6.802/6.874/20.390/20.490/HST.506 Exam

April 11, 2017

Answer the questions in the spaces provided. When appropriate, neatly show your work for partial credit cases.**We will only grade answers that appear inside the answer boxes.**

If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.

You are permitted one 8.5” × 11” sheet (front and back) of notes to refer to during the exam. **No other resources are allowed.**

Write your name on every page.

Name: ____________________ Email: ____________________

Question	**Points**	**Score**
1	28	
2	26	
3	28	
4	18	
Total	100	

Problem 1 (Short Answer Problems) (28 Points)

a) (4 Points) For RNA-seq data you decide to model the data as distributed according to a negative binomial distribution and compute the following likelihood ratio test for a gene of interest where D_1 and D_2 are the data as observed in condition 1 and condition 2, and D are the data from both conditions. The values of $\hat{\mu_1}$, $\hat{\mu_2}$, $\hat{\mu}$, and $\hat{\sigma}$ are maximum likelihood estimates from the relevant data:

$$T = 2 \ln \frac{P(D_1|\hat{\mu_1}, \hat{\sigma}^2)P(D_2|\hat{\mu_2}, \hat{\sigma}^2)}{P(D|\hat{\mu}, \hat{\sigma}^2)}$$

How do you expect T to be distributed if the null hypothesis is true and the observed ratio occured by chance?

b) (4 Points) You have just built two models of observed biological data D, and you compute $P_1(D|\vec{\theta_1})$ and $P_2(D|\vec{\theta_2})$ from Model 1 and 2 respectively. The parameters $\vec{\theta_1}$ and $\vec{\theta_2}$ are the maximum likelihood estimates for their respective models. You wish to select Model 1 or Model 2 as best describing the data. If k_1 is the number of parameters in $\vec{\theta_1}$, k_2 is the number of parameters in $\vec{\theta_2}$, and n is the number of observations in D, provide a equation to compare the two models that accounts for the difference in the number of their parameters k_1 and k_2.

c) (4 Points) Assume that we wish our false positive rate to be α, and we are performing n tests. Provide the p-value threshold at which we would start accepting the null hypothesis.

d) (4 Points) Draw a plot of expected test set error (y-axis) as a function of model capacity (x-axis). Assume the test set is much larger than the training set.

e) (4 Points) You intialize a weight vector $\vec{w}$ to [0.3, 0.4, 0.1, 0.7] before training, and after training the value of $\vec{w}$ is [0.0, 0.0, 0.9, 0.0]. Was $\|\vec{w}\|_1$ or $\|\vec{w}\|_2$ most likely used to penalize w? Describe why you came to that conclusion.

f) (4 Points) t-SNE uses KL divergence to minimize the differences between pairwise similarities in high-dimensional space P to pariwise similarities in low-dimensional space Q. What optimization method is used to choose the location of points in low dimensional space to minimize the cost $KL(P||Q)$?

g) (4 Points) Imagine you wish to assign 104 tissue samples to one of four clusters based upon the expression of the 4323 genes you have measured for each tissue sample. However, you do not have any idea what the mean expression of the clusters should be or how to assign genes to the clusters. Describe the two steps you would iterate between to discover these latent variables and the name of this method.

Problem 2 (Convolutional Neural Networks) (28 Points)

This problem makes use of the STL-10 dataset, which contains 500 training images and 800 test images for each of 10 different classes: airplane, bird, car, cat, deer, dog, horse, monkey, ship, and truck. You decide to construct a one-layer convolutional neural net for the STL-10 dataset. Your network takes an input RGB image of dimensions 96 × 96 × 3 and outputs a probability vector over the 10 classes. There are 16 convolutional filters followed by 16 pools in the pooling layer, and each convolutional filter is connected to a single pooling operation.

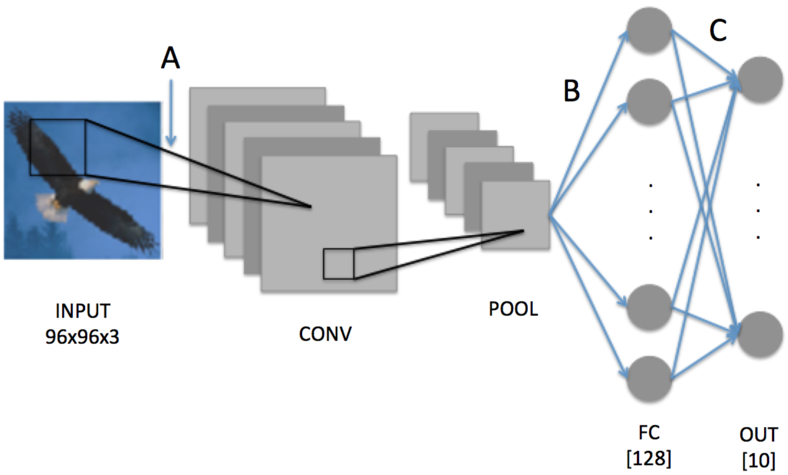


Figure 1: CNN for Problem 2

(a) (3 Points) Assume each of the 2D convolutional filters is of size 3 × 3. What is the shape of a single convolutional filter's weight matrix (label A)?

(b) (4 Points) What is the resulting shape of the output of a single convolutional filter? Assume all strides are 1, and there is no zero-padding

(c) (4 Points) You are employing a 2 × 2 max pooling layer with a stride of 2 in both directions with no zero-padding. What is the shape of the weight matrix that connects the output of all 16 pooling units in the pool layer to the 128 units in the fully connected layer (labeled B)?

(d) (3 Points) What is the shape of the weight matrix between the two fully connected layers (labeled C)?

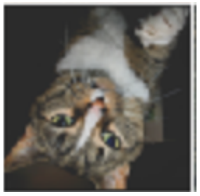
(e) (3 Points) What activation function of the final 10 outputs will you need to use to produce a vector of probabilities that sum to 1?

f) There are many architectural decisions that must be made when designing a neural network, and each one has its pros and cons. For each of the proposed changes below, give at least one negative impact it might have on the network.

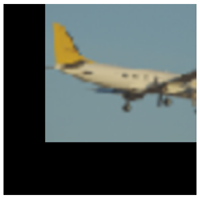
(i) (3 Points) Removing the 128-node fully connected layer

(ii) (3 Points) Replacing the convolutional layer with a fully connected layer

g) (3 Points) Suppose we wanted to test the robustness of our network by feeding it a modified version of the test set. Which of following images do you think the network would struggle more with? Why?



(a) A rotated image



(b) A translated image

Problem 3 (Recurrent Neural Networks) (28 Points)

Consider a simplified recurrent neural network architecture that operates on 1-D inputs $x_t \in \mathbb{R}$ and updates a 1-D hidden state $h_t \in \mathbb{R}$ at each time-step $t = 1, \ldots, T$ as follows:

$$h_t = v \cdot x_t + w \cdot h_{t-1} \tag{1}$$

where the RNN-parameters $w, v \in \mathbb{R}$ are both simply scalars rather than vectors/matrices. Each training sequence of inputs, denoted as $(x_1, \ldots, x_T)$, will be associated with a single scalar label $y \in \mathbb{R}$. To predict y given $(x_1, \ldots, x_T)$, we will simply use the final RNN hidden state h_T (at time-step T corresponding to the final input), and our prediction will incur mean-squared error loss $(y - h_T)^2$. For this problem, you *must* always assume each input $x_t \in \{0, 1\}$ takes either value 0 or 1, and we always set the initial hidden state $h_0 = 0$ before the RNN operates on a given input sequence.

a) (4 Points) Is there a setting of the parameters w, v such that this RNN will output a predicted value $h_T = 1$ when given the input sequence (of length $T = 7$): (0,0,0,0,0,0,0) where each $x_t = 0$ for $t = 1, \ldots, 7$? If yes, specify the values of w, v that would result in this prediction. Otherwise, explain why no such parameter-values exist.

b) (4 Points) Suppose we have a training set of (sequence, label) pairs where we always have $y = 2 \cdot x_T$ (the label associated with the sequence is always twice the value of the input at the final position). Is there a setting of the parameters w, v such that this RNN model could achieve zero training loss on all datasets (of arbitrary sample-size)? If yes, specify the values of w, v. Otherwise, provide an example of dataset (you are free to choose the sample-size and sequence-length T) where our RNN would not be able to achieve zero training loss.

c) (4 Points) Suppose we have a training set of (sequence, label) pairs where we always have $y = 2 \cdot x_1$ (the label associated with the sequence is always twice the value of the input at the first position). Is there a setting of the parameters w, v such that this RNN model could achieve zero training loss on all datasets (of arbitrary sample-size)? If yes, specify the values of w, v. Otherwise, provide an example of dataset (you are free to choose the sample-size and sequence-length T) where our RNN would not be able to achieve zero training loss.

Recall the following identities from differential calculus:

$$f(x) = c \cdot x \implies \frac{\partial f}{\partial x} = c \qquad f(x) = x + c \implies \frac{\partial f}{\partial x} = 1$$

$$f(x) = c \implies \frac{\partial f}{\partial x} = 0 \qquad f(x) = (y(x))^c \implies \frac{\partial f}{\partial x} = c \cdot (y(x))^{c-1} \frac{\partial y}{\partial x}$$

For some sequence-length T, suppose we are given the input sequence (1,0,0,0,…, 0), in which the first input $x_1 = 1$ and all other $x_t = 0$ for $t = 2, \ldots, T$. Suppose the corresponding label is $y = 2$. For the following questions, your answers should depend on T but no other variables (you do not need to simplify any numerical calculations).

d) (4 Points) Suppose the simplified RNN described in (1) is applied to the given input-label pair $x_{1:T} = (1, 0, 0, 0, \ldots, 0), y = 2$, and suppose the current parameter-values are $v = 1, w = 10$. In this case, what is $\frac{\partial L(v,w)}{\partial w}$, the partial derivative (with respect to w) of the resulting MSE loss $L = (y - h_T)^2$ evaluated at the current parameter configuration $(v = 1, w = 10)$?

e) (4 Points) What is $\frac{\partial L(v,w)}{\partial w}$ if the current parameter configuration is instead $v = 1, w = \frac{1}{10}$?

f) (4 Points) If T is very large (we have a long input sequence), what is the problem with derivative-based updates for training the w-parameter of this model in the case where we currently have $w = 10$ or when $w = 1/10$?
What is a practical remedy for this problem that can be applied when $w = 10$ but not for the case where $w = 1/10$?

We now consider a highly simplified version of the LSTM model which performs the following updates at each time-step utilizing a basic gating procedure:

$$\begin{aligned} h_t &= (1 - g_t) \cdot h_{t-1} + g_t \cdot s_t \\ s_t &= v \cdot x_t + w \cdot h_{t-1} \end{aligned} \tag{2}$$

where the parameters of this model $w, v \in \mathbb{R}$ are all still scalar values. At each time-step in this simplified model, we are free to choose any value for the scalar gate g_t from the interval [0,1], which may vary across different t (traditional LSTM-style models instead use a parameterized gating function which operates on the inputs and previous hidden-state).

g) (4 Points) Like in part (c), suppose we have a training set of (sequence, label) pairs where we always have $y = 2 \cdot x_1$ (the label associated with the sequence is always twice the value of the input at the first position). Is there a setting of the parameters w, v and the gates $g_1, \ldots, g_T$ such that this simplified LSTM could achieve zero training loss on any dataset (of arbitrary sample-size) of this form? If yes, specify these values. Otherwise, provide an example of dataset (you are free to choose the sample-size and sequence-length T) where our simplified LSTM would not be able to achieve zero training loss.

Problem 4 (Neural Network Interpretability) (16 Points)

Determine which of the following statements are true:

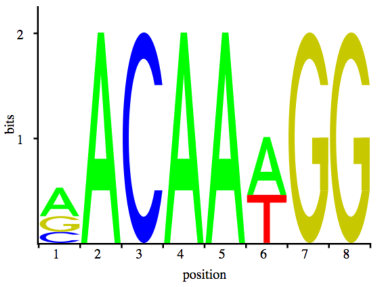
a) (2 Points) Given a first-layer convolutional filter from a CNN trained to predict TF binding from DNA sequences, we can get the nucleotide probability at each position by normalizing each column of the filter to sum to one ? (Yes / No)

b) (2 Points) We can select the best set of hyper-parameters by finding the combination that gives the lowest training loss? (Yes / No)

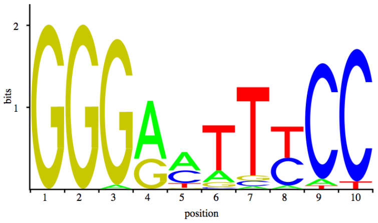
Suppose we have the following convolutional kernel from a CNN trained to predict TF binding from DNA sequences.

$$\begin{bmatrix} A \\ C \\ G \\ T \end{bmatrix} = \begin{bmatrix} -1.1 & -0.5 & 1.3 & 0.3 & 0.9 & 0.19 & 0.09 & 0.22 & -0.09 & 0.01 \\ 4.1 & 2.5 & -0.3 & -0.08 & 1.3 & -0.13 & -0.05 & 0.36 & 0.03 & 0.1 \\ 0.5 & 0.1 & -0.1 & 0.13 & 0.01 & 0.91 & 0.13 & 0.11 & 0.01 & -0.01 \\ 0.6 & 0.5 & 1.6 & 2.5 & 3.1 & -0.57 & 0.73 & 0.21 & -0.05 & 0.12 \end{bmatrix}$$

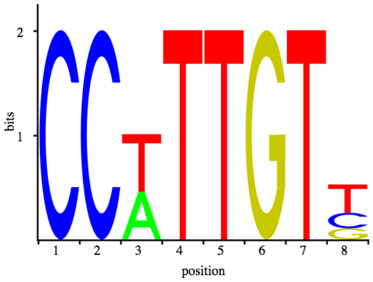
And we have the forward-strand motifs of the following TFs represented in bit-information logo as follows:



(a) TF 1



(b) TF 2



(c) TF 3

c) (4 Points) The sequence determinants of which of above TF(s) can be characterized by this convolutional kernel?

One way to interpret a specific neuron in a neural network is to fix the network weights, and find the input that can maximize the output of that neuron. Consider a one-layer linear network with weight $W = [w_1, w_2, w_3]^T$. Suppose the input is $X = [x_1, x_2, x_3]^T$, then the output is $y = W^T X = w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3$. In the following questions, assume W is fixed and $W = [1, 2, 3]^T$.

d) (2 Points) Is there a finite input X that can maximize Y? If yes, specify the optimal input vector. If not, explain why.

e) (4 Points) If we force the L_2 norm of X to be 1, i.e. $(x_1^2 + x_2^2 + x_3^2)^{\frac{1}{2}} = 1$, is there a finite input X that can maximize Y? If yes, specify the optimal input vector. If not, explain why.

f) (4 Points) In practice, due to the complexity of the network, we usually search the optimal input using gradient ascent (not "descent" as here we try to maximize the target function). With a learning rate of 0.5, if $X = [4, 5, 6]^T$ at iteration t, what would be the value of X at iteration $t + 1$? Here we don't assume any constraint on the L_2 norm of X.