Final Exam Long Answer Questions

Introduction to Deep Learning
CMPT 980
Simon Fraser University
Spring 2020

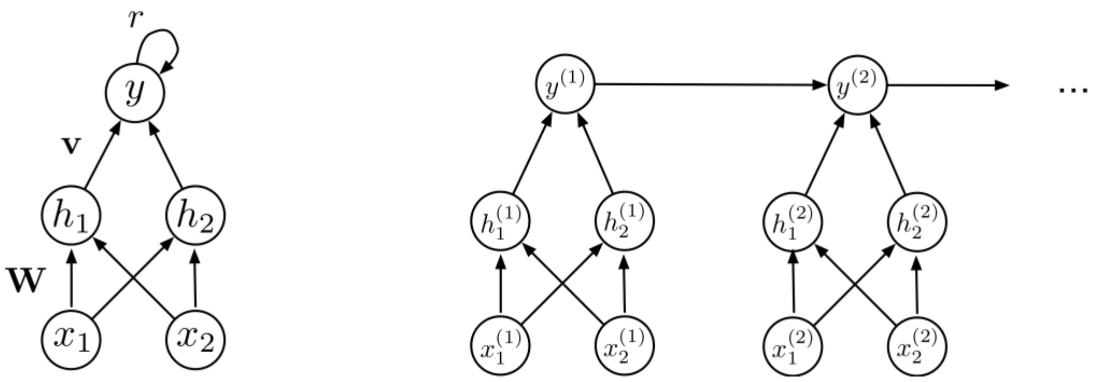
Instructions: Below are listed 6 questions. Exactly 2 of these questions will appear on your exam. You can upload two files, one for each answer. You should explain your answers, even if not explicitly asked to do so. For example, if a question asks: “what is the derivative of equation E”, you should not only give the derivative, but also outline how you obtained it.

The exam is open-book, in the sense that you can consult the textbook and on-line resources like Wikipedia. The university policy on academic dishonesty and plagiarism (cheating) will be taken very seriously in this course. *Everything submitted should be your own writing or coding.* You must not let other students copy your work. If I determine that you have copied, you will receive 0 marks.

Group Work: Discussing the problems is okay, for example to understand the concepts involved. If you work in a group, put down the name of all members of your group. There should be no group submissions. *Each group member should write up their own solution* to show their own understanding.

Question on Recurrent Neural Networks

We want to process two binary input sequences with 0-1 entries and determine if they are equal. For notation, let $x_1 = x_1^{(1)}, x_1^{(2)},\ldots,x_1^{(T)}$ be the first input sequence and $x_2 = x_2^{(1)}, x_2^{(2)},\ldots,x_2^{(T)}$ be the second. We use the RNN architecture shown in the Figure.



The corresponding update equations are as follows.

$\mathbf{h}^{(t)}=g(\mathbf{Wx}^{(t)}+\mathbf{b})$	
$y^{(t)}=g(\mathbf{v}^T\mathbf{h}^{(t)}+ry^{(t-1)}+c)$	for t>1
$y^{(t)}=g(\mathbf{v}^T\mathbf{h}^{(t)}+c_0)$	for t=1

Where $\mathbf{v}^T$ is the transpose of vector **v** and the activation function *g* is defined as follows.

$g(z)=1$	for $z > 0$
$g(z)=0$	for $z \leq 0$

Described in words, the parameters are as follows.

W	2x2 weight matrix
b	2-dimensional bias vector
v	2-dimensional weight vector
r	Scalar recurrent weight
c	Scalar bias for all time steps except the first
c_0	Scalar bias for the first time step

I suggest the following strategy for solving this problem.

- At time step *t*, the neural network is fed two inputs $x_1^{(t)}$ and $x_2^{(t)}$.
- Use the two hidden units $h_1^{(t)}$ and $h_2^{(t)}$ to determine if the current inputs match.
- Use the output unit $y^{(t)}$ to compute whether all inputs have matched up to the current time.

Specify parameter values that correctly implement this function, like in the table shown. (You do not have to write your answer in the table). Justify why you think your parameter values are correct.

W	Your solution
b	Your solution
v	Your solution
r	Your solution
c	Your solution
c_0	Your solution

Question on Gated Recurrent Neural Networks

Suppose we want to build an RNN cell that sums its inputs over time.

1. For the LSTM architecture as explained in Section 4.6. of the text, what should be the value of the input gate and the forget gate?
2. For the GRU architecture, what should be the value of the reset gate and the update gate? The GRU architecture is described in the slides and in on-line sources like this one https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

Question on Variational Auto-Encoders

1. Compare the loss function for the Variational Auto-Encoder (Figure 7.8 in the text) to the loss function for an associative auto-encoder (Figure 7.1 in the text). Which parts are similar and which are different?
2. How does the VAE architecture allow it to generate new data points, especially compared to associative auto-encoder, which cannot generate new data points?
3. Let *d* be the latent embedding dimension. The VAE encoder outputs a mean vector $\boldsymbol{\mu}=(m_1,m_2,..,m_d)$ and a variance vector $\boldsymbol{\sigma}=(s_1,..,s_d)$, where each $s_i \geq 0$. The variational loss function for this output is given by

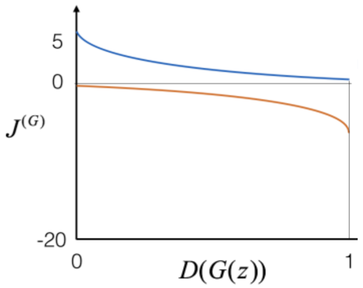
   $$½ \sum_{i=1,..d} \{(s_i)^2+(m_i)^2-\ln[(s_i)^2]-1\}.$$

   (This equation is somewhat different from the book.) Show that this variation loss is minimized when $\boldsymbol{\mu}=\boldsymbol{0}$ and $\boldsymbol{\sigma}=\boldsymbol{1}$ (i.e. all the means are 0 and all variances are 1).

Question on Generative Adversarial Networks

Considering training a GAN with generator *G(z)* and discriminator *D(G(z)).* Figure 1 shows the training losses for two generator loss functions. In detail, for the first m generated points, i = 1,…,m. Each point in the plot shows the value of $D(G(z_i))$, $J_1(G)$ and $J_2(G)$, defined as follows.

1. $J_1(G) = -1/m \sum_{i=1,..m} ln(D(G(z_i)))$. Shown as the blue curve.
2. $J_2(G) = 1/m \sum_{i=1,..m} ln(1-D(G(z_i)))$. Shown as the orange curve.



1. Early in the training, is the value of *D(G(z))* closer to 0 or closer to 1? Explain why.
2. Which of the two cost functions would you choose to train your GAN? Justify your answer.
3. A GAN is successfully trained when *D(G(z))* is close to 1. True or False? Explain your answer.

Question on Seq2Seq Problems

Consider multiple-length seq2seq for an French-to-English MT program, as described in the text. We have decided on two sentence sizes

1. Up to 8 words (and STOPs) for English, 10 for French
2. Up to 10 words for English, 13 for French.

Write out the input if the French sentence is “AB C D E F" and the English is “M N O P Q R S T".

Question on Attention

Consider the original transformer architecture as described in the paper "Attention is All You Need" (https://arxiv.org/abs/1706.03762) and also in this blog post http://jalammar.github.io/illustrated-transformer/. Both the encoder and the decoder each use a stack of 6 self-attention modules. For the answers below, *assume 1 attention head only* (not 8 as in the paper). As shown in the notebook https://colab.research.google.com/github/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/hello_t2t.ipynb#scrollTo=OJKU36QAfqOC the input sentence and output translation are as follows.

Input: "The animal didn't cross the street because it was too tired"
Output: "Das Tier überquerte die Straße nicht, weil es zu müde war"

All the following questions pertain to this example. Treat each word as a separate token, so the input sequence contains 11 items, and the output sequence contains also 11 items.

1. Consider the third output word "überquerte". Describe how the representation of this output word is computed. List the key vectors used (e.g. key vector for first input word), query vectors used (e.g. key vector for first input word), and value vectors used (e.g. value vector for first input word)? How is each of these key/query/value vectors used computed?
2. To produce the input and output shown, how many key, query, and value vectors are computed in total by the encoder? How many by the decoder *in total* (not just for the 3rd output word)? Fill in the following table and also explain your answer.

Vector Type	Key	Query	Value
Encoder	#?	#?	#?
Decoder	#?	#?	#?

3. An attention weight connects an input word to other words. For each word in the input sequence:
   a. how many attention weights for other words are computed for each word in the input sequence during *encoding*?
   b. How many attention weights for other words are computed during *decoding*?