LAB 07

NAME: Yana Kaveramma A A

ROLL NO: 21BDA32

1.Have you come across Grid Search Cross Validation? Fit any two models covered in previous classes and optimize them using Grid search CV.

Yes, by using cross validation and grid search a more meaningful result can be achieved when compared to our original train/test split with minimal tuning. Cross validation is a very important method used to create better fitting models by training and testing on all parts of the training dataset.

Dataset used: Diabetes Prediction in PIMA Women

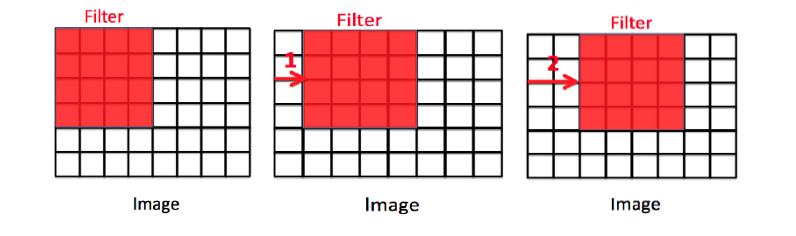
Models fit: Decision Tree Classifier and Random Forest Classifier

Result: The accuracy is reduced in decision tree after applying grid search cv (from 0.70 to 0.61) whereas it has increased by.004 in random forest (from 0.763 to 0.767)

2.What is Stride, Padding & Pooling? Explain with an example.

STRIDE:

When the array is created, the pixels are shifted over to the input matrix. The number of pixels turning to the input matrix is known as the strides. When the number of strides is 1, we move the filters to 1 pixel at a time. Similarly, when the number of strides is 2, we carry the filters to 2 pixels, and so on. They are essential because they control the convolution of the filter against the input, i.e., Strides are responsible for regulating the features that could be missed while flattening the image. They denote the number of steps we are moving in each convolution. The following figure shows how the convolution would work.

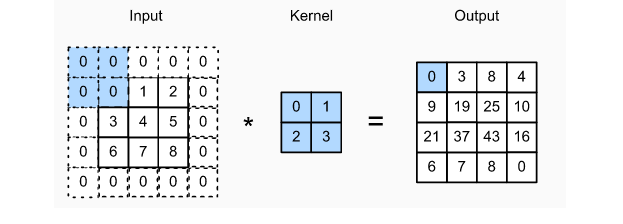


PADDING:

The padding plays a vital role in creating CNN. After the convolution operation, the original size of the image is shrunk. Also, in the image classification task, there are multiple convolution layers after which our original image is shrunk after every step, which we don’t want.

Secondly, when the kernel moves over the original image, it passes through the middle layer more times than the edge layers, due to which there occurs an overlap.

To overcome this problem, a new concept was introduced named padding. It is an additional layer that can add to the borders of an image while preserving the size of the original picture. For example:

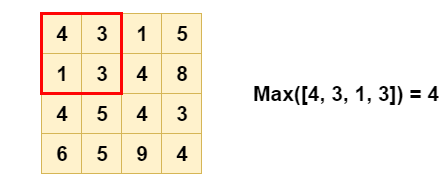


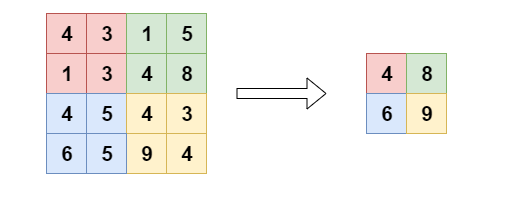
POOLING:

The pooling layer is another building block of a CNN and plays a vital role in pre-processing an image. In the pre-process, the image size shrinks by reducing the number of parameters if the image is too large. When the picture is shrunk, the pixel density is also reduced, the downscaled image is obtained from the previous layers. Basically, its function is to progressively reduce the spatial size of the image to reduce the network complexity and computational cost. Spatial pooling is also known as down sampling or subsampling that reduces the dimensionality of each map but retains the essential features. A rectified linear activation function, or ReLU, is applied to each value in the feature map. Relu is a simple and effective nonlinearity that does not change the values in the feature map but is present because later subsequent pooling layers are added. Pooling is added after the nonlinearity is applied to the feature maps. There are three types of spatial pooling:

1. Max Pooling

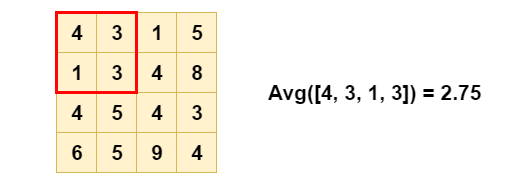
Max pooling is a rule to take the maximum of a region and help to proceed with the most crucial features from the image. It is a sample-based process that transfers continuous functions into discrete counterparts. Its primary objective is to downscale an input by reducing its dimensionality and making assumptions about features contained in the sub-region that were rejected.

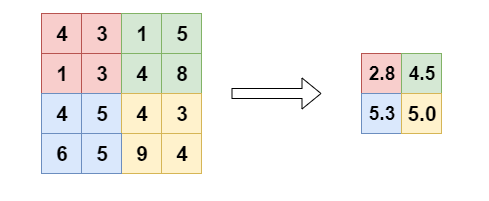




2. Average Pooling

It is different from Max Pooling; it retains information about the lesser essential features. It simply downscales by dividing the input matrix into rectangular regions and calculating the average values of each area.



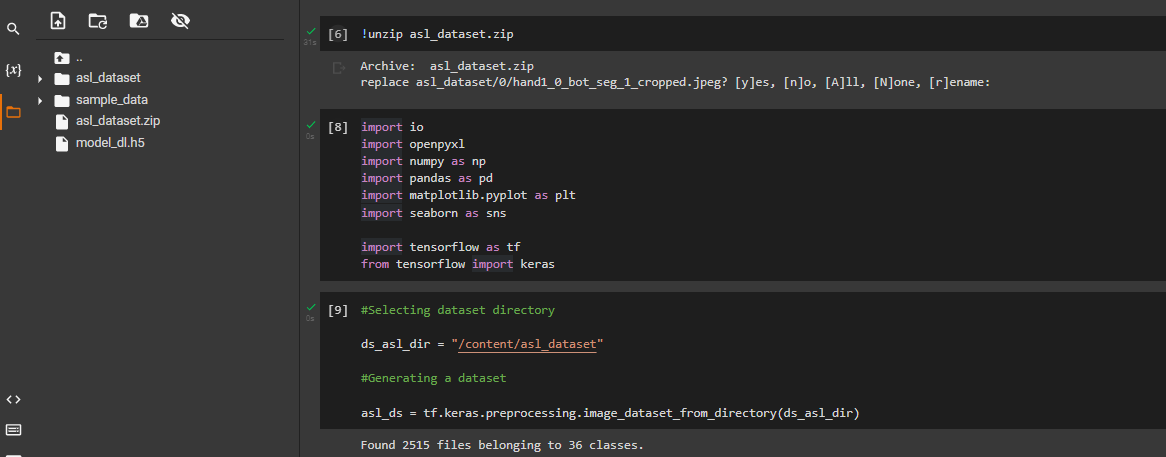


### 3. Sum Pooling

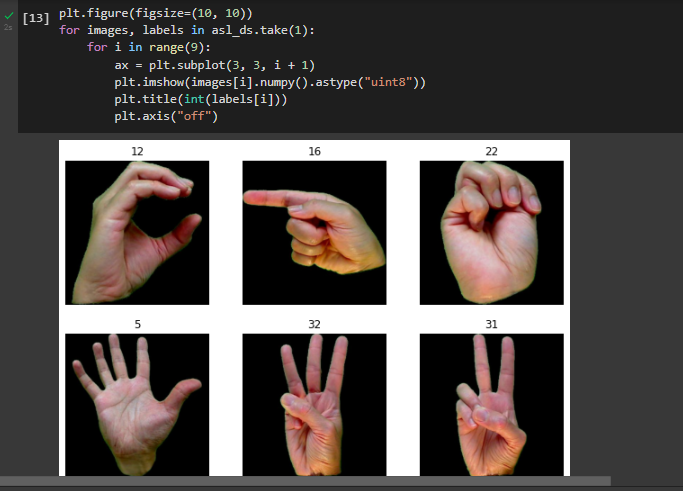
It is similar to Max pooling, but instead of calculating the maximum value, we calculate the mean of each sub-region.

3. Fit a CNN model on the dataset which has been assigned to you. Print a classification report to see the model metrics on train and test datasets.

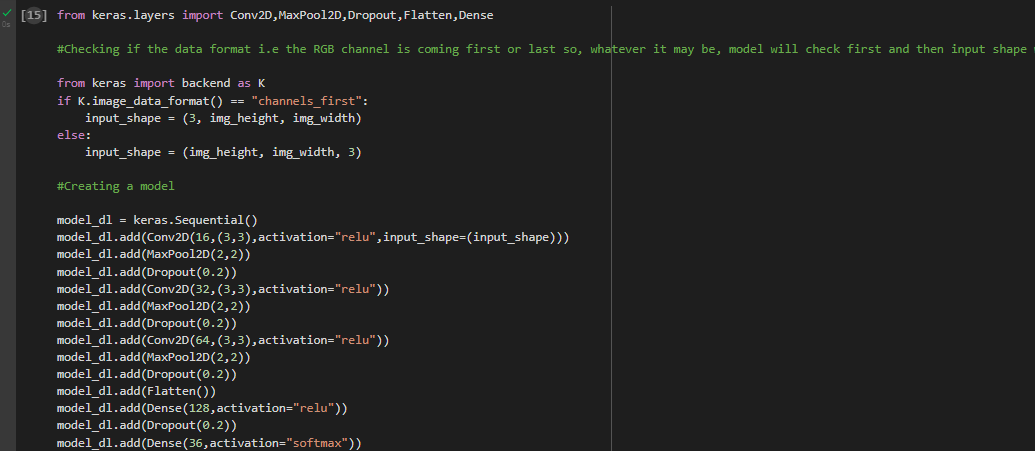
(Worked on google collab)

1. Importing zip folder, then unzipping the folder. Loading libraries, generating dataset. , 

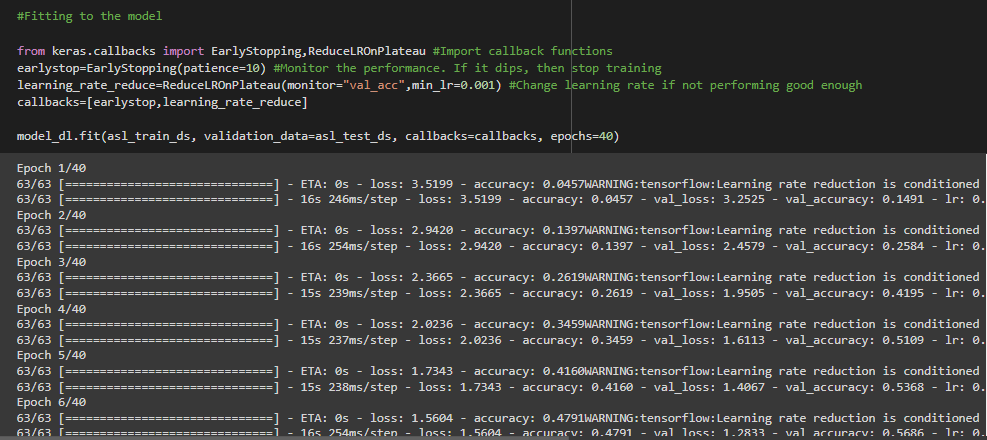
2. Displaying sample data



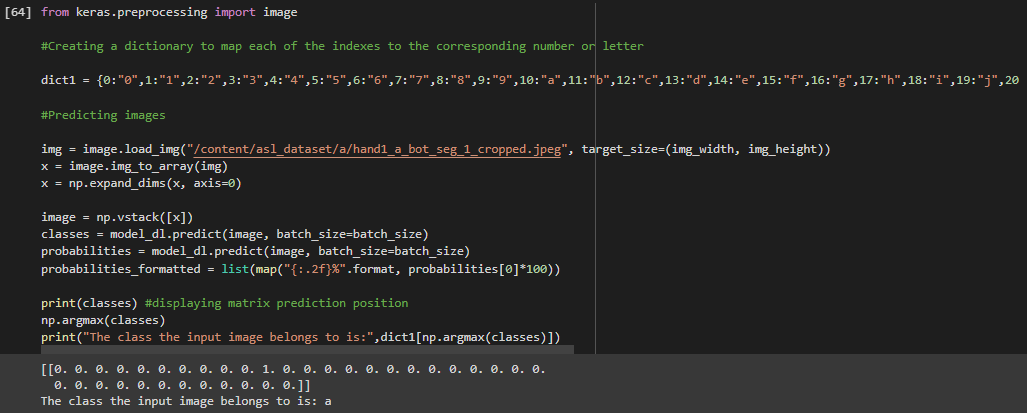
1. Creating a model



1. Compiling neural network



5.Predicting images



4.What is overfitting? How to overcome overfitting in an ML model?

Overfitting means that the machine learning model trains too specifically on our training dataset and causes higher levels of error when applied to our test/holdout datasets.

Measures to overcome overfitting in an ML model:

1. Hold-out (data)

Rather than using all of our data for training, we can simply split our dataset into two sets: training and testing. A common split ratio is 80% for training and 20% for testing. We train our model until it performs well not only on the training set but also for the testing set. This indicates good generalization capability since the testing set represents unseen data that were not used for training. However, this approach would require a sufficiently large dataset to train on even after splitting.

2. Cross-validation (data)

We can split our dataset into k groups (k-fold cross-validation). We let one of the groups to be the testing set and the others as the training set, and repeat this process until each individual group has been used as the testing set (e.g., k repeats). Unlike hold-out, cross-validation allows all data to be eventually used for training but is also more computationally expensive than hold-out.

3. Data augmentation (data)

A larger dataset would reduce overfitting. If we cannot gather more data and are constrained to the data we have in our current dataset, we can apply data augmentation to artificially increase the size of our dataset. For example, if we are training for an image classification task, we can perform various image transformations to our image dataset (e.g., flipping, rotating, rescaling, shifting).

4. Feature selection (data)

If we have only a limited amount of training samples, each with a large number of features, we should only select the most important features for training so that our model doesn’t need to learn for so many features and eventually overfit. We can simply test out different features, train individual models for these features, and evaluate generalization capabilities, or use one of the various widely used feature selection methods.

5. L1 / L2 regularization (learning algorithm)

Regularization is a technique to constrain our network from learning a model that is too complex, which may therefore overfit. In L1 or L2 regularization, we can add a penalty term on the cost function to push the estimated coefficients towards zero (and not take more extreme values). L2 regularization allows weights to decay towards zero but not to zero, while L1 regularization allows weights to decay to zero.

6. Remove layers / number of units per layer (model)

As mentioned in L1 or L2 regularization, an over-complex model may more likely overfit. Therefore, we can directly reduce the model’s complexity by removing layers and reduce the size of our model. We may further reduce complexity by decreasing the number of neurons in the fully-connected layers. We should have a model with a complexity that sufficiently balances between underfitting and overfitting for our task.

7. Dropout (model)

By applying dropout, which is a form of regularization, to our layers, we ignore a subset of units of our network with a set probability. Using dropout, we can reduce interdependent learning among units, which may have led to overfitting. However, with dropout, we would need more epochs for our model to converge.

8. Early stopping (model)

We can first train our model for an arbitrarily large number of epochs and plot the validation loss graph (e.g., using hold-out). Once the validation loss begins to degrade (e.g., stops decreasing but rather begins increasing), we stop the training and save the current model. We can implement this either by monitoring the loss graph or set an early stopping trigger. The saved model would be the optimal model for generalization among different training epoch values.