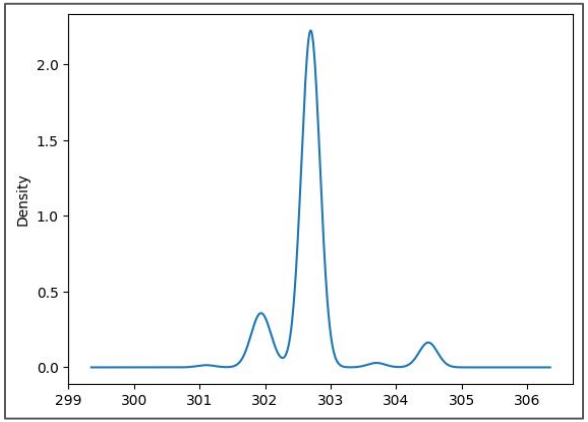
Structure Variant 2: Data requirement in Regression vs Classification

Achsah Marlene Aruva
2019med1001
Submitted as part of ME504: Deep Learning for Physical Systems

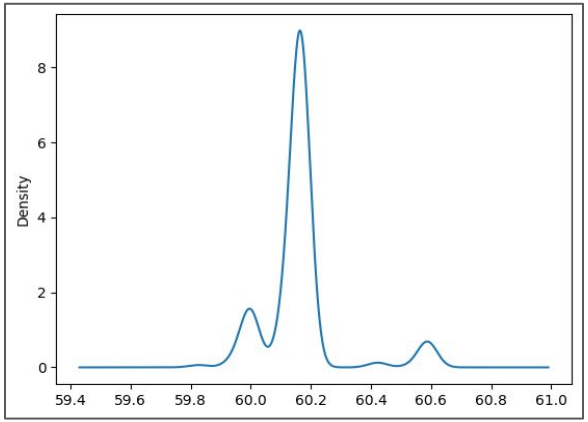
Dataset Generation

- Creating dataset by shuffling n times (n=1000, tqdm = 6 min approx)
- We should only shuffle the E values
- Writing the new shuffled values in prop file
- Running the command which outputs stress and displacement
- Read stress and displacement files
- output.extend

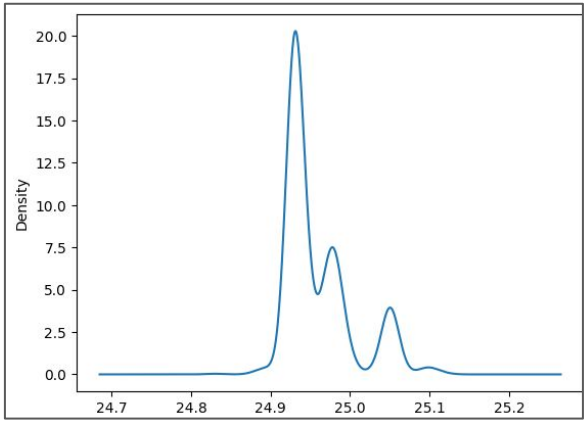
Kernel Density Estimation - Stress



σ_x

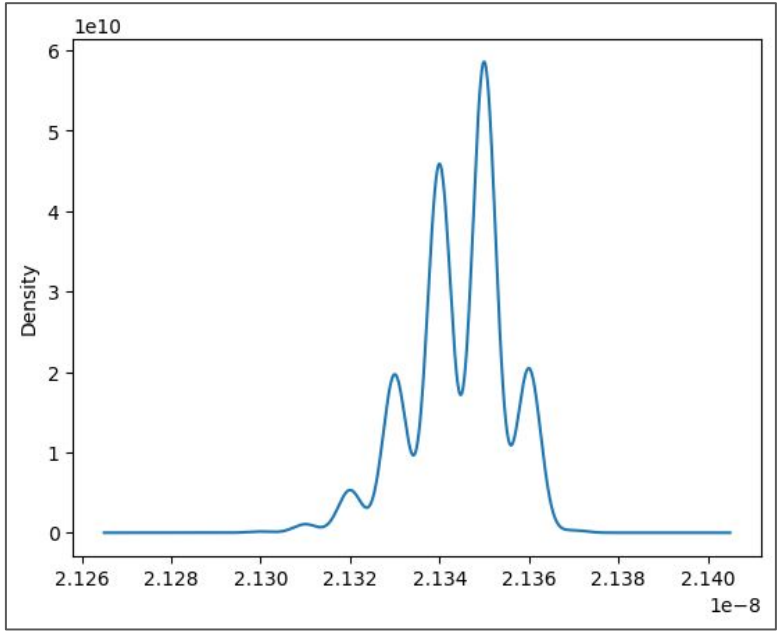


σ_y

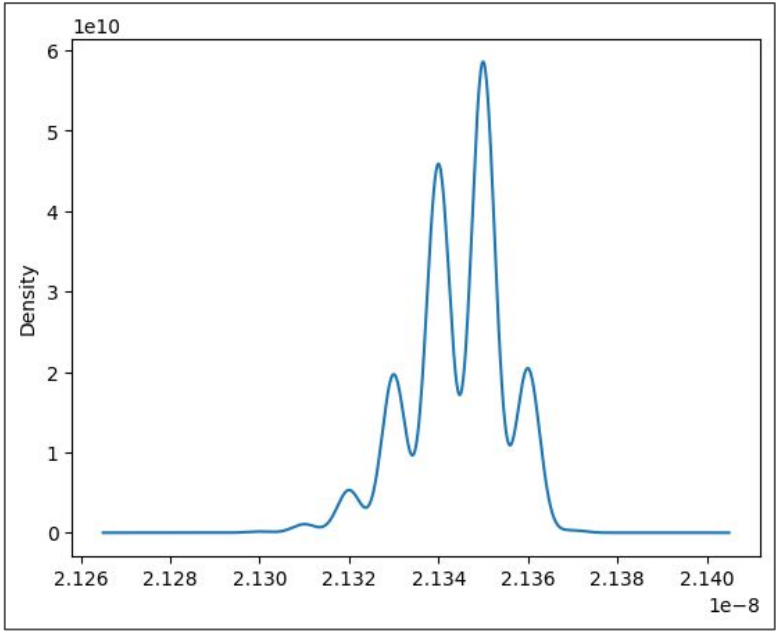


σ_{xy}

Kernel Density Estimation - Displacement



u_x



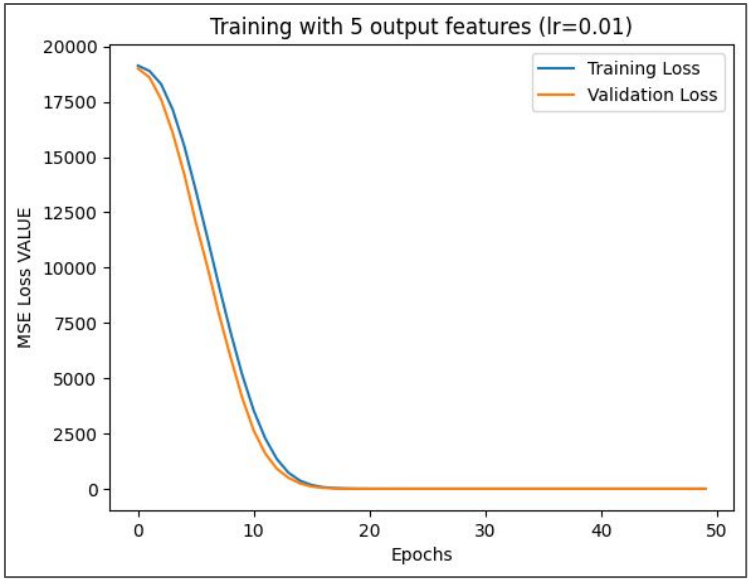
u_y

NN performs poorly with five output features

Mean Square: 0.00016444173343403522

R2 Score: 0.938858076814235

loss: 0.0216 - r_square: 5153364828487680.0000 - val_loss: 0.0188 - val_r_square: -5623192374738944.0000



Regression

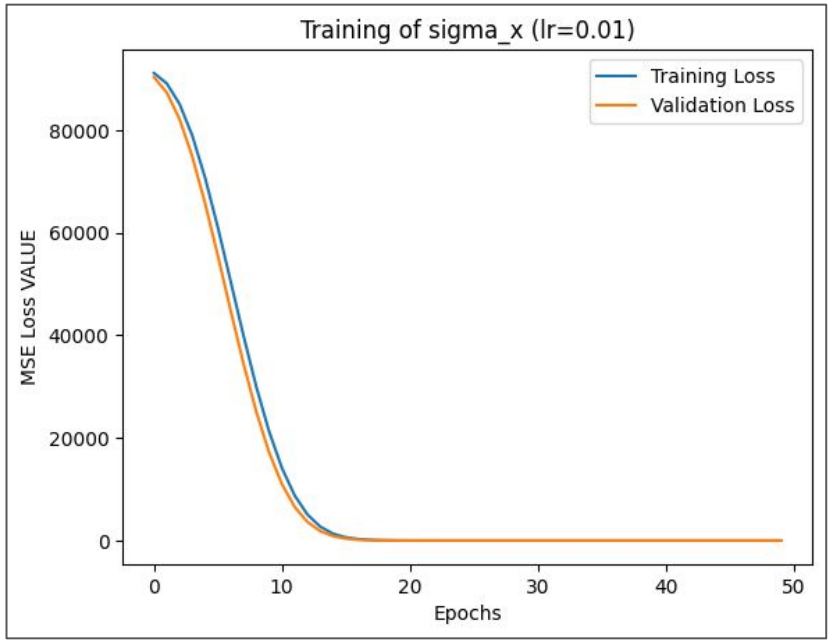
```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_12 (Dense)            (None, 10)                20

 dense_13 (Dense)            (None, 1)                 11

=================================================================
Total params: 31
Trainable params: 31
Non-trainable params: 0
```

sigma_x

loss: 0.0841 - r_square: 0.7443 - val_loss: 0.0951 - val_r_square: 0.6933

Mean Square: 0.0951004919034810 9

R2 Score: 0.6951233064252034

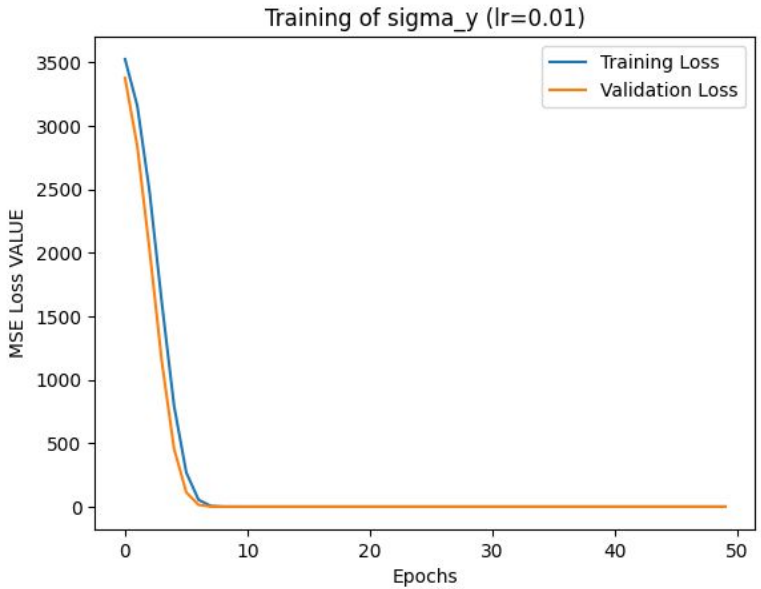


sigma_y

loss: 0.0149 - r_square: 0.1370 - val_loss: 0.0081 - val_r_square: 0.0722

Mean Square: 0.016076994466257658

R2 Score: 0.05800700379343193

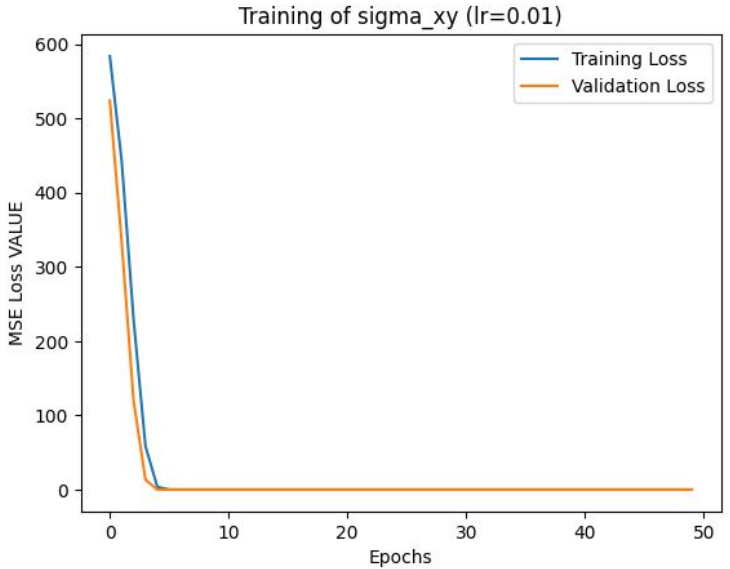


sigma_xy

loss: 0.0019 - r_square: 0.0435 - val_loss: 0.0019 - val_r_square: 0.0335

Mean Square: 0.0020566565319063744

R2 Score: -0.005818057819181321 5

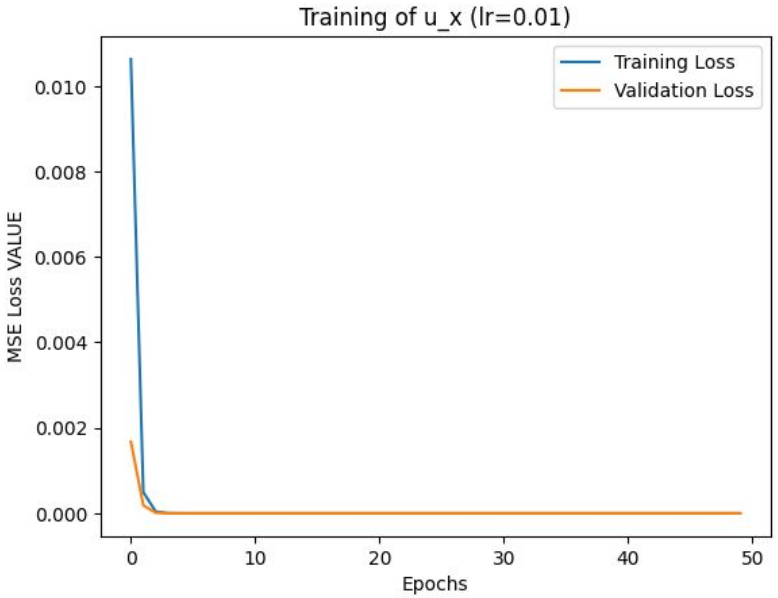


u_x

loss: 4.8271e-14 - r_square: -267130624.0000 - val_loss: 3.1345e-14 - val_r_square: -462566112.0000

Mean Square: 1.2628106244333813e-22

R2 Score: -0.005742771928517465

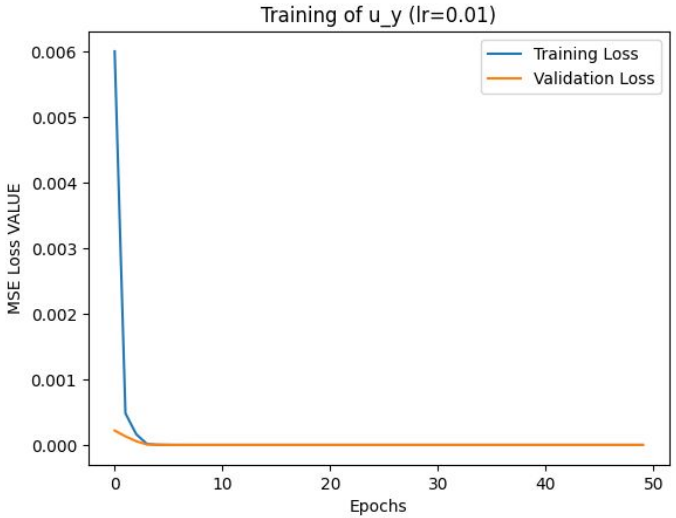


u_y

loss: 1.8137e-16 - r_square: -11761.2500 - val_loss: 1.6479e-17 - val_r_square: -948.9506

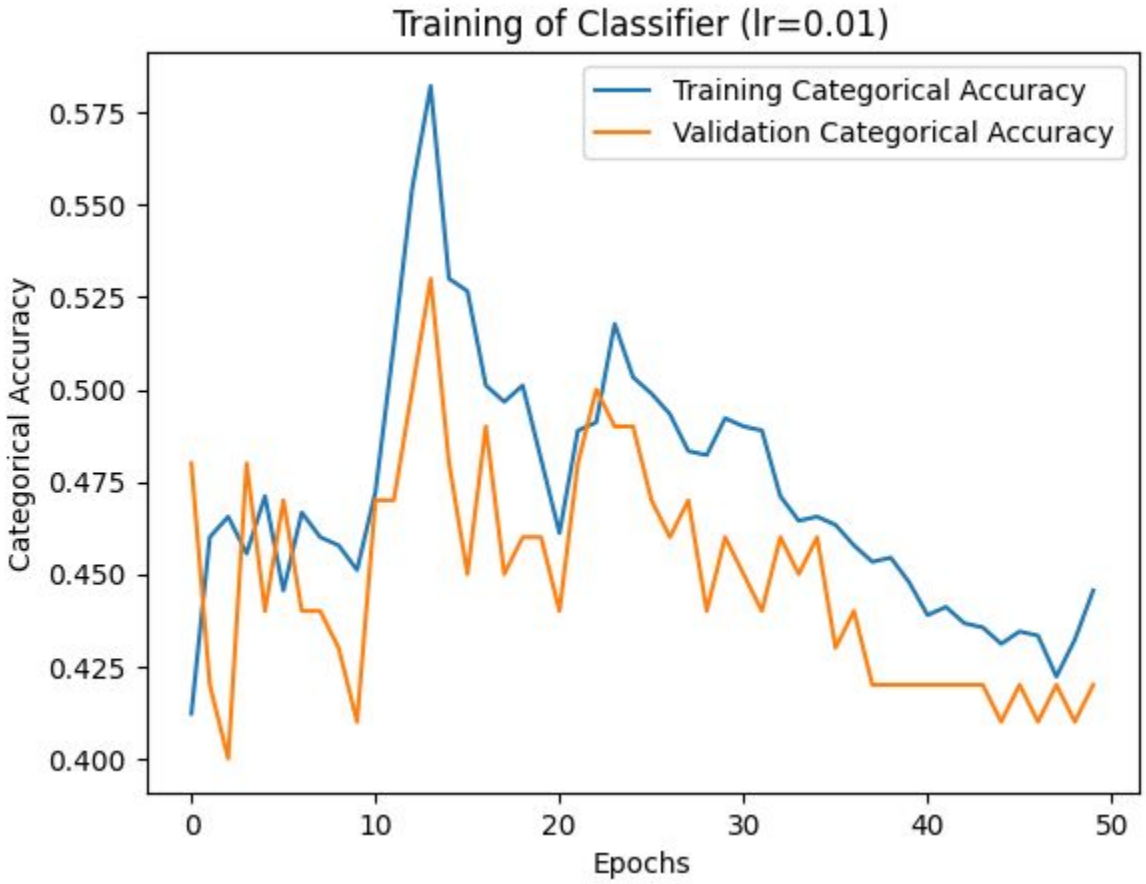
Mean Square: 1.988048187519702e-20

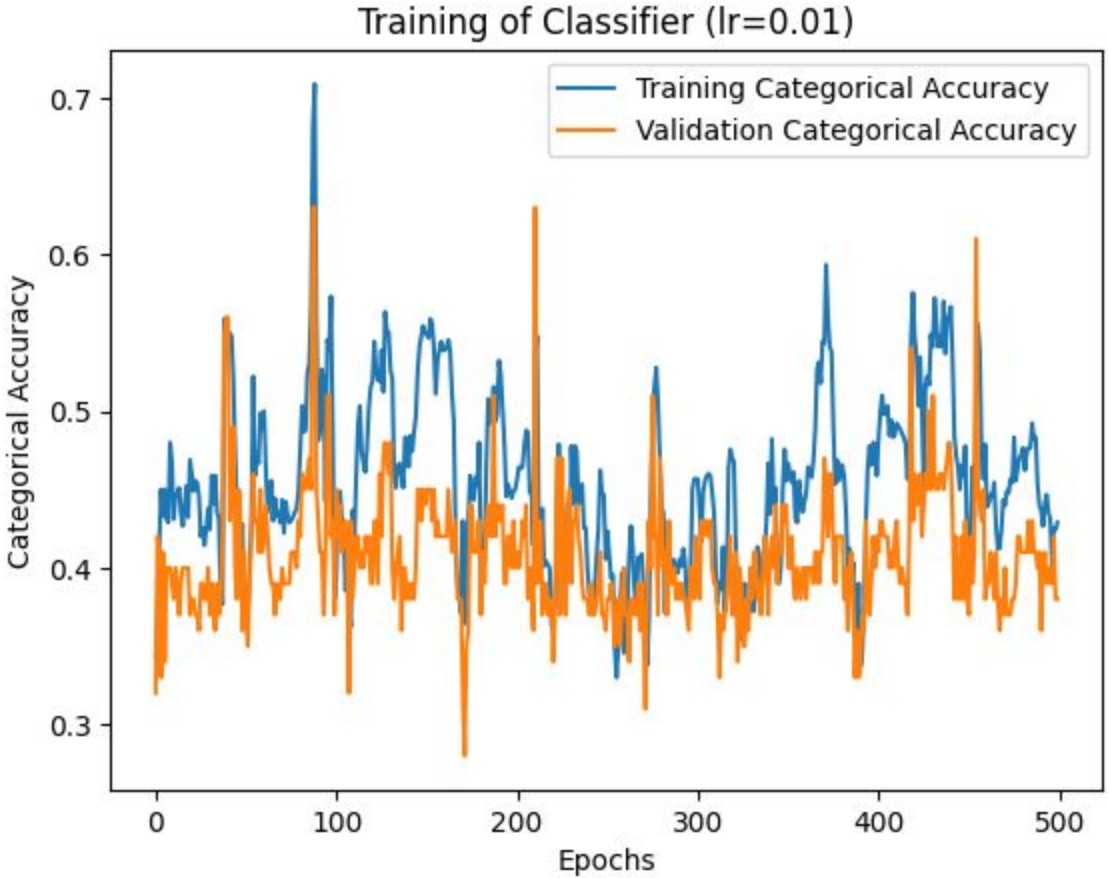
R2 Score: 0.01868394909915893 6



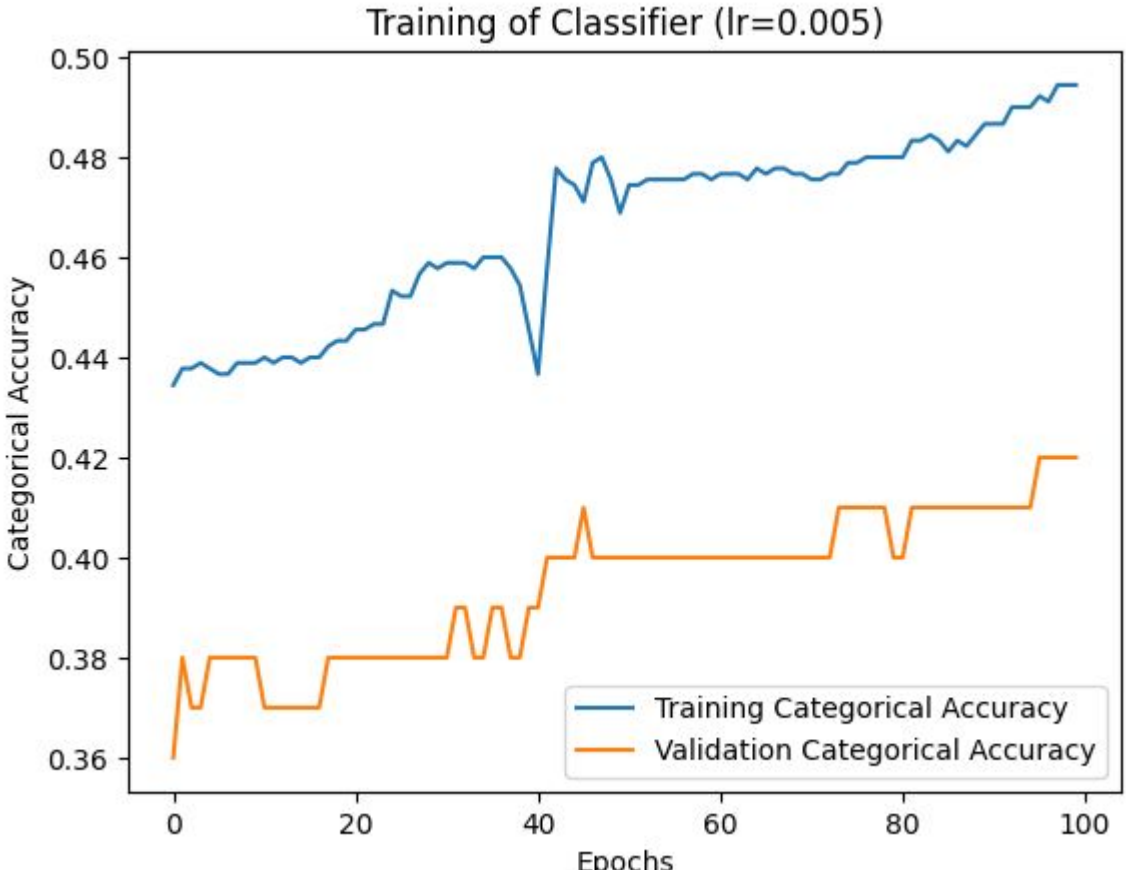
Classification

```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_30 (Dense)            (None, 50)                5050

 dense_31 (Dense)            (None, 80)                4080

 dense_32 (Dense)            (None, 50)                4050

 dense_33 (Dense)            (None, 5)                 255

 activation_6 (Activation)   (None, 5)                 0

=================================================================
Total params: 13,435
Trainable params: 13,435
Non-trainable params: 0
```

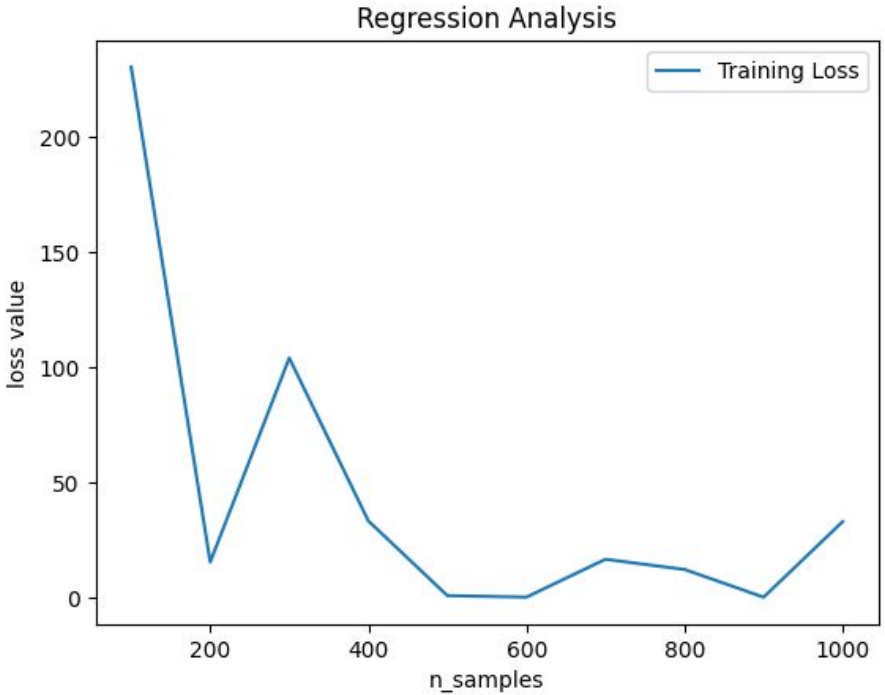

Training of Classifier (lr=0.01)
Training Categorical Accuracy
Validation Categorical Accuracy
Categorical Accuracy
0.575
0.550
0.525
0.500
0.475
0.450
0.425
0.400
0
10
20
30
40
50
Epochs


Training of Classifier (lr=0.01)
Training Categorical Accuracy
Validation Categorical Accuracy
Categorical Accuracy
0.7
0.6
0.5
0.4
0.3
0
100
200
300
400
500
Epochs

Batch size = 16



```
model.compile(optimizer= keras.optimizers.Adam(learning_rate=0.01), loss='mse', metrics=tfa.metrics.r_square.RSquare())

history = model.fit(training_input, training_prediction, epochs=50 , validation_data = (testing_input , testing_prediction), batch_size=8)
```



```
model.compile(optimizer= keras.optimizers.Adam(learning_rate=0.01), loss='binary_crossentropy', metrics=['categorical_accuracy'])

history = model.fit(training_input, training_prediction, epochs=50 , validation_data = (testing_input , testing_prediction), batch_size=8)
```