Model hyper parameter tuning procedure and results

With respect to the training of the classification algorithms, the first matter addressed pertains to finding appropriate hyper parameter values for the models. In the case of k-NN, a suitable value for k is found by varying the value of k from 1 to 50 and documenting the AUC performance scores. Based on the results, a value of $k = 10$ is chosen for the silent-sound classification analysis. In the case of the RF model, the number of trees constituting the forest was set to 128, in accordance with the results and recommendation of Oshiro *et.al.* (2012).

In the case of the SVM model, the results of the well-known *grid-search* hyper parameter tuning approach indicated that suitable hyper parameter values (or settings) for the SVM learning model are an RBF kernel with a regularisation parameter value of C = 1 and the kernel coefficient set to *auto* (*i.e.* $\gamma = 1 / n_{features} = 1 / 12 = 0.083$). The DT algorithm achieved the best test AUC scores with the minimum number of instances in nodes set to twenty. In the case of the LR model, the following regularisation parameter values were considered: $C \in \{10,1,0.1,0.01,0.001\}$. The LR model was ultimately implemented with L_2 regularisation and a regularisation parameter value of C=0.01, which delivered improved classification performance.

In the case of the MLP model, the well-known *random search* and *Bayesian optimisation* methods were employed the purpose of navigating the hyper parameter search space (Abadi *et. al.,* 2015, Chollet, 2015). A total of 100 candidate models were evaluated and the best-performing architecture and hyper parameter configuration of the MLP model with respect to the silent-sound film style data comprises the input layer, one hidden layer with 22 neurons and the *rectified linear unit* (ReLu) activation function, and one output layer with a single neuron that uses the *sigmoid* activation function. Furthermore, L_2 regularisation was implemented with a regularisation factor of 0.01, effectively mitigating the risk of over-fitting the training data. The model is based on an implementation of the bounded *limited memory Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS-B) solver function with simple bound constraints imposed on input variables for weight optimisation (Fletcher, 2013). The L-BFGS-B solver function tends to perform well and converge fast on small data sets (Scikit-Learn, 2020). Due to this observation and based on preliminary analysis results, the final MLP model was trained for a maximum of only 12 epochs with a batch size of 32.

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Corrado G, Citro C, Davis A, Dean J, Devin M, Ghemawat IS Goodfellow, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Man´e D, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vi´egas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y & Zheng X, 2015, *TensorFlow: Large-scale machine learning on heterogeneous systems*, [Online], [Cited April 2022], Available from https://www.tensorflow.org/.

Chollet F, 2015, *Keras*, [Online], [Cited May 2021], Available from https://github.com/fchollet/keras.

Fletcher R, 2013, *Practical methods of optimization*, 2nd Edition, John Wiley & Sons, New York (NY).

Oshiro T, Perez P & Baranauskas J, 2012, *How many trees in a random forest?*, Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition, Berlin, pp. 154–168.

Scikit-learn, 2020, *Multi-layer perceptron classifier*, [Online], [Cited May 2021], Available from https://scikit-learn.org/stable/modules/generated/sklearn.neuralnetwork.MLPClassifier.html.