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

April 23, 2019

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	22	
2	30	
3	18	
4	30	
Total	100	

Problem 1 (Short Answer Problems) (22 Points)

a) (5 Points) You observe that Gene A and Gene B are highly correlated in expression data for 20 different conditions. Whenever Gene A is expressed Gene B is expressed. To investigate further you knock out Gene A, and note that Gene B is still expressed in certain conditions. Explain how this could be happening given your data was consistent with the causal regulation of Gene B by Gene A.

There is a latent confounder, such as a gene C that regulates both A and B

b) (6 Points) You hypothesize that five transcription factors $\mathrm{TF}_1, \ldots \mathrm{TF}_5$ may regulate the transcription of gene G. You perform intervention experiments on the transcription factors one at a time, and calculate on an individual basis the probability that you would observe the changes in gene G transcripts in your data at random (null hypothesis) for $\mathrm{TF}_1, \ldots \mathrm{TF}_5$ as 0.003, 0.006, 0.020, 0.045, and 0.600, respectively. Using Bonferonni multiple hypothesis correction for which TFs can you reject the null hypothesis? Using Benjamini-Hochberg for an expected false discovery rate (FDR) of 0.05, for which TFs can you reject the null hypothesis that the observed expression change is occurring at random?

Bonferonni up to TF_2, Benjamani-Hochberg up to TF_3

c) (6 Points) We have observed we can regularize weight values by adding a term onto a loss function that penalizes weight magnitude. In Figure 1 below we show $\|w\|\|_1$ (L1 norm, left) and $\|w\|_2$ (L2 norm, right). The point w* in both figures is on the same loss contour but makes different weight choices that minimize the respective norm. Describe which method will choose the sparsest set of non-zero weights and why.

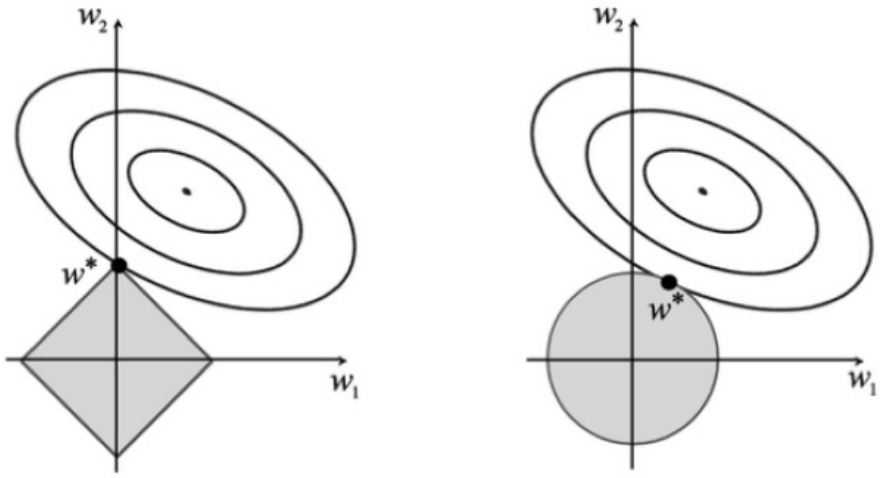


Figure 1: Weight regularization at a fixed loss contour

$\|w\|_1$ will choose the sparsest set of weights as the point of intersection w* will typically be where w_1 or w_2 is zero

Problem 2 (Neural Networks) (30 Points)

For subproblems (a) and (b) we consider a neural network with the following architecture:

1. Input layer: grid of 5 x 5 real-valued input features (e.g., images with 5 x 5 grayscale pixels)
2. Convolutional layer with 10 filters of size 2 x 2, stride of 1, zero-padded (*same* padding)
3. Flatten layer that concatenates the results of all convolutional filters
4. Output layer: fully connected layer with 2 units

There are no bias terms.

(a) What are the output shapes of each layer? E.g., the shape of the input (layer) is `(?, 5, 5)` or `(?, 5, 5, 1)` in TensorFlow's NHWC format.

(i) (2 Points) Output shape of convolutional layer:
(?, 5, 5, 10)

(ii) (2 Points) Output shape of flatten layer:
(?, 250)

(iii) (2 Points) Output shape of output layer:
(?, 2)

(b) How many trainable parameters (i.e., weights) does each layer of the network have? As mentioned above, there are **no bias terms** in this network.

(i) (2 Points) Number of trainable parameters in the input layer:
0

(ii) (2 Points) Number of trainable parameters in the convolutional layer:
2 * 2 * 10 = 40

(iii) (2 Points) Number of trainable parameters in the flatten layer:
0

(iv) (2 Points) Number of trainable parameters in the output layer:
250 * 2 = 500

For subproblems (c) and (d) we change the previous network architecture by adding a max-pooling layer after the convolutional layer:

1. Input layer: grid of 5 x 5 real-valued input features (e.g., images with 5 x 5 grayscale pixels)
2. Convolutional layer with 10 filters of size 2 x 2, stride of 1, zero-padded (*same* padding)
3. Pooling layer with a 2 x 2 pool size, stride of 2 in both directions, zero-padded (*same* padding)
4. Flatten layer
5. Output layer: fully connected layer with 2 units

Again, we assume there are no bias terms.

(c) How do the output shapes of subsequent layers change?

(i) (2 Points) Output shape of pooling layer:
(?, 3, 3, 10)

(ii) (2 Points) Output shape of flatten layer:
(?, 90)

(iii) (2 Points) Output shape of output layer:
(?, 2)

(d) How does the number of trainable parameters of each layer change? As mentioned above, there are **no bias terms** in this network.

(i) (2 Points) Number of trainable parameters in the input layer:
0

(ii) (2 Points) Number of trainable parameters in the convolutional layer:
40

(iii) (2 Points) Number of trainable parameters in the pooling layer:
0

(iv) (2 Points) Number of trainable parameters in the flatten layer:
0

(v) (2 Points) Number of trainable parameters in the output layer:
90 * 2 = 180

Problem 3 (Model uncertainty and Bayesian optimization) (18 Points)

a) (6 Points) Suppose you want to predict TF binding intensity to DNA sequences and wish to model both the aleatoric and epistemic uncertainty for that problem. You collect data for TF binding intensity via genomic Chip-Seq, where you observe pairs of {DNA sequence, Observed read count}. Assume the higher the read count for a DNA sequence, the stronger the binding. Give an example of (i) a source of aleatoric uncertainty for these data. [As an example, for images this could be noise in the light to the camera] (ii) an example of what kind of training data may give rise to epistemic uncertainty [As an example, if you have a classifer for dogs and cats, and the all the dogs are brown while all the cats are white, there is going to be model uncertainty as to whether the color is right feature for classification or the shape of the animals as you could get equally good classifiers on the training data using either feature].

(i) PCR errors for example for sequencing or biological stochastic noise in gene expression (ii) a motif is causative for binding but highly correlated with another motif in the training data. There would be model uncertainty between the two motifs.

b) (6 Points) Suppose you have trained your above network and want to understand whether the uncertainties it is estimating are accurate. How would you expect the (1) aleatoric and (2) epistemic uncertainties to behave on test sequences (increase/decrease) that are close versus far away in terms of edit distance in sequence space from the training data sequences? Explain your answer for both types of uncertainties separately to get credit.

(1) Epistemic uncertainty should increase for far off sequences compared to close by ones as far off sequences are likely out of distribution on which model uncertainty should be higher (2) Aleatoric uncertainty should remain the same as it should not depend on whether data is out of distribution. It is possible however that the training data has a weird bias that leads to higher aleatoric uncertainty on OOD. So if the student writes that no particular behavior should be expected, that's fine too.

c) (6 Points) Assume your aleatoric noise is homoscedastic (does not vary with the input). If you are performing Bayesian optimization would modeling aleatoric uncertainty in your ensemble then make any difference if you were to use the upper confidence bound (UCB) as your acquisition function? Explain your answer for full credit.

The set of solutions for a neural network will remain the same no matter the aleatoric uncertainty if it is homoscedastic. So modeling it shouldn't make a difference to the mean prediction or the epistemic uncertainty which is what UCB uses. If aleatoric uncertainty is made part of UCB however, depending on how it is incorporated, it could have an effect on acquisition as it could effectively change the weighting of the epistemic uncertainty term relative to the mean prediction term.

Problem 4 (Neural Network Interpretability) (30 Points)

Determine which of the following statements are true:

a) (3 Points) During the generation of a sufficient input subset (SIS) with threshold τ, we keep removing pixels until the predicted score on the remaining pixels become smaller than τ and use the remaining pixels as an SIS? (Yes / No) **No**

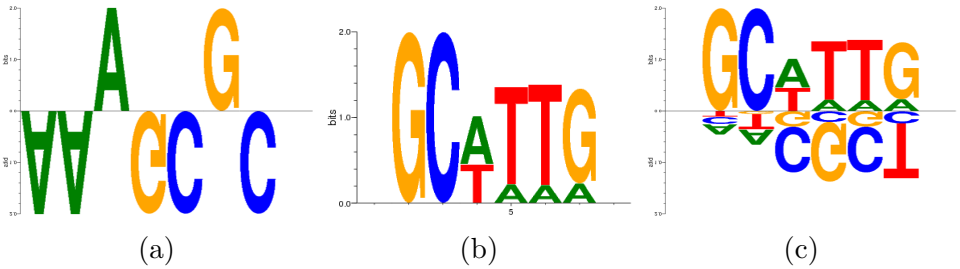
b) (3 Points) SIS can produce multiple SISs sequentially for a given input x. Might these SISs contain overlapping features? (Yes / No) **No**

c) (3 Points) Integrated gradients (IG) computes the path integral of the gradients along the straight-line path from a baseline to the given input. Will IG remain the same no matter what reference baseline input is used? (Yes / No) **No**

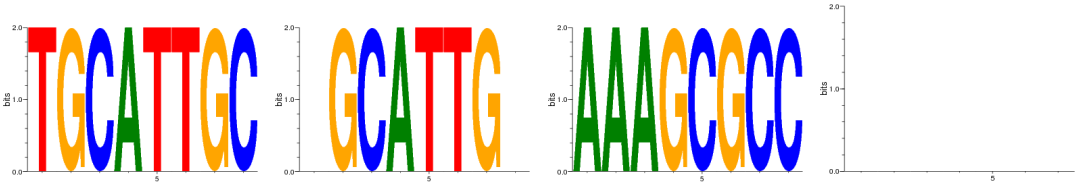
Suppose we trained a convolutional neural network to predict TF binding from DNA sequences with one convolutional kernel (weights are as following) and no fully connected layer. *Valid* padding is used for convolutional layer and thus no zero-padding is needed. The output of the convolutional layer is follwed by ReLUs and then global max-pooling. The bias is zero for the ReLU. Assume the output from global max-pooling is directly used as the prediction.

$$\begin{bmatrix} A \\ C \\ G \\ T \end{bmatrix} = \begin{bmatrix} -1.1 & -0.5 & 1.4 & 0.3 & 0.2 & 0.19 \\ -0.8 & 2.5 & -0.3 & -0.08 & -0.3 & -0.13 \\ 4.1 & -0.1 & -0.1 & -0.43 & -0.1 & 0.91 \\ -0.6 & -0.5 & 1.1 & 1.5 & 1.1 & -0.57 \end{bmatrix}$$

We interpret this network with different methods and produce following visualization results, which are seqLogo of the attribution scores.The flipped parts below the center line represent the SeqLogo of negative values if they exist.



(a) (b) (c)



(d) (e) (f) (g)

d) (5 Points) The Saliency Map method interprets a network by computing the gradients of the network's output with respect to its **input**. Unlike gradients used to modify parameters during training, in the Saliency Map method gradients are back-propagated all the way to the input layer with fixed model parameters that are not updated. The rules for back propagation are exactly the same (recall the chain rule and special rules for back-propagating gradients through max-pooling and ReLU units). Which of them is most likely to be the sequence logo visualization of the Saliency Map of the input sequence TGCATTGC?

c

e) (6 Points) One variation of the saliency map is to multiply the gradients with the input such that the visualization is cleaner. Please write down the saliency map $\times$ input result for input sequence TGCATTGC as a 4×8 matrix. Which one from the above is most likely to be the sequence logo visualization of this matrix?

e

$$\begin{bmatrix} 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.1 & 0 & 0 & 0 & 0 & 0.91 & 0 \\ 0 & 0 & 0 & 0 & 1.5 & 1.1 & 0 & 0 \end{bmatrix}$$

f) (5 points) Another variation of the saliency map only back-propagate positive gradients (recall the de-convolutional neural networks). Which one from the above for input sequence TGCATTGC is most likely to be the sequence logo visualization of this method?

b

g) (5 points) Which of the above visualizations is most likely to be the sequence logo visualization of the Saliency Map of input sequence AAAGCGCC? Recall this is a convolutional network! Assume VALID and no padding.

g