**Gesture Recognition**

**Project & Model Report**

**Neural Network Architecture development and training:**

* We started by building a basic CNN+RNN model without using any dropouts. GRU was selected as it has fewer parameters than LSTM, as it lacks an output gate.
* We experimented with different models by using different model configurations and hyperparameters such as introducing dropouts, using L2 regularization in combination with different optimizers.
* We also experimented with SGD() and Adam() optimizers but finally used Adam() as it lead to improvement in model’s accuracy.
* Due to the limited computational capacity we could not experiment with other optimizers such as Adagrad and Adadelta as these take a lot of time to run.
* We also played around with different learning rates and ReduceLROnPlateau was used to decrease the learning rate .
* To overcome the issue of overfitting we used Batch Normalization, pooling and dropout layers.
* Then we used Conv3D and experimented with different model configurations to arrive at the final model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Model Description** | **Accuracy** | **Result** | **Decision & Explanation** |
| 1 | Conv2D layers + GRU Layer+ Adam optimizer + No dropout | **TRAINING:**  **VALIDATION**: | Overfitting | Training accuracy was high. So, we introduced dropouts |
| **ADDING MORE LAYERS** | | | | |
| 2 | Conv2D layers + GRU Layer+ Adam optimiser + Dropout | **TRAINING:**  **VALIDATION**: | The overfitting slightly reduced but was still high | We thought of employing L2 regularization in Model 2 in next ste |
| 3 | Conv2D layers + GRU Layer+ Adam optimiser + dropout + L2 regularization | **TRAINING:**  **VALIDATION**: | Val Accuracy Dropped | Still overfitting. We need to change the optimizer to SDG for the next steps |
| 4 | Conv2D layers + GRU Layer+ SDG optimiser + dropout + L2 regularization | **TRAINING:**  **VALIDATION**: | Overfitting problem resolved to some extent; however training / validation accuracy is quite low | We reverted the SDG optimizer as it reduced the accuracy on training / validation data and continued experimenting with Adam optimizer |
| **CHANGING FILTER** | | | | |
| 5 | Conv3D layers + Dropout + Batch Training .59 Validation The accuracy on train and validation data becomes almost As the Accuracy got reduced, we experimented Normalization + L2 Regularization | **TRAINING:**  **VALIDATION**: | The accuracy on train and validation data becomes almost equal, but its seems to be underfitting | As the Accuracy got reduced, we experimented with Transfer Learning in next step |
| **TRANSFER LEARNING** | | | | |
| 6 | Transfer Learning (Resnet50) + dropout + GRU Layer | **TRAINING:**  **VALIDATION**: | Number of trainable parameters became quite high (25,126,46 9) | Let Experiment with Another NN : VGGNE T |
| 7 | Transfer Learning (VGGNET) + dropout + GRU Layer | **TRAINING:**  **VALIDATION**: | Underfitting observed | Let Experiment with Another NN : mobilenet |
| 8 | Transfer Learning (mobilenet) + dropout + GRU Layer | **TRAINING:**  **VALIDATION**: | High Accuracy both on train / validation data | Accuracy Improved with MobileNet Arch, as its has lightweight design & high speed performance as compared to other used Arch. |
| **INCREASE BATCH SIZE** | | | | |
| 9 | Transfer Learning (mobilenet) + dropout + GRU Layer + Increased epoch (30 ) | **TRAINING:**  **VALIDATION**: | Increasing epoch does not increased validation accuracy. | Let experiment with increasing batch size |
| 10 | Transfer Learning (mobilenet) + dropout + GRU Layer + Increased epoch + Increased Batch Size(20) | **TRAINING:**  **VALIDATION**: | High Accuracy on Training / validation Data | MODEL SELECTED |
| **INCREASE BATCH SIZE** | | | | |
| 11 | Transfer Learning (mobilenet) + dropout + GRU Layer + Increased Batch Size(25) | **TRAINING:**  **VALIDATION**: | High Accuracy on Training / validation Data | Still Model 10, seems to be better. |

**Observations:**

* The training time increased in proportion with the number of trainable parameters.
* A large batch size value was giving GPU Out of memory error. So we started with a batch size of 15 and gradually moved till batch size of 25.
* A small batch helped reducing training time but it had a trade off on model accuracy.
* Transfer learning boosted the overall accuracy of the model. We made use of the Resnet50 ,VGGNET and MobileNet Architecture .

We took the model with MobileNet architecture as the final model due to its light weight design and high speed performance coupled with low maintenance as compared to other well-known architectures like VGG16, AlexNet, GoogleNet etc.

**Result**: Model 10 -- Transfer Learning + mobilenet + dropout + GRU Layer + Increased epoch + Increased Batch Size(20), which performed well.

**Reason**:

1. (Training Accuracy: 98%, Validation Accuracy: 96%)

2. Number of Parameters (3,420,549) less according to other models’ performance