UNIVERSITY OF TORONTO
FACULTY OF ARTS AND SCIENCE

FINAL EXAMINATION, APRIL 2016

DURATION: 3 hours

CSC 321 H1S — Introduction to Neural Networks and Machine Learning

Aids allowed: Non-programmable calculators
Examiner(s): M. Guerzhoy

Please detach aid sheet if necessary

Student Number: |_|_|_|_|_|_|_|_|_|_|_|_|_|

Family Name(s): _____

Given Name(s): _____

---

*Do **not** turn this page until you have received the signal to start.*
In the meantime, please read the instructions below *carefully*.

This final examination paper consists of 8 questions on 30 pages (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy is complete, fill in the identification section above, and write your student number where indicated at the bottom of every odd-numbered page (except page 1).*

Answer each question directly on this paper, in the space provided, and use the reverse side of the previous page for rough work. If you need more space for one of your solutions, use the reverse side of a page or the pages at the end of the exam and *indicate clearly the part of your work that should be marked.*

Write up your solutions carefully! In particular, use notation and terminology correctly and explain what you are trying to do—part marks *will* be given for showing that you know some aspects of the answer, even if your solution is incomplete.

Code documentation is not required, although they may help us mark your answers, particularly when parts of the code are missing. They may also be worth part marks if you cannot figure out how to complete your code, but completed some part of it.

A mark of at least **40%** (after adjustment, if there is an adjustment) on this exam is required to obtain a passing grade in the course.

MARKING GUIDE

# 1:	_____/ 15
# 2:	_____/ 15
# 3:	_____/ 20
# 4:	_____/ 10
# 5:	_____/ 5
# 6:	_____/ 10
# 7:	_____/ 15
# 8:	_____/ 10
TOTAL:	_____/100

Good Luck!

OVER...

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 1. [15 MARKS]

Part (a) [10 MARKS]

Sketch the *typical* learning curves for the training and validation sets, for a setting where overfitting occurs at some point. Assume that the training set and the validation set are of the same size. Label all the axes, and label the curves that you sketch. Make sure that your sketch is as complete as possible in terms of demonstrating the training process.

Part (b) [5 MARKS]

State how to apply early stopping in the context of learning using Gradient Descent. Why is it necessary to use a validation set (instead of simply using the test set) when using early stopping?

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 2. [15 MARKS]

When learning using Stochastic Gradient Descent, it sometimes happens that the training cost goes *up* after performing an update iteration.

Part (a) [8 MARKS]

What are two different issues that can cause this to happen? (Assume that there are no outright bugs in the code). For each issue, state how it could be addressed to make it so that the training cost wouln't often go up. Detailed explanations are not required.

Use this page for rough work—clearly indicate any section(s) to be marked.

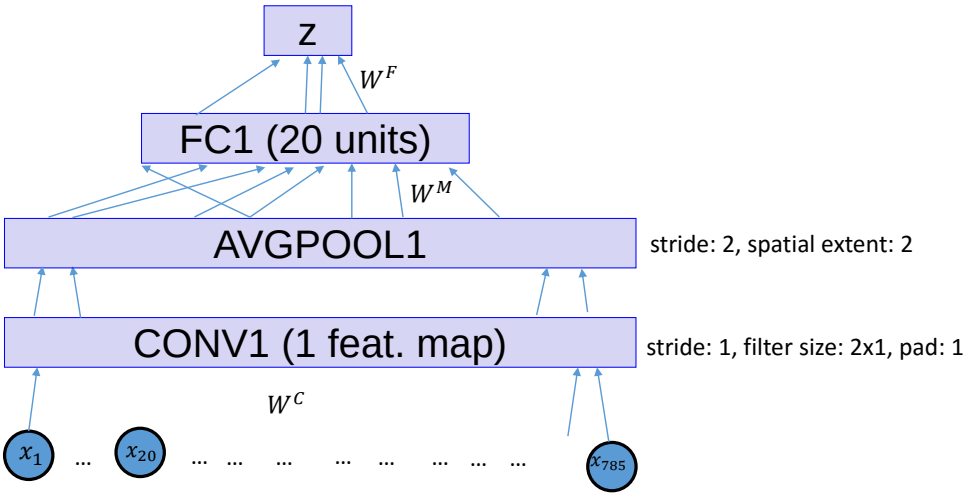
Part (b) [7 MARKS]

Provide a numerical example that illustrates one of the issues from Part (a). You should provide the model, the training set, the values of the parameters before the update, the cost before the update, and show that the update increases the training cost.

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 3. [20 MARKS]

Consider the Convolutional Neural Network below.



The network takes in input of dimension 785×1, and its output is of dimension 1×1. The network consists of an input layer X (with a 0-pad of witdth 1), a convolutional layer CONV1 which consists of one feature map with a 2×1 filter which uses the *ReLU* nonlinearity, an average-pooling layer AVGPOOL1, a fully-connected layer FC1 which uses a *ReLU* nonlinearity, and an output layer Z of size 1×1, which is fully connected to the FC1 layer and uses a *sigmoid* nonlinearity. Recall that $\sigma'(t) = \sigma(t)(1 - \sigma(t))$.

Denote the weight that connects the i-th unit in FC1 to Z by W_i^F and the bias for Z by b^F. Denote the weight that connects the j-th unit in AVGPOOL1 to the i-th unit in FC1 by W_{ji}^A and the bias of the i-th unit in FC1 by b_i^M. Let $W^C = [W_1^C, W_2^C]$ and the bias for the CONV1 layer be b^C.

Part (a) [4 MARKS]

How many parameters are there in this network? Briefly show your work.

Use this page for rough work—clearly indicate any section(s) to be marked.

Let the inputs in the training set be $X = [X^{(1)}, X^{(2)}, ..., X^{(N)}]$ and the expected outputs be $Y = [Y^{(1)}, Y^{(2)}, ..., Y^{(N)}]$.

Let the outputs of the layers in the network be denoted using $c(X^{(i)})$, $a(X^{(i)})$, $f(X^{(i)})$, and $z(X^{(i)})$ for the CONV1, AVGPOOL1, FC1, and Z layers, respectively (you may use notation such as z_i, f_j, etc.). You may use those without explicitly telling us how to compute them.

The cost function is

$$cost(X, Y) = \sum_n cost(X^{(n)}, Y^{(n)}) = \sum_n (-Y^{(n)} \log(z(X^{(n)})) - (1 - Y^{(n)}) \log(1 - z(X^{(n)})).$$

Part (b) [8 MARKS]

Compute $\partial Cost / \partial W^A_{ji}$, for a single training case. Show the details of the computation.

Use this page for rough work—clearly indicate any section(s) to be marked.

Part (c) [8 MARKS]

Compute $\partial Cost/\partial W_1^C$, for a single training case. Show the details of the computation. Note: the padding is significant.

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 4. [10 MARKS]

A ReLU neuron is considered "dead" if it doesn't activate across the entire dataset, for a given set of weights and biases. Under what circumstances might a "dead" neuron become active again during training, and under what circumstances will a neuron definitely stay "dead?" Assume no regularization is used during learning.

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 5. [5 MARKS]

Suppose we have a ConvNet trained on ImageNet, and we would like to understand/visualize the role of a particular neuron. Describe a way of doing that. (Assume the neuron is e.g. in the 4-th layer, in a 20-layer network.)

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 6. [10 MARKS]

Suppose that the outputs (before they've been passed through a softmax) of a character-level RNN at all times are stored in the array `y` (so that the t-th output is `y[:,t]`), and suppose that the one-hot encodings of all the characters in the string we are trying to model are stored in the array `x` (so that the encoding of the k-th character is `z[:,k]`). Write Python code to compute the cost function we used for training the character-level RNN. `y[:,0]` is computed using `x[:,0]` and a supplied initial hidden state.

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 7. [15 MARKS]

Part (a) [5 MARKS]

Write down the formula for the energy function (E) of a Restricted Boltzmann Machine (RBM).

Part (b) [5 MARKS]

Define the probability of the sample x in terms of the energy function of an RBM.

Use this page for rough work—clearly indicate any section(s) to be marked.

Part (c) [5 MARKS]

Explain how the weights of an RBM can be interpreted as (partial) "templates" for the training set samples. Make sure to explain the role that the hidden units play in the interpretation.

Use this page for rough work—clearly indicate any section(s) to be marked.

Question 8. [10 MARKS]

Part (a) [5 MARKS]

Sketch the architecture of an autoencoder network.

Part (b) [5 MARKS]

Describe how to train an autoencoder network. (You do not need to show how to compute any derivatives.)

Use this page for rough work—clearly indicate any section(s) to be marked.

This page was intentionally left blank

Use this page for rough work—clearly indicate any section(s) to be marked.

This page was intentionally left blank

PLEASE WRITE NOTHING ON THIS PAGE