* **Project Description:**

In this model we have to find if the patient whose facial image we submit have stroke or not. We use a CNN model with multiple Convolutional and Maxpooling layer everything is done using keras to build a model.

* **Problem formulation:**

The dataset is divided into two classes which are:

1)Stroke 2) Non-Stroke

* **Dataset:**

The dataset contains 69 facial images out of which 21 are strokes patient image while the rest of the are non-stroke images. Link below for the used dataset in model:

<https://drive.google.com/drive/folders/1Bc6E_ELJ6qcvLhL5FBrxsCIaPCd_FVVt?usp=share_link>

* **Hyperparameter tuning**

A model hyperparameter is a characteristic of a model that is external to the model and whose value cannot be estimated from data. The value of the hyperparameter must be set before the learning process begins.

* Learning rate ῃ
* epochs

Run your neural network for tuning Learning rate and epochs using train test split validation. Fill in *Table 3.* Use the epoch and the learning rate with the best accuracy for further experimentation of the model.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| epochs | 10 | 25 | 50 | 100 | 500 | 1000 | 5000 |
| ῃ = 0.1 | 67.7% | 67.7% | 67.9% | 68.05% | 70.93% | 69.4% | 68.1% |
| ῃ = 0.01 | 67.8% | 68.3% | 73.2% | 69.5% | 72.3% | 77% | 62.1% |
| ῃ = 0.001 | 72.3% | 79.6% | 78.33% | 81.09% | 81.76% | 79.9% | 82.7% |

*Table 3.* Average Accuracy over folds

Plot train test graph for each of the learning rate and epoch value in Table 4.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| epochs | 10 | 25 | 50 | 100 | 500 | 1000 | 5000 |
| ῃ = 0.1 |  |  |  |  |  |  |  |
| ῃ = 0.01 |  |  |  |  |  |  |  |
| ῃ = 0.001 |  |  |  |  |  |  |  |

*Table 4. Train Test Learning Curves for each of the learning rate and epoch*

* **Cross validation:**

Use k fold cross validation procedure for reporting results in your dataset. The following code would help you split the data for cross validation.

Use the value k=3. Populate *Table 1* during experimentation.

|  |  |  |  |
| --- | --- | --- | --- |
| Epoch | First fold validation | Train Loss | Test Loss |
| 1 |  | 0.9763 | 0.5331 |
| 2 |  | 0.5399 | 0.4842 |
| 3 |  | 0.3875 | 0.5231 |
| 4 |  | 0.3498 | 0.6097 |
| 5 |  | 0.1592 | 0.6709 |
| 6 |  | 0.1766 | 0.3806 |
| 7 |  | 0.0636 | 0.5256 |
|  | Second fold validation | Train Loss | Test Loss |
| 1 |  | 0.9096 | 0.7903 |
| 2 |  | 0.8074 | 0.571 |
| 3 |  | 0.4408 | 0.3736 |
| 4 |  | 0.3593 | 0.5491 |
| 5 |  | 0.2608 | 0.4494 |
| 6 |  | 0.1906 | 0.358 |
| 7 |  | 0.0683 | 0.3719 |
|  | Third fold validation | Train Loss | Test Loss |
| 1 |  | 0.7348 | 0.7821 |
| 2 |  | 0.5295 | 0.6566 |
| 3 |  | 0.3348 | 0.5599 |
| 4 |  | 0.2315 | 0.5361 |
| 5 |  | 0.2068 | 0.4945 |
| 6 |  | 0.0853 | 0.6486 |
| 7 |  | 0.0715 | 0.4968 |

*Table 1: Train and Test error for each epoch in each fold*

* **Generate the Train Test Error Learning Curve** **5+5 points]**

Please note that you would take the average of errors obtained in each fold of validation (yellow row in *Table 1*) in each epoch computed in *Table 1* and plot the learning curve like *Figure 1*. Provide the legend and label the axis properly. Each curve carries 5 points.



*Figure 1: Average Train error/Average Validation Error vs epochs*



* **Metrics to use for model evaluation:**

For a balanced dataset and multiclass, you must use accuracy, F1 score, precision, recall, ROC AUC for the developed deep learning model evaluation. After each cross-validation fold, compute all the metrics above and fill in the values in *Table 2*.

Learning rate= 0.001, Batch size=1, fold=3, epoch=100

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train Set | Accuracy | F1 score | precision | recall | ROC AUC |
| Fold 1 | 78.2% | 0.7059 | 0.6 | 0.857 | 0.804 |
| Fold 2 | 82.6% | 0.7413 | 0.625 | 0.834 | 0.828 |
| Fold 3 | 79.21% | 0.615 | 1 | 0.444 | 0.722 |
| Average | 80.03% | 0.6874 | 0.742 | 0.712 | 0.785 |

*Table 2: Train set/Validation set Accuracy, F1 score, Precision, Recall, ROC AUC for each fold*