CS 6886: Systems Engineering for Deep Learning
Mini Assignment 1
Total Marks: 5

Q1: Mention 5 sources of complexity in a machine learning / deep learning system.

1. The most important and well-known complexity in machine learning systems comes in the form of the tradeoff between bias and variance of the model. The machine learning model we develop uses the training data to make some generalized function which works on producing an output given a set of inputs. The tradeoff is between how generalized the model will turn out, will it be overly generalized, thus being useful for nothing (high bias), or the generalization is too low, which makes the model work on only specific data (high variance). To find a balance between these two features is a major complexity in machine learning. (**Design Tradeoff**)

2. We do not exactly know what set of hyperparameters will work for any given machine learning problem. Since the hyperparameters by themselves affect the models in a complex way, choosing the right set of parameters which will extract the best results from a machine learning system adds to the complexity of the task.

3. Usually, we often face situations where the amount of data which we have is not enough for properly training the model. This situation usually occurs in the case of deep learning training which requires millions of data points to effectively give a proper solution. The data requirement also increases when we increase the deepness of the neural network, as deeper neural networks require more data to reduce the bias in the model. (**Incommensurate Scaling**)

4. There are cases where there are errors in the labelling of data. If the mislabelling dataset size is small, it is usually ignored, but relatively ample worse of mislabelling might go unnoticed during the training or testing period but will become apparent when we use the machine learning model in real-time. Thus finding out whether data has been mislabelled is a complexity. (**Propagation of Errors**)

5. Sometimes a model which works perfectly on the real time data might not be good enough for deployment since it might fail other objectives which we might not have foreseen like latency for prediction is too high, or cost for preprocessing live data is high, and many more. (**Emergent Properties**)

Q2: Mention 5 ways (with examples) to limit or manage the complexity in a ML/DL system

1. To tackle the problem of bias-variance tradeoff, first we split the data into train data and validation data. We train the model using train data while simultaneously testing the model using the validation data (which the model does not know about, and thus will simulate real-time data). When the error of the model on both train data and validation data is high, we know that the model is generalizing too much and thus we need to train the model more. When the error on training data is low but the error on validation data is much higher, then the model is not generalizing enough. Thus to tackle this case we try to use regularization and other techniques to mitigate the high variance in the model.

2. To tackle the hyperparameter selection problem, we first create a grid of hyperparameters and iterate over this grid by selecting hyperparameters from this grid to train the model on a smaller subset of data. Whichever set of parameters from this grid work best in this case will be used for further training. We also have to make sure the best set of parameters lie somewhere in the middle of the grid, otherwise we have to increase the size of the grid, since the best possible case might be somewhere outside the grid which we chose.

3. To tackle the problem of lack of data, we try to increase data by manually adding changes like noise, rotation, etc. to existing data to create new data which looks somewhat different from the original data but still belongs to the class of data we are dealing with. Using this method of data augmentation can somewhat mitigate the lack of data issue which is generally present in deep learning.

4. Tackling mislabelled data is one of the more difficult tasks in machine learning. Low level methods like checking random samples of data for mislabelling is one way in which it can be tackled, as it is effective for datasets with either low number of classes or manageable dataset size. Also filtering out outliers is a good way to counter mislabelling. But if the dataset size is huge with many classes, the above methods are not feasible. Research is being done on model giving feedback when the training data for a particular class changes abruptly, and might prove to be a good method for tackling mislabelling.

5. Tackling unforeseen downfalls of the model could be mitigated by taking into account the required specifications of the model in real time during the model development part itself and whenever the model is being changed, we have to make sure that the model is conforming to the limits set by the specifications to work in real time.