Industrial Engineering (Deep Learning) 874
Post-block Assignment 3: State of the Art Deep Learning
Department of Industrial Engineering

Deadline: 24 July 2020, 23:59

Total: 139

Instructions

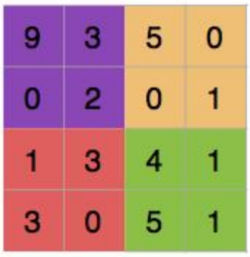
The focus of this assignment is to test your understanding of the concepts covered in the lectures from days 4-5. In addition, your implementation of these concepts on real-world data will also be tested in this assignment.

- Answer all the questions below.
- Where asked to provide all calculations and steps in the methodology you used, please make sure that you do so. 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 answers. For instance, submit your Python (or other programming language) script 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.

Convolutional Neural Network (CNN) **[44]**

1. What are the four most important layered concepts in convolutional neural networks (CNNs)? Briefly discuss the functionality of each of these in a CNN. **[8]**

2. What does Pooling do in a CNN? Apply max pooling to the feature map in the figure below with a 2x2 filter and stride of 2. Create a figure showing the resulting pooled feature map. **[2]**



9	3	5	0
0	2	0	1
1	3	4	1
3	0	5	1

3. Suppose, for a CNN, the input size is 6×6 and the filter size is 8×8. What would the size of the output be? **[2]**

4. Explain what the difference is between valid padding and same padding is in a CNN? **[2]**

5. In this question, we will develop a CNN to detect whether a plane is present in an image or not. The data set consists of 32000 images of which 8000 have planes (of different shapes and sizes) in the image. Images with planes have a label of 1, whereas images without a plane have a label of 0. An 80/20 split should be used for training and testing your developed CNN model. Numpy arrays with the training and testing data are provided if you are working in Python and can be loaded using *loaded_array = np.load('file_name.npy')*. These arrays are called *x_train_planes.npy, x_test_planes.npy, y_train_planes.npy, y_test_planes.npy* and can be downloaded from SunLearn. Alternatively, download the data set called *planes.json* from SunLearn and remember to shuffle the data set before generating the train and test data sets.

i. Develop a base CNN model to accurately distinguish between the two classes of the response variable. **[10]**

ii. Improve the model developed in (i) by implementing at least 3 techniques which are commonly used to improve network performance in CNNs. Discuss and illustrate the effect of the techniques used on the performance of the model. Consider other classification performance metrics (such as F1 score, recall) in addition to training and testing accuracy (and loss) when discussing the performance of the various candidate models. **[15]**

iii. Create a classification report for the final model which includes accuracy, recall, specificity, and F1 score. An accuracy and F1 score of around 98% should be aimed for. **[5]**

Long Short Term Memory (LSTM) networks **[45]**

1. Why do LSTMs (and RNNs) work better with text data than other neural networks? **[2]**

2. Briefly discuss how LSTM addresses the vanishing gradient problem. **[3]**

3. In this question, we consider a multi-class text classification problem. The well-known Consumer Complaints data set is used. A smaller version (consisting of about 39000 complaints) has been created for the purpose of this exercise. The data consists of 11 categories of complaints. Complete the following steps:

   i. Load the data set (‘*Complaints.csv*’) into your programming environment.

   ii. Apply text pre-processing techniques (such as tokenizing) to the data set in order for it to be used in the RNN and LSTM models. **[5]**

   iii. Use an 80/10/10 split for training/validation/testing in the following steps.

iv. Create an RNN model to predict the category of complaints. Generate a classification report for predictions on the test set with your best RNN model. **[15]**

v. Now create an LSTM model to predict the category of complaints. Similarly, generate a classification report for predictions on the test set with your best LSTM model. **[15]**

vi. Comment on your observations from *iv* and *v* by elaborating on why these results were obtained. **[5]**

Asynchronous Actor-Critic Agent (A3C) and Generative Adversarial Network (GAN) [50]

1. The Asynchronous Advantage Actor-Critic (A3C) algorithm has gained popularity in recent years and has become the go-to Deep RL algorithm for new challenging problems with complex state and action spaces. Provide a high-level overview of:

i. What the A3C algorithm is by elaborating on the mechanics behind the algorithm with respect to its name. **[5]**

ii. How the A3C algorithm work. Elaborate on how the algorithm trains the A3C model. **[5]**

iii. Contrast the A3C algorithm with the Deep Q-Network (DQN) algorithm. **[5]**

2. In this question, we will explore Generative Adversarial Networks (GANs). Specifically, we will use the MNIST data set (where we have 28×28 pixel images of digits 0 through 9) to build a GAN.

i. Load the mnist data set into your programming environment.

ii. Develop a GAN that has the ability to learn by itself how to synthesize new images. You are free to use any architecture and hyperparameters when developing the discriminator and adversarial models. Clearly describe your logic and add comments to your code. **[25]**

iii. Produce sample generative outputs of the images when you train the network and discuss your observations. **[10]**