CSE 253
Winter 2018
Midterm 2
SECTION B
VERSION 1

NO NOTES ALLOWED!!!
Thank you!
DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO START!!!!

March 7, 2018

Name	
PID	

- Once the exam has started, SORRY, NO TALKING!!
- There are 5 problems: Make sure you have all of them - AFTER YOU ARE TOLD TO START!
- Read each question carefully.
- Remain calm at all times!

Problem	Type	Points	Score
1	True or False	10	
2	Multiple Choice	10	
3	Recurrent Neural Networks	10	
4	Long-Short Term Memory	10	
5	Reinforcement Learning	10	
	Total	**50**	

1 True or False

(10 pts: +1 for correct, -0.5 for incorrect, 0 for no answer) If you would like to justify an answer, feel free. PLEASE *clearly mark in the provided blank* T or F.

a. F Dropout randomly drops some of the neurons in a layer with predefined probability. The same effect can also be achieved by reducing the number of neurons in that layer with no dropout.

? b. One way to avoid "dead" ReLU units is to set the initial bias to a positive number. A ReLU neuron is "dead" if it's stuck in the negative side and always outputs 0. Because the slope of ReLU in the negative range is also 0, once a neuron gets negative, it's unlikely for it to recover.

c. T An autoencoder learns to reproduce its input on its output, and can be used for dimensionality reduction.

d. F An LSTM unit is just a simplified version of a GRU unit.

? e. For an LSTM cell, the forget gate $f_t \in (0, 1)$ represents how much information it can forget from the last state.

f. F While generating music in your last programming assignment, using the ~~argmax~~ of the outputs lead to a larger variety of music produced.

? g. In language translation (English to French) tasks using encoder-decoder architecture, the output sentence should be of same length as the input sentence.

? h. In the Neural Turing Machine, computing the address of an item is a simple process.

? i. While training the policy network, we discourage all of the good actions in an action sequence if it leads to a negative outcome because of some bad actions.

j. T Wally used a Recurrent neural network to write his PhD thesis. However the grammar of report was so bad, Wally finally dumped his RNN and chewed up his thesis himself.

2 Multiple Choice

(10 pts, 2 pts each. 2 extra credit questions). There is only *one* best answer. PLEASE *CIRCLE* your answer.

1. When calculating Δw_{ij}, what does the update rule depend on (consider the activation at layer k to be z_k)?

   i. z_j

   ii. w_{ij}

   iii. δ_j

   iv. Δw_{jk}

2. Which of the following is *crucial* for a convnet to exhibit translational invariance?

   i. Average Pooling

   ii. Convolutions with stride > 1

   iii. Max Pooling

   iv. ReLU activation function

   v. (i, iii, and iv)

   vi. (ii, iii, and iv)

3. You notice that the loss of your autoencoder does not decrease in the first few epochs. It has been initialized with Xavier Initialization. The reasons for this could be:

   i. The learning is rate is too high

   ii. The regularization parameter is too low

   iii. The network is on a plateau

   iv. The activation function is always positive

   v. (i and ii)

   vi. (i, ii, and iii)

4. Which following statements is NOT true for recurrent neural networks?

   i. The state of the network is used to encode the network's place in the sequence.

   ii. Without LSTMs, RNNs struggle to encode very long-term patterns

   iii. They must be symmetrically connected

   iv. LSTMs can be stacked into multiple layers to process more complex sequences.

5. The optimal Q function $Q^*(s,a)$ represents

    i. The maximum expected cumulative reward from state s.

    ii. The maximum expected cumulative reward from state s, assuming action a is taken.

    iii. The maximum expected cumulative reward from state s, assuming a is the optimal action at state s

6. Which of the following does NOT help combat the vanishing gradients problem

    i. Using ReLU activations.

    ii. Clipping the gradients.

    iii. Using LSTM units.

    iv. Using batch normalization.

7. Which of the following is TRUE

    i. In policy gradients, we increase the probability of every action in an action sequence that resulted in positive reward, therefore possibly rewarding 'bad' intermediate actions.

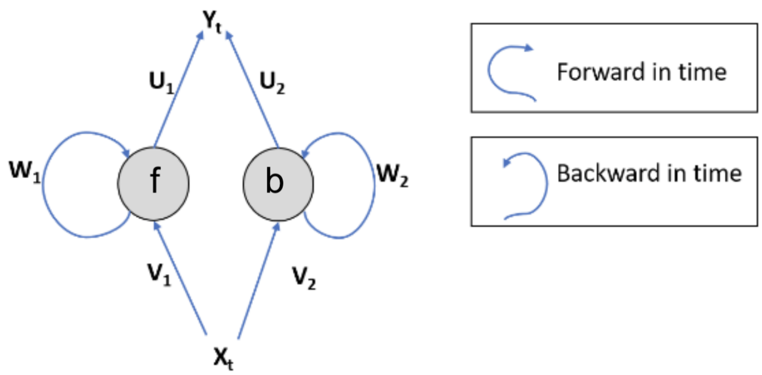
    ii. In policy gradients, we increase the probability of only those actions in an action sequence that directly contribute to the positive reward, therefore rewarding only 'good' actions.

    iii. A neural network that attempts to learn a Q-function needs to have one output unit for every state-action pair.

    iv. AlphaGo Zero is initially trained on a database of human expert moves.

3 Recurrent Neural Networks

In this question we are going to explore a variant of Recurrent Neural Networks called Bidirectional Recurrent Neural Networks. The figure below shows a Bidirectional RNN. It has the following hidden layers (with only one unit in each, so *everything here is a scalar!*): (1) a "forward layer" (unit f) that processes input from left to right, modeling prior information (same as a simple RNN), and (2) a "backward layer" (unit b) that processes input from right to left, modeling future information.



In a BRNN, *these two hidden layers are independent of one another*, but both get the same inputs. The output layer is obtained by combining the results of the above two layers.

1. **Draw the network unrolled in time.** (5 pts) Draw the architecture unrolled for two timesteps (i.e., show the network at t, $t+1$, and $t+2$), and label the weights, making explicit which weights are the same.

2. **Activation Propagation.** (3 pts) Fill in the blanks below, writing the expression for the hidden state and the output. $h_f(t)$, $h_b(t)$ are the forward and backward hidden states respectively at time t. g_f and g_b are their activation functions respectively.

   $h_f(t) = g_f($________________________$)$ $\qquad$ $h_b(t) = g_b($________________________$)$

   $y_t = softmax($________________________$)$

3. **Thought question** (2 pts) Give one situation where having information propagate back from the future might be useful (one or two sentences).

4 Long-Short Term Memory

Here are the operations involved in an LSTM cell. The hidden state of the LSTM at any time-step t is given by the vectors c_t and h_t. x_t is the input at time-step t (this LSTM cell is just above the input). i, f, and o are the input, forget, output gates, respectively, while g is the input to the cell.

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1})$$
$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1})$$
$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1})$$
$$g_t = tanh(W^{(g)}x_t + U^{(g)}h_{t-1})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot tanh(c_t)$$

$\odot$ denotes the element-wise multiplication.

a) (2 pts) Recall that h_t, x_t are vectors. Let d_x and d_h be dimensions of x_t and h_t respectively. What are the dimensions of the matrices $W^{(i)}$, $U^{(i)}$, $W^{(g)}$ and $U^{(g)}$

b) (2 pts) Explain how the LSTM overcomes the vanishing/exploding gradients problem faced by an RNN.

c) (2 pts) What is the idea behind using TWO vectors (c_t, h_t) to represent the hidden state of the LSTM at any time-step?

d) (2 pts) Explain what f_t does?

e) (2 pts) What is the reason for choosing $tanh$ as an activation for computing g_t and sigmoid (σ) for computing o_t?

5 Reinforcement Learning

a) (4 pts) Describe how reinforcement learning differs from supervised learning. Be sure to mention what the system is trying to learn. Quoting Russell and Norvig's definition is not what I'm looking for here!

b) (3 pts) Briefly describe how *policy gradient* works. Be sure to describe what happens on each move, and what happens at the end of the game.

c) (3 pts) *AlphaGo Lee* (the first version of AlphaGo) used three networks to play Go. Without going into great detail, what were they and what was their function?