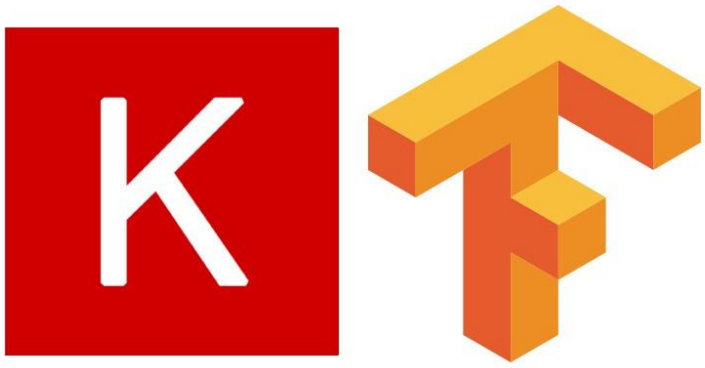
4.0 Model Estimation

Now let's talk about model estimation. Here we have used Keras tuner for hyperparameter search. Keras tuner easily configures your search space with a define-by-run syntax, then leverage one of the available search algorithms to find the best hyperparameter values for your models. It comes with different search algorithms Bayesian Optimization, Hyperband, and Random Search. First, we specify a range number of layers and nodes, so that the search algorithm will return the best suitable number of layers and nodes for our model. For that, we need to specify a function that will do the number of nodes and layers estimation. This function will return the desired output. In this case, we have given the range for nodes as 2 to 20 and the range for layers as 32 to 512. The search algorithm we used is RandomSearch which randomly tries a combination of hyperparameters from a given search space. We need to define the RandomSearch inside which we need to specify our function for estimation, define objective as val_accuracy for classification and val_mean_absolute_error for regression, the maximum number of trials is 5 and we specify 3 execution per trial. Finally, we initialize our algorithm using search_space_summary. Then we need to specify the search function with training and validation data for the model Using result_summary we get 10 best trials with minimum loss. Finally, we get the best some of the best values for the number of layers and nodes with minimum loss.

4.1 Model implementation



For our classification and prediction operation, we use the Keras library of TensorFlow. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. Keras runs TensorFlow2. It is an approachable, highly productive interface for solving machine learning problems. It mainly focuses on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

One of the most important data structures of Keras is layers and models. The simplest type of model is the Sequential model, a linear stack of layers. Next, we use add function to create the required number of layers. Inside we specify a necessary number of nodes. The activation function we use is relu. Finally, our final layer must contain the same number of nodes as our output column. Once our model is ready we configure its learning process using compile function. The loss function used is mean_square_error for regression and categorical_crossentropy for classification purposes. The optimizer

function used is adam. All of these needs to be assigned in function and after that need, we declare our function to a variable e.g. model.

4.2 Model training

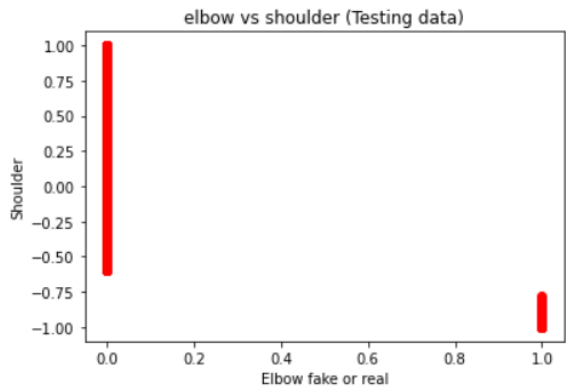
Once our model is ready we need to train our model. For that purpose, we need to split our data into training and testing data using train_test_split from sklearn.

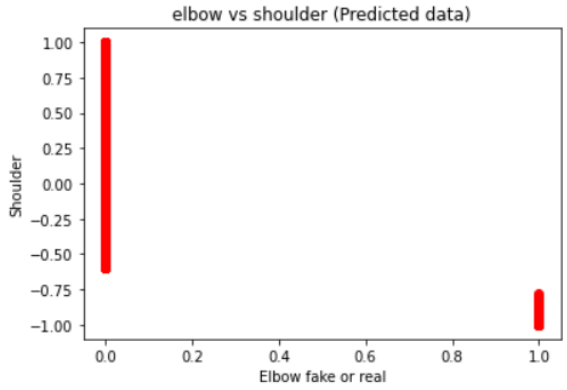
We separate 30% of the dataset as testing data and the rest for training purposes. Once we get our training data we use the fit function to fit ours data with a validation split of 0.3 and 100 epochs. After training for 100 epochs we get 100% validation accuracy for classification and 0.05 validation loss for regression. Once that is done we predict our output using predict function and pass in our testing data.

4.3 Accuracy estimation

Once we get our output data we inorder to make sure our model is making the right predictions and classification we evaluate our model using evaluate function. For classification, we get 100% accuracy and for regression, we get 0.05 loss. Hence we can conclude that our model is working with good efficiency.

For our classification model, we plotted two diagrams, one which represented elbow is fake or real vs shoulder using predicted data and second diagram with same using testing data.





The result is obvious, as we got 100% accuracy the right exercise prediction for testing data was the same as predicted data as the values of testing and predicted data were accurate.