Duration: **110 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 5 questions on 18 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

1: ______/ 5

2: ______/ 5

# 3: ______/10

4: ______/ 5

5: ______/ 3

TOTAL: ______/28

Good Luck!

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

CONT'D...

Question 1. [5 MARKS]

The M rows of data_x, an $M \times N$ NumPy array of type int32, contain unnormalized images of faces. The face labels are stored as strings in the $M \times 1$ array data_y.

For an unnormalized image x, stored as an $1 \times N$ NumPy array, write Python code to find and print its label using 1-Nearest Neighbours, with data_x as the training set. Use any reasonable normalization scheme.

Marking scheme:

- Distance function (any reasonable function is OK): 2 pts
- Argmin (i.e., finding the label that correspond to the minimum of the distance function): 2 pts
- General idea is right: 1 pts

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

CONT'D...

Question 2. [5 marks]

Suppose you have a training set with N data points in it, and you are using k-Nearest Neighbours. For which value of k are you most likely to observe overfitting? Circle one choice, and explain your answer. You do not have to use all of the space provided.

k = 1 **k ≈ N/2** **k = N** **No value of *k* is more likely to cause overfitting than another**

Marking scheme:

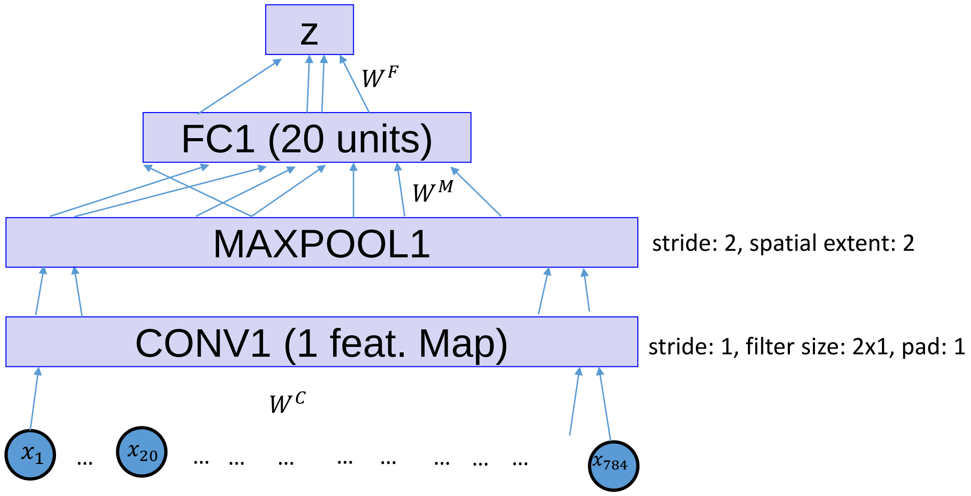
- 5 pts for the correct answer + a reasonable explanation
- 3-4 pts for correct answer + significantly flawed explanation
- 1-2 pts for incorrect answer + explanation that shows evidence of understanding the issues involved

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

CONT'D...

Question 3. [10 MARKS]

Consider the Convolutional Neural Network below.



The network takes in input of dimension 784 × 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 *tanh* nonlinearity, a max-pooling layer MAXPOOL1, a fully-connected layer FC1 which uses a *tanh* nonlinearity, and an output layer Z of size 1 × 1, which is fully connected to the FC1 layer and uses a *tanh* nonlinearity. Recall that $tanh'(t) = 1 - tanh^2(t)$.

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 MAXPOOL1 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.

(Note: it would be better if the last x coordinate were x_{785}. This did not affect marking.)

Part (a) [1 MARK]

How many units are there in the CONV1 layer? Provide the details of the computation, but there is no need for an English explanation.

Marking scheme: 1 pt for 784 or 785. 0.5 pts for an answer that evinces some understanding of what's going on.

Part (b) [1 MARK]

How many units are there in the MAXPOOL1 layer? Provide the details of the computation, but there is no need for an English explanation.

Marking scheme: 1 pt for 374 (i.e. basically $PartA/2$. 0.5 pts for an answer that evinces understanding of what's going on. (0.5 pts was pretty rare.)

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 $c(X^{(i)})$, $m(X^{(i)})$, $f(X^{(i)})$, and $z(X^{(i)})$ for the CONV1, MAXPOOL1, 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.

Denote the correct/expected output for $X^{(n)}$ as $Y^{(n)}$. Let the cost function be

$$cost(X, Y) = \sum_n cost(X^{(n)}, Y^{(n)}) = \sum_n (z(X^{(n)}) - Y^{(n)})^2$$

.

Part (c) [5 MARKS]

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

Marking scheme:

- Chain rule, applied to layers: 1 pt
- Chain rule, applied to activataions: 1 pt
- Cost function derivative: 1 pt
- Combining the above: 1 pt
- Getting things right overall: 1 pt

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

Part (d) [3 MARKS]

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

Marking scheme:

- $\frac{\partial Cost}{\partial c_i}$: 1 pt
- $\frac{\partial Cost}{\partial W_1^C}$ using the above: 1 pt
- Getting things right overall: 1 pt

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

Question 4. [5 MARKS]

Write Python code that finds a local **maximum** of $f(x, y) = \exp(-x^2 - y^2) - x^2 - y^2 - x + 2y$.

Marking scheme:

- Overall idea: 3 pts
- Computing partials: 1 pt
- Combining the partial into a gradient vector somehow (can be done implicitly): 1 pt

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

Question 5. [3 MARKS]

A 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?" Provide as complete an answer as possible.
Marking scheme:

- 0.5 pts for something reasonable that doesn't really get at the heart of the issue
- 1.5 pts per good answer (Note: there were two questions.)

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

cont'd. . .

Additional page for answers

On this page, please write nothing except your name.

Family Name(s): ______________________________

Given Name(s): ______________________________