Theo Guidroz
CS 577 Assignment

1. **Explain the difference between training, validation, and testing data sets. Explain the need for such datasets.**
   Training set: Used to train the model and is fed to the algorithm that generates a model.
   Validation set: Smaller than training set and used to improve the model by changing parameters.
   Test set: Used to evaluate the performance of the model
   It is important to divide the data to allow for generalization and reduce the chances of overfitting since we want the model to be accurate on all data sets and not only the one used to train the model.

2. **Explain the 4 steps used in writing a network program using Keras (data, model, learning process, fitting).**

   Define the training data: input tensors and target tensor.

   Define a model: Network of layers that maps input target. Two common types of models are sequential class and functional API.

   Configure the learning process: Specify the loss function, optimizer, and metrics to monitor.

   Train the model using fit() and search hyper-parameters.

3. **Explain the basic parameters used to define a dense layer (number of units, and activation).**

   In a dense layer, all the neurons are interconnected. The number of units refers to the output dimensionally and the activation function defines the output node.

4. **Explain the network configuration aspects when compiling the model (optimizer, loss, metrics). Explain the difference between loss and metric.**

   Optimizer: Finds the weight values that outputs the lowest loss function.

   Loss: Evaluates the performance of the algorithm by comparing output to the true values and is this value is used to optimize the model.

   Metrics: Evaluates the final performance of the model. It can be represented as a confusion matrix which is used to calculate the accuracy and precision of a model.

   The difference between loss and metric is that loss is calculated during the training process and is used to optimize the model whereas metric is only used to evaluate the performance of the final model and is not involved in the optimization process.

5. **Explain the 5 basic arguments provided to fit (input, output, batch size, epochs, validation data).**

   Input: The training data for the model.

   Output: The prediction made by the model on data after it has been optimized.

   Batch size: The amount of data trained and tested before changing the parameters.

   Epochs: The number of cycles that the training data is trained.

   Validation data: A smaller set than the training data and is used to evaluate the model of being trained on the training data. The evaluation is used to modify the parameters to improve the model.

6. **Explain the steps used to convert a variable length text string into a binary feature vector.**

   Each character is associated with a unique binary string of 1's and 0's. The location of a particular character will be represented by 1 in the overall vector. Start with an integer sequence with each character assigned to an integer and then incode it to a binary matrix. The matrix will be filled up with 0's and 1's. 1 will represent the indexes of the integers that represent the characters.

7. **Explain possible conclusions when observing training and validation loss graphs over epochs (underfitting and overfitting).**

   The more a model is optimized for a specific set of data, the more likely it is to overfit. Optimization adjusts parameters to get the best performance on the training data. This means that the model might perform extremely well on trained data but poorly on unseen data because of overfitting. When observing training versus epochs graph, the loss will go down as the number of epochs increases. This is because the parameters are modified to reduce the loss function on the training set. However, when observing validation graphs versus epochs, the loss might decrease while the model is overcoming underfitting but if the model is iterated too many times on the data set, we could see an increase in loss due to overfitting.

8. **Explain possible hyper-parameters that can be tuned (layers, units per layer, activation functions, loss).**

   Layers: This refers to the number of layers between the input and output layers. The number of layers can be decreased to overcome overfitting or increased to come up with a more complex network which could increase metrics such as accuracy.

9. **Explain how a vector of predictions from a binary classifier with a logistic function in the output layer can be converted to class decisions.**

   Coming up with a decision boundary that separates the different classes.

10. **Explain how one-hot-encoding is used to encode class labels of a multi-class classification problem.**

    One hot encoding is a process by which categorical variables are converted into a binary form that could be provided to a model to do a better job in prediction. Each class can be mapped to integers then represented as a binary vector which can be optimized by certain algorithms.

11. **Explain the meaning of the output layer when using softmax as an activation function in it.**

    Softmax converts a real vector to a vector of categorical probabilities. The elements of the output vector are in range (0, 1) and sum to 1. This represents the probability distribution of the outputs.

12. **Explain the difference between sparse-categorical-crossentropy and categorical-crossentropy.**

    Both are loss functions that compute the crossentropy loss between the labels and the output function. The main difference is that sparse-categorical-crossentropy is used with one-hot encoded labels while categorical-crossentropy is used with integer tensors.

13. **Assuming a dataset with 5 classes where each class is represented equally, what will be the accuracy of a random classifier?**

    ⅕ x100 so 20%. If chosen randomly, the classifier will average one good answer every five tests.

14. **Explain how to normalize vectors to have equal mean and standard deviation. Explain the purpose of such normalization.**

    Normalization converts values to a range of 0 to 1 inclusive. Each value is normalized and the mean and standard deviation is taken. To normalize a value, subtract by the mean and divide by the standard deviation. This process ensures that features with larger values do not overweight other features. Example: Normalizing the price of a house when it was built and normalizing the age of the house so that the bigger value (the price) does not dominate the smaller value (the age) when predicting its current value.

15. **Explain the difference between the MSE and MAE metrics. Which is easier to interpret?**

    Mean Squared Error (MSE) represents the difference between the original and predicted values extracted by squared the average difference over the data set whereas Mean Absolute Error (MAE) represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set. Both values are always positive and the closer the value is to zero, the better the performance. Since MSE squares the value, the difference between the estimated value and the actual value is harder to interpret because the square function exaggerates the error. Therefore, MAE is easier to interpret.

16. **Explain how to perform a k-fold cross validation. When is k-fold cross validation needed?**

    K-fold validation is the process of splitting data into k-parts, leaving one part out, training on the remaining k-1 parts, computing the average error, and repeating this process k times. K-fold cross-validation is needed when the amount of data available is relatively small since, in this validation process, all the data is used for training. For the final model, train on all examples.

17. **Explain when performing k-fold cross-validation how to report the validation error and how to train the final model.**

    Compute the validation error of each fold k times and average all the numbers calculated to find the average validation error. For the final model, train on the entire dataset.