CS5242 Assignment 1

August 21st, 2018

ASSIGNMENT DEADLINE l – 4TH SEPTEMBER 2018 11.59 P.M.

Grades will be given based on:

1. Explanation of results
2. Spotting and explaining of any anomalies in your results

Submit to: IVLE/workbin/assignment 1 submission

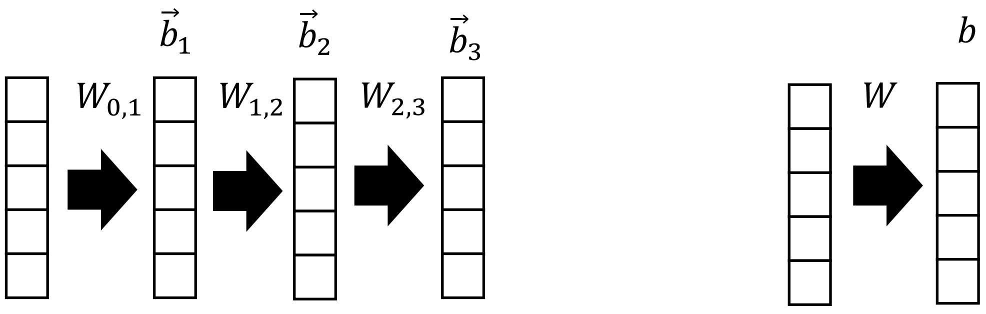
Question 1

Two networks are said to be equivalent if, they have the same number of input and output nodes, for all inputs, the outputs of both networks are identical.

Given two neural networks as shown below, all activation functions are identity functions,

$$z = \sigma(z)$$

If the weights and biases are given in the first network with two hidden layers, write a program to find the weights and biases of the second network with no hidden layers to make the two networks equivalent.



Given files: Question_1 folder contains five folders a, b, c, d, and e. Each contains example weights and biases for the first network. To ease your understanding, in each csv, there's one heading column introducing the weights/biases.

Submission: In a folder named "Question_1" save your results for W and b as csv files. Please prefix each file name with the folder name. For example, results for the inputs in folder "a" should be named as "a-w.csv" and "a-b.csv".

Your result files should **NOT** contain a heading column.

Each file should be formatted to the format explained in **Appendix I.**

Round your results to at least **16** digits.

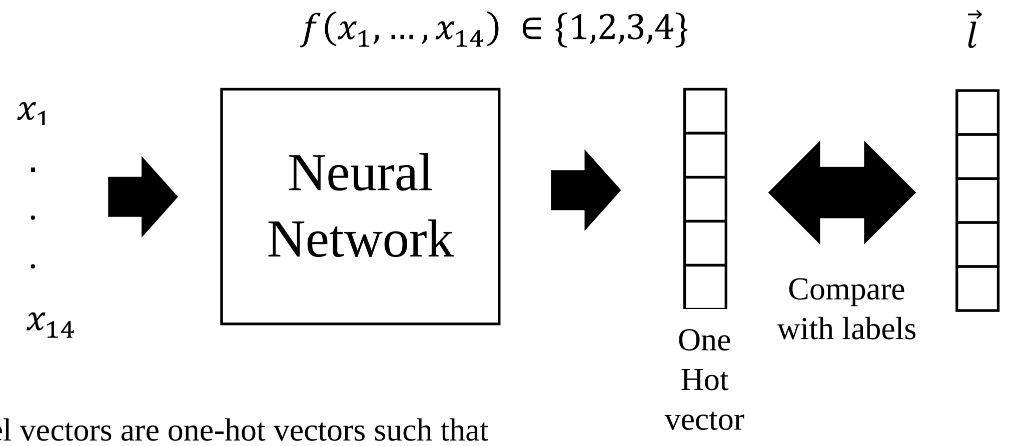
Your submissions will be automatically graded using a script. Be sure to format your output according to instructions. Incorrect formats will be graded as incorrect answers.

Question 2

Given a data set that represents a function that takes in 14 binary digits and output one of four numbers, 0,1,2,3. Construct a neural network to train for this function.

$$f: \{-1,1\}^{14} \rightarrow \{1,2,3,4\}$$

$$f(x_1, \dots, x_{14}) \in \{1,2,3,4\}$$

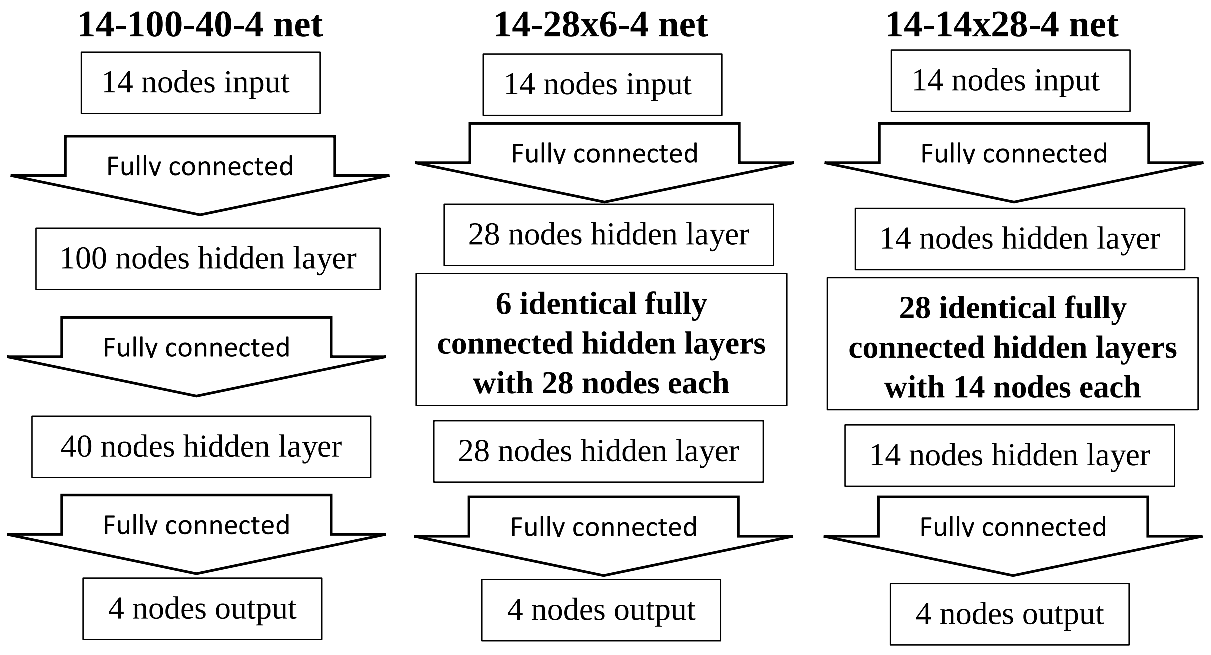


The label vectors are one-hot vectors such that

$$\vec{l} = \begin{cases} (1,0,0,0) & \text{if } f(x_1, \cdots x_{14}) = 0 \\ (0,1,0,0) & \text{if } f(x_1, \cdots x_{14}) = 1 \\ (0,0,1,0) & \text{if } f(x_1, \cdots x_{14}) = 2 \\ (0,0,0,1) & \text{if } f(x_1, \cdots x_{14}) = 3 \end{cases}$$

For this assignment, construct three networks given bellow. Implement the back-propagation algorithm and use gradient descent to train the networks. Use **cross entropy cost function** on a softmax function for training.

All but the last fully connected layers have ReLU as the activation function.



1.
    a. For each network
        i. Plot the training cost w.r.t. iterations
        ii. Plot the testing cost w.r.t. iterations
        iii. Plot the train & test accuracy scores w.r.t. iterations
    b. Give a half page discussion on why the three networks 14-100-40-4 net,14-28x6-4 net,14-14x28-4 net perform differently. Which one performs better and why.
2. Check your back propagation intermediate results against known answers:
    a. You will be given one special data point d.
    b. You will be given one weight & bias set, W_0 and b_0 together with correct gradients computed using data point d and W_0 and b_0.

$$\frac{\partial ouput}{\partial W_0} \qquad \frac{\partial ouput}{\partial b_0}$$

Note: Output means the loss function

    c. You will be given another weight & bias set W_1 and b_1 with no corresponding gradients given.
    d. Use (a) and (b) to compute gradients and compare to the given correct gradients.
    e. Use (a) and (c) to compute gradients and submit your gradient values for grading.

Given files:

For part 1.a,

- The data given to you is in folder Question_2_1.
- It contains 4 csv files, (i.e. x_train, x_test, y_train, y_test). In the x_* files, each line is a data point of 14 dimensions. Each y_* file contains respective labels. The n^{th} entry in the y_* file is the corresponding label of the n^{th} entry in the respective x_* file.

For part 2,

- The data given to you is in folder Question_2_2.
- You will be given weights and gradients for verification, which is under the ‘b’ folder, where the weights you will work on is in the ‘c’ folder. To ease your understanding, in the weights csv, there’s one heading column introducing the weights/biases, while there is **NO** such column for the gradients. And you should **NOT** include the headings in your submission as well.
- The given data point x and label y is in the ‘**a.txt**’ file. For both verification and test, we use the same data point and label.

Your submission should be the same format as the given ‘true-d*’ files. For more details please check **Appendix I**.

If possible, use **np.float32** to control the granularity of your gradients, otherwise, round your results to at least **16** digits.

Your submissions will be automatically graded using a script. Be sure to format your output according to given instructions. Incorrect format will be graded as incorrect answers.

Final Submission

- Your submission is a single .zip file. Other compressions are NOT acceptable.
- The naming of the .zip file is your ID as shown on the IVLE/class. We will use it for grading. For most cases, it's your NUS NETID, which will be e******.zip ('e' in lowercase).
- Inside the .zip file, there are 4 files. They are 1 pdf and three folders; Question_1 which contains results for Question 1, Question_2 which contains results for Question 2 and code which contains your code.
- For Question 1 and Question 2.2, the output files should be csv files, with comma (i.e.',') as the delimiter. Using space or other delimiters are **NOT** acceptable.
- For Question 2.2 csv naming are as follow: dw-100-40-4.csv, db-100-40-4.csv, dw-28-6-4.csv, db-28-6-4.csv, dw-14-28-4.csv, db-14-28-4.csv, which correspond to the gradients of weights and biases for the three network configurations: 14-100-40-4, 14-28-6-4, 14-14-28-4. Other naming is **NOT** acceptable.

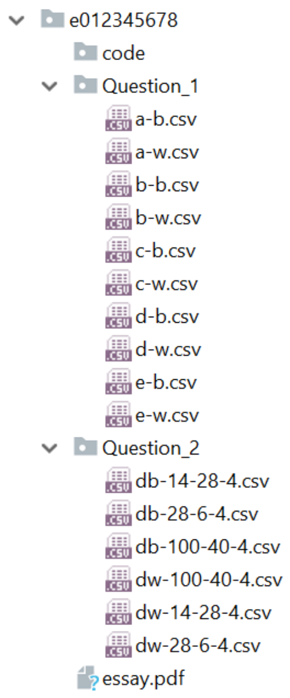

```
e012345678
    code
    Question_1
        a-b.csv
        a-w.csv
        b-b.csv
        b-w.csv
        c-b.csv
        c-w.csv
        d-b.csv
        d-w.csv
        e-b.csv
        e-w.csv
    Question_2
        db-14-28-4.csv
        db-28-6-4.csv
        db-100-40-4.csv
        dw-100-40-4.csv
        dw-14-28-4.csv
        dw-28-6-4.csv
    essay.pdf
```

Figure 1: Expected folder structure

Appendix I: More details for Question_1 and Question_2_2

Suppose that in a given neural network layer t has N_t number of nodes,

- The weights from layer t to layer $t+1$ form a N_t by N_{t+1} matrix, where the $(ij)^{th}$ entry of this matrix represents the weight, w_{ij} connecting the i^{th} node of layer t to the j^{th} node of layer$(t+1)$.
- The bias is a length-N_t vector as for layer t. Note that the input layer 0 has no corresponding bias.
- Softmax is not considered to be a layer in this context, so the output layer output logits.

The given weights files have the following formatting:

- Totally $\sum_{t=0}^{\#\,layers-1} N_t$ rows
- The first N_0 rows are the matrix from input layer 0 to hidden layer 1, the following N_1 rows are the matrix from layer 1 to layer 2, and so on. There's **NO blank line** between matrix(t) and matrix(t+1)
- Then of each matrix, the (ij) item is w_{ij} (For Question_2_2 corresponding gradients file are also given. There the (ij) item is the derivative w.r.t. w_{ij}, which is $\frac{doutput}{dw_{ij}}$).

The given bias files also have the same formatting:

- Totally $\sum_{t=0}^{\#\,layers-1} 1$ rows
- The first row is the bias vector for layer 1 and has N_1 items. The second line is the bias vector for layer 2 with N_2 items, and so on.
- Of each row, the i item is either b_i. (For Question_2_2 corresponding gradients file are also given. There the (ij) item is the derivative w.r.t. b_i $\frac{doutput}{db_i}$).

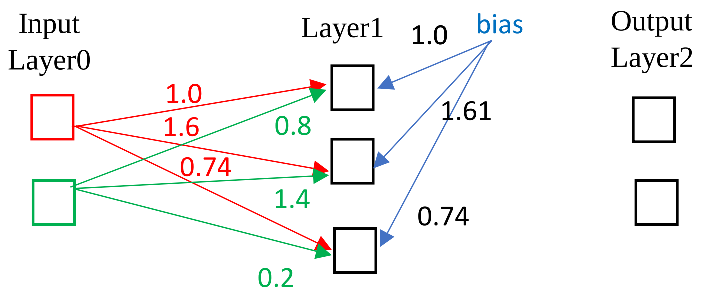


Figure 2: An example fully connected network

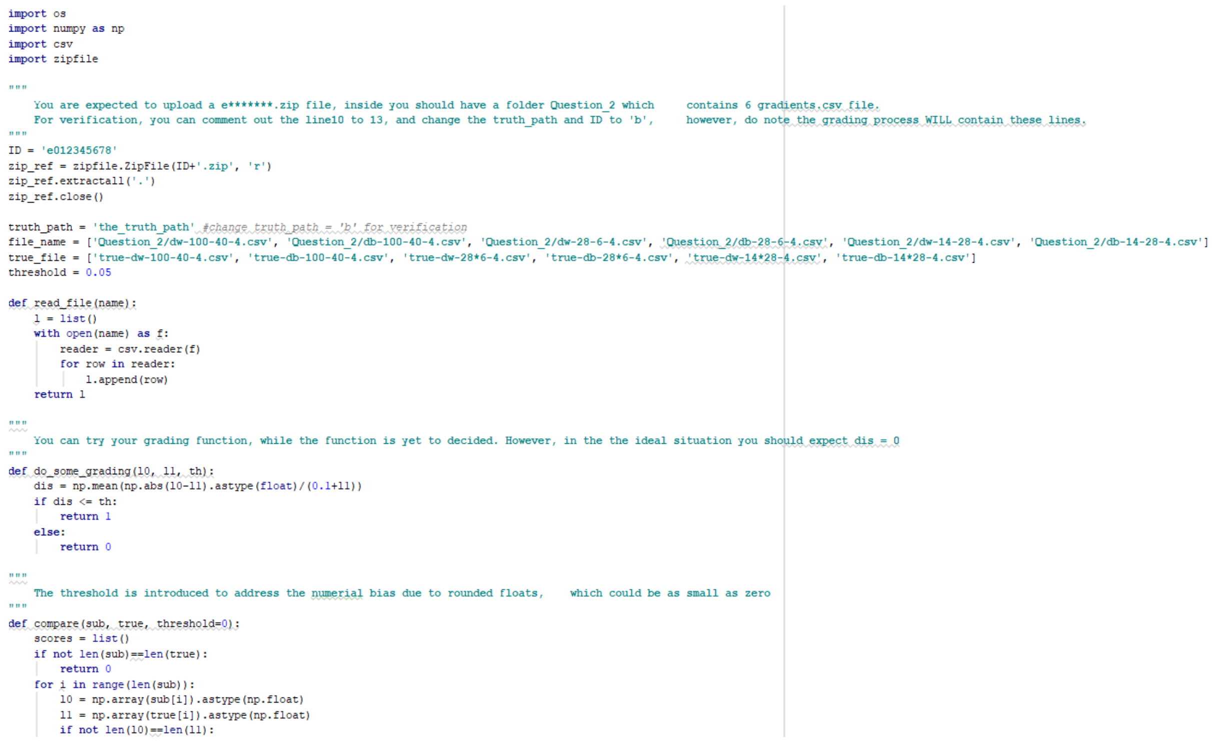
For example, the weights and biases file for the network in Figure.2 would be as follows:

Weights	Biases
1.0, 1.6, 0.74	1.0, 1.61, 0.74
0.8, 1.4, 0.2	0.01, −0.99
0.01, -1.87	
0.0.1, -1.23	
0.0, -0.34	

The csv file you would submit should have the same format as above.

Appendix II: Example Evaluation Script

- Your result will be graded using a script like the one bellow. If you run it for verification for ‘b’, you would get an output 781.
- This script will be given in Question_2_2 folder.
- Do try running your code with this script to adjust your formatting.
- Otherwise you are likely to receive no points.



```
import os
import numpy as np
import csv
import zipfile

"""
    You are expected to upload a e*******.zip file, inside you should have a folder Question_2 which     contains 6 gradients.csv file.
    For verification, you can comment out the line10 to 13, and change the truth_path and ID to 'b',     however, do note the grading process WILL contain these lines.
"""
ID = 'e012345678'
zip_ref = zipfile.ZipFile(ID+'.zip', 'r')
zip_ref.extractall('.')
zip_ref.close()

truth_path = 'the_truth_path' #change truth_path = 'b' for verification
file_name = ['Question_2/dw-100-40-4.csv', 'Question_2/db-100-40-4.csv', 'Question_2/dw-28-6-4.csv', 'Question_2/db-28-6-4.csv', 'Question_2/dw-14-28-4.csv', 'Question_2/db-14-28-4.csv']
true_file = ['true-dw-100-40-4.csv', 'true-db-100-40-4.csv', 'true-dw-28*6-4.csv', 'true-db-28*6-4.csv', 'true-dw-14*28-4.csv', 'true-db-14*28-4.csv']
threshold = 0.05

def read_file(name):
    l = list()
    with open(name) as f:
        reader = csv.reader(f)
        for row in reader:
            l.append(row)
    return l

"""
    You can try your grading function, while the function is yet to decided. However, in the the ideal situation you should expect dis = 0
"""
def do_some_grading(l0, l1, th):
    dis = np.mean(np.abs(l0-l1).astype(float)/(0.1+l1))
    if dis <= th:
        return 1
    else:
        return 0

"""
    The threshold is introduced to address the numerial bias due to rounded floats,    which could be as small as zero
"""
def compare(sub, true, threshold=0):
    scores = list()
    if not len(sub)==len(true):
        return 0
    for i in range(len(sub)):
        l0 = np.array(sub[i]).astype(np.float)
        l1 = np.array(true[i]).astype(np.float)
        if not len(l0)==len(l1):
```