Assignment #: 1

Syed Alle Mustafa

September 20, 2022

1. *Difference between training, validation and testing datasets.* All three of the datasets are same as the nature of data is considered since all of them are partitioned from the same dataset. As far as training dataset is considered its the largest them of all because it is the dataset we use to feed the model to learn the parameters and predictions are made on logic learnt from this data points. Validation set is the dataset used for setting and tuning the hyper-parameters. They are used once training is done and to find optimal values for hyper-parameters. Testing set is used once the final model is trained and validated, this dataset is to evaluate the performance of model on unseen data.

2. *Four steps for writing network program using Keras.*

   (a) The first step is to **Load and prepare the data**. In this step data is loaded from the the data source and data preparation is done on the it (one-hot encoding, removing useless features). The the alter data is separated in to 3 sets, i.e, training set, testing set and validation set.

   (b) The next step is to **train the model**, the training set used in this step. Number of layers, and type of layers, activation and loss functions are defined in this phase. Then the model is trained on the training set.

   (c) The third step is to **evaluate the model** based on results from validation set. The evaluation is done by changing the hyper parameters which include number of layers, activation/loss functions.

   (d) The last step is to **Test the model on unseen data**. The testing data is now used on the final evaluated model. The accuracy of the model is found out on testing set. After this set, model is set for real time predictions.

3. *Basic parameters used to define dense layer.*

   (a) The most basic parameter that is used to define the dense layer is **units**. It basically defines the output size of the dense layer. It is necessary for it to be a positive integer, and of that size, the output vector is given from the layer.

   (b) The another basic parameter is the **activation function**, which is the function through which a neural network maps or transfer the input values into an output to like yes/no, 0-1, -1-1 or anything depending upon the nature of a given activation function.

4. *Network configuration while compiling the model.*

   Compiling the model is the last phase of creating the network, after which the training is starting following the the main parameters for compiling the model:

   (a) **Optimizer** is the argument which defines the kind of optimizer to use, they are algorithms that modifies weights and learning rate of the neural network to minimize overall loss and improve the accuracy.

   (b) **Loss** is also an argument we give during compiling the model. It is a numeric value, which is to be minimized while the training process. If the loss value is lesser, then our model makes close predictions. There are multiple loss functions for example, categorical, mean squared error and many more.

   (c) **Metric** is a argument that compiling model, it is item or list of item, that is monitored during the training and can be used to see the learning rate of the model after the training is done. Some of the metrics are accuracy metrics, probabilistic metrics, regression metrics, and many more.

   (d) The loss function is used to optimize your model. This is the function that will get minimized by the optimizer. The main difference between a loss function and metric is that loss function is used by optimizer during the training process to increase performance of the model, where evaluation metric is used after the training is done to check the learning rate of the model, for example if it overfit or underfit. A metric is used to judge the performance of your model. This is only for you to look at and has nothing to do with the optimization process.

5. *Five basic arguments to provided to fit a model*

   Following are the arguments that is to be provided to fit a model:

   (a) **Input or (X)** is a single or multi dimensional array given as input to the model. This array is the features of the data that would be given while making prediction of the target value.

   (b) **Output or (Y)** is an vector of the feature that is to be predicted once the training is done. The sole purpose of training the model is to make it predict y using the values of x.

   (c) **Batch size** is the number of samples from the total data set that might be used in one training propagation. The higher the batch size is the more memory is consumed during the training process.

   (d) **Epochs** is the number of times model is iterated over the entire training data set (x and y).

   (e) **Validation data** is the validation dataset that is given in order to evaluate the model after every epoch. This dataset is not used for the training, its only for optimization using the loss function and optimizers.

6. *How to convert variable length text string in to binary feature vector.*

   Firstly, The string should be traversed once and the list of words containing in the string can be converted in to a dictionary. this step can be done using python dictionary. A binary vector can now be defined having a same length as the dictionary keys. Each word can be assigned an index based upon the dictionary, now the list of words can be traversed again and whenever a word is found its index assigned should be changed from 0 to 1. this is how sentence containing string of variable sizes can be converted in to a binary feature vector.

7. *Possible conclusion when observing training and validation graphs.*

   The graph found after training and validation can easily determine whether the model is overfit or underfit, It the loss of the training set decrease very much but the loss of validation set increases, or the accuracy increases of the training set but decreases of the validation set, both these cases determine that the model is overfitted, so, the cycle where loss or accuracy of both the training and validation set is minimum that cycle and params are chosen for the final model. Similarly, if the accuracy is low and loss is high of both sets then the model is underfitted, and it needs more epochs or data for training.

8. *Possible conclusion when observing training and validation graphs.*

   The graph found after training and validation can easily determine whether the model is overfit or underfit, It the loss of the training set decrease very much but the loss of validation set increases, or the accuracy increases of the training set but decreases of the validation set, both these cases determine that the model is overfitted, so, the cycle where loss or accuracy of both the training and validation set is minimum that cycle and params are chosen for the final model. Similarly, if the accuracy is low and loss is high of both sets then the model is underfitted, and it needs more epochs or data for training.

9. *Vector of predictions from binary classifier with a logistic function can be converted to class decisions.*

   To get the class decisions, the distance from the decision boundary can be find out of the vector of predictions and the closest class determined by the distance can be chosen as the final output. While doing so, activation functions can also play a major role. The probability of possible outcomes would be given as an output and the one with max probability can be chose as the class decision.

10. *Explain one-hot encoding for multi-class classification.*

    One-hot encoding is the technique labels in to numeric data in order to make it easier for model to make predictions. for instance, if there are 5 or (n) categories, then we will make a vector of length 5 or n. Then 6 binary classes would be made where every bit determines a class. If the bit is on so it signals the class or category assigned to that bit.

11. *Output layer when using softmax as an activation function*

    Softmax is an activation function that gives the probability of classes. The main advantage of using softmax as an output layer is that it gives range of probabilities (0-1) as an output. The range is for multiclasses and sum of all probabilities would be always 1. The target class or predicted class would be the one with highest probability.

12. *Difference between sparse-categorical-crossentropy and categorical-crossentropy*

    The main difference between both crossentropy is that sparse categorical crossentropy gives a category as a final output that is predicted, where categorical crossentropy gives probability of each class that have possibility to be the actual class. In SCC the classes are mutually exclusive, and in CC one sample can have soft probabilities of multiple classes.

13. *Accuracy of a random classifier when 5 classes each having same probability.*

    Formula for classification accuracy of a random classifier is as follows:

    $$Accuracy = 1/k$$

    where K is the number of classes, so, as there are 5 classes so the accuracy is follow:

    $$Accuracy = 1/5 => 0.2$$

14. *How to normalize vectors to have same mean and standard deviation.*

    In order to do so, we can use batch normalization. In batch normalization the input vector is normalized so that its mean and standard deviation becomes zero. This helps in gradient descent.

15. *Difference between MSE and MAE, which is easier to interpret.*

    Mean average error is the mean distance between the real data and the predicted data, where mean squared error is the mean squared difference between the estimated values and the actual value. MSE is easier to interpret because it is easy to perform mathematical operations in comparison to a non-differentiable function like MAE.

16. *How to perform K-Fold cross validation, why is it needed*

    K-Fold Cross validation is a technique for validating the model on a limited sample data and to make sure that each validation data is only validated once. So the validation set does not get mixed with the training set. To Perform K-Fold Validation:

    (a) Pick number of folds i.e k.
    (b) split data set into k equal parts.
    (c) assign k-1 for training and remaining one for validation.
    (d) using the training set train the model. On each K iteration train a new model independent from the previous model.
    (e) validate the model through validation set and save the result.
    (f) now you have validation score on each k iteration so to find the accuracy, get the mean of the accuracy of each model.

17. *How to report k-fold cross validation results, how to train the final model.*

    After every running k-fold cross validation, we get an array of size k showing the accuracy, loss or some other metric of every fold, we can take a MSE of all of them to find the accuracy of final model. Although k-fold cross validation is only for validating of how will our model perform on real world data, when we have less sample data, but the process can help to figure out or tuning the hyper parameters for the final model.