CS230: Deep Learning

Winter Quarter 2020

Stanford University

Midterm Examination

180 minutes

	Problem	Full Points	Your Score
1	Neural Network	12	
2	Loss	12	
3	Optimization	14	
4	Batch Normalization	11	
5	DL Strategy	12	
6	Adversarial Attacks and GANs	7	
7	CNNs	10	
Total		78	

The exam contains 22 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 (Neural Network, 12 points)

You want to build a model to predict whether a startup will raise funding (1) or not (0) in its first year. You have access to a dataset of examples with 300 input features, including the number of founders, the number of employees, the variance in the age of the employees, etc. You will be setting up a logistic regression and a simple neural network for the task.

(a) **(1 point)** You implement a logistic regression that takes a vector input of shape $(n_x, 1)$. What should be n_x for the task at hand? (1 word)

(b) **(2 points)** A logistic regression has trainable weights w and bias b.

(i) What should be the dimension of w? (1 word)

(ii) What should be the dimension of b? (1 word)

(c) **(2 points)** You want to predict the probability $\hat{y}^{(i)}$ that a new startup $x^{(i)}$ raises funding in its first year.

(i) What is the dimension of $\hat{y}^{(i)}$? (1 word)

(ii) How is $\hat{y}^{(i)}$ computed using the parameters and the input? (1 formula)

Your trained logistic regression serves as a baseline model. To compare, you decide to train a neural network on the same dataset.

(d) **(2 points)** You implement a single hidden layer neural network with 80 hidden neurons. How many weights and biases does this neural network have? (2 sentences)

(e) **(1 point)** Recall that a neural network with a single hidden layer can approximate any continuous function (with some assumptions on the activation). Why would you use neural networks with multiple layers? (1 sentence)

(f) **(3 point)** Chris forgets to include non-linearities in his hidden layer. You want to demonstrate to him that a logistic regression model can perform the same computation as his neural network.

(i) Let y represent the variable for the probability output of the network, and x the input. What is the formula for the computation performed by the neural network? (1 formula)

(ii) What is the weight parameter, w, for a logistic regression model performing the same computation as Chris' neural network in terms of the neural network parameters? (1 formula)

(iii) What is the bias parameter, b, for a logistic regression model performing the same computation as Chris' neural network in terms of the neural network parameters? (1 formula)

(g) **(1 point)** Chris now adds ReLU non-linearities, but also adds one on the final output, right before the sigmoid activation. $\hat{y} = \sigma(\text{ReLU}(z))$. Chris classifies all inputs with a final value $\hat{y} \geq 0.5$ as positive. What problem is he going to encounter? (1 sentence)

Question 2 (Loss, 12 points)

In the early hours of the morning, you find your friend Vineet in the dining hall. He explains that he is trying to build a network to predict whether a watermelon is sweet or not given an image of it. (He also explains that a yellow creamy spot on the outside means that the watermelon is sweet – you are fascinated and immediately agree to help!) You will first Vineet set up his loss function.

(a) **(1 point)** Vineet has already set up a neural network architecture, and decided to use an L2 loss. Give Vineet one reason why he should not use the L2 loss? (1 sentence)

(b) **(1 point)** Using your advice, Vineet decides to switch to the binary cross-entropy loss. Write down the formula for this loss (for a single example) in terms of the label y and prediction $\hat{y}$! (1 formula)

(c) **(1 points)** You want to sanity check Vineet's implementation of binary cross-entropy loss. What should the loss be for a probability of $\hat{y} = 0.9$ on a *negative* ($y = 0$) example. *You can simply fill in the formula with the numbers, without needing to calculate it.* (1 formula)

(d) **(3 points)** Compute the value of the cost function, J, of the network on the following dataset of 4 examples using the binary cross entropy loss. $Y^T = (1, 0, 0)$, and $\hat{Y}^T = (0.1, 0.2, 0.7)$. *You can simply fill in the formula, without needing to simplify it. There is no penalty on the weights.* (1 formula)

(e) **(3 points)** You recommend adding L2-regularization to the cost function.

(i) What is the update rule for the weights, W, using vanilla gradient descent? Use the symbols $\alpha, \lambda, \frac{\partial J}{\partial W}, W$. (1 formula)

(ii) Looking at your update rule, Vineet guesses that many of the resulting weights will be small. True or False? (1 word)

(iii) Which regularization method, L1, or L2, leads to weight sparsity? (1 word)

(f) **(1 point)** Vineet also knows that there is a relationship between how the watermelon looks and how heavy it is. He wants to build a neural network to also output the weight of a watermelon given the image. Explain one advantage of having the same network output both the weight of the watermelon, and whether it is sweet. (1 sentence)

(g) **(2 point)** Vineet decides to go for a single neural network for both tasks.

(i) Write down the dimension of a new label y for the new network. (1 word)

(i) Write down the formula for the updated loss function using the new y and updated network $\hat{y}$ (1 formula)

Question 3 (Optimization, 14 points)

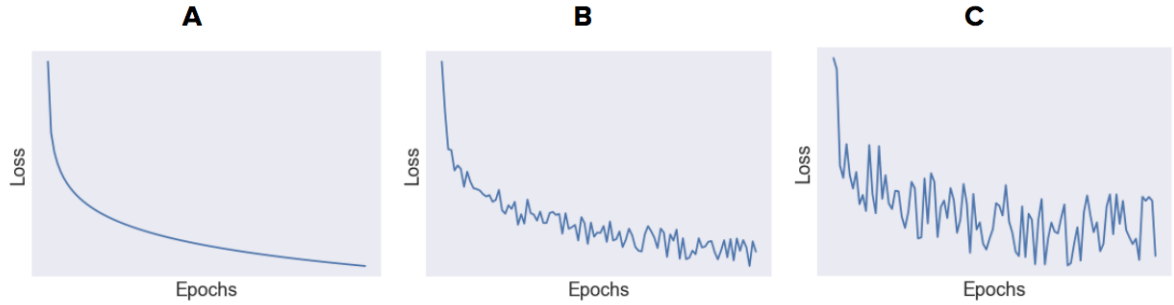
Jon is preparing a bird-watching trip at El Polín Spring. He wants to build a neural network to predict the species of a bird given a photo. He knows that there are 300 species of bird that have been spotted. Jon has collected a dataset of images of those 300 species, first collecting images of bald eagles (type of bird), and then images for herons, and so on for the 300 species. Jon is hours away from his trip, and needs your help optimizing his neural network.

(a) **(1 point)** Jon has started by using batch gradient descent. The network has been giving him poor training loss, and he's just realized he's not shuffling the training data. Would shuffling help? (1-2 sentences)

(b) **(1 point)** Jon is deciding whether he should train his network using mini-batch gradient descent versus stochastic gradient descent (batch size of 1). Give one reason why Jon should use mini-batch gradient descent. (1-2 sentences)

(c) **(1 point)** Jon asks you why you would use minibatch gradient descent over batch gradient descent. Give him one reason. (1-2 sentences)

(d) **(1 point)** Jon trains with all of (stochastic, minibatch, and batch) gradient descent, and saves the training curves, but forgets to name them. Remind him which curve refers to which. (1 sentence)



(e) **(2 points)** Jon knows that learning rate is one of the most important hyperparameters. He shows you the learning rate, and asks you whether his learning rate is too large or too small. You explain to him that you'll be able to detect that when the model is training.

(i) How will you detect during model training that the learning rate is too large? (1 sentence)

(ii) How will you detect during model training that the learning rate is too small? (1 sentence)

(f) **(2 points)** Jon is not happy with vanilla gradient descent, and decides to try out gradient descent with momentum.

(i) Explain to Jon what momentum does to the update rule. (1 sentence)

(ii) Explain to Jon how momentum speeds up learning. (1 sentence)

(g) **(3 points)** Jon also decides to try out RMSProp, as shown below:

$$S = \beta S + (1 - \beta)\frac{\partial L}{\partial W}^2$$
$$W = W - (\frac{\alpha}{\sqrt{S + \epsilon}})\frac{\partial L}{\partial W}$$

(i) What happens to the model updates when the model gradients are large (compared to vanilla gradient descent)? (1 sentence)

(ii) What happens to the model updates when the model gradients are small (compared to vanilla gradient descent)? (1 sentence)

(iii) Based on your answers to i) and ii) why might this be desirable for learning? (1 sentence)

(h) **(3 points)** Jon decides to start implementing Adam for fun, before realizing he needs to go pack for his trip. He asks you to complete his numpy implementation. He's left you with the update rule for Adam, and some notes on the lines of code for you to fill out.

$$W = W - \alpha \frac{V_{dW}}{\sqrt{S_{dW} + \epsilon}}$$

```
def optim_adam(weights_dict, gradients_dict, cache_dict, step):
    """
    v is VdW, s is SdW, v_corr is VcorrdW, s_corr is ScorrdW.
    """
    lr, beta1, beta2, eps = 1e-3, 0.9, 0.999, 1e-8
    for weight_name in weights_dict:
        w = weights_dict[weight_name]
        grad = cache_dict[weight_name]
        v = cache_dict["v" + weight_name]
        s = cache_dict["s" + weight_name]

        # TODO: Exp weighted avg of grad
        v =

        # TODO: Exp weighted avg of grad^2
        s =

        # TODO: Bias correction. divide by (1 beta1^step))
        v_corr =

        # TODO: Bias correction. divide by (1 - beta2^step))
        s_corr =

        #TODO: Update rule for Adam
        w =

        cache_dict["v" + weight_name] = v
        cache_dict["s" + weight_name] = s
        weights_dict[weight_name] = w
```

Question 4 (Batch Normalization Questions, 11 points)

Leo has been working all day with one goal only: to understand batch normalization. He finds you to ask your help!

(a) **(2 points)** Leo is curious to hear from you why batch normalization works! **(Fill in the circles for all that apply)**

- (A) Batch normalization makes processing of a single batch faster, which also reduces the training time when keeping the number of updates fixed. This allows us to spend the same amount of time performing more updates to reaching the minima.
- (B) Batch normalization weakens the coupling between earlier/later layers, which allows for independent learning.
- (C) Batch normalization normalizes the output distribution to be more uniform across dimensions.
- (D) Batch normalization mitigates the effects of poor weight initialization and allows us to initialize our weights to smaller values close to zero.

(b) **(4 points)** You decide that the best way to teach Leo about Batch Normalization is to implement it. Complete the following implementation of batch normalization's forward propagation in numpy.

Assume X is a tensor where the first dimension corresponds to the batch. X is the input to the layer, gamma and beta are the batchnorm parameters, and any intermediate calculations can be stored in the cache dictionary. In addition, beta_avg can be used as a beta term to calculate exponential moving averages.

The following formulas may be helpful:

$$z_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$
$$y_i = \gamma z_i + \beta$$

The code and space to enter your answers is on the next page.

```
def forward_batchnorm(X, gamma, beta, eps, cache_dict, beta_avg, mode):
    """
    Performs the forward propagation through a BatchNorm layer.
    X:          input data that we will normalize.
                shape (num_examples, num_channels, ...)
    gamma:      parameter for the BN layer
    beta:       parameter for the BN layer
    eps:        epsilon
    beta_avg:   the beta value to use for calculating moving averages
    mode:       if we are performing BN at 'train' time or at 'test' time

    """

    if mode == 'train':
        # TODO: Mean of X across first dimension
        mu =

        # TODO: Variance of X across first dimension
        var =

        # TODO: Take moving average for cache_dict['mu']
        cache_dict['mu'] =

        # TODO: Take moving average for cache_dict['var']
        cache_dict['var'] =

    elif mode == 'test':
        # TODO: Load moving average of mu
        mu =

        # TODO: Load moving average of var
        var =

    # TODO: Apply z_i transformation
    Zi =

    # TODO: Apply gamma and beta transformation
    out =

    return out
```

(c) **(1 point)** Leo doesn't understand why when using batch normalization, the bias parameters on the layers being normalized are removed. Explain to him why this is the case! (1-2 sentences)

(d) **(1 point)** Leo is curious why you use epsilon as part of the formula. Explain to him the purpose of epsilon! (1 sentence)

(e) **(1 point)** Leo decides to tackle a new problem where he is processing high-resolution MRI data. However, because this data is high-resolution, he can only use a minibatch size of 1. Why can Leo not use batch normalization for this problem? (1-2 sentences)

(f) **(2 points)** Leo wants to confirm his understanding of number of parameters. Leo is applying batch normalization to a fully connected (dense) layer with an input size of 10 and output size of 20. He asks you how many trainable parameters this layer, including batch normalization, contains. Show your work.

Question 5 (Deep Learning Strategy, 12 points)

Your friend Jingbo has been working with Dr. Sherry in the medical school on a project aimed at detecting the risk of premature birth for pregnant women using ultrasound images. Jingbo thinks he is done building the algorithm now, and wants to run his decisions by you before he presents them to Dr. Sherry.

(a) **(1 point)** Jingbo tells you that he has 500 examples in total, of which only 175 were examples of preterm births (positive examples, label = 1). To compensate, Jingbo decides to duplicate all of the positive examples, and then splits the data into train, validation and test. Explain to Jingbo why there is a problem with this approach. (1-2 sentences)

(b) **(1 point)** Jingbo fixes the mistake, and now finds that the model is still doing really well. Jingbo explains that Dr. Sherry has explained that the model should not miss preterm births, but false positives are okay. Jingbo explains that he is thus selecting models based on the recall metric. His top selected model has a perfect recall of 1.0! Explain to Jingbo why there is a problem with this approach. (1-2 sentences)

(c) **(2 points)** Jingbo fixes the mistake. Jingbo is bothered by his training loss curve, which shows that the loss is oscillating. What should Jingbo consider changing? **(Fill in the circles for the two best options.)**

- (a) Increase Learning Rate
- (b) Decrease Learning Rate
- (c) Increase Minibatch Size
- (d) Decrease Minibatch Size

(d) **(1 points)** You ask Jingbo whether he is using transfer learning on his ultrasound images, and he explains he thinks they may not be useful because ultrasound images look very different from natural images. Give him a reason why transfer learning may still be a good idea. (1-2 sentences)

(e) **(1 point)** Jingbo asks Dr. Sherry about how well experts can do the task, and measures the F1 score for different people. Jingbo also measures the F1 score of the label, which is not determined using ultrasound, but determined after observing whether the patient had a premature birth. Which experiment would give the best estimate of Bayes error on this task using ultrasound data? (1 sentence)

	Experiment	F1 Score
Option A	Jingbo	0.20
Option B	Group of Doctors	0.88
Option C	Single Doctor	0.80
Option D	Label	0.90

(f) **(1 points)** Jingbo finds that his training loss is much lower than his dev loss. He decides to use dropout to regularize the network. Explain how dropout acts a regularizer. (1-2 sentences)

(g) **(1 points)** Jingbo still sees that his dev loss is higher than his train loss. What is something he could change in the cost function or optimization (without changing the architecture)? (1 sentence)

(h) **(1 point)** Jingbo still sees that his dev loss is higher than his train loss. What is something he could change with the data? (1 sentence)

(i) **(1 point)** Jingbo tries to run a grid search over the hyperparameters, but feels that there are too many options. What is another method he can use for hyperparameter search? (1 sentence)

(j) **(2 points)** Write some code to randomly sample 5 values for alpha (the learning rate, between 1e-5 and 1e-1), and the number of hidden layers (between 1 and 7) in numpy, using `np.random.rand()`. Hint: use an appropriate scale!

```
# TODO
alpha =
# TODO
layers =
```

Question 6 (Adversarial Attacks and GANs, 7 points)

(a) **(2 points)** Which of the following are true regarding adversarial attacks and GANs? **(Fill in the circles for all that apply.)**

(i) If you forge an adversarial example to fool a cat classifier A, there's a chance it will fool another cat classifier B.

(ii) The Fast Gradient Sign Method can be used for a black box attack.

(iii) If the loss associated with the generator of a GAN is low, the samples it generates will look realistic.

(iv) For a GAN using saturating loss, the loss function for the generator and discriminator are identical except one is the negation of the other.

(b) **(1 point)** Recall the Fast Gradient Sign Method for generating adversarial examples. Given $x = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^\top$, $\nabla_x \mathcal{L}(x, y, \theta) = \begin{bmatrix} 0.5 & -0.5 & 1 \end{bmatrix}^\top$, and $\varepsilon = 0.01$. What would our resulting adversarial example be? Show your work.

(c) **(1 point)** The magnitude of ε needed to forge the adversarial example increases with the dimension of x. Do you agree with this statement? Explain why in 1-2 sentences.

(d) **(1 point)** Describe whether the saturating or non-saturating loss would be a preferable optimization objective for training a GAN, and why this choice would be considered preferable. (1-2 sentences)

(e) **(1 point)** Consider a binary classifier where the output activation is a sigmoid. To resist adversarial attacks, you replace the sigmoid with the following function:

$$a(x) = 0 \text{ if } x < 0 \text{ else } 1$$

Since the network is no longer differentiable, would you expect the model to be resistant to adversarial attack? (1 sentence)

(f) **(1 point)** You choose to train a standard GAN and, at iteration t, you read off values for the loss of the generator and discriminator. At iteration $t' > t$, you read off the same exact values. Why are the quality of generated images at iteration t and t' not necessarily similar? (1-2 sentences)

Question 7 (CNNs, 10 points)

(a) **(2 points)** Which of the following are true concerning CONV layers? **(Fill in the circles for all that apply)**

(i) If a CONV layer configuration (kernel size, stride, etc.) is compatible with an input volume $W \times H \times D$ (meaning it does not throw an error), then it will be compatible with any input volume $W \times H \times D'$, where $D' \geq 1$.

(ii) Max pooling layers have exactly one tunable parameter, regardless of the input volume dimensionality.

(iii) The output of a (3×3) kernel applied to a (3×3) region in an input image is yielded by a matrix multiplication between the two matrices.

(iv) A CONV layer with the padding set to 'valid' corresponds to setting the padding to zero.

(b) **(2 points)** Which of the following are benefits associated with parameter sharing in CNNs? **(Fill in the circles for all that apply)**

(i) Helps to mediate the vanishing gradient problem.

(ii) Significantly reduces the number of parameters as compared to fully connected layers.

(iii) Allows for resistance to adversarial attacks.

(iv) Allows for useful kernels to not have to be relearned in different regions of the image.

(c) **(1 point)** What could be a potential motivation for using a (1×1) convolution? Hint: what does performing a (1×1) convolution achieve in terms of the resulting output volume? (1 sentence)

(d) **(1 point)** What would you set the padding of a CONV layer to be (as a function of the filter width f) to achieve an output volume with the same dimension? Assume the stride is 1. (1 formula)

(e) **(1 point)** You have an input volume of $(32 \times 32 \times 3)$. What is the dimensionality of the resulting volume after convolving a (5×5) kernel with the padding set to 0, the stride set to 1, and 2 filters? Show your work.

(f) **(1 point)** How many parameters would this layer have, including biases? Show your work.

(g) **(1 point)** Now, say we wanted to process time-series data, modeled as three fluctuating values (the channels) over time i.e. a volume of size $(T \times 3)$. If the kernel was length 5 and convolved over the input, what would be the new number of parameters for this layer, including biases? Assume the same configuration (padding, stride, number of filters) and show your work.

(h) **(1 point)** We now model a video as sequential images indexed by time i.e. a volume of size $(W \times H \times T \times 3)$. If the kernel was extended to $(5 \times 5 \times 5)$ and convolved over the input, what would be the new number of parameters, including biases? Once again, assume the same configuration (padding, stride, number of filters) and show your work.

Extra Page 1/2

Extra Page 2/2

END OF PAPER