

Report

I initially started off by using simple neural networks with many dense layers & each containing many neurons. I used ReLU on all the first layers and softmax on the last layer(I used linear and then set logits to true). I also used the adam optimizer. But that didn't work out and predicted that all the images in the test set were happy. It didn't work on the train set either(it again predicted all the inputs in the training set were happy), so it wasn't a case of overfitting. I then proceeded to try augmenting the data using opencv and increased the brightness, contrast etc of the training set. But there was no improvement. I also tried playing around with the model framework by adding more layers and more neurons etc, but that was also not doing anything to the accuracy.

Then I read a bit about Convolutional neural networks and tried to implement that. I used techniques like batch normalization, dropout, maxpooling 2d etc to improve the model. I used the RMSprop instead of Adam here. I then ran the code and saw a significant rise in accuracy. I tried changing the epochs to 4 5 and 6 and saw that the optimum result was given by epochs=5. I then also split the dataset into train and validation set to ensure that the data wouldn't overfit. This resulted in my final model.