CS230: Deep Learning

Spring Quarter 2019
Stanford University

Midterm Examination

180 minutes

	Problem	Full Points	Your Score
1	Multiple Choice Questions	20	
2	Short Answer Questions	24	
3	Case study (Face Verification)	13	
4	Adversarial Examples	20	
5	Backpropagation	28	
6	Data Augmentation	6	
Total		111	

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, showing your calculations and derivations!** We'll give partial credit for good explanations of what you were trying to do.

Name: ORION DARLEY

SUNETID: ORION D @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, 20 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) **(2 points)** Which of the following is the derivative of sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$?

(i) $\sigma(x)(\sigma(x) - 1)$

(ii) $1 - \sigma(x)$

(iii) $\sigma(x)(1 - \sigma(x))$

(iv) $\sigma(-x)$

(b) **(2 points)** A convolutional layer has filter size of 5×5, stride of 2, and no padding. What is the output shape of this layer, if an image of size 15×15 is fed as input? You only need to compute the width and height of the image (no need to account for the depth dimension).

(i) 5×5

(ii) 6×6

(iii) 9×9

(iv) 10×10

(c) **(2 points)** Which of the following statements is true about stochastic gradient descent (SGD) with batch size of 1?

(i) For small datasets ($10 \sim 100$ examples), SGD is equivalent to gradient descent over the entire dataset.

(ii) Each step of SGD is guaranteed to decrease the cost.

(iii) When the cost reaches a saddle point, SGD cannot escape it.

(iv) None of the above.

(d) **(2 points)** In the context of a Convolutional Neural Network, which of the following techniques usually alleviate the problem of *underfitting*? **(Circle all that apply.)**

(i) Removing some of the activations (i.e. nonlinearities).

(ii) Increasing the number of max pooling layers.

(iii) Increasing the number of convolutional layers.

(iv) Increasing the number of parameters.

(v) Decreasing the learning rate.

(e) **(2 points)** You are performing transfer learning by fine-tuning a large pre-trained network on a new task. When compared to fine-tuning the entire network, what usually happens to the bias and variance of the model when you freeze most of the weights and fine-tune only the last few layers?

(i) Higher bias, higher variance

(ii) Higher bias, lower variance

(iii) Lower bias, higher variance

(iv) Lower bias, lower variance

(f) **(2 points)** Which of the following is true about regularization techniques?

(i) Regularization speeds up training by reducing the bias of the model.

(ii) Dropout works by normalizing the output of a layer with respect to the mini-batch.

(iii) Convolutional neural networks do not benefit from regularization.

(iv) L_1 regularization causes weights to be sparse.

(g) **(2 points)** How many parameters are there for a 2D convolutional layer with filter size 3×3, input depth of 4, output depth of 8, and with bias?

(i) 72

(ii) 80

(iii) 288

(iv) 296

(h) **(2 points)** Which of the following statements about initialization is *false*?

(i) Xavier initialization guarantees that the variances of layer outputs do not change during training.

(ii) Initializing all weights to zero may lead to slow training.

(iii) Randomly initializing weights helps training via symmetry breaking.

(iv) Different activation functions may benefit from different initialization techniques.

(i) **(2 points)** Which of the following statements are true about Generative Adversarial Networks (GANs)? **(Circle all that apply.)**

(i) After successfully training a GAN, you can use the discriminator to create new data samples.

(ii) The generator's goal is to map random noise to samples indistinguishable from real data points.

(iii) In order to stabilize training, it is common to first train the discriminator before running the usual GAN training procedure.

(iv) It may help to train the discriminator for multiple steps for each step of generator training.

(j) **(2 points)** Which of the following are true about activation functions? **(Circle all that apply.)**

(i) Activation functions are applied element-wise to each neuron in a layer.

(ii) Activation functions are necessary for a fully-connected neural network to represent non-linear functions.

(iii) Activation functions must be monotonically increasing.

(iv) Activation functions must have non-zero gradient at all points to avoid vanishing gradient issues.

Question 2 (Short Answer Questions, 24 points)

Please write concise answers.

(a) **(2 point)** You have trained a fully-connected neural network, and you observe high variance. State two methods that solve this problem (no need to explain how they work).

① DATA AUGMENTATION

② L1 / L2 Regularization + other Reg. techniques

(b) **(2 points)** Let p be the probability of keeping neurons in a layer with dropout. With inverted dropout, you need to scale the activations by dividing them by p during training time.

(i) Why do you need to scale the activations during training time?

Generally the analyst wants to diminish the hidden unit activations, 1-P, where P = # of hidden units. This is done with the "keep-prob" COMMAND IN TENSORFLOW

(ii) Bob forgot to scale the activations during training. What can Bob do during test time to resolve this issue?

Bob should multiply the activations by P, where P is the # of hidden units.

(c) **(4 points)** When performing batch normalization during training, you need to do the following calculations:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

$$y_i = \gamma \odot \hat{x}_i + \beta$$

where $\odot$ represents element wise multiplication

(i) List the variables which are trainable parameters.

(1) Beta (2) gamma

(ii) Let's say that the input to this batch normalization layer is a matrix $X \in \mathbb{R}^{m \times n}$ where m is the number of examples and n is the number of features. Give the dimensions of each trainable parameter.

Each trainable parameter are arrays of dim n, (# of features)

(iii) During training, you need to keep track of the exponential moving average of the mean, μ_R, and exponential moving average of the variance, σ_R^2. State, for σ_R^2, (1) its dimensions and (2) the equation to update it.

Array of dim n

$\sigma_R^2 = (1 - m)\sigma_R^2 + m\sigma^2$

(iv) How does the implementation of batchnorm differ between training time and test time?

Training	Test
mu	muR
Sigma^2	Sigma^{2}R

(d) **(2 points)** Your friend Alice wants to tune a neural network. She wants to tune both the learning rate and the regularization constant. She can use either random search or grid search for tuning. Which method should she pick and why?

GRID suffers from being computationally ineffecient especially in high-dim scenarios. A random search typically finds "close-enough" values in fewer iterations than grid searching, and its more effecient. RANDOM search is best here.

(e) **(2 points)** Your friend Eve is tuning her learning rate using the test set.

(i) Why is this a bad idea? Her model could be dispositioned to overfitting. She should evaluate tuning her learning rate on the validation set.

(ii) Which set should she use to tune her learning rate?

Not the test set... Eve should use the validation set or set b/w Training & test. ie 80/10/10 Tr V Te 60/20/20 Tr V Te

(f) **(2 points)** Your friend Ron got 99% accuracy on a fraud detection problem. 99% of the dataset contained examples labeled 'Not fraud'. Why shouldn't he be satisfied with his results? What evaluation metric should he use instead?

The dataset is imbalanced, so the accuracy metric is insufficient in evaluating model performance.

F1 Score $= 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

(g) **(3 points)** You are training a neural network with the following forward propagation equations using full batch gradient descent:

$$Z = WX + b$$

$$A = ReLU(Z)$$

All of the elements of Z are negative.

(i) What is the problem with using ReLU as an activation function?

The ReLU will produce zeros for every negative value in Z.

(ii) Would using the activation function max(0.4z, z) solve the problem? Explain briefly.

This is a "Leaky Relu" and it can help solve the issue of $z < 0$

(iii) Would the activation function g(z) = 4z be a good choice to solve the problem? Explain briefly.

This is a linear function and is generally not a good idea to use this as it will only multiply the output by four. It went decrease the complexity of network architecture. This activation function should not be used.

(h) **(2 points)** How does backpropagation differ between a convolutional neural network (CNN) used for image classification and a CNN used for neural style transfer?

Backpropagation updates the weights for ConVnets for image recognition.

Neural style/content transfer models...

(i) **(2 points)** Give an example of a data augmentation technique which could be useful for cat classification but would not give good results for handwritten digits classification.

Mirroring is a sufficient technique for the cat library, however not for the ~~MNIST~~ / hand-written library, due to the digits' orientation on the left and right sides. For example IF 7 or 9 is mirrored, its not longer A 7 or 9. A 1, 8, 0 ~~will~~ cannot be mirrored, but a cat can.

(j) **(1 points)** Give one advantage of RMSprop over SGD.

RMSPROP tries to dampen oscillations In concept, it is similar to having an adaptive learning rate.

(k) **(2 points)** Your friend Bob trained a CNN for a binary classification task with the following architecture:

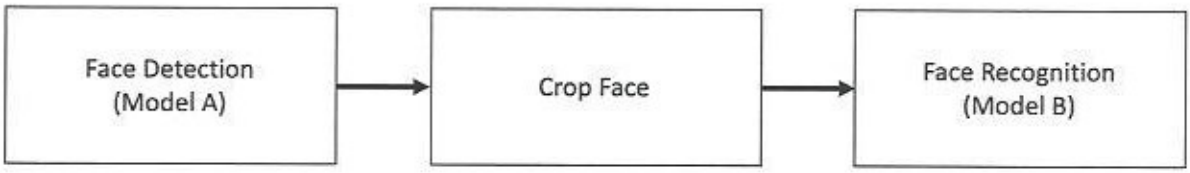
[Input] → [Conv] → [ReLU] → [Linear] → [Sigmoid].

The input is of shape (15,15). The conv layer is a 3x3 filter with padding 0 and a stride of 12. The training accuracy is low. Suggest one likely reason why.

STRIDE of 1 is usually default. Using a stride of 1 and pooling can remedy the poor training accuracy. Compared to the image size, it could be too large. The output map could be too small, so part of the picture could be lossed.

Question 3 Case Study (Face Verification), 13 points

You have been tasked with building a fast facial verification system for the Happy House. A face verification system consists of two parts. First, a face detection model (model A) runs on each frame to return a tight bounding box around each face. Second, a face recognition model (model B) is applied to each cropped face image to convert it to a 128-dimensional feature vector.



(a) **(2 points)** First, you need to build a fast face detection model (model A). How can you use a slow face detection model, and access to the camera footage (which contains a lot of faces) to build a labelled training set for your model?

This slow CNN will be useful help build the labeled dataset. The model must have seperate neurons to create the bounding box for each image (24 or 30 FPS) * Frames per second. Images should be cropped, specifically the face P=1 and non-faces P=0, Sliding windows can help due to effeciency.

(b) **(2 points)** Next, you need to train a fast face detection model (model A). However, the only fast, pretrained, detection model that you have is to detect cats. How can you efficiently re-purpose this model to detect faces?

Remove the layer that the CNN trained cats for (fully-connected layers), run the model to extract the attributes for all images it was designed to learn on, And retrain the model on the faces dataset. - TRANSfer Learning

(c) **(2 points)** Now you need to pick a face recognition model (model B). Consider the following models:

Model Name	Processing time per face	Accuracy
AlphaFace	100ms	99.5%
DeeperFace	20ms	94.6%
FaceGo	10ms	91.2%

Your house will be happy with any model that offers real-time performance (i.e. processing time allows for 20 faces to be processed each second). Which model would you pick? Explain briefly.

DeeperFace is the best choice

DeeperFace - 50 faces/sec @ 94.6% acc. ← Best accuracy

FaceGo - 100 faces/sec @ 91.2% acc.

Assuming Criteria is constant

(d) **(2 points)** Face recognition models are trained using the triplet loss function discussed in class. As a refresher, triplet loss is defined over three images: an anchor (A), a positive (P) and a negative (N). The anchor and positive images are of the same person, and the anchor and negative images are of different people. If E_X is the embedding of image X, the triplet loss is defined as:

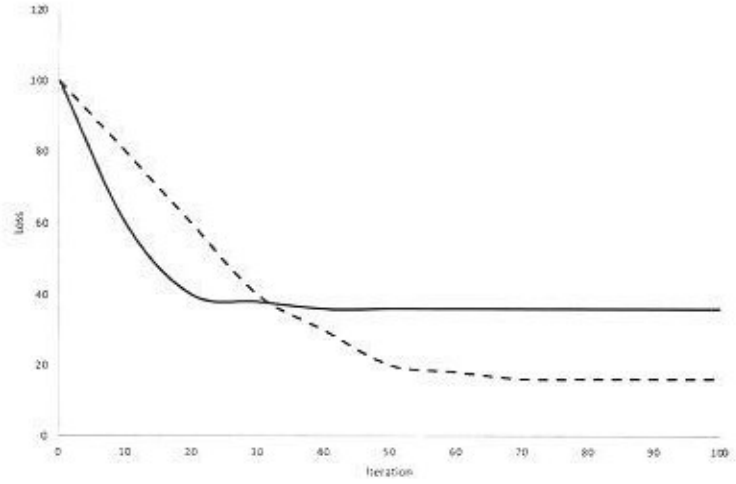
$$\texttt{Loss} = \max(||E_A - E_P||_2^2 - ||E_A - E_N||_2^2, 0) \quad (1)$$

While training your model, you realize that triplets with a low loss empirically contribute little to the overall training. Choosing triplets with higher loss, tend to improve the model faster after backpropagation. Why might this be the case?

The training dataset might not have an adequate number of the same person's face (anchors and pairs), Triplet loss could be too high, Hard tripplets in tripplet mining ie triplets where negatives are closest to the anchor

$d(a,n) < d(a,p)$

(e) **(2 points)** As you train your model, you see the training loss evolve according to the solid line. You lower your learning rate and retrain the model; now, your training loss evolves according to the dashed line. Why might a lower learning rate result in a lower loss like in the graph?



When using the higher learning rate in this instance, the gradient updates don't improve the loss over iterations, INSTEAD the PARAMETERS ARE JUMPING around. On the other hand, the lower learning rate could have allowed it to converge.

(f) **(3 points)** Once your face recognition system is deployed to the Happy House, you are told that it sometimes does not let people into the house. How might you detect whether or not it is the face detection model (model A) that is faulty, or whether the face recognition model (model B) is faulty?

The analyst must examine the face detection model first, see if the images are labeled correctly, respecify the dimensions of the bounding boxes + experiment w/ the sliding windows parameters. If it classifys the faces correctly, the problem is in the recognition model. If not, the problem still exists in the detection model. Triplet mining should be examined in the recognition model

Question 4 (Adversarial Examples, 20 points)

You have a binary classification problem. The data $(X^{(i)}, Y^{(i)})|_{i=1}^{m}$ is generated from the following underlying distribution:
$P(Y = 0) = P(Y = 1) = 0.5$
$X|Y = 0$ comes from a normal distribution with mean -1 and variance 1.
$X|Y = 1$ comes from a normal distribution with mean 2 and variance 1.
where $X \in \mathbb{R}$ and $Y \in \{0, 1\}$.

(a) **(3 points)** You generate a large number of distinct training datapoints (i.e. $m \to \infty$). Is it possible to train a logistic regression model that achieves 100% training accuracy? Explain in one sentence why or why not.

NO, because of the nonlinear distribution and the logistic regression model choice. Another model should be used here.

(b) **(2 points)** You train a logistic regression classifier of the form $\hat{y} = \sigma(wx+b)$ where σ is the sigmoid function. The predicted label is 1 if $\hat{y} \geq 0.5$ and 0 otherwise. Assuming w is positive, for which range of values is x classified as 1? Express your answer in terms of w and b.

(c) **(2 points)** Give an expression for the binary cross entropy (BCE) loss, L, in terms of the label $y^{(i)}$ and prediction $\hat{y}^{(i)}$ for all examples from $i = 1$ to $i = m$.

$$L(\hat{y}, y) = -\sum_{i=1}^{m_y} y_i \log \hat{y}_i$$

where

$$\hat{y} = (\hat{y}_1 \dots \hat{y}_m)$$
$$y = (y_1 \dots y_m)$$

For the remaining parts, assume that you have trained the logistic regression network by minimizing the BCE loss and found the optimal parameters to be $w = 2$ and $b = -1$.

(d) **(2 point)** Let's say that humans have a heuristic for labelling the class of an example x_0 based on its sign. Concretely, if x_0 is negative, humans will label the class of x_0 as 0 and vice versa.

Your objective now is to find an adversarial example x_A such that humans will label the point as 1 but your trained logistic regression classifier from (b) will label the point as 0. You start off by choosing a point x_B such that both humans and your trained logistic regression classifier from (b) label the point as 1. Give a value of x_B you could start with.

$0.5 \leq x_0 \leq z$

where z = any value

(e) **(2 points)** Give the range of possible values for x_A.

$0 \leq x_A < 0.5$

(f) **(2 points)** Instead of directly deriving possible values for x_A, Alex wants to find x_A from x_B using gradient descent. Alex suggests the following method for finding x_A. We create a dataset of one example $\{(x_B, 0)\}$. At each step of training, we forward propagate the single example and calculate the binary cross entropy loss on that single example. We then perform gradient descent on the example x_B until convergence. Write the update rule for x_B at each step with respect to learning rate α and the loss L.

?

(g) **(2 points)** After running Alex's method, we get $x_B = -0.8$. Both humans and the trained logistic regression classifier classifies x_B as 0 and hence Alex has failed to find an adversarial example. Suggest a tweak to Alex's method that would potentially make it work.

Opt for regularization hyperparameter

(h) **(2 points)** This method can be classified as a form of White Box attack. What property of this method makes it a White Box attack?

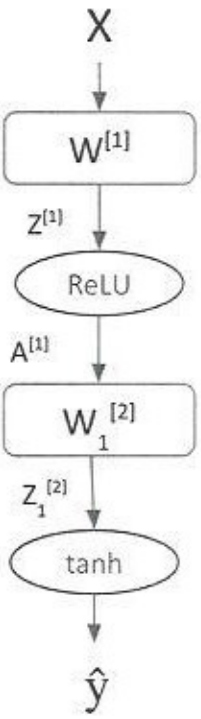
~~A logistic regression is not a black box~~

The Adversary knows the logistic regression model?

(i) **(3 points)** Explain how you would perform a Black Box attack to guess the parameters w and b of the logistic regression model if you can only run inference on it (assume that model inference outputs the value of $\hat{y}$).

Question 5 (Backpropagation, 28 points)

You are trying to build a two-layer neural network using the architecture shown in the figure below.
You decide to use the following loss function, where $\hat{y}$ is the output from your model, y is the label and m is the number of examples:
$\mathcal{L} = \frac{1}{m}||\hat{y} - y||^2$



(a) **(2 points)** The shape of the input matrix X is (n, m) and the shape of the label y is $(1, m)$, where m is the number of examples and n is the number of features. Assume that the shape of the weight matrix $W^{[1]}$ is (k, n).
Write the shape of the matrices $W_1^{[2]}$ and $\hat{y}$

$W^{[1]} \Rightarrow (k, n)$
$W^{[2]} \rightarrow (1, k)$
~~$\hat{Y} = \tanh(\quad) = W \quad A \quad + b$~~
$\hat{Y} \longrightarrow (1, m)$

(b) **(4 points)** Write the expressions for forward propagation through your neural network for $Z^{[1]}$, $A^{[1]}$, $Z_1^{[2]}$ and $\hat{y}$.

$$Z^1 = W^1 X + b^1$$
$$A^1 = \text{ReLU}(Z^1)$$
$$Z^2 = W^2 A^1 + b^2$$
$$\hat{y} = \tanh(Z^2)$$

(c) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial W_1^{[2]}}$.

Hint: You can use an indicator function $\mathbb{1}\{.\}$ that takes on a value 1 if its arguments are true and 0 otherwise.

$$\frac{\partial \mathcal{L}}{\partial W_1^2} = \left(\frac{\partial \mathcal{L}}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial Z_1^2}\right)\left(\frac{\partial Z_1^2}{\partial W_1^2}\right)$$
$$= \left(\frac{2}{m}(\hat{y}-y)\right)\left(1-\hat{y}^2\right)\left(A^1\right)$$

(d) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial W^{[1]}}$

$$= \left(\frac{\partial \mathcal{L}}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial Z^2}\right)\left(\frac{\partial Z_1^2}{\partial A^1}\right)\left(\frac{\partial A^1}{\partial Z^1}\right)\left(\frac{\partial Z^1}{\partial W^1}\right)$$

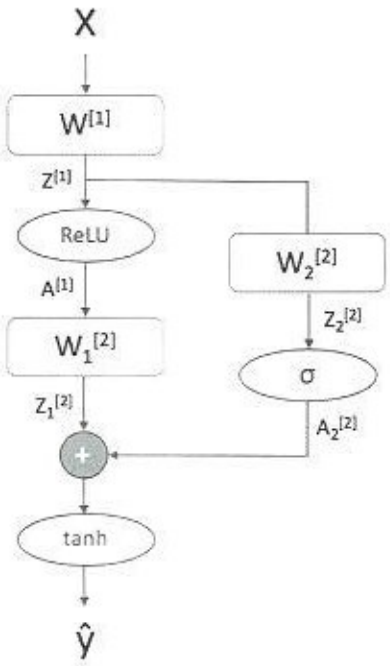
$$= \left(\frac{2}{m}(\hat{y}-y)\right)\left(1-\hat{y}^2\right)\left(W_1^2\right)\left(\mathbb{1}[Z^1]\right)\left(X\right)$$

Let $\tilde{T}$ be indicator function

(e) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial X}$ $\left(\frac{\partial \mathcal{L}}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial Z^2}\right)\left(\frac{\partial Z^2}{\partial A^1}\right)\left(\frac{\partial A^1}{\partial Z^1}\right)\left(\frac{\partial Z^1}{\partial X}\right)$

$= \left(\frac{2}{m}(\hat{y}-y)\right)\left(1-\hat{y}^2\right)\left(W_1^2\right)\left(\tilde{T}[Z^1]\right)\left(W^1\right)$

You decide to modify your network and add $Z^{[1]}$ to the output of the second layer $Z_1^{[2]}$ by passing it through another layer with a sigmoid activation function, as shown in the figure below.
You decide to use the same loss function as before.
Assume that you are using this network for the remaining parts of this question.



(f) **(2 points)** As before, the shape of the input matrix X is (n, m), the shape of y is $(1, m)$ and the shape of the weight matrix $W^{[1]}$ is (k, n).Write the shape of the matrices $W_2^{[2]}$ and $\hat{y}$

(g) **(6 points)** Write the expressions for forward propagation through your neural network for $Z^{[1]}$, $A^{[1]}$, $Z_1^{[2]}$, $Z_2^{[2]}$, $A_2^{[2]}$ and $\hat{y}$.

$Z^1 = W^1 X + b^1$

$A^1 = ReLU(Z^1)$

$Z_1^2 = W^2 a^1 + b^2$

$\rightarrow$

$Z_2^2 = W_2^2 Z^1 + b_2^2$

$A_2^2 = \sigma(Z_2^2)$

$\hat{y} = \tanh(A_2^2 + Z_1^2)$

(h) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial W_1^{[2]}}$

$$\left(\frac{\partial L}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial Z_1^2}\right)\left(\frac{\partial Z_1^2}{\partial W_1^2}\right)$$

$$= \left(\frac{2}{m}(\hat{y}-y)\right)\left(1-\hat{y}^2\right)\left(A^1\right)$$

(i) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial W_2^{[2]}}$

$$= \left(\frac{\partial L}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial A_2^2}\right)\left(\frac{\partial A_2^2}{\partial Z_2^2}\right)\left(\frac{\partial Z_2^2}{\partial W_2^2}\right)$$

$$= \left(2m(\hat{y}-y)\right)\left(1-\hat{y}^2\right)\left(Z_2^2\left[1-Z_2^2\right]\right)\left(Z^1\right)$$

Let $\tilde{I}$ be indicator function

(j) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial W^{[1]}}$

$$= \left(\frac{\partial \mathcal{L}}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial Z_2^2}\right)\left(\frac{\partial Z_2^2}{\partial A^1}\right)\left(\frac{\partial A^1}{\partial Z^1}\right)\left(\frac{\partial Z^1}{\partial W^1}\right)$$

$$I = \left(\frac{2}{m}\hat{y}-y\right)\left(1-\hat{y}^2\right)\left(W_2^2\right)\left(\tilde{I}[Z^1]\right)(X)$$

$$II = \left[\left(\frac{2}{m}\hat{y}-y\right)\left(1-\hat{y}^2\right)\left(Z_2^2[1-Z_2^2]\right)\left(W_2^2\right)\right](X)$$

$$\left(\frac{\partial \mathcal{L}}{\partial \hat{y}}\right)\left(\frac{\partial \hat{y}}{\partial A_2^2}\right)\left(\frac{\partial A_2^2}{\partial Z_2^2}\right)\left(\frac{\partial Z_2^2}{\partial Z^1}\right)\left(\frac{\partial Z^1}{\partial W^1}\right)$$

II

(k) **(2 points)** Compute $\frac{\partial \mathcal{L}}{\partial X}$

$$\left(\frac{\partial \mathcal{L}}{\partial W^1}\right)\left(\frac{\partial W^1}{\partial Z^1}\right)\left(\frac{\partial Z^1}{\partial X}\right)$$

$$= I + II\left(\frac{1}{X}\right)\cdot(W^1)$$

Question 6 (Data Augmentation, 6 points)

In this question, you will study techniques such as adding noise to feature vectors and adding noise to labels. For this purpose, consider a simple regression task on the dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, where $y^{(i)} \in \mathbb{R}$ is a label and $x^{(i)} \in \mathbb{R}^n$ is a feature vector. You have decided to use mean squared error as your cost function, that is,

$$J = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - {x^{(i)}}^\top w)^2,$$

where $w \in \mathbb{R}^n$ is a column vector.

(a) **(2 points)** Write the cost function J in a vectorized form in terms of the matrix $X = \begin{pmatrix} {x^{(1)}}^\top \\ \vdots \\ {x^{(m)}}^\top \end{pmatrix}$ and the label vector $Y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$.

Recall that the L2 norm of a vector $v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \in \mathbb{R}^n$ is $\|v\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$

(b) **Extra Credit (1 point)** Your friend Alice suggests you corrupt the labels $y^{(i)}$. That is, she suggests adding Gaussian noise $\eta^{(i)} \in \mathbb{R}$ with mean zero and variance σ^2 to each label $y^{(i)}$; hence, minimizing the following cost function

$$J_1 = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} + \eta^{(i)} - {x^{(i)}}^\top w)^2.$$

Show that $\mathbb{E}[J_1] = J + \sigma^2$.
Hint: For a Gaussian random variable η with mean zero and variance σ^2 we have $\mathbb{E}[\eta^2] = \sigma^2$ and $\mathbb{E}[\eta] = 0$.

(c) **(2 points)** Based on the expression from part (b), is Alice's suggestion of corrupting the labels useful? Explain why.

(d) **Extra Credit (1 point)** Now, your friend Bob suggests you add Gaussian noise vectors $\epsilon^{(i)} \in \mathbb{R}^n$ with mean zero and covariance matrix $\sigma^2 I$ to each feature vector $x^{(i)}$ and you minimize the following cost function

$$J_2 = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - (x^{(i)} + \epsilon^{(i)})^\top w)^2.$$

Show that $\mathbb{E}[J_2] = J + \sigma^2\|w\|_2^2$.
Hint: For a Gaussian random vector ϵ with mean zero and covariance matrix $\sigma^2 I$ we have $\mathbb{E}[\epsilon\epsilon^\top] = \sigma^2 I$ and $\mathbb{E}[\epsilon] = 0$.

(e) **(2 points)** Based on the expression from the part (d), is Bob's suggestion useful? Explain why.

Extra Page 1/3

Extra Page 2/3

Extra Page 3/3

END OF PAPER