Assignment-3

Abstract

The report explains about implementation of convolution neural network (CNN) to extract features from the images in a dataset. The dataset used in this model contains totally 1500 CT scans of various parts of body such as abdomen, chest, and head. In this report, CNN is built from scratch to extract features from the dataset. Here the features are extracted from intermediate dense layer. After extracting the features, the K -nearest neighbor (KNN) algorithm is applied to the extracted features and determining the optimal KNN model. At last, Random forest algorithm is applied to the extracted features and tuning the two hyperparameters using random search to obtain better performance.

Introduction

Convolution Neural Network (CNN)

Convolution Neural Network is an Artificial Neural Network, it is most popularly used for analyzing images. It also can be used in data analysis and classification problems. CNN has ability to detect pattern and makes sense of patterns, pattern detection making CNN most useful.

CNN has input layer, output layer and hidden layers which are called as convolution layers. Convolution layer makes CNN most powerful network. The convolution layers contain some filters which helps them to abstract features or patterns. The main advantage of CNN is its architecture. The architecture of CNN is very simple, it contains input layer, convolution layers, pooling, RELU (rectified linear unit layer) acts as activation function ensuring nonlinearity and fully connected layer. The main function of RELU is conversion of non-linear model into linear model. Training and testing of model are simple with CNN. The CNN is used in many applications such as analysing the images, image classification, image and video recognition and in Natural Language Processing.

K-Nearest Neighbour (KNN)

K-Nearest Neighbour algorithm is simplest classification algorithm. The KNN algorithm assumes that similar things lie close to each other.

Random Forest Algorithm

Random forest algorithm is classification algorithm. It creates a decision trees on data samples and performs the prediction on each of them and finally selects the best solution through the voting.

Proposed Model

Convolution Neural Network (CNN) Model

The steps involved in this model are

1)Pre-Setup (Importing all the required libraries)

2)Dataset Pre-Processing

3)Model Creation

1. Pre-Setup:

Libraries used in this model are

NumPy: It is used for performing mathematical calculations on multi-dimensional arrays.

Sklearn: Sklearn supports Regression, classification and clustering algorithms. It also supports SVM (Support Vector Machine).

Matplotlib: Matplotlib is used for plotting mathematical values in forms of graphs.

TensorFlow: TensorFlow is an open source machine learning library developed by Google. It supports C++, Python and Cuda

Flatten: Flatten is mainly used to convert a different size grid into a straight one.

Maxpoo1d: It reduces the dimensionality of an input by providing maximum value in each patch of feature map.

Linear layer: linear layer has a capability to learn correlation between input and output.

RELU: RELU (rectified linear unit layer) acts as activation function makes sure nonlinearity and fully connected layer. The main function of RELU is converting non-linear model into linear model.

SoftMax: SoftMax is an activation function. It maps an output to range from [0,1] and sum of values in output is equal to one. Therefore, output of SoftMax is probability distribution. In logistic regression model, SoftMax is used for Multi-classification.

2. Dataset Pre-Processing

The dataset used in this model contains totally 1500 CT scans of various parts of body such as abdomen, chest, and head. Firstly, we load the dataset for all three class labels using python list directory function. After loading the dataset, Open CV function such as cv2.resize is used to resize the image from 64*64 to 32*32. After that dataset is separated into features and labels. The feature dataset is converted into NumPy array using np array function. All the values in the dataset is converted to zero's and ones using feature scaling.

Data Split

The dataset is spilt into 80% for training and 20% for testing using sklearn train test split function.

3. Model Creation

In this step CNN model is created. The CNN model is built using keras. First, we build sequential model, then we add all the layers to it. The CNN model contains convolution layers,

maxpooling2D, flatten layer, dense layer, activation function layer. Here we are using SoftMax activation function. The CNN model contains 2 convolution layers, 1 flatten layer, and 3 dense layers. We extract the features from intermediate dense layer and covert them into NumPy array.

KNN Model

First, we import all the required libraries for the model. We load the KNeighborsClassifier from sklearn neighbours and cross-val-score from sklearn model selection. The KNN algorithm is applied to the extracted features from CNN model. Here we train the knn model for different k values such as (3,5,7,9) and determine the accuracy using cross validation score.

K=3

Accuracy	Precision	Recall	F-1 Score
1.00	1.00	0.99	1.00

K=5

Accuracy	Precision	Recall	F-1 Score
0.99	0.99	0.98	0.98

K=7

Accuracy	Precision	Recall	F-1 Score
0.993	0.993	0.994	0.993

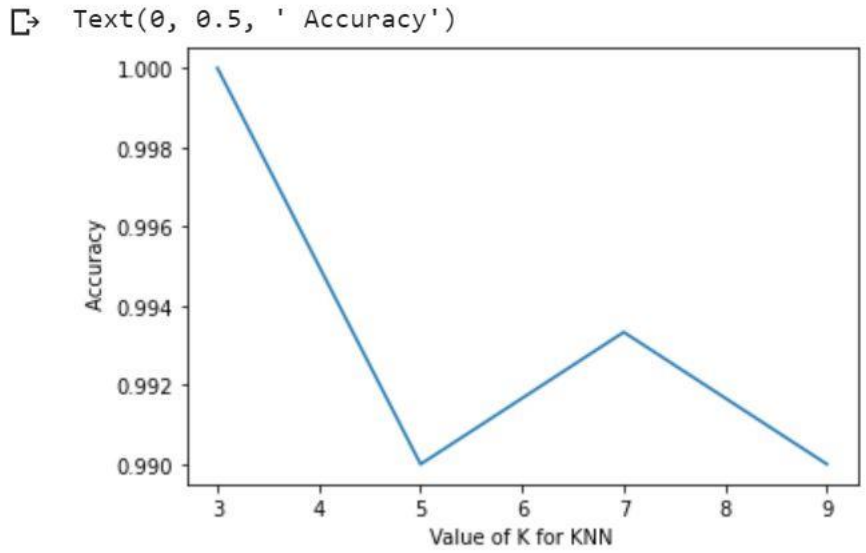
K=9

Accuracy	Precision	Recall	F-1 Score
0.990	0.991	0.990	0.990

At k=3, the model generated an accuracy = 0.993 which is better than all other k values. So, at k=3 is the optimal model.

Graph

K value v/s accuracy



Overfitting/Underfitting: In Overfitting model performs well in training but not performs good at testing time. In KNN algorithm, when k value is small, it leads to overfitting and when k value increases or k value is high it leads to underfitting. So, we need to choose k value such that it should neither be overfitting or underfitting. At k=9, the KNN model results underfitting and at k=2, it results underfitting. So, I choose k=3, it fits the model.

Random Forest Model

First, I imported all the required libraries. I loaded Randomforestclassifier from sklearn and created base model for the extracted features from CNN model.

Base Model

In base model, the hyper parameters used are n_estimators = 20 and max_depth = 2, mini_samples_split = 9

The model generated an **accuracy** = 0.946

Accuracy	Precision	Recall	F-1 Score
0.946	0.92	0.94	0.95

Performed random search using Randomizedserach to tune the hyperparameters.

Hyper Parameters

The Hyperparameters tuned in this model are n_estimators, mini_samples_split and max_depth

Here I an array of n_estimators and max_depth is taken for random search

max_depth = [2,4,6,8,10]

n_estimators=[10,20,30,40,50]

mini_samples_split=[2,5,7,9,10]

After perfoming randomized serach it results best parameters

The best parameters are max_depth = 8, mini_samples_split = 9, and n_estimators=20

Model with best parameters results

The random forest model by substituting best parameters results an **accuracy = 0.999**

Accuracy	Precision	Recall	F-1 Score
0.999	0.99	0.99	0.99