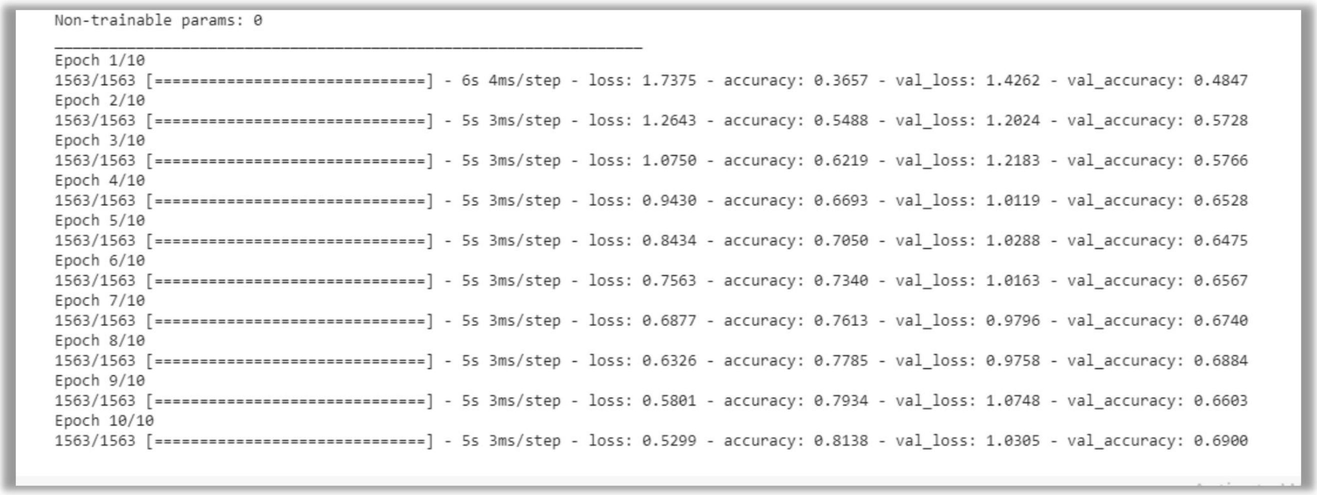
Samar Ibrahim Antar

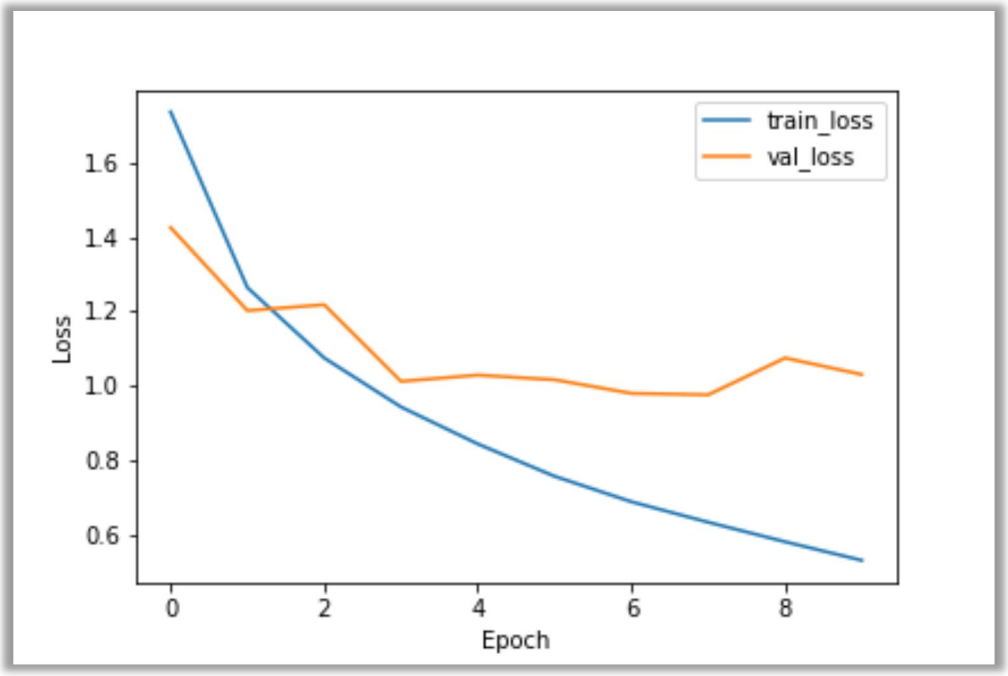
Homework#4: Convolution Neural Networks

- **<u>the learning and testing error of point(a) with relu at all layers except softmax in last layer :</u>**

✓ loss: 1.0305 - accuracy: 0.6900

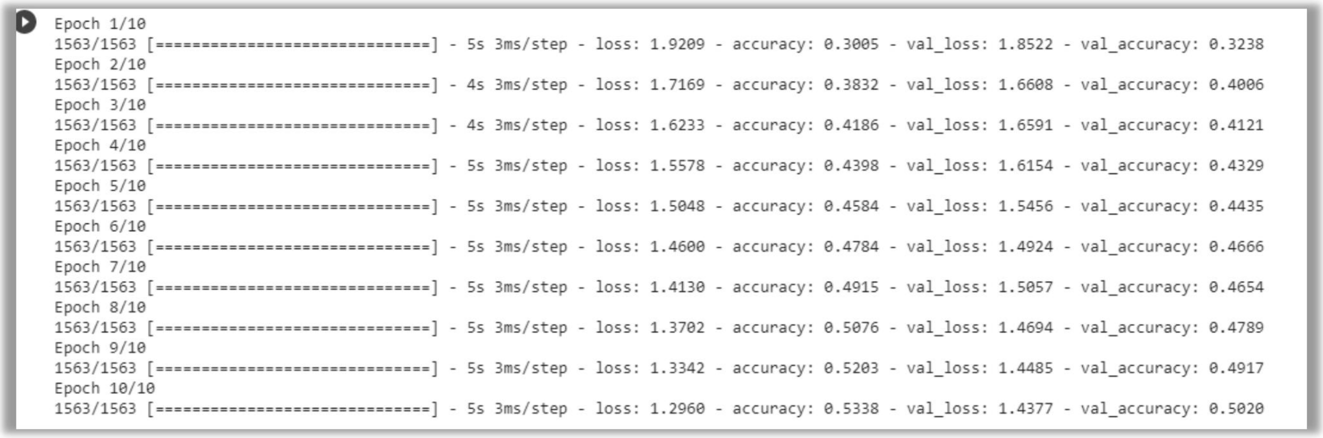


```
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.7375 - accuracy: 0.3657 - val_loss: 1.4262 - val_accuracy: 0.4847
Epoch 2/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.2643 - accuracy: 0.5488 - val_loss: 1.2024 - val_accuracy: 0.5728
Epoch 3/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.0750 - accuracy: 0.6219 - val_loss: 1.2183 - val_accuracy: 0.5766
Epoch 4/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.9430 - accuracy: 0.6693 - val_loss: 1.0119 - val_accuracy: 0.6528
Epoch 5/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.8434 - accuracy: 0.7050 - val_loss: 1.0288 - val_accuracy: 0.6475
Epoch 6/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.7563 - accuracy: 0.7340 - val_loss: 1.0163 - val_accuracy: 0.6567
Epoch 7/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.6877 - accuracy: 0.7613 - val_loss: 0.9796 - val_accuracy: 0.6740
Epoch 8/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.6326 - accuracy: 0.7785 - val_loss: 0.9758 - val_accuracy: 0.6884
Epoch 9/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.5801 - accuracy: 0.7934 - val_loss: 1.0748 - val_accuracy: 0.6603
Epoch 10/10
1563/1563 [==============================] - 5s 3ms/step - loss: 0.5299 - accuracy: 0.8138 - val_loss: 1.0305 - val_accuracy: 0.6900
```

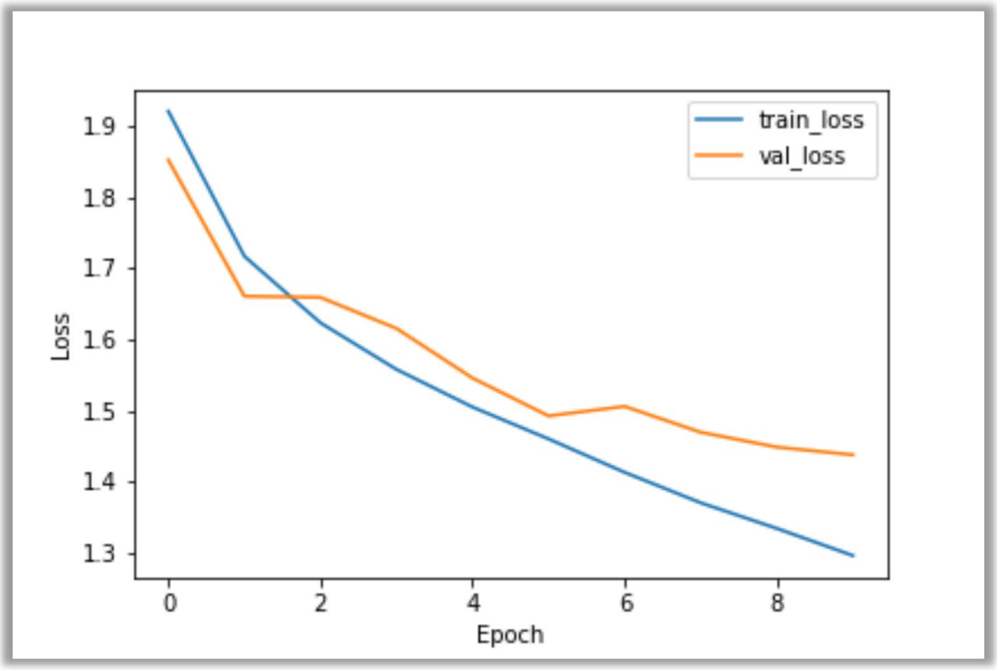


- **<u>the learning and testing error of point(b) with relu at all layers except softmax in last layer :</u>**

✓ loss: 1.4377 - accuracy: 0.5020



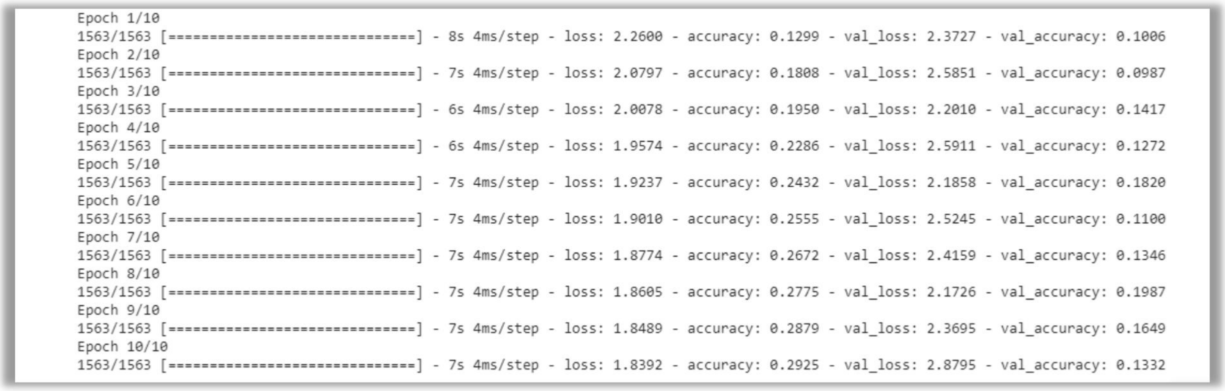
```
Epoch 1/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.9209 - accuracy: 0.3005 - val_loss: 1.8522 - val_accuracy: 0.3238
Epoch 2/10
1563/1563 [==============================] - 4s 3ms/step - loss: 1.7169 - accuracy: 0.3832 - val_loss: 1.6608 - val_accuracy: 0.4006
Epoch 3/10
1563/1563 [==============================] - 4s 3ms/step - loss: 1.6233 - accuracy: 0.4186 - val_loss: 1.6591 - val_accuracy: 0.4121
Epoch 4/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.5578 - accuracy: 0.4398 - val_loss: 1.6154 - val_accuracy: 0.4329
Epoch 5/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.5048 - accuracy: 0.4584 - val_loss: 1.5456 - val_accuracy: 0.4435
Epoch 6/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.4600 - accuracy: 0.4784 - val_loss: 1.4924 - val_accuracy: 0.4666
Epoch 7/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.4130 - accuracy: 0.4915 - val_loss: 1.5057 - val_accuracy: 0.4654
Epoch 8/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.3702 - accuracy: 0.5076 - val_loss: 1.4694 - val_accuracy: 0.4789
Epoch 9/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.3342 - accuracy: 0.5203 - val_loss: 1.4485 - val_accuracy: 0.4917
Epoch 10/10
1563/1563 [==============================] - 5s 3ms/step - loss: 1.2960 - accuracy: 0.5338 - val_loss: 1.4377 - val_accuracy: 0.5020
```



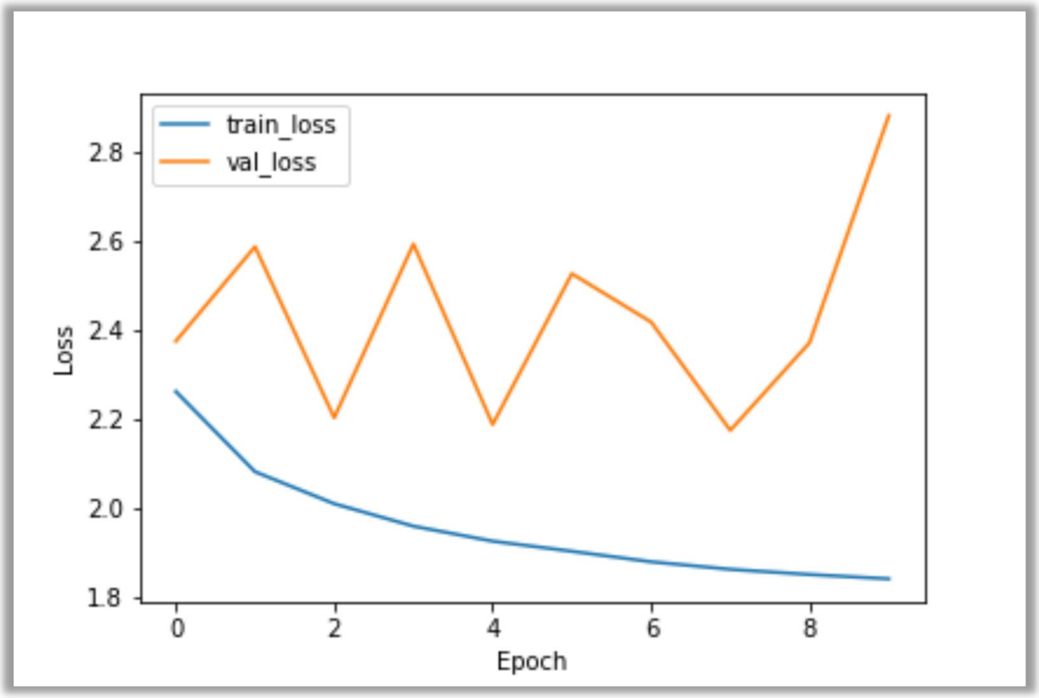
- **<u>the learning and testing error of point(c) with relu at all layers except softmax in last layer :</u>**

✓ Dropout rate(0.5) after every convolution layer and after every Dense layer

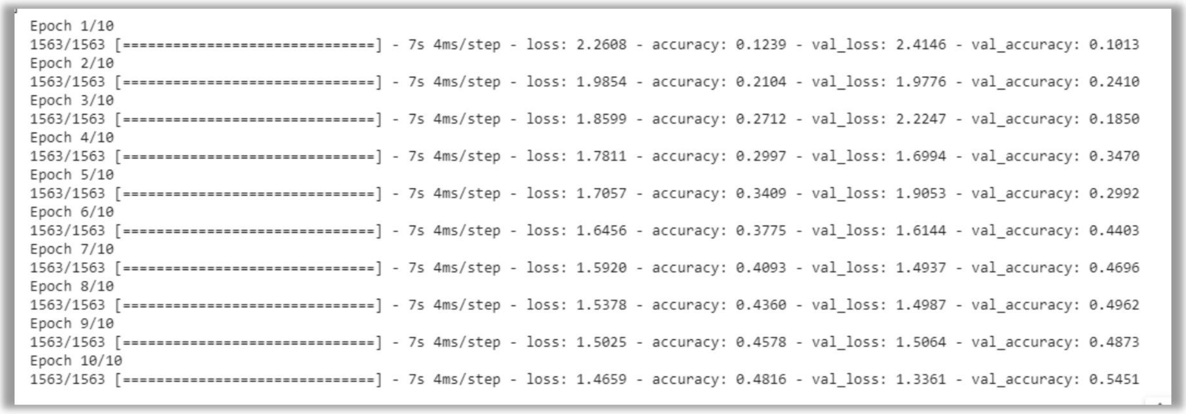
✓ loss: 2.8795 - accuracy: 0.1332



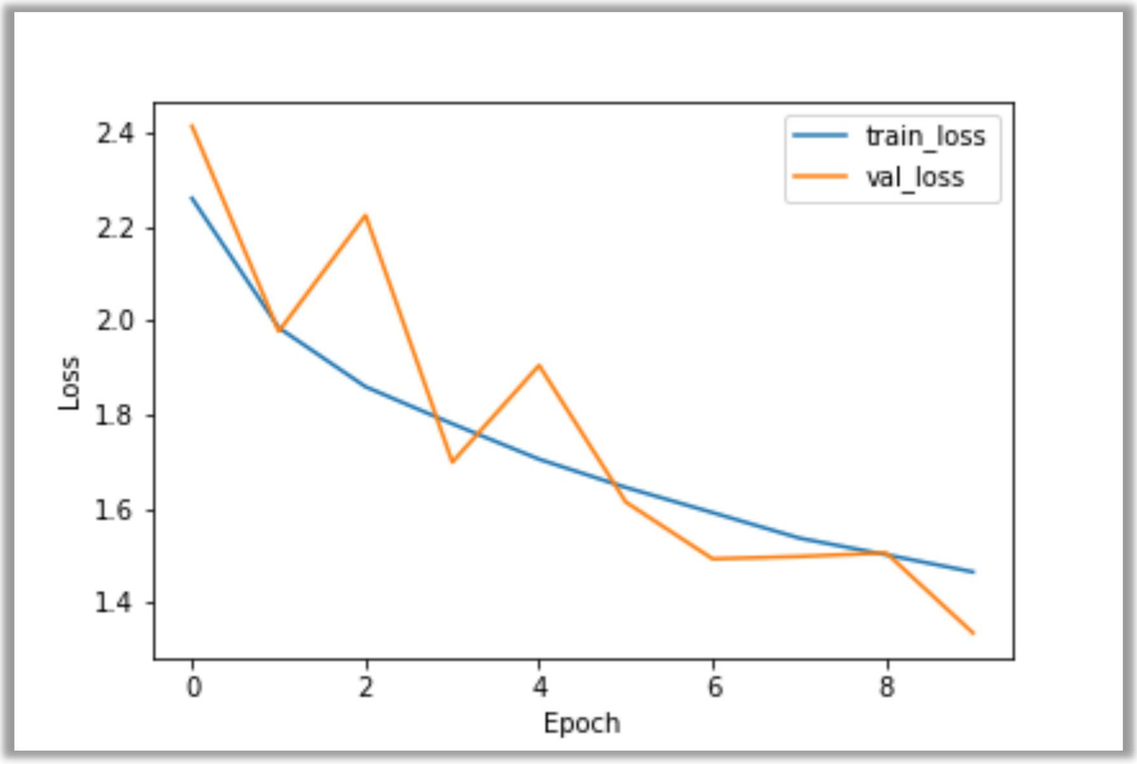
```
Epoch 1/10
1563/1563 [==============================] - 8s 4ms/step - loss: 2.2600 - accuracy: 0.1299 - val_loss: 2.3727 - val_accuracy: 0.1006
Epoch 2/10
1563/1563 [==============================] - 7s 4ms/step - loss: 2.0797 - accuracy: 0.1808 - val_loss: 2.5851 - val_accuracy: 0.0987
Epoch 3/10
1563/1563 [==============================] - 6s 4ms/step - loss: 2.0078 - accuracy: 0.1950 - val_loss: 2.2010 - val_accuracy: 0.1417
Epoch 4/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.9574 - accuracy: 0.2286 - val_loss: 2.5911 - val_accuracy: 0.1272
Epoch 5/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.9237 - accuracy: 0.2432 - val_loss: 2.1858 - val_accuracy: 0.1820
Epoch 6/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.9010 - accuracy: 0.2555 - val_loss: 2.5245 - val_accuracy: 0.1100
Epoch 7/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.8774 - accuracy: 0.2672 - val_loss: 2.4159 - val_accuracy: 0.1346
Epoch 8/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.8605 - accuracy: 0.2775 - val_loss: 2.1726 - val_accuracy: 0.1987
Epoch 9/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.8489 - accuracy: 0.2879 - val_loss: 2.3695 - val_accuracy: 0.1649
Epoch 10/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.8392 - accuracy: 0.2925 - val_loss: 2.8795 - val_accuracy: 0.1332
```



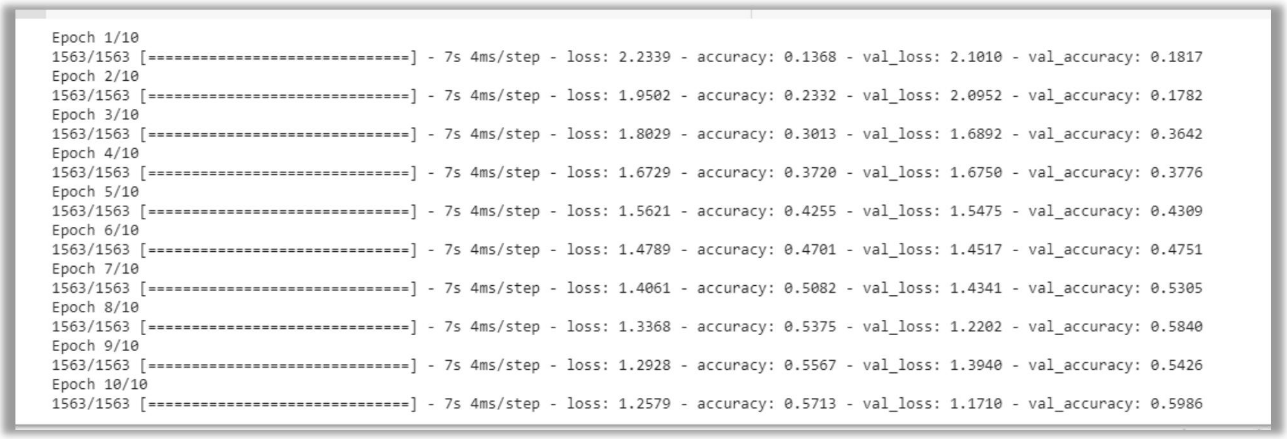
- ✓ Dropout rate(0.5) after only last convolution layer and after every Dense layer
- ✓ loss: 1.3361 - accuracy: 0.5451



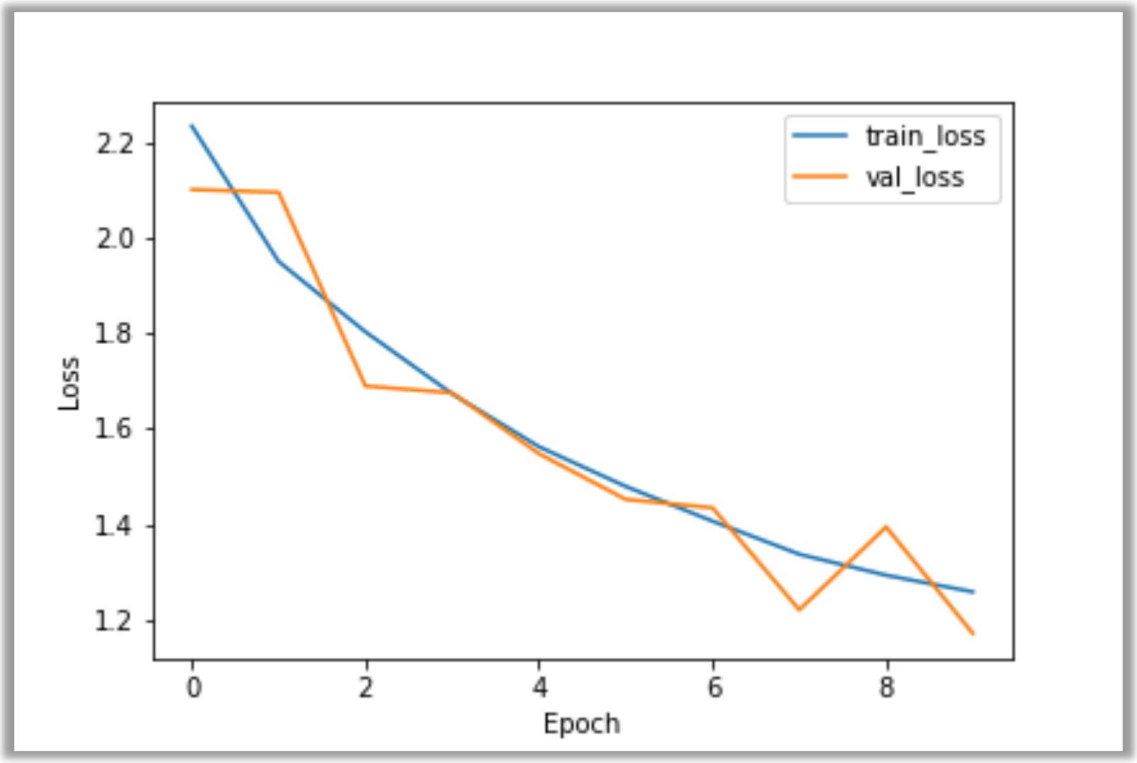
```
Epoch 1/10
1563/1563 [==============================] - 7s 4ms/step - loss: 2.2608 - accuracy: 0.1239 - val_loss: 2.4146 - val_accuracy: 0.1013
Epoch 2/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.9854 - accuracy: 0.2104 - val_loss: 1.9776 - val_accuracy: 0.2410
Epoch 3/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.8599 - accuracy: 0.2712 - val_loss: 2.2247 - val_accuracy: 0.1850
Epoch 4/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.7811 - accuracy: 0.2997 - val_loss: 1.6994 - val_accuracy: 0.3470
Epoch 5/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.7057 - accuracy: 0.3409 - val_loss: 1.9053 - val_accuracy: 0.2992
Epoch 6/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.6456 - accuracy: 0.3775 - val_loss: 1.6144 - val_accuracy: 0.4403
Epoch 7/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.5920 - accuracy: 0.4093 - val_loss: 1.4937 - val_accuracy: 0.4696
Epoch 8/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.5378 - accuracy: 0.4360 - val_loss: 1.4987 - val_accuracy: 0.4962
Epoch 9/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.5025 - accuracy: 0.4578 - val_loss: 1.5064 - val_accuracy: 0.4873
Epoch 10/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.4659 - accuracy: 0.4816 - val_loss: 1.3361 - val_accuracy: 0.5451
```



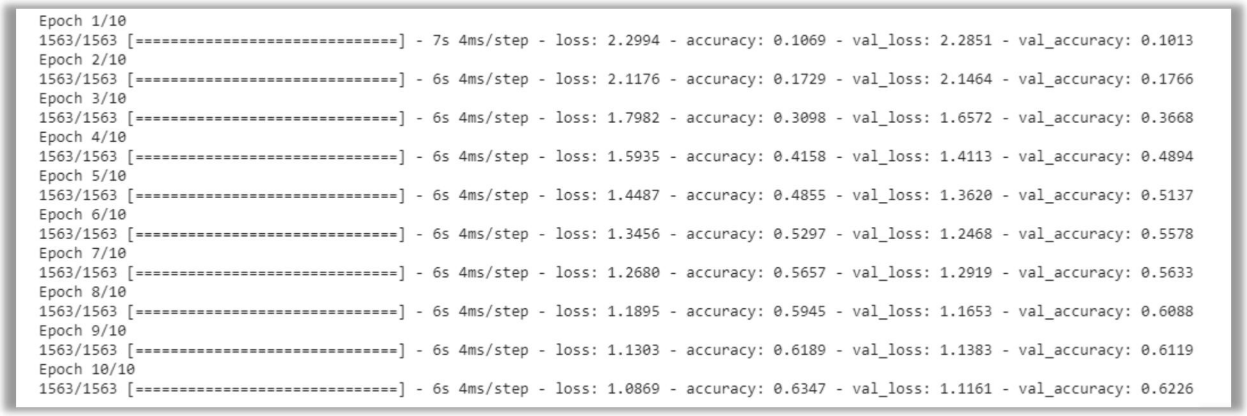
- ✓ Dropout rate(0.5) after only every Dense layer
- ✓ loss: 1.1710 - accuracy: 0.5986



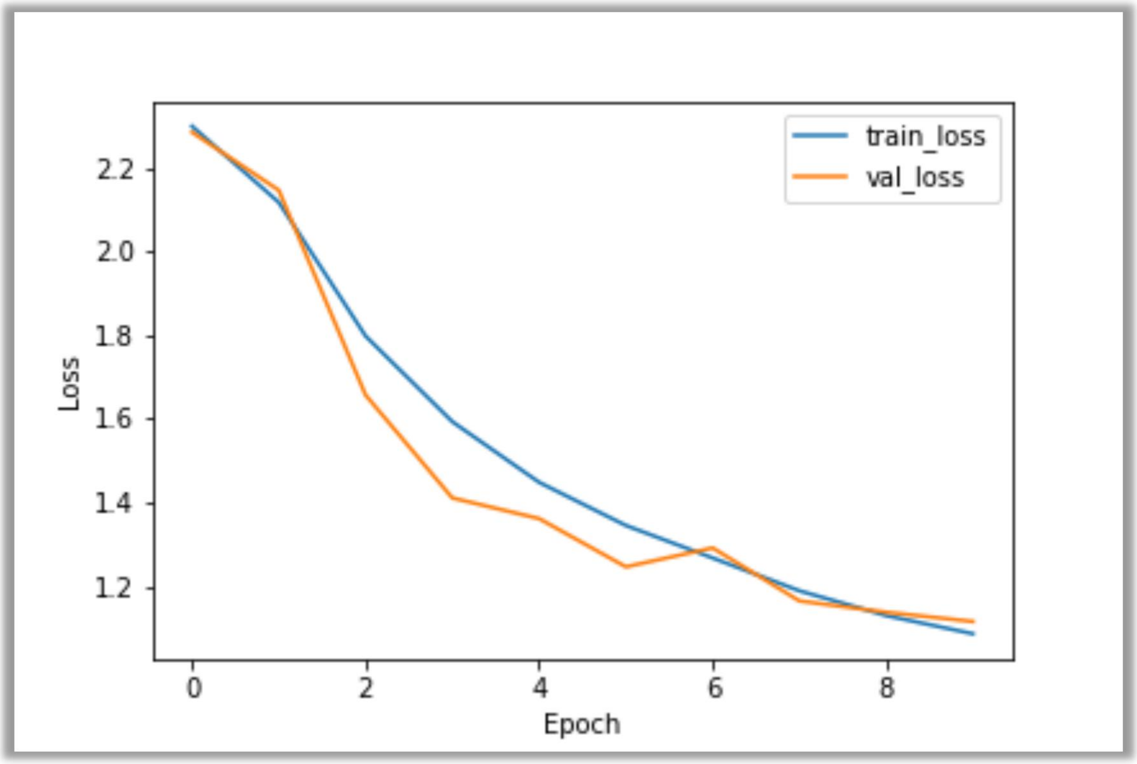
```
Epoch 1/10
1563/1563 [==============================] - 7s 4ms/step - loss: 2.2339 - accuracy: 0.1368 - val_loss: 2.1010 - val_accuracy: 0.1817
Epoch 2/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.9502 - accuracy: 0.2332 - val_loss: 2.0952 - val_accuracy: 0.1782
Epoch 3/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.8029 - accuracy: 0.3013 - val_loss: 1.6892 - val_accuracy: 0.3642
Epoch 4/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.6729 - accuracy: 0.3720 - val_loss: 1.6750 - val_accuracy: 0.3776
Epoch 5/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.5621 - accuracy: 0.4255 - val_loss: 1.5475 - val_accuracy: 0.4309
Epoch 6/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.4789 - accuracy: 0.4701 - val_loss: 1.4517 - val_accuracy: 0.4751
Epoch 7/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.4061 - accuracy: 0.5082 - val_loss: 1.4341 - val_accuracy: 0.5305
Epoch 8/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.3368 - accuracy: 0.5375 - val_loss: 1.2202 - val_accuracy: 0.5840
Epoch 9/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.2928 - accuracy: 0.5567 - val_loss: 1.3940 - val_accuracy: 0.5426
Epoch 10/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.2579 - accuracy: 0.5713 - val_loss: 1.1710 - val_accuracy: 0.5986
```



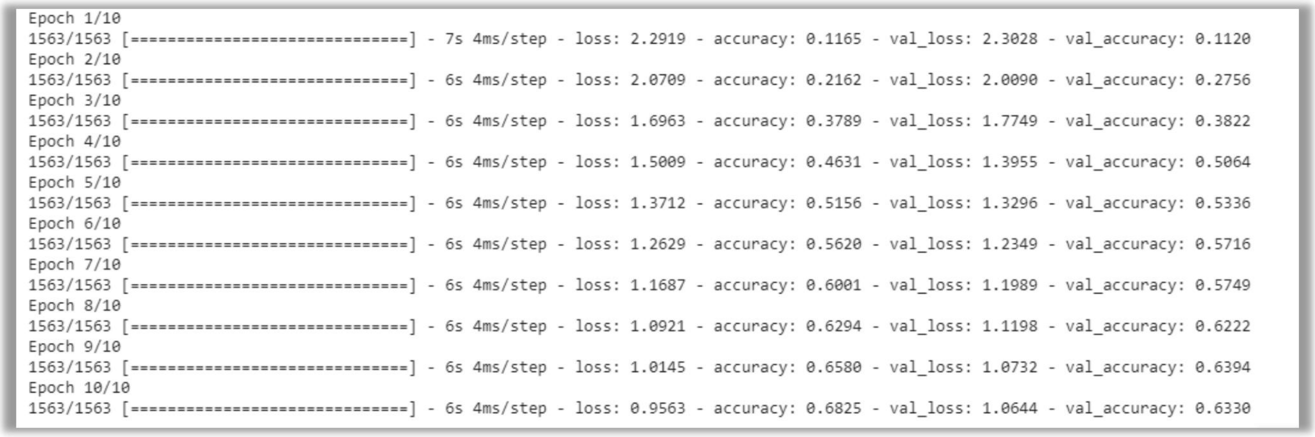
✓ Dropout rate(0.5) after only the last 4 Dense layers



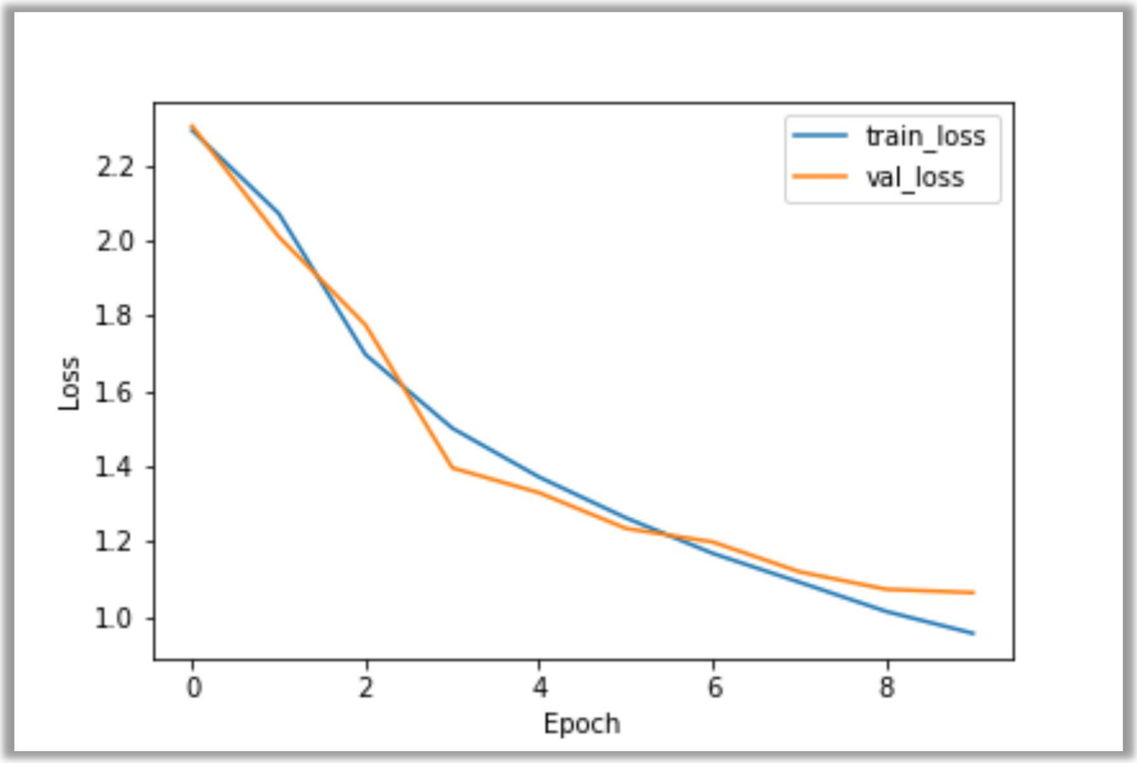
```
Epoch 1/10
1563/1563 [==============================] - 7s 4ms/step - loss: 2.2994 - accuracy: 0.1069 - val_loss: 2.2851 - val_accuracy: 0.1013
Epoch 2/10
1563/1563 [==============================] - 6s 4ms/step - loss: 2.1176 - accuracy: 0.1729 - val_loss: 2.1464 - val_accuracy: 0.1766
Epoch 3/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.7982 - accuracy: 0.3098 - val_loss: 1.6572 - val_accuracy: 0.3668
Epoch 4/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.5935 - accuracy: 0.4158 - val_loss: 1.4113 - val_accuracy: 0.4894
Epoch 5/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.4487 - accuracy: 0.4855 - val_loss: 1.3620 - val_accuracy: 0.5137
Epoch 6/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.3456 - accuracy: 0.5297 - val_loss: 1.2468 - val_accuracy: 0.5578
Epoch 7/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.2680 - accuracy: 0.5657 - val_loss: 1.2919 - val_accuracy: 0.5633
Epoch 8/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.1895 - accuracy: 0.5945 - val_loss: 1.1653 - val_accuracy: 0.6088
Epoch 9/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.1303 - accuracy: 0.6189 - val_loss: 1.1383 - val_accuracy: 0.6119
Epoch 10/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.0869 - accuracy: 0.6347 - val_loss: 1.1161 - val_accuracy: 0.6226
```



- ✓ Dropout rate(0.5) after only the last 3 Dense layers
- ✓ loss: 1.0644 - accuracy: 0.6330



```
Epoch 1/10
1563/1563 [==============================] - 7s 4ms/step - loss: 2.2919 - accuracy: 0.1165 - val_loss: 2.3028 - val_accuracy: 0.1120
Epoch 2/10
1563/1563 [==============================] - 6s 4ms/step - loss: 2.0709 - accuracy: 0.2162 - val_loss: 2.0090 - val_accuracy: 0.2756
Epoch 3/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.6963 - accuracy: 0.3789 - val_loss: 1.7749 - val_accuracy: 0.3822
Epoch 4/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.5009 - accuracy: 0.4631 - val_loss: 1.3955 - val_accuracy: 0.5064
Epoch 5/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.3712 - accuracy: 0.5156 - val_loss: 1.3296 - val_accuracy: 0.5336
Epoch 6/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.2629 - accuracy: 0.5620 - val_loss: 1.2349 - val_accuracy: 0.5716
Epoch 7/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.1687 - accuracy: 0.6001 - val_loss: 1.1989 - val_accuracy: 0.5749
Epoch 8/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.0921 - accuracy: 0.6294 - val_loss: 1.1198 - val_accuracy: 0.6222
Epoch 9/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.0145 - accuracy: 0.6580 - val_loss: 1.0732 - val_accuracy: 0.6394
Epoch 10/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.9563 - accuracy: 0.6825 - val_loss: 1.0644 - val_accuracy: 0.6330
```

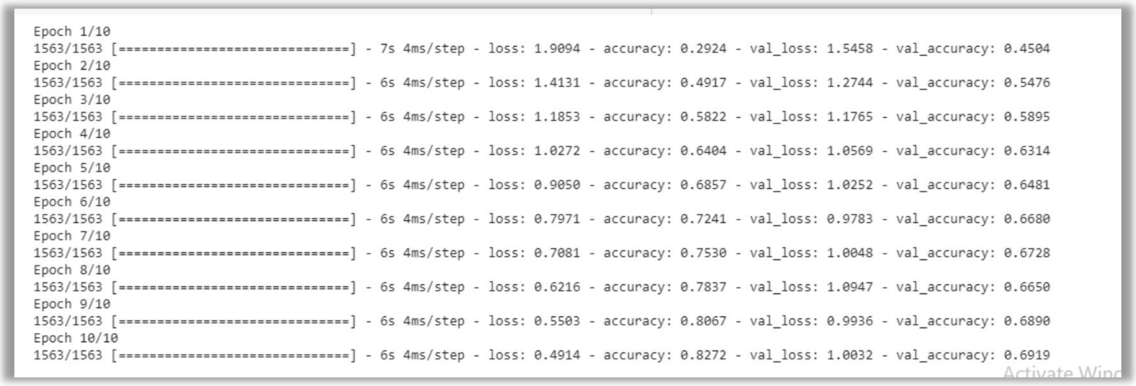


- ✓ Dropout rate(0.5) after only the last 2 Dense layers, that produces loss: 0.9690 - accuracy: 0.6856
- ✓ Dropout rate(0.5) after only the last Dense layer, that produces loss: 0.9775 - accuracy: 0.6911

- According to results above of points (a) & (b) >>
- ✓ Accuracy of **point(a)** is **0.69** & accuracy of **point(b)** is **0.5020**
- ✓ they have shown us that convolution neural networks is better than DNN for classify images, and I think if we increase layers of CNN that will improve the accuracy more. this is because of filters that are applied to extract specific features that makes CNN learn fast for classification.

- Point (c) >> this figure below is with 3 CNN layers , 5 DNN layers & without any Dropout layers, we will note that accuracy of training increases gradually until reach(0.82) compared to accuracy (0.69)of testing or validation data through 10 epochs, and this achieved also in figure of training with 10 epochs in point (a), I think this increasing in accuracy of training than accuracy of testing, means overfitting problem**.**

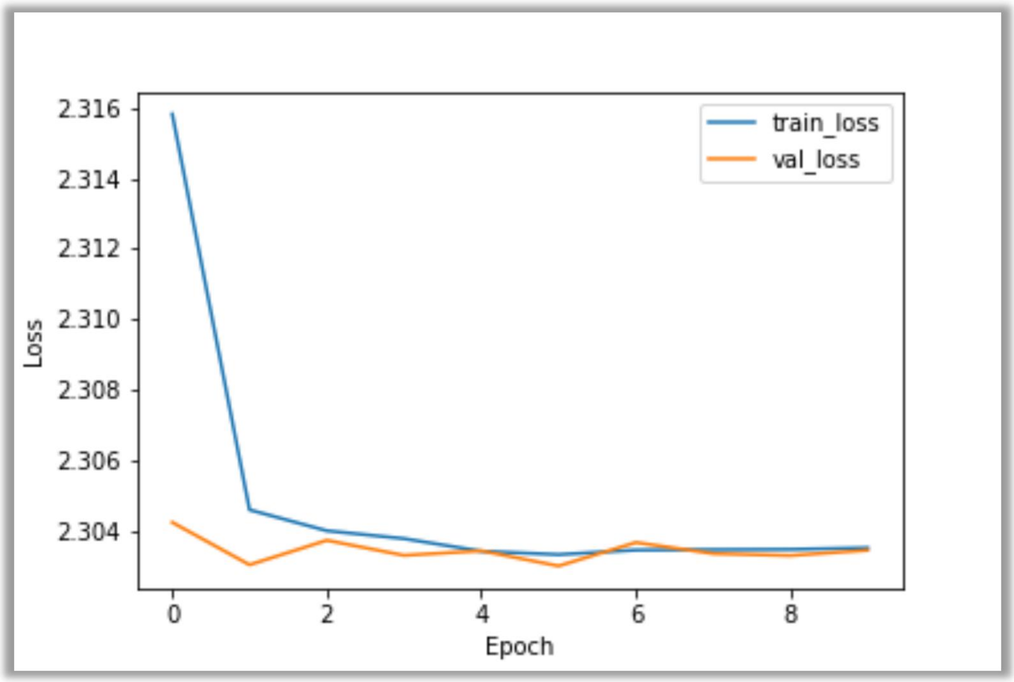
- ✓ So , at point (c), I tried different positions for dropout layers and recorded results above to overcome overfitting and improve accuracy of testing with ReLU at all layers except softmax at output layer.



```
Epoch 1/10
1563/1563 [==============================] - 7s 4ms/step - loss: 1.9094 - accuracy: 0.2924 - val_loss: 1.5458 - val_accuracy: 0.4504
Epoch 2/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.4131 - accuracy: 0.4917 - val_loss: 1.2744 - val_accuracy: 0.5476
Epoch 3/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.1853 - accuracy: 0.5822 - val_loss: 1.1765 - val_accuracy: 0.5895
Epoch 4/10
1563/1563 [==============================] - 6s 4ms/step - loss: 1.0272 - accuracy: 0.6404 - val_loss: 1.0569 - val_accuracy: 0.6314
Epoch 5/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.9050 - accuracy: 0.6857 - val_loss: 1.0252 - val_accuracy: 0.6481
Epoch 6/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.7971 - accuracy: 0.7241 - val_loss: 0.9783 - val_accuracy: 0.6680
Epoch 7/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.7081 - accuracy: 0.7530 - val_loss: 1.0048 - val_accuracy: 0.6728
Epoch 8/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.6216 - accuracy: 0.7837 - val_loss: 1.0947 - val_accuracy: 0.6650
Epoch 9/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.5503 - accuracy: 0.8067 - val_loss: 0.9936 - val_accuracy: 0.6890
Epoch 10/10
1563/1563 [==============================] - 6s 4ms/step - loss: 0.4914 - accuracy: 0.8272 - val_loss: 1.0032 - val_accuracy: 0.6919
```

- **<u>the learning and testing error of point(d) with sigmoid at all layers except softmax in last layer :</u>**

✓ loss: 2.3034 - accuracy: 0.1000



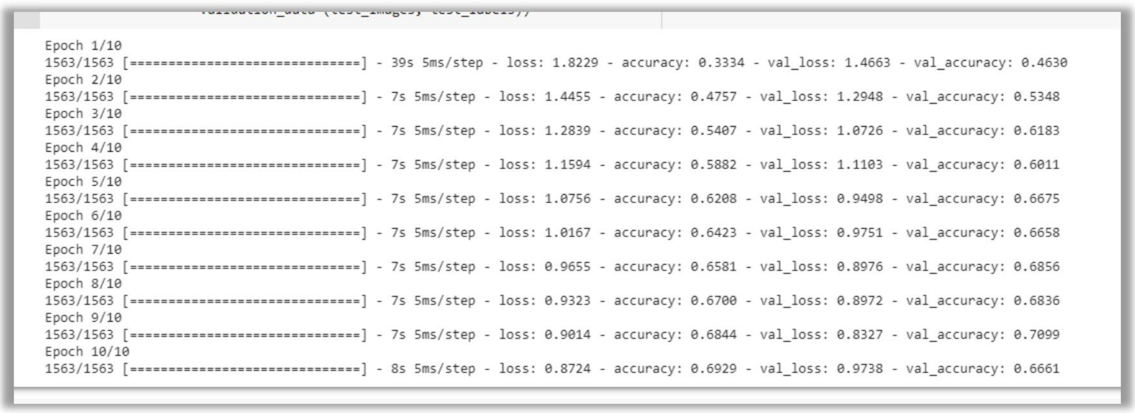
✓ Accuracy of point(a) with ReLU (0.69)is much better than accuracy at point(c) with sigmoid (0.10).

✓ A general problem with sigmoid is that it saturates, This means that large values snap to 1.0 and small values snap 0.0, this function is only really sensitive to changes around their mid-point of their input, such as 0.5, the limited sensitivity and saturation of sigmoid happen regardless of whether the summed activation from the node provided as input contains useful information or not. Once saturated, it becomes challenging for the learning algorithm to continue to adapt the weights to improve the performance of the model.

✓ So, layers deep in large networks using these nonlinear activation functions like sigmoid fail to receive useful gradient information. Error is back propagated through the network and used to update the weights. The amount of error decreases dramatically with each additional layer through which it is propagated, given the derivative of the chosen activation function. This is called the **vanishing gradient problem** and prevents deep (multi-layered) networks from learning effectively.

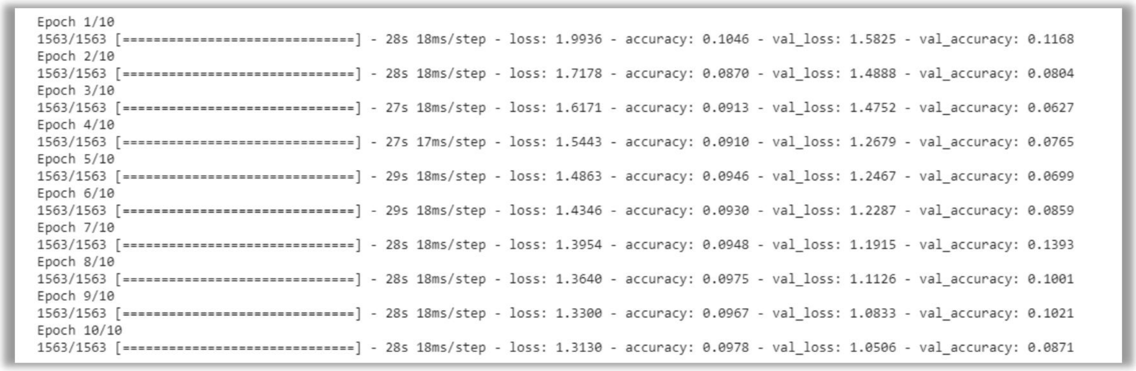
- **<u>Point(e) with dropout of rate 0.3 after last CNN layer and dropout of rate 0.4 after dense layer that is before the output layer & relu at all layers except softmax in last (output) layer :</u>**

✓ According to this figure : the problem of overfitting is solved and the difference between training and test accuracy has became better compered to point (a).



```
Epoch 1/10
1563/1563 [==============================] - 39s 5ms/step - loss: 1.8229 - accuracy: 0.3334 - val_loss: 1.4663 - val_accuracy: 0.4630
Epoch 2/10
1563/1563 [==============================] - 7s 5ms/step - loss: 1.4455 - accuracy: 0.4757 - val_loss: 1.2948 - val_accuracy: 0.5348
Epoch 3/10
1563/1563 [==============================] - 7s 5ms/step - loss: 1.2839 - accuracy: 0.5407 - val_loss: 1.0726 - val_accuracy: 0.6183
Epoch 4/10
1563/1563 [==============================] - 7s 5ms/step - loss: 1.1594 - accuracy: 0.5882 - val_loss: 1.1103 - val_accuracy: 0.6011
Epoch 5/10
1563/1563 [==============================] - 7s 5ms/step - loss: 1.0756 - accuracy: 0.6208 - val_loss: 0.9498 - val_accuracy: 0.6675
Epoch 6/10
1563/1563 [==============================] - 7s 5ms/step - loss: 1.0167 - accuracy: 0.6423 - val_loss: 0.9751 - val_accuracy: 0.6658
Epoch 7/10
1563/1563 [==============================] - 7s 5ms/step - loss: 0.9655 - accuracy: 0.6581 - val_loss: 0.8976 - val_accuracy: 0.6856
Epoch 8/10
1563/1563 [==============================] - 7s 5ms/step - loss: 0.9323 - accuracy: 0.6700 - val_loss: 0.8972 - val_accuracy: 0.6836
Epoch 9/10
1563/1563 [==============================] - 7s 5ms/step - loss: 0.9014 - accuracy: 0.6844 - val_loss: 0.8327 - val_accuracy: 0.7099
Epoch 10/10
1563/1563 [==============================] - 8s 5ms/step - loss: 0.8724 - accuracy: 0.6929 - val_loss: 0.9738 - val_accuracy: 0.6661
```

- **<u>Point(e) with data augmentation & relu at all layers except softmax in last (output) layer :</u>**

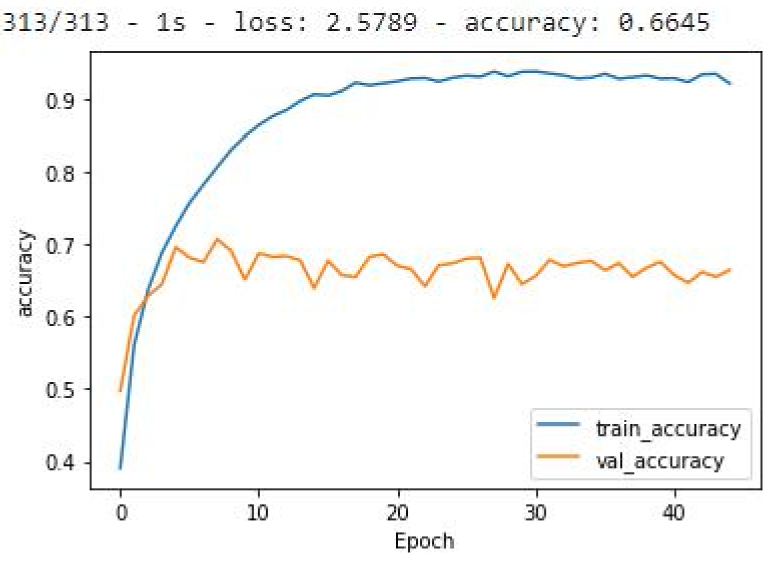


```
Epoch 1/10
1563/1563 [==============================] - 28s 18ms/step - loss: 1.9936 - accuracy: 0.1046 - val_loss: 1.5825 - val_accuracy: 0.1168
Epoch 2/10
1563/1563 [==============================] - 28s 18ms/step - loss: 1.7178 - accuracy: 0.0870 - val_loss: 1.4888 - val_accuracy: 0.0804
Epoch 3/10
1563/1563 [==============================] - 27s 18ms/step - loss: 1.6171 - accuracy: 0.0913 - val_loss: 1.4752 - val_accuracy: 0.0627
Epoch 4/10
1563/1563 [==============================] - 27s 17ms/step - loss: 1.5443 - accuracy: 0.0910 - val_loss: 1.2679 - val_accuracy: 0.0765
Epoch 5/10
1563/1563 [==============================] - 29s 18ms/step - loss: 1.4863 - accuracy: 0.0946 - val_loss: 1.2467 - val_accuracy: 0.0699
Epoch 6/10
1563/1563 [==============================] - 29s 18ms/step - loss: 1.4346 - accuracy: 0.0930 - val_loss: 1.2287 - val_accuracy: 0.0859
Epoch 7/10
1563/1563 [==============================] - 28s 18ms/step - loss: 1.3954 - accuracy: 0.0948 - val_loss: 1.1915 - val_accuracy: 0.1393
Epoch 8/10
1563/1563 [==============================] - 28s 18ms/step - loss: 1.3640 - accuracy: 0.0975 - val_loss: 1.1126 - val_accuracy: 0.1001
Epoch 9/10
1563/1563 [==============================] - 28s 18ms/step - loss: 1.3300 - accuracy: 0.0967 - val_loss: 1.0833 - val_accuracy: 0.1021
Epoch 10/10
1563/1563 [==============================] - 28s 18ms/step - loss: 1.3130 - accuracy: 0.0978 - val_loss: 1.0506 - val_accuracy: 0.0871
```

✓ At beginning i expected that the accuracy would increase but it has been get worse and when i searched about the reason, i understand that this may be because the model has simply too small capacity and it's not able to learn all the patterns in the data, so I think if we increase number of CNN and max pooling layers, the problem may be solved.

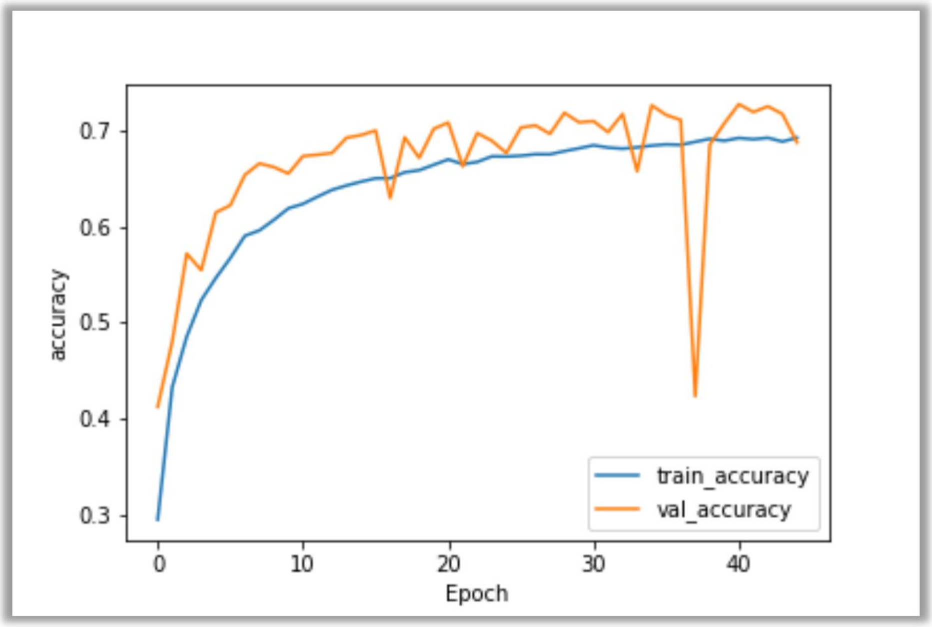
- **<u>Point(f):</u>** accuracy of training and testing without dropout and data augmentation until 45 epochs , I can't reach 100 , the system of colab is crashed because of using all available RAM.

✓ According to this figure is the almost the same as point (a) and also I think there is overfitting problem

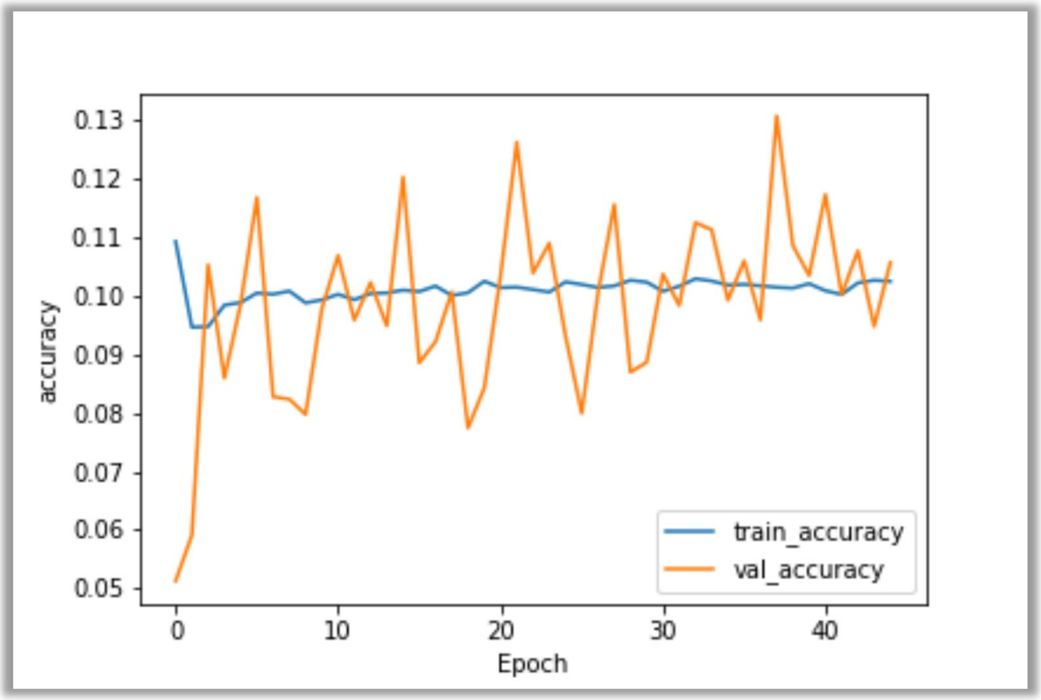


```
313/313 - 1s - loss: 2.5789 - accuracy: 0.6645
```

- **<u>Point(f):</u>** accuracy of training and testing with dropout, I think problem of overfitting is bit solved according to this figure



- **<u>Point(f):</u>** accuracy of training and testing with data augmentation, is the same as **point (e) with data augmentation,** the accuracy has became worst because the model has simply too small capacity and it's not able to learn all the patterns in the data.


accuracy
Epoch
train_accuracy
val_accuracy