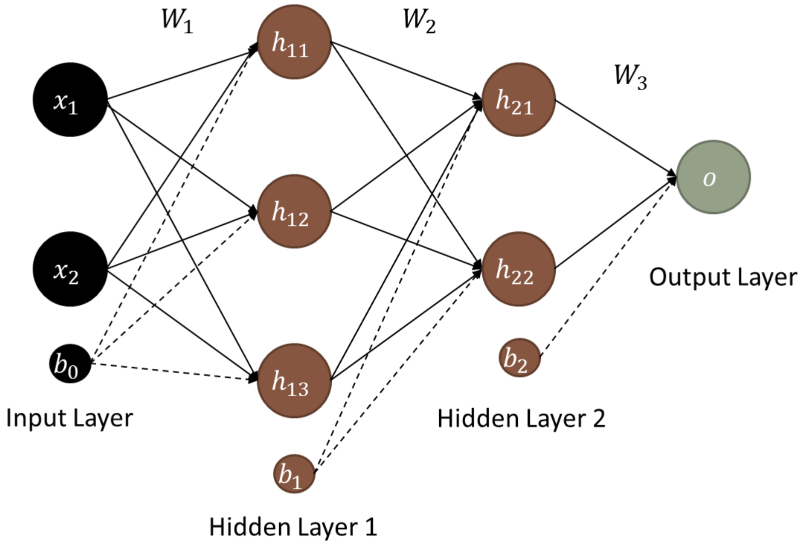
1111 Deep Learning – Homework 2

Due: 12/5, 2022, 11:59 pm

For the following questions, please upload the source code to moodle and show the results in your report.

1. (20%) Please construct a neural net (with its architecture shown below) and train it on ‘train.mat.” The activation function must be used in the two hidden layers and the output layer. You can construct and optimize your network model by using any off-the-shelf functions. Report the test error on the test set ‘test.mat’ (percentage of misclassified test samples).



2. (80%) The MNIST dataset contains handwritten digits, with a training set of 60,000 samples and a test set of 10,000 samples.
   Please download it here: http://yann.lecun.com/exdb/mnist/

   2.1 (20%) You are asked to construct a classification model based on multi-layer convolutional neural networks (at least five layers) for digit recognition. Please report the prediction accuracy for the test set. (Hint: its loss function could be cross entropy)

   2.2 (15%) Please randomly set 5%, 10%, and 15% of pixels to 255 for each image and evaluate the test set using the model you trained on **clean images** from 2.1. Report the prediction **accuracies** for the three different corruption rates. Compare your results with those in Question 2.1. What do you find?

   2.3 (15%) Following Question 2.2, please **re-train** your model with the corrupted data (5%, 10%, and 15% separately) and re-evaluate the test set.
   Report the prediction **accuracies** also for the three different corruption rates. Compare your results with those from 2.1 and 2.2. What do you find?

2.4 (15%) Following Question 2.2, please construct **ONE** restoration model (also convolutional neural networks) that inputs a corrupt image and outputs its restored image. (Hint: its loss function could be MAE or MSE)

2.5 (15%) Following Question 2.4, please evaluate the restored test set using the model you trained on **clean images** in Question 2.1. Report the prediction **accuracies** for restored images separately at the three different corruption rates. What do you find?

To corrupt an image, you could use the example code below:

```
import random
import numpy as np

corrupt_lv = 0.05    % using 5% and training data as an example
img_size = 28*28
for i in range(len(X_train)):
  ran_seq = random.sample([n for n in range(img_size)], np.int(img_size*corrupt_lv))
  x = X_train[i].reshape(-1, img_size)
  x[0, ran_seq]=255
```