function:

$$\begin{aligned} \mathcal{L}_A(f(x^{(i)}), y^{(i)}) &= \max\{0, 1 - y^{(i)} f(x^{(i)})\}, \\ \mathcal{L}_B(f(x^{(i)}), y^{(i)}) &= \log_2(1 + \exp(-y^{(i)} f(x^{(i)}))). \end{aligned}$$

(e) Consider \mathcal{L}_B .

(i) (2 points) Rewrite \mathcal{L}_B in terms of the sigmoid function.

Solution:

$$\mathcal{L}_B(f(x^{(i)}), y^{(i)}) = -\log_2(\sigma(y^{(i)} x^{(i)}))$$

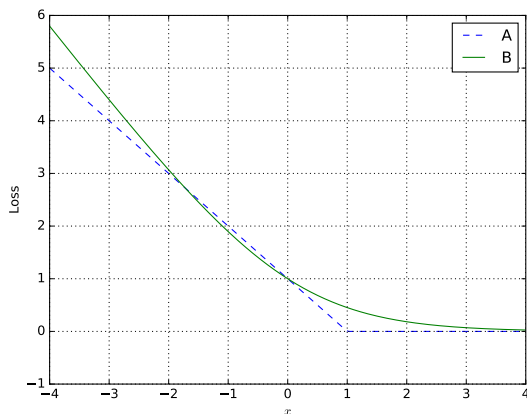
(ii) (2 points) You are given an example with $y^{(i)} = -1$. What value of $f(x^{(i)})$ will minimize \mathcal{L}_B ?

Solution: $f(x^{(i)}) = -\infty$

(iii) (2 points) You are given an example with $y^{(i)} = -1$. What is the greatest value of $f(x^{(i)})$ that will minimize \mathcal{L}_A ?

Solution: $f(x^{(i)}) = -1$

The following is the plot of $L_A(x, 1)$ and $L_B(x, 1)$ versus $x \in \mathbb{R}$.



- (f) **(2 points)** Based on these plots, which method would work better when there is a large number of outliers in the dataset? Explain why.

Solution: Method A. Because it penalizes less than method B for outliers.

- (g) **(3 points)** You would like a classifier whose output can be interpreted as a probability. Which loss function is better to optimize? Justify your choice.

Solution: \mathcal{L}_B . From the previous part, we know that \mathcal{L}_B can be written as minimizing the log of the sigmoid of the output. This means that the sigmoid of the output can be interpreted as a probability. \mathcal{L}_A has no such interpretation. (Also recognizing that \mathcal{L}_B is the logistic loss helps).

Question 6 (The Optimizer, 20 points)

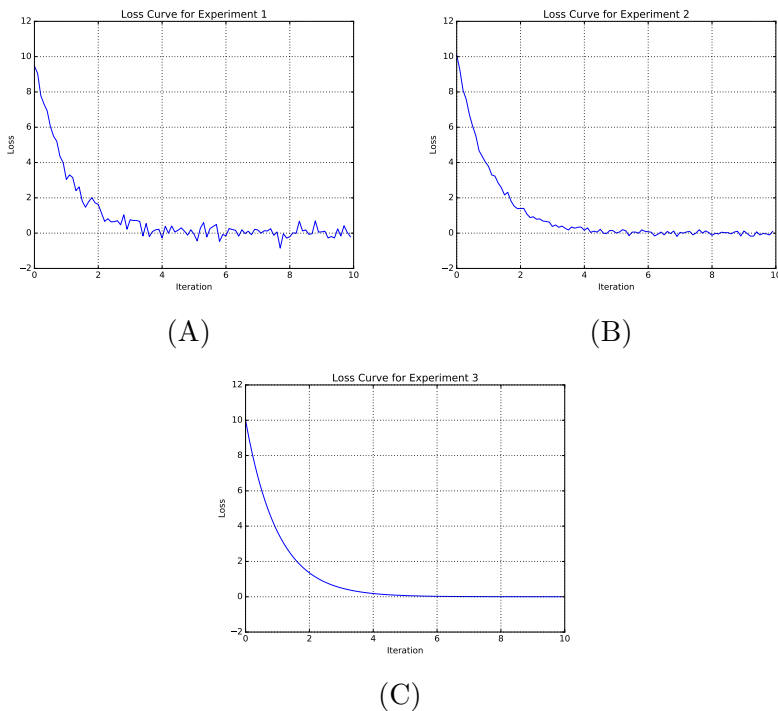


Figure 3: Loss curves for Experiment A, Experiment B, and Experiment C

You are given a black box optimizer which produces the loss curve shown in Figure 3A. You see a big red button on the optimizer and decide to push it. After doing this, you notice the loss curve shown in Figure 3B. You press the button one more time and finally notice the loss curve shown in Figure 3C.

- (a) The red button modifies a single hyperparameter.
- (i) (1 point) Which hyperparameter is likely to be modified by pressing the button?

Solution: The red button is likely to change the batch size.

- (ii) (1 point) Of experiments A, B and C, which corresponds to largest magnitude of the hyperparameter?

Solution: The experiment C is likely to have the largest batch size. tm

- (iii) **(2 points)** The loss curve for experiment C seems to be the most desirable. Despite this, give two reasons why you would choose the hyperparameter in experiment B for training your model.

Solution: B: Minibatches

- All the data might not fit in memory
- Faster updates

You're tired of using the black box and decide instead to use an Adam Optimizer.

For the following questions, assume that each experiment begins with the same initial weights. You keep the default values of β_1 (0.9) and β_2 (0.999), and run the optimization for 100 updates. The distribution of gradient updates $\left(\frac{m_t}{\sqrt{s_t+\epsilon}}\right)$ is shown in Figure 4.

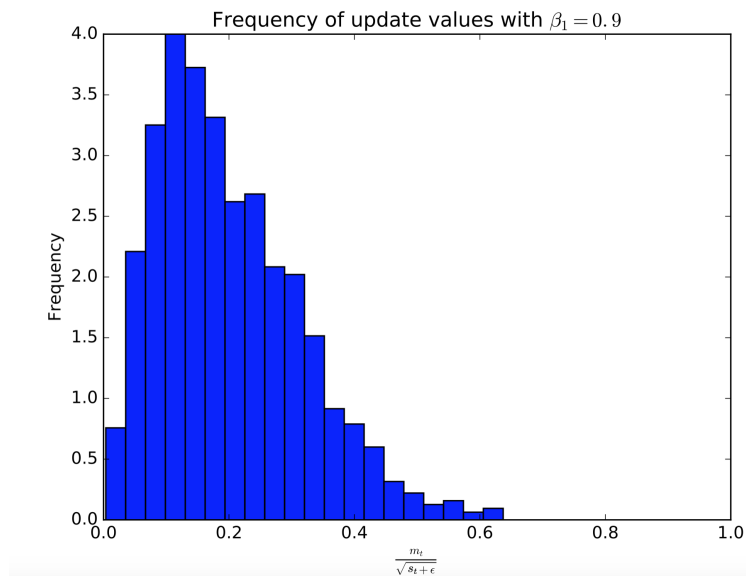


Figure 4: Distribution of updates with $\beta_1 = 0.9$.

Note that the area under the curve for a distribution like this is always 1.

- (b) **(3 points)** You change the value of β_1 from the default 0.9 to 0.4, while keeping β_2 at 0.999 and run the optimization for 100 steps. Which of the following are likely attributes of the curve to change in this new experiment?

- The peak value
- The width (the spread)

Justify.

Solution: The peak value, and width are both likely to change. The width is likely to change because when the value of β_1 goes down, the value of m_t is likely to have high variance. If this is the case, the width of the above distribution will increase. To maintain that the area under the curve of the distribution is 1, the peak value is likely to reduce.

- (c) Given the gradient calculated at a point, the Adam optimizer has three distinct steps. First, update the moving averages. Second, apply the bias correction. Third, update the parameters.

Consider the moving average of the square of the gradients. It is given by the recursive formula:

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) g_t^2$$

- (i) **(4 points)** Write down the expression for s_t only in terms of the gradients g_0, g_1, \dots, g_t

Solution: Since $s_t = \beta_2 s_{t-1} + (1 - \beta_2) g_t^2$, this implies that $s_{t-1} = \beta_2 s_{t-2} + (1 - \beta_2) g_{t-1}^2$ and so on.

Therefore, replacing s_{t-1} in the first equation gives us:

$$s_t = \beta_2 (\beta_2 s_{t-2} + (1 - \beta_2) g_{t-1}^2) + (1 - \beta_2) g_t^2$$

$$s_t = \beta_2^2 s_{t-2} + (1 - \beta_2) (g_t^2 + \beta_2 g_{t-1}^2)$$

$$s_t = (1 - \beta_2) (g_t^2 + \beta_2 g_{t-1}^2 + \dots + \beta_2^{t-1} g_1^2)$$

Or equivalently

$$s_t = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} g_i^2$$

- (ii) **(5 points)** Given your expression in part (i), what is $E[s_t]$ in terms of $E[g_t^2]$ and β_2 ? You may assume that g_i 's are independent and identically distributed. The following formulas might be useful:

$$\sum_{i=0}^{n-1} (a + id) = \frac{n}{2} (2a + (n-1)d)$$

$$\sum_{i=0}^{n-1} ar^i = \frac{a(r^n - 1)}{r - 1}$$

Solution: Starting from $s_t = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} g_i^2$

$$E[s_t] = E \left[(1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} g_i^2 \right]$$

$$E[s_t] = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} E[g_i^2]$$

Using the i.i.d. assumption,

$$E[s_t] = E[g_t^2] (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i}$$

Which can be computed to be (using the given formula)

$$\mathbb{E}[s_t] = \mathbb{E}[g_t^2] (1 - \beta_2) \frac{\beta_2^t - 1}{\beta_2 - 1}$$

$$\mathbb{E}[s_t] = \mathbb{E}[g_t^2] (1 - \beta_2^t)$$

This is the motivation behind the bias correction step.

- (iii) **(2 points)** Using your result in the previous part, explain what would happen if you did not perform the bias correction step.

Solution: In the long term - both expectations would converge. However, initially, the value is biased toward zero. This bias is worse with larger values of β_2 .

- (d) **(2 points)** You have two weights w_1 and w_2 within a model such that initially, $|w_1| \gg |w_2|$. What can you say about the magnitude of $\frac{m_t}{\sqrt{s_t + \epsilon}}|_{w_1}$ compared to the magnitude of $\frac{m_t}{\sqrt{s_t + \epsilon}}|_{w_2}$ for the first update (greater than, equal to, less than)? Explain.

Solution: The first update will not depend on the magnitude of the parameters. The gradient only depends on the impact each parameter has on the loss. Therefore, we can draw no conclusions about the update values.

Extra Page 1/3

Extra Page 2/3

Extra Page 3/3

END OF PAPER