NOTTINGHAM TRENT UNIVERSITY

SCHOOL OF SCIENCE AND TECHNOLOGY

COMP40511: Applied AI & Data Mining 202122

by

Manav Rachen N0797516

in

2022

MSc Computer Science

I hereby declare that I am the sole author of this report. I authorize Nottingham Trent University to lend this report to other institutions or individuals for scholarly research.

I also authorize Nottingham Trent University to reproduce this report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for scholarly research.

Signature

Manav Rachen – N0797516

Contents

Introduction

This document will be discussing all the tasks that is required for this AI module. The FaceRec file has been given to us to adapt and use to complete the following tasks. This file allowed us to download the data set of faces, using these data set multiple classifier models have been made. These models will be compared by an accuracy score, parameters used will be different in each model providing a different accuracy in each model. A tuned version of a model will also be discussed as this will show the best optimised parameters that can be used for the best accuracy for the data set. The clusters will be discussed showing the optimum clusters for the data set as well as the cluster that is optimised for George bush and his pictures within the data set. An elbow graph will show a visual view of all the clusters and the highest and lowest optimised cluster.

Task 1 Random Forest

"A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True(default), otherwise the whole dataset is used to build each tree" (Pedregosa, 2011).

This segment of the code represents the random forest classifier. First, the code imports the random forest classifier. The max_depth is set to two which means that it contains two levels of decision trees in the model. Random state was used to determine the random generation for weights and building the trees. As we have specified the number of trees should be at least two, each tree has an equal chance of being selected as a leaf node or the final prediction. We then fit the data using X_train_pca and y_train this then allows the code to predict the accuracy of the random forest classifier against the data. The overall accuracy of this classifier was 0.45.

Task 2 Multi-Layer Perceptron Classification

MLPClassifier relies on an underlying Neural Network to perform the task of classification (Nair, 2019).

This MLP classifier section starts off with importing of the MLP classifier. Next, we determine the object parameters for the section the solver what handles the weight optimization is adam. This parameter was chosen as it is suitable with large datasets. Next alpha was chosen for the next parameter it constrains the changes of the weighting resulting in a low bias result. Next parameter that was selected is hidden_layer_sizes**,** 15 and 10 are the neurons that are being used within this hidden layer. Random state has a value of 2 this "determines random number generation for weights and bias initialization, train-test split if early stopping is used, and batch sampling when solver='sgd' or 'adam'. Pass an int for reproducible results across multiple function calls" (Pedregosa, 2011). The code above will take the training data and create a model that predicts the target variable, y_test, based on the input variable, X_train_pca. The accuracy overall for this is 0.74 what is an increase from the initial 0.45 that the random forest classifier produced as its result.

Task 2 MLP Classification extended high parameter tuning

This extended functionally determines the best parameters for the mlp classifier then rebuilding the model to give a new accuracy with the optimised parameter.

The MLP begins with a selection of different types of parameters. The parameters used are hidden layer, activasion, solver, alpha and learning rate. "The initial learning rate used. It controls the step-

size in updating the weights." (Pedregosa, 2011). Within these parameters contains multiple values for each parameter that will be searched through to find the optimised parameter the method used for this was grid search.

"There is GridSearchCV method which easily finds the optimum hyperparameters among the given values" (Panjeh, 2020). The optimised values for each parameter were listed, a new mlp model was built using the new values gaining an accuracy of 0.81 percent what's an increase as the previous mlp had an accuracy of 0.74.

Task 3 Convolutional Neural Network Classification

"Convolutional Neural Network is a Deep Learning algorithm specially designed for working with Images and videos. It takes images as inputs, extracts and learns the features of the image, and classifies them based on the learned features" (Deepanshi, 2021).

The code first changes the shape of all images within the data set what the values 50, 37, 1. We use the KERAS import library to build the model. The layers used for this model were input, this allowed the reshaped numbers to be inserted into the model. Conv2D is the second layer used this creates a convolution kernel this allows the model to produce the outputs. We use 2 filters 32 and 64 this represent the number of filters that should be learnt by the conv layer and RELU as the activation because it's one of the most used activations for CNN. We use maxpooling because this will downscale the image if its not being used this will the replace it with convolution to extract the most important features according to the filter sizes. Theres a flatten layer to get rid of the last fully connected layer in the network before dropping out 0.5% of the data from every batch to prevent overfitting on training data. The dense layer receives input from all neurons from all the previous layers in the model, this layer is also commonly used in models. "The softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution" (Brownlee, 2020).

we train the model by declaring the number of epochs that the model will run through, and the batch size is also stated for the model.

We compile the model using the optimiser adam as this is good for large data sets. The new set up the metrics for accuracy to be calculated from predictions made by the model.

Once the epochs and batch size has been stated the evaluation stage can begin printing out the accuracy of the model. The accuracy that the CNN model present was 85 percent this is an increase from the base MLP classification. The parameters are outputted in a table shown below.

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 48, 35, 32)	320
max_pooling2d_4 (MaxPooling2	(None, 24, 17, 32)	0
conv2d_5 (Conv2D)	(None, 22, 15, 64)	18496
max_pooling2d_5 (MaxPooling2	(None, 11, 7, 64)	0
flatten_5 (Flatten)	(None, 4928)	0

_____

dropout_2 (Dropout) (None, 4928) 0

_____

dense_8 (Dense) (None, 10) 49290

==========

Total params: 68,106
Trainable params: 68,106
Non-trainable params: 0

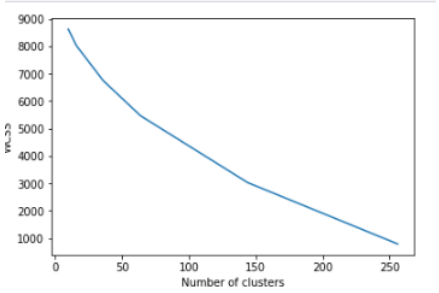
Task 4 Clustering

KMeans was used for the clustering task. "*K*-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K" (Trevino, 2016) .

The data first is normalised and then reshaped because the data is currently in a 3 dimensions format what needs to be changed to a 2-dimension format. The kmeans function is created to hold the number of clusters that will be in our model. This searches through all the clusters to find the best possible combinations. The output of the code is listed out that represents each cluster in the dataset.

The clusters are listed out in an array that will print out the accuracy for each of these clusters. The code begins with creating a dictionary called reference labels this allows the code to iterate through the list of labels in the kmeans clustering algorithm. A list is created called cluster labels that will hold all labels generated by kmeans. The variable number labels store the number displayed in the images. A loop is created to run through each label of the clusters. Another for loop is created that contains the kmeans function, the total clusters, putting the model to the training set and calculating the metrics. The metrics involves the number of clusters that is being printed. The accuracy for the clusters were printed out these were 10,16,36,64,144,256. The accuracy for 10 was 0.42, 16 was 0.43, 36 was 0.43, 64 was 0.46, 144 was 0.50 and 256 was 0.65.

An elbow graph has been created to determine the optimised cluster that presents the most accuracy based on the dataset. This has been done by the matplot import and the keras import.



This is the chart shows the clusters that were searched through as you can see the higher amount of cluster had the most accuracy.

For the next section of the clustering task the code first finds out how many clusters contains an image of George bush and what clusters they are contained in. Once the clusters have appeared

the next section of code will go through the dataset and predict which cluster contains the most images of George bush. This is done by passing through the X_test, the code then loops through this list and checks if any point has a value equal to 3 as this is the optimum cluster for George bush and if so, it appends that point into X_bush. This is then reshaped and printed through the kmeans .predict function giving us the optimised cluster for George bush which is 3.

Task 5 Ethical

Many organisations use image recognition from autonomous cars that detect road signs and road lanes. "Industrial manufacturers and utilities, machines have learned how to recognize defects in things like power lines, wind turbines, and offshore oil rigs through the use of drones" (Noble, 2019). This reduces humans from dangerous environments, improving safety in dangerous work environments. However, is machine AI ethics important well for example an autonomous car image recognition system fails or bugs out and crashes against a car in the next lane who is liable. Well, the liability of this issue can go to 3 choices the designer, the owner, or the system itself all these things must be considered what can cause many issues on who should be liable. Face recognition can be used for good however it can be used for bad. "The most obvious example of the misuse of image recognition is deepfake video or audio" (Noble, 2019). The AI creates a fake audio or video providing mis leading content that can change views or opinions of certain communities that the video is targeting.

Task 5 Social

Privacy is a major issue due to the fact of the neural network learns what kind of person you are. This means that supermarkets for example use their AI to find out what their customers shopping behaviours are. Some people many are not happy with the storage of personal data without consent of the person as this can cause a privacy breach. Another social issue that AI is likely to replace many white-collar workers the reason for this is because companies can use AI to become more independent from workers and frees up the human workforce for better equipped and important jobs. The databases containing facial scans and identities are being used by various parties such as banks, police forces, government, and other defense firms and are hence prone to misuse (B, 2021).

References

Brownlee, J., 2020. Softmax Activation Function with Python. *Making Developers Awsome at Machine Learning.*

B, S., 2021. *Future Impacts of AI on Image Recognition.* [Online]
Available at: https://techaffinity.com/blog/impact-of-ai-on-image-recognition/
[Accessed 01 06 2022].

Deepanshi, 2021. Beginners Guide to Convolutional Neural Network with Implementation in Python. *Data Science Blogathon.*

Nair, A., 2019. *A Beginner's Guide To Scikit-Learn's MLPClassifier,* s.l.: DEVELOPERS CORNER.

Noble, B., 2019. *The Ethics of AI Image Recognition.* [Online]
Available at: https://blog.cloudera.com/the-ethics-of-ai-image-recognition/
[Accessed 01 06 2022].

Panjeh, 2020. *scikit learn hyperparameter optimization for MLPClassifier.* [Online]
Available at: https://panjeh.medium.com/scikit-learn-hyperparameter-optimization-for-mlpclassifier-4d670413042b

Pedregosa, F. e. a., 2011. sklearn.ensemble.RandomForestClassifier¶. *Journal of Machine Learning Research,* Volume 12, pp. 2825-2830.

Trevino, A., 2016. *Introduction to K-means Clustering.* [Online]
Available at: https://blogs.oracle.com/ai-and-datascience/post/introduction-to-k-means-clustering