| Duration:     | 120 minutes                 |
|---------------|-----------------------------|
| Aids Allowed: | Non-programmable calculator |

| Student Number:  |                   |
|------------------|-------------------|
| Family Name(s):  |                   |
| Given Name(s):   |                   |
| Lecture Section: | Afternoon Section |
|                  | Evening Section   |

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

This term test consists of 6 questions on 22 pages (including this one), printed on both sides of the paper. When you receive the signal to start, please make sure that your copy of the test is complete, fill in the identification section above, and write your name on the back of the last page.

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

You will not lose marks for minor syntax errors. If you cannot complete a helper function that is required to solve a problem, you should clearly indicate what the function that you cannot complete is supposed to do. You will receive part marks as appropriate for knowing what kind of helper function is needed and for using it correctly.

Write up your solutions carefully! Comments and docstrings are *not* required to receive full marks. However, they may help us mark your answers, and part marks *will* be given for showing that you know the general structure of an answer, even if your solution is incomplete.

#### Marking Guide



## Question 1. [15 MARKS]

Part (a) [5 MARKS]

Give an example of a dataset to which you could apply linear regression. State what the goal of applying linear regression to that dataset is, and make up a few numerical values for datapoints to make it clear what the data could look like.

### **Part (b)** [10 MARKS]

Describe how you would obtain the parameters of the linear regression and how you would use those parameters to obtain predictions for new data. You **do** need to give details such as the cost function, but you **do not** have to give the details of Gradient Descent.

## Question 2. [20 MARKS]

Part (a) [10 MARKS]

Draw the typical learning curves for the training and the validation set when training a neural network using stochastic gradient descent. Plot the cost on the y-axis and the iteration number on the x-axis. List the properties of the learning curves that you were trying to demonstrate in your sketch. Make sure to demonstrate all the properties that you would typically observe.

### Part (b) [10 MARKS]

Draw the graphs of the performance when applying k-Nearest Neighbours on the training and validation sets (on the y-axis) versus k (on the x-axis) that you would typically observe. List the properties of the graphs that you were trying to demonstrate in your sketch. Make sure to demonstrate all the properties that you would typically observe.

## Question 3. [20 MARKS]

You are given a Python function def f(X) which you can call, but whose source code you cannot look at. The function takes in a 42-dimensional NumPy array and returns a float. Write Python code that finds and prints a local **maximum** of the function f using Gradient Descent.

## Question 4. [20 Marks]

Consider the Convolutional Neural Network below.



The network takes in an input of dimension  $119 \times 1$ , and its output is of dimension  $1 \times 1$ . The network consists of the input layer X (with a 0-pad of witdth 1), a convolutional layer CONV1 which consists of one feature map with a  $2 \times 1$  filter which uses the *ReLU* nonlinearity, a median-pooling layer MEDIANPOOL1, a fully-connected layer FC1 which uses a *ReLU* nonlinearity, and an output layer Z of size  $1 \times 1$ , 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 MEDIANPOOL1 to the i-th unit in FC1 by  $W_{ji}^M$  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  $b^C$ .

A unit in a median-pooling layer outputs the median value of the neurons in its receptive field (i.e., the neurons connected to the unit).

### Part (a) [4 MARKS]

How many parameters (i.e., values that specify how the network computes its output) are there in this network? Briefly show your work.

#### Winter 2017

#### Midterm Test

Let the training set inputs 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  $c(X^{(i)})$ ,  $m(X^{(i)})$ ,  $f(X^{(i)})$ , and  $z(X^{(i)})$  for the CONV1, MEDIANPOOL1, 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  $\partial Cost/\partial W_{ji}^M$ , for a single training case  $(x, y) = (X^{(1)}, Y^{(1)})$ . Show the details of the computation. Use Backpropagation to obtain the final answer: show how you would compute the gradients layer-by-layer.

### Part (c) [8 MARKS]

Compute  $\partial Cost/\partial W_1^C$ , for a single training case  $(x, y) = (X^{(1)}, Y^{(1)})$ . Show the details of the computation. Note: the padding is significant. Use Backpropagation to obtain the final answer: show how you would compute the gradients layer-by-layer.

#### $CSC\,411\,H1$

### Question 5. [15 MARKS]

A coin is flipped 100 times, and the results (1's and 0's) are stored in the array flips. The result of a flip is 1 with probability  $\theta$ , and 0 with probability  $1 - \theta$ . We have the prior belief that  $\theta \sim N(0.5, 0.15^2)$ . Recall that the Gaussian pdf is

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Part (a) [5 MARKS]

Write Python code to obtain the Maximum Likelihood estimate of  $\theta$  given the training set flips.

Part (b) [10 MARKS]

Write Python code to obtain the Maximum A-Posteriori estimate of  $\theta$  given the training set flips.

# Question 6. [10 MARKS]

For what kind of dataset does the expected performance **on the test set** of k-Nearest Neighbours always grow as k grows? Describe a way to randomly generate that kind of dataset (use pseudocode or Python). The dataset should have two possible labels (i.e., y = 0 or y = 1).

Additional page for answers

On this page, please write nothing except your name.

Family Name(s):

Given Name(s):