UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

MIE 324 — Introduction to Machine Intelligence
Midterm Examination
November 23, 2018
10:15 a.m. – 11:45 p.m.
(90 minutes)

Examiner: J. Rose

This is a "closed book" examination; no aids are permitted.

No calculators or other electronic devices are allowed.

All questions are to be answered on the examination paper. If the space provided for a question is insufficient, you may use the last page to complete your answer. If you use the last page, please direct the marker to that page and indicate clearly on that page which question(s) you are answering there. It is acceptable to use a pencil.

The grades associated with each question are given in square brackets next to each question number, and for portions of questions.

The examination has 20 pages, including this one.

First Name: _____ Last Name: _____

Student Number: _____

MARKS

1	2	3	4	5	6	7	8	9	Total
/8	/4	/6	/6	/10	/10	/12	/5	/10	/71

This course is about the understanding and use of neural networks as one example of an approach to machine learning. The first three questions are broad questions that apply to the general structure and use of neural networks.

Question 1 [7 Marks]

(a) [2] What is the difference between 'running' a neural network in *training* mode versus *inference* mode. Explain, in 1 or 2 sentences.

Please write your answer here:

(b) [1] In general, in neural network training, and in all the assignments in this course, you used two or three *groupings* of the dataset. What are the names of those three groupings of data?

Please write your answer here:

Question 1, continued

(c) [1] For each of the three groupings listed above in part (b), in which mode of the neural network (training or inference) is the grouping used?

Please write your answer here:

(d) [3] What is the specific purpose of each of the three groupings of the dataset from part (b)?

Please write your answer here:

Question 2 [4 Marks]

a) [2] What does it mean when a Neural Network has been **over-fit**? Explain.

Please write your answer here:

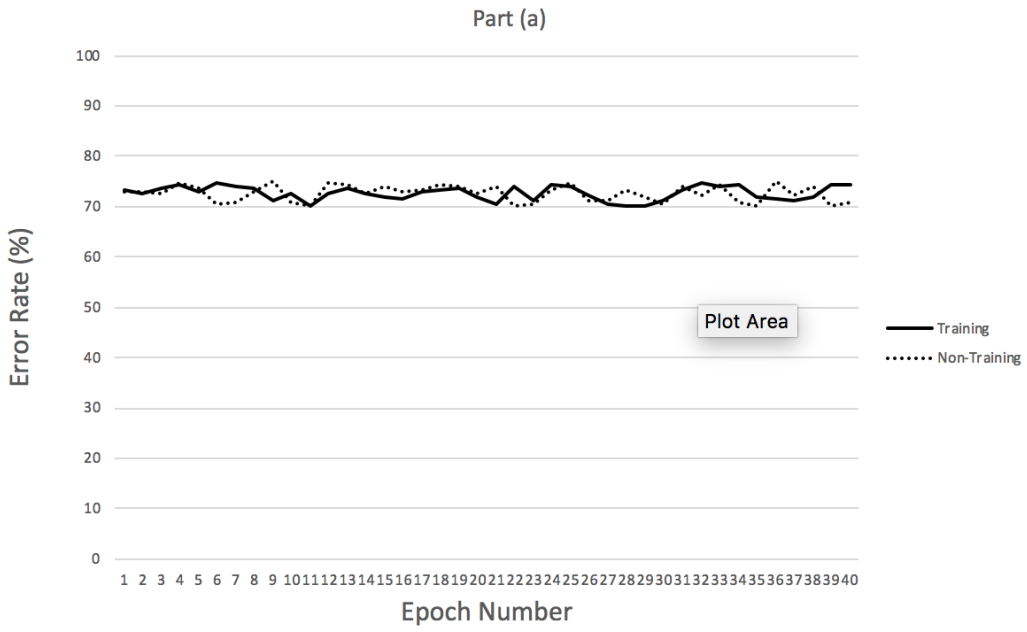
b) [2] List two possible negative (bad) effects of having a learning rate that is too small, when training a neural network.

Please write your answer here:

Question 3 [6 Marks]

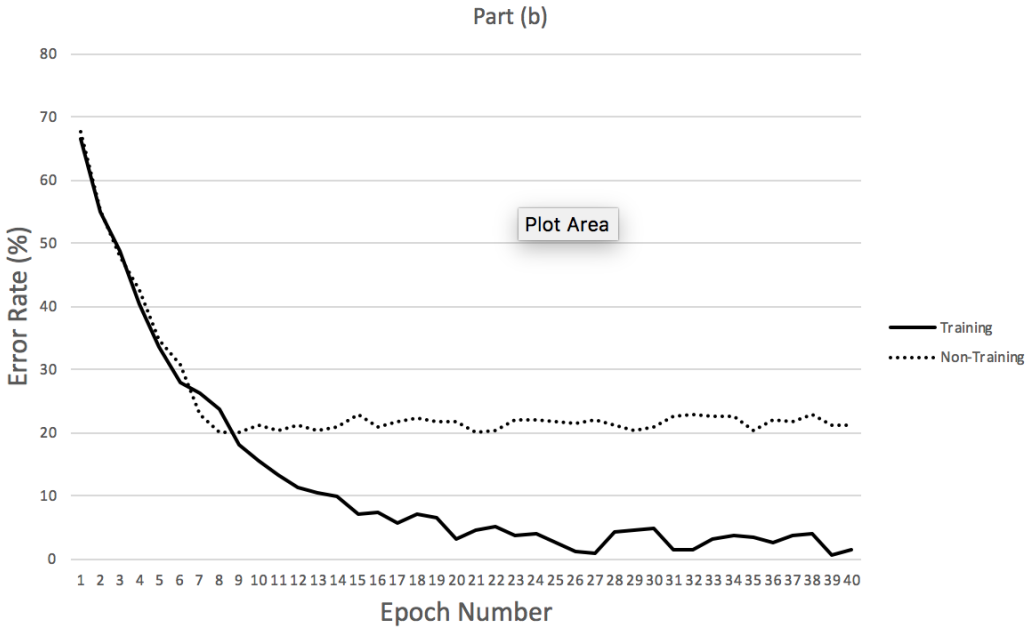
Consider the following learning curves, which plot the error rate of a neural network during training, versus Epoch number, for both training data and non-training data. This specific network's goal is to **classify its input into one of four classes**. In each case, give the name associated with the plot (that describes how well the network is learning), and explain why.

3(a) [2] Consider this plot:



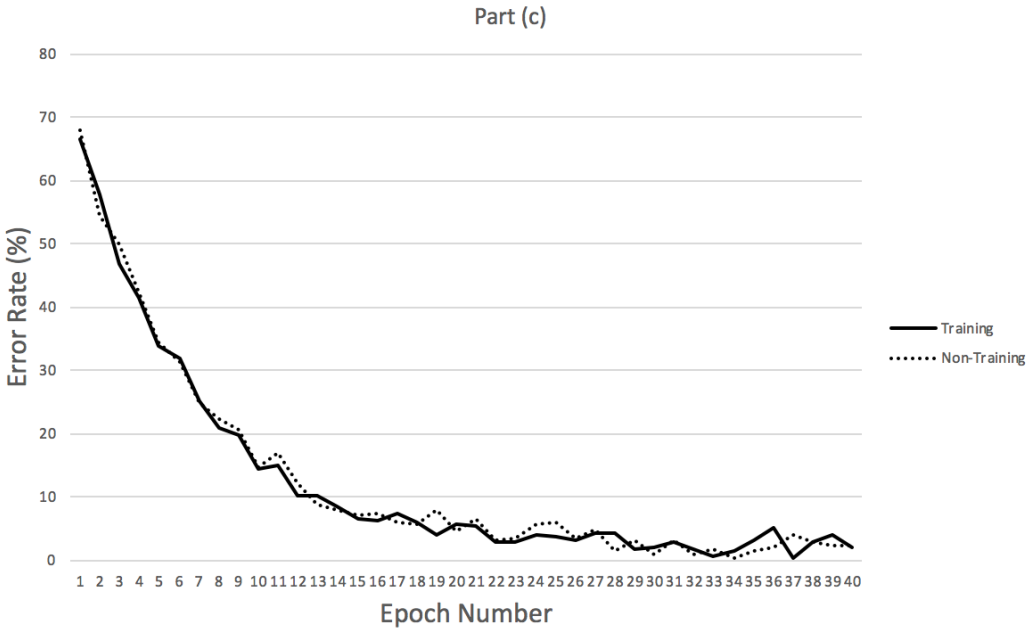
Please write your answer here:

3(b) [2] Consider this plot:



Please write your answer here:

3(c) [2] Consider this plot:



Please write your answer here:

Question 4 [6 Marks]

Consider the problem of image classification on a 100x100 pixel 3-channel (colour) image, similar to what was done in Assignment 1, Cats vs. Dogs. It is possible to do image classification using different kinds of neural networks; in Assignment 1 you used a Convolutional Neural Network (which had two convolutional layers followed by three fully-connected layers, together with various pooling layers and activation functions). It could also be done with a series of (say 4) fully-connected linear layers as well, with appropriate activation functions.

(a) [2] List two advantages that a CNN would have over a fully-connected linear version.

Please write your answer here:

(b) [1] List one advantage the fully-connected linear network might have over the CNN.

Please write your answer here:

Question 4, continued

(c) [3] Consider just the *first* layer of the two networks. Compute the number of parameters in the first layer for each network, which have the following specifications:

(1) The first layer of the CNN uses 10 kernels of size 4x4.

(2) The first layer of the fully-connected linear network has 20 neurons (hidden units).

(3) Assume that all neurons and kernels also use a bias parameter.

Please put your rough work directly below, and put your answers in the appropriate line down below.

Number of Parameters in first layer of CNN:

Number of Parameters in first layer of Linear Network:

Question 5 [10 Marks]

In this question you are to write the Pytorch code for the class definition of a convolutional neural network (CNN) model, and the code to instantiate an object in that class. The input and CNN have the following specifications:

(a) The input is a 100x100 pixel *grayscale* image - i.e. it has only one 'colour.'

(b) The CNN will be trained classify the image into five different classes. The outputs should be in the form of a probability that the input image is one of the five classes.

(c) The network should use two 2D convolutional layers (with no padding) each of which are followed by a ReLU activation and a 2x2 max pool layer, with a stride of 2. (Note that, to be helpful, the first line of the PyTorch documentation of class definitions for nn.Conv2d, nn.MaxPool2d and nn.Linear are given below). There should be n1 kernels on the first layer and n2 kernels on the second layer, where n1 and n2 are variables available to the program and should be given as arguments to the model object initialization/instantiation method. Similarly, the size of all the kernels on the first layer should be k1 x k1, where k1 is also a given variable. The size of the kernels on the second layer are similarly k2 x k2.

(d) The second convolutional layer should be followed by two fully-connected linear layers. The first layer should have a parameterized number of neurons (hidden units), L, which are also passed to the object instantiation method. The first linear layer should use a ReLU activation function. The second linear layer should be the output layer, as described above.

(e) After defining the class, also provide the code to instantiate the model. You can assume that the five parameters - k1, k2, n1, n2 and L are available as variables and don't need to be set to specific values.

Here are first line of the definitions of the three relevant PyTorch methods from the PyTorch documentation:

```
class torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1,
                      padding=0, dilation=1, groups=1, bias=True)

class torch.nn.MaxPool2d(kernel_size, stride=None, padding=0, dilation=1,
                         return_indices=False, ceil_mode=False)

class torch.nn.Linear(in_features, out_features, bias=True)
```

If you need to make any further assumptions to write your code, state them clearly. Finally, you should assume that your code comes after the following statements:

```
import torch.nn as nn
import torch.nn.functional as F
```

Please write your answer on the next page:

Place answer to the Question 5 here:

Question 6 [10 Marks]

In Assignment 2 you learned about methods of coding the inputs that are presented to a neural network. In particular, you learned about the 1-hot method of coding categorical inputs as distinct from a fully-encoded input. For example, if you had three categories they could be presented as a 1-hot codes `100`, `010` and `001` to a neural network, with three separate inputs. If, instead, these three possible inputs were presented as fully encoded, they could appear as a single input, with the codes `1`, `2` and `3` respectively.

(a) [1] When an input to a neural network is categorical, why is it better to code it as a 1-hot rather than fully-encoded?

Please write your answer here:

(b) [9] You are to determine the full design of neural network that converts fully-encoded inputs into a 1-hot encoded form, for the specific example codes given above. That is, the network has one input, X, which will only ever take on the value of one of the three codes – `1`, `2` and `3` – and produces three outputs, H_1, H_2 and H_3, where $H_i = 1$ if $X = i$ and $H_i = 0$ otherwise. (Thus the output $H_1H_2H_3$ represents the 1-hot encoding of the input.)

The network should have 3 hidden layers, and on each layer, if an activation function is needed, it should use the ReLU function. The design of the network should include all of the weights and biases of all neurons used. You should draw a picture of the network, which clearly shows its parameters, and include enough information that shows exactly what the computation being performed.

Please place your answer on the next page.

Please place the Answer to Question 6(b) here:

Question 7 [12 Marks]

Consider the following PyTorch training loop code, which is taken directly from Assignment #1, Cats vs. Dogs.

```
 . . .
    # Instantiate  Network, defined in model.py
    net = Net()

    criterion = nn.BCEWithLogitsLoss()
    optimizer = optim.SGD(net.parameters(), lr=learning_rate, momentum=0.9)

    train_err = np.zeros(num_epochs)
    train_loss = np.zeros(num_epochs)
    val_err = np.zeros(num_epochs)
    val_loss = np.zeros(num_epochs)

    for epoch in range(num_epochs):
        total_train_loss = 0.0
        total_train_err = 0.0
        total_epoch = 0

        for i, data in enumerate(train_loader, 0):
            inputs, labels = data
            labels = normalize_label(labels)

            optimizer.zero_grad()

L1          outputs = net(inputs)
L2          loss = criterion(outputs, labels.float())
L3          loss.backward()
L4          optimizer.step()

L5          corr = (outputs > 0.0).squeeze().long() != labels
            total_train_err += int(corr.sum())
            total_train_loss += loss.item()
            total_epoch += len(labels)

            train_err[epoch] = float(total_train_err) / total_epoch
            train_loss[epoch] = float(total_train_loss) / (i+1)

L6      val_err[epoch], val_loss[epoch] = evaluate(net, val_loader, criterion)
```

Question continues on next page

In the following table, describe what is happening in each of the lines labelled **L1, L2, L3, L4, L5 and L6** on the left hand side in the above code.

Line	Description of Function of Code in the Line
L1	
L2	
L3	
L4	
L5	
L6	

Question 8 [5 Marks]

Recall the concept of Word Vectors (also called Word embeddings) that were used in Assignment 4. These are generated by through the training of a neural network on a corpus of many sentences.

a) [1] The training of a neural network to produce word vectors relies on the *distributional hypothesis*. What is the distributional hypothesis?

Answer:

b) [4] Consider a neural network that is going to be used to create a word vector of size `10` as described in class. The network is being trained to predict the word that will come after a given (single) input word, on a vocabulary of size 2000 words. The network has two layers of neurons – a hidden layer and an output layer. Determine the number of parameters in the hidden layer, and the number of parameters in the output layer. Assume that neither layer uses a bias. Explain your answer.

Answer:

Question 9 [10 Marks]

Imagine that you are a full-fledged machine learning engineer, working for HealthyBrain (HB) Corporation, which seeks to develop tools to monitor and improve mental health. Scientists at HB have determined that anxiety has been increasing in young people using social media, leading to a direct worsening of mental health. Your boss comes to you one day and tells you to design, build and train a neural network that detects anxiety in speech from sound recordings of a person talking. The plan is to sell this product to companies that develop video/audio communication tools like those found in Messenger from Facebook, WeChat from Tencent or FaceTime from Apple. This would enable those companies to measure anxiety levels in their customers, and perhaps do something about it, in some kind of intervention.

HB Corp is worried about the competition and is rushing to get this product ready. Other HB workers have collected the dataset you are to use which consists of 1 minute instances of speech from 100 of its employees. Those instances of speech each consist of time-series data samples (similar, in a way, to the instances of gestures in Assignment 3). Your boss says that you can only use these 100 instances, which come from the following demographics of the HB employees:

Demographic		Number of Instances
Gender	Female	81
	Male	19
Age Range	20-30	10
	31-40	60
	41-50	30

(a) [2] **Aside** from its small size of the dataset, list two problems with this dataset, and give the reason why each is a problem.

Please write your answer here:

Question 9 continues.

(b) [3] In developing the neural network, you come to realize that there just isn't enough data (particularly given the problems you identified in part (a)) to properly train a neural network from scratch. So, you decide to employ **data augmentation** techniques to address this situation. List three kinds of transformations (in English, using mathematical notation only if you feel it is necessary) on the instances you're given that could improve the amount and quality of the data. For each transformation you list, describe why it makes the dataset better.

Answer:

(c) [3] Suppose you decide that training a new neural network from scratch just isn't going to work, because of the insufficient data. Suggest an alternative approach, one that was described in class, that is a different way to deal with insufficient data, and briefly describe how that approach works in this context. **Do not use data augmentation** as part of this answer as that was already covered in part (b) of this question.

Answer:

Question 9 continues.

(d) [2] Consider the product being designed, and its final deployment at the potential customers. List and explain two ethical issues that might arise.

Answer:

This page has been left blank intentionally. You may use it for answers to any question in this examination.