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

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)

As a χ^2 distribution with one degree of freedom

b) (4 Points)

Choose Model 1 if $P_1(D|\vec{\theta}_1) - \frac{k_1}{2}\log n > P_2(D|\vec{\theta}_2) - \frac{k_2}{2}\log n$

c) **Accept the null hypothesis if $p > \frac{\alpha}{n}$**

d) (4 Points)

Test set error that decreases with capacity and then increases

e) (4 Points)

We expect $\|\vec{w}\|_1$ as it favors parameter sparsity while $\|\vec{w}\|_1$ favors parameter magnitude

f) (4 Points)

Gradient descent

g) (4 Points)

Step 1 - Assign genes to clusters. A gene can be assigned to a single cluster or probabilities can be associated with cluster membership. Step 2 - Compute cluster parameters (such as mean) based upon cluster membership. The method used is expectation Maximization.

Problem 2 (Convolutional Neural Networks) (28 Points)

(a) (3 Points) $\boxed{3 \times 3 \times 3}$

(b) (4 Points) $\boxed{94 \times 94}$

(c) (4 Points) $\boxed{(47^2 \times 16, 128) \text{ or } (35344, 128)}$

(d) (3 Points) $\boxed{(128, 10)}$

(e) (3 Points) $\boxed{\text{Softmax}}$

f) (i) (3 Points)

(One possibility) Removing the fully connected layer removes the possibility of learning any non-linear combinations from the feature space learned by the convolutional layers to the output space

(ii) (3 Points)

(One possibility) Convolutional layers can learn 2D features with much fewer parameters than a fully-connected layer, so in order to achieve the same complexity as a convolutional layer, the fully connected layer would have to be extremely large, greatly increasing the size of the network

g) (3 Points)

The rotated image. The sliding window nature of a convolution means that it is translation invariant, not rotation invariant. In fact, there are even some cases where rotation invariance is bad (ex. a rotated 6 is a 9)

Problem 3 (Recurrent Neural Networks) (28 Points)

a) (4 Points)

Not possible. We will always have $h_T = 0$ regardless of the choice of w, v. For example: $h_1 = v \cdot x_1 + w \cdot h_0 = 0$ since $x_1 = h_0 = 0$. Similarly, h_2 will always equal 0 and so on...

b) (4 Points)

Yes, choose $v = 2, w = 0$.

c) (4 Points)

Not possible. Consider dataset consisting of two sequences: (0, 1) and (1,1). Under our rule, the first sequence will receive label 0, the second label 2. For $T = 2$: we have $h_T = v \cdot x_2 + w \cdot v \cdot x_1$ (recursively applying the RNN update with $h_0 = 0$). Any parameter-values which would give zero training loss, would have to satisfy the following unsolvable system of equations:

$$0 = 1 \cdot v + 0 \cdot v \cdot w$$
$$2 = 1 \cdot v + 1 \cdot v \cdot w$$

d) (4 Points)

Recursively applying the update equation, we have: $h_T = w^{T-1}v = w^{T-1}$ when $v = 1, x_1 = 1, x_2 = x_3 = \cdots = x_T = 0$.
Thus: $\frac{\partial h_T}{\partial w} = (T-1) \cdot w^{T-2}$
Also: $\frac{\partial L}{\partial h_T} = -2 \cdot (2 - h_T) = -2(2 - w^{T-1})$
We therefore have: $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial h_T} \cdot \frac{\partial h_T}{\partial w} = -2(T-1)w^{T-2}(2 - w^{T-1})$
Finally, we plug-in the value $w = 10$ to obtain the answer.

e) (4 Points)

Plug-in the value $w = 1/10$ in our derivative expression obtained in (d).

f) (4 Points)

From (d), we see that the partial derivative with respect to w involves w raised to the power T. Thus, when $w = 10$, we will have exploding gradients, and when $w = 1/10$, we will have vanishing gradients. To remedy the case of exploding gradients, we can simply clip the derivatives before taking a gradient-step in the optimization, as done in Problem Set 2.

g) (4 Points)

Yes. Choose $v = 2, w = 0, g_1 = 1, g_t = 0$ for all $1 < t \leq T$. The resulting model will no longer alter its hidden state after time step $t = 1$, and will have the correct hidden-state at $t = 1$.

Problem 4 (Neural Network Interpretability) (18 Points)

a) (2 Points)

No. The actual values of the weights result from optimizing a highly non-linear transformation of the input, and thus do not correspond to the ratio of nucleotide frequency at each position of a binding sequence.

b) (2 Points)

No, we should pick based on *validation* loss so that these hyper-parameters can generalize to unseen test data.

c) (4 Points)

TF 1 (1 point) and TF 3 (3 points). It's easy to see why TF 3 matches by just comparing the nucleotide with the largest weight / bit information. Note that logo (a) and (c) are reverse-complement to each other. So TF 1 and 3 essentially bind to the same sequences (DNA are double-helix and have two strands that are reverse-complement to each other). Thus the kernel can model the binding of TF 1 too.

d) (2 Points)

No, y is a linear function of each of x_1,x_2, and x_3 independently.

e) (4 Points)

Yes (1 point). $x_1^* = \frac{1}{\sqrt{14}}$, $x_2^* = \frac{2}{\sqrt{14}}$, $x_3^* = \frac{3}{\sqrt{14}}$. One possible solution: First get x_1^* and x_2^* by taking the derivatives of $Y = w_1x_1 + w_2x_2 + w_3\sqrt{1 - x_1^2 - x_2^2}$ with respect to x_1 and x_2 and set them to zero. Then $x_3^* = \sqrt{1 - x_1^{*2} - x_2^{*2}}$. (2 points for proposing a sound way to calculate the optimal value, and 1 point for the right answer)

f) (4 Points)

$$\begin{aligned} X^{t+1} &= X^t + \alpha\frac{\partial Y}{\partial X}(2points) \\ &= X^t + 0.5W(2points) \\ &= [4.5, 6, 7.5]^T \end{aligned}$$