Industrial Engineering (Deep Learning) 874
Post-block Assignment 1: Building Blocks of Deep Learning
Department of Industrial Engineering

Deadline: 6 July 2020, 23:59

Total: 147

Instructions

The focus of this assignment is to test your understanding of the concepts covered in the lectures from days 1-2. In addition, your implementation of these concepts on real-world data will also be tested in this assignment.

- Answer all the questions below.
- Where asked to provide all calculations and steps in the methodology you used, please make sure that you do. Without these calculations and insights to your methodology, no marks will be given.
- Submit your typed answers as a pdf document. Please also submit all other documents required to obtain your answer. For instance, submit your Python script(s) that you used for any of the questions.
- Please make sure that you do and submit your own work. Plagiarism will not be tolerated.
- Note that late submissions cannot be accepted and that no extensions to the deadline can be provided.

General Deep Learning Concepts **[24]**

1. A neural network is an approximate mathematical representation of the brain, consisting of smaller components called neurons (perceptrons). A neuron has an input, a processing/activation function, and an output. To approximate a function, these neurons are stacked together to form a network. **[2]**

Considering this description, when does a neural network model become a *deep* learning model?

A. When you add more hidden layers and increase depth of neural network

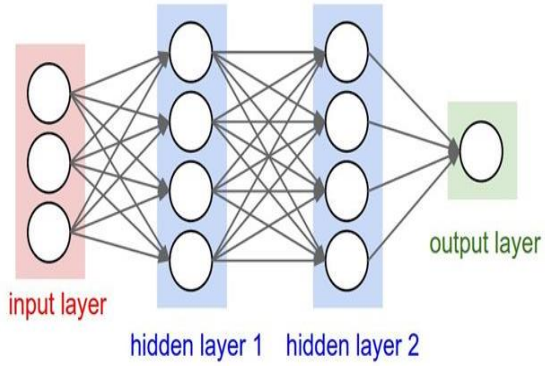
B. When there is higher dimensionality of data

C. When the problem is an image recognition problem

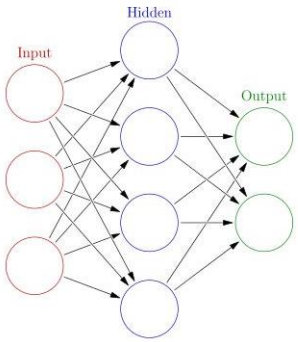
D. None of these

2. Consider a problem where we want to classify a given picture as either a cat or a dog, which of the following architecture would you choose? **[2]**

1



2



A) 1
B) 2
C) Any one of these
D) None of these

3. Consider a neural network where the number of nodes in the input layer is 10 and the hidden layer is 10. What is the maximum number of connections from the input layer to the hidden layer? **[2]**

A) 100

B) Less than 100

C) More than 100

D) It is an arbitrary value

4. Consider a simple multi-layer perceptron (MLP) model with the following architecture. There are 6 neurons in the input layer, 10 neurons in the hidden layer and a single neuron in the output layer. From the options below, what is the size of the weight matrices between the hidden layer and the output layer, as well as the input layer and the hidden layer? **[2]**

A) [1 X 10], [10 X 6]

B) [6 X 10], [1 X 10]

C) [6 X 10], [10 X 1]

D) [10 x 1], [6 X 10]

5. There are multiple steps in the training process of a neural network. What is the correct sequence of the following training tasks in a network? **[2]**

1. Initialize weights of neuron randomly
2. Go to the next batch of dataset
3. If the prediction does not closely approximate the output, change the weights
4. For a sample input, compute an output

A. 1, 2, 3, 4

B. 4, 3, 2, 1

C. 3, 1, 2, 4

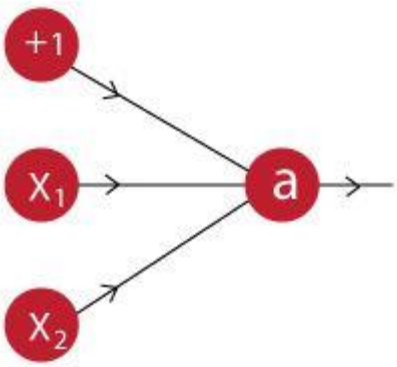
D. 1, 4, 3, 2

6. Consider a representation where we implement an AND function to a single neuron. Below is a tabular representation of an AND function: **[4]**

X1	X2	X1 AND X2
0	0	0
0	1	0
1	0	0
1	1	1

The activation function of our neuron is denoted as:

$$f(x) = \begin{cases} 0, & for\ x < 0 \\ 1, & for\ x \geq 0 \end{cases}$$



What would the weights and bias be for the neuron?

(Hint: For which values of w1, w2 and b does our neuron implement an AND function?)

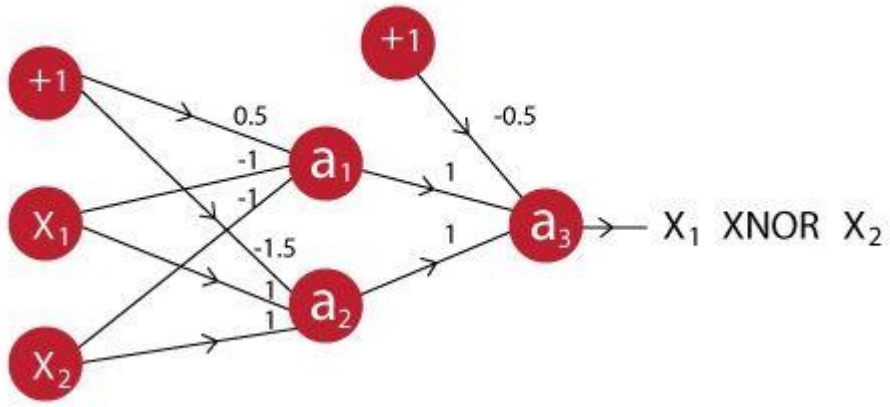
A. Bias = -1.5, w1 = 1, w2 = 1

B. Bias = 1.5, w1 = 2, w2 = 2

C. Bias = 1, w1 = 1.5, w2 = 1.5

D. None of these

7. When we stack neurons together, we form a neural network. Consider the example of a neural network simulating an XNOR function as indicated below. **[4]**



You can see that the last neuron takes input from two neurons before it. The activation function for all the neurons is given by:

$$f(x) = \begin{cases} 0, & for\ x < 0 \\ 1, & for\ x \geq 0 \end{cases}$$

Suppose X1 is 0 and X2 is 1, what will the output be for the given neural network?

A. 0

B. 1

8. Model capacity refers to the ability of a neural network to approximate complex functions. Which of the following is statements true about model capacity? **[2]**

A. As number of hidden layers increase, model capacity increases

B. As dropout ratio increases, model capacity increases

C. As learning rate increases, model capacity increases

D. None of these

9. Will the classification error of test data always decrease if you increase the number of hidden layers in a MLP model. **[2]**

A. Yes

B. No

10. You want to map every possible image of size 64 × 64 to a binary category. Suppose an image has 3 channels and each pixel in each channel can take an integer value between (and including) 0 and 255. How many bits do you need to represent this mapping? **[2]**

(i) $256^{3\,64\times64}$

(ii) $256^{3\times64\times64}$

(iii) $(64\ \times\ 64)^{256\times3}$

(iv) $(256\ \times\ 3)^{64\times64}$

Vectorization **[24]**

1. Describe what vectorization in programming is? **[3]**

2. Consider the two vectors $a\ =\ [0,1,2]$ and $b\ =\ [3,4,5]$. Using: 1) a *for loop*, and 2) vectorization, add the two vectors to form a new vector *c*. Illustrate how you use vectorization versus using the *for loop* to add the two vectors by code *and* by visual means as part of your answer. Any programming language can be used. **[6]**

3. Why is vectorization important in *deep learning*? As part of your answer, discuss how vectorization is used in the domain of *natural language processing (NLP)* within the field of deep learning. **[5]**

4. In this question, we will investigate the speed-up of vectorization versus *for-loops* when generating probabilities for a Gaussian function.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Using the above equation, create a list of *n* probabilities where $n \in [1000, 10000, 100000, 1000000, 10000000]$ and where x is sampled from a uniform distribution between 10 and 20, i.e., $x \in \mathrm{U}(10, 20)$. Any values may be used for μ and σ as long as they are consistent across the two methods (for loops versus vectorization). Plot the time it takes to create the list of *n* probabilities for the various values of *n* for the two different approaches and comment on the difference in time as the magnitude of *n* increases. **[10]**

Vanishing gradient problem and activation functions **[49]**

In this question we will explore the vanishing gradient problem encountered in a binary classification problem. Using the train and test data sets, implement the following:

1. Load the data sets into your environment (trainX, trainy, testX, testy)
2. Develop an MLP model with the following specifications: **[10]**
   - The input layer should have 2 inputs (consistent with the dimensions of the training dataset), a single hidden layer with 5 nodes, and lastly the output layer should have one node to predict the class.
   - Use a Random Uniform kernel initializer with a range of [0, 1] (hint, Keras has a built-in initializer RandomUniform(minval=0, maxval=1)).
   - Use a stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and a momentum of 0.9.
   - Use a binary cross entropy loss function.
   - Use a hyperbolic tangent activation function (tanh) in the hidden layer.
   - Use 500 training epochs.
   - Use the test data as the validation dataset so we can get an idea of how the model performs on the holdout set during training
3. Train and test the model and plot the training and testing accuracy versus the training epochs of the respective data sets.
   a. Report the training and testing accuracy and illustrate the plots. **[7]**
   b. Studying the plots and the accuracy metrics, do you think the current model suffers from a vanishing gradient problem? Provider reasons for your answer. **[3]**
4. Adding more hidden layers often improves the prediction capabilities of a model. Alter the existing model **only** by adding an additional 5 hidden layers (using the same specifications as the current hidden layer). The new model will therefore

have 6 hidden layers, each with a hyperbolic tangent activation function and a Random Uniform kernel initializer with a range of [0, 1].

a. Report on the new model's training and testing accuracy as well as illustrating the plots (training and testing accuracy versus the training epochs) **[7]**
b. Studying the plots and the accuracy metrics, do you think the current model suffers from a vanishing gradient problem? Provider reasons for your answer. **[3]**

5. Alter the latest (as developed in the previous question) model by implementing the following: 1) Change the activation functions of all the hidden layers to a ReLu activation function. 2) Change the kernel initializer to the He initializer (`he_uniform`). Alternatively, replace the kernel initializer with an initializer which will prevent the vanishing gradient problem. As a result, the latest model will have 6 hidden layers with ReLu activation functions and a new kernel initializer. All other model hyper-parameters should stay the same as in the previous questions. **[3]**
   a. Report on the new model's training and testing accuracy as well as illustrating the plots (training and testing accuracy versus the training epochs) **[7]**
   b. What impact did changing the activation function and kernel initializer have? **[3]**
   c. Studying the plots and the accuracy metrics, do you think the current model suffers from a vanishing gradient problem? Provider reasons for your answer. **[3]**
   d. Compare the performance of the latest model with the initial 3-layer MLP model. **[3]**

Overfitting/Underfitting in Deep Learning **[50]**

1. Explain what effect the following operations generally will have on the bias and variance of your model. Fill in one of 'increases', 'decreases' or 'no change' in each of the cells: **[12]**

	Bias	Variance
Regularizing the weights		
Increasing the size of the layers		
Using dropout to train a deep neural network		
More training data		
Implementing early stopping		
Combining multiple outputs from neural networks trained in parallel		

2. Which of the following techniques perform similar operations as dropout in a neural network? **[2]**

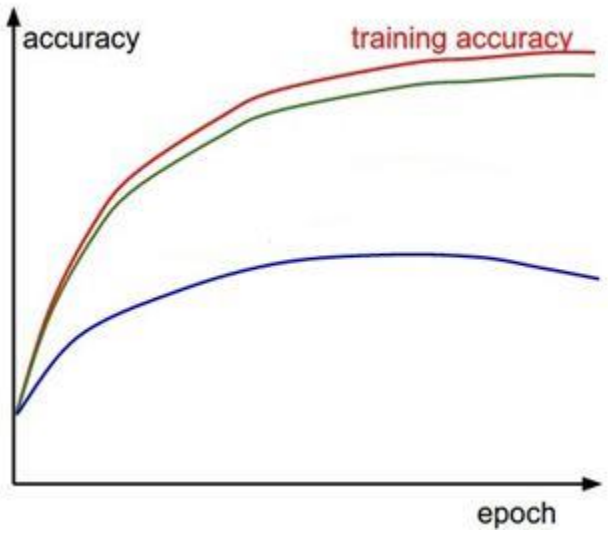
A. Bagging

B. Boosting

C. Stacking

D. None of these

3. The red curve above denotes training accuracy with respect to each epoch in a deep learning algorithm. Both the green and blue curves denote validation accuracy. **[2]**



Which of these indicate overfitting?

A) Green Curve
B) Blue Curve

4. Suppose you are using early stopping mechanism with patience as 2, at which point will the neural network model stop training? **[2]**

Epoch	Training Loss	Validation Loss
1	1.0	1.1
2	0.9	1.0
3	0.8	1.0
4	0.7	1.0
5	0.6	1.1

A) 2
B) 3
C) 4
D) 5

5. In this question, we will investigate a dataset that is prone to overfitting due to some of its features. You will start with a base model and implement certain features to prevent overfitting.

i) Load the training set called trainX_overfitting.csv into your environment. You should use an 80/20 split for training/validation when training and validating the model.
ii) For the base model, develop the following deep network. **[10]**
    - An input layer of size 300 (matching the number of features in the data). Add 3 hidden layers, each with a Relu activation function. The first hidden layer should have 300 nodes, the second hidden layer should have 128 nodes, and the third hidden layer should have 64 nodes. The output layer should have one node to predict the class (with a suitable activation function for a binary classification problem).
    - Any kernel initializer may be used (Keras uses a default initializer if none is specified).
    - Use the 'adam' optimizer
    - Use a suitable loss function
iii) Train and evaluate the model by printing the training and validation accuracy. What observations can you make from the plot? Is the model currently overfitting? If so, provide reasons for your answer. **[5]**
iv) Implement **at least** 3 measures preventing overfitting. Provide reasons for implementing the techniques you decide on. **[7]**
v) Print the training and validation accuracy of the new model. Have you improved the overfitting nature of the original model? **[6]**
vi) Name and discuss 2 techniques to prevent overfitting in addition to the techniques you implemented. **[4]**