

## Task1 & Task2

### Homework3\_Task1.ipynb

[https://colab.research.google.com/drive/1\\_X3Zv2IMm-KZ9XzSYEyRrQZKiJLE3Ib?usp=sharing](https://colab.research.google.com/drive/1_X3Zv2IMm-KZ9XzSYEyRrQZKiJLE3Ib?usp=sharing)

In these 2 tasks after downloading the cat-vs-dog train and test pictures, I also downloaded the pre-trained models SqueezeNet\_v1 and MobileNetV2 including model\_json and params (layers of the model and in the params weights and biases). I used the code in the lectures as mentioned in the homework.

The shape of the output vector is (1, 1000) , for first time , I do train for N= 50 with LogisticRegression and the result was

N= 50

Test set score: 0.96

And the for N: 10, 100, 500, and 1000, with RandomForestClassifier , the result were: (I did in CPU runtime type)

Results for N = 10:

Accuracy: 1.00

Computational Time (seconds): 4.41

Best Hyperparameters:

{'max\_depth': 10, 'n\_estimators': 10}

Results for N = 100:

Accuracy: 0.85

Computational Time (seconds): 5.27

Best Hyperparameters:

{'max\_depth': 20, 'n\_estimators': 50}

Results for N = 500:

Accuracy: 0.95

Computational Time (seconds): 26.51

Best Hyperparameters:

{'max\_depth': 20, 'n\_estimators': 100}

Results for N = 1000:

Accuracy: 0.95

Computational Time (seconds): 61.14

Best Hyperparameters:

{'max\_depth': None, 'n\_estimators': 50}

### Homework3\_Task2.ipynb

[https://colab.research.google.com/drive/1Sq57O4zmyg4bGNp7CXF\\_leAunLIQVcTJ?usp=sharing](https://colab.research.google.com/drive/1Sq57O4zmyg4bGNp7CXF_leAunLIQVcTJ?usp=sharing)

For the second pre-trained model MobileNetV2 in task 2 , the output vector is shape: (1, 1280). I run the code in GPU runtime type and with LogisticRegression for

N=5000

Test set score: 0.97, lasting about 18 min

And with RandomForestClassifier for N: 10, 100, 500, and 1000

Results for N = 10:

Accuracy: 1.00

Computational Time (seconds): 7.46

Best Hyperparameters:

{'max\_depth': 10, 'n\_estimators': 100}

Results for N = 100:

Accuracy: 0.90

Computational Time (seconds): 5.38

Best Hyperparameters:

{'max\_depth': 20, 'n\_estimators': 50}

Results for N = 500:

Accuracy: 0.94

Computational Time (seconds): 24.32

Best Hyperparameters:

{'max\_depth': None, 'n\_estimators': 100}

Results for N = 1000:

Accuracy: 0.93

Computational Time (seconds): 58.76

Best Hyperparameters:

{'max\_depth': 10, 'n\_estimators': 100}

Just look at the computational time, we have 0.9 with high computational time for MobileNetV2 toward SqueezeNet\_v1 which it shows 0,85 considering with less computational time. I tested N=5000 for MobileNetV2 and the result of accuracy is 0.97.

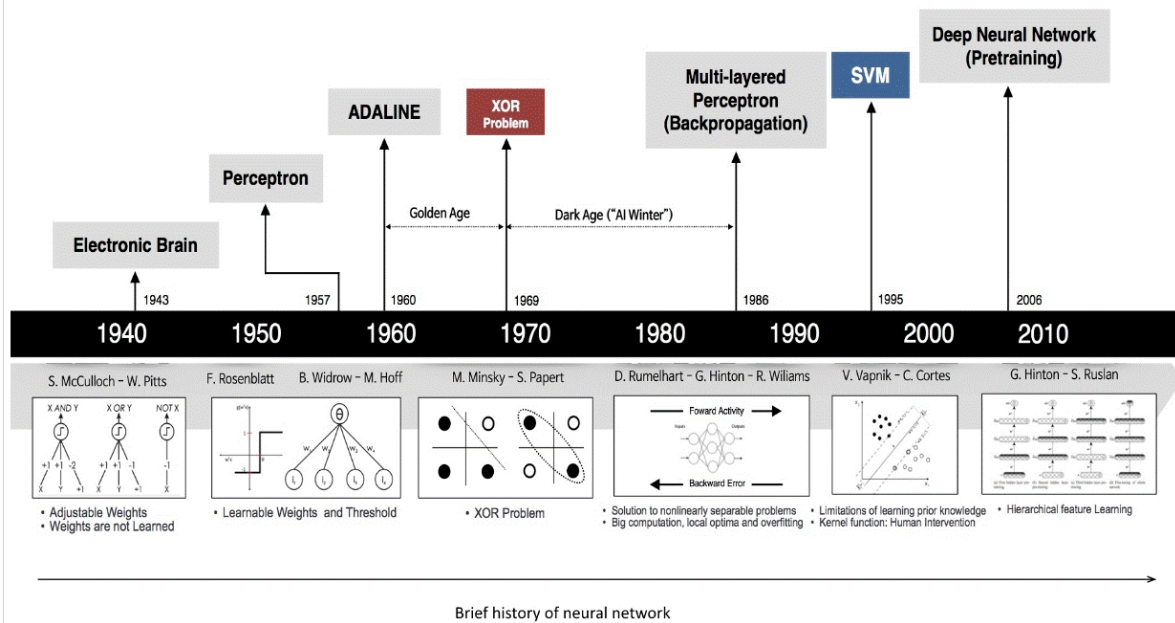
### Task3

#### Homework3\_Task3.ipynb

[https://colab.research.google.com/drive/1qCwcXNK\\_W\\_94oTsw7xrAYd-A3OPJkG\\_6?usp=sharing](https://colab.research.google.com/drive/1qCwcXNK_W_94oTsw7xrAYd-A3OPJkG_6?usp=sharing)

I did Machine learning and also deep learning , as I remembered from the lecture, I used SVM.

Theory is relatively old (from the 40's),

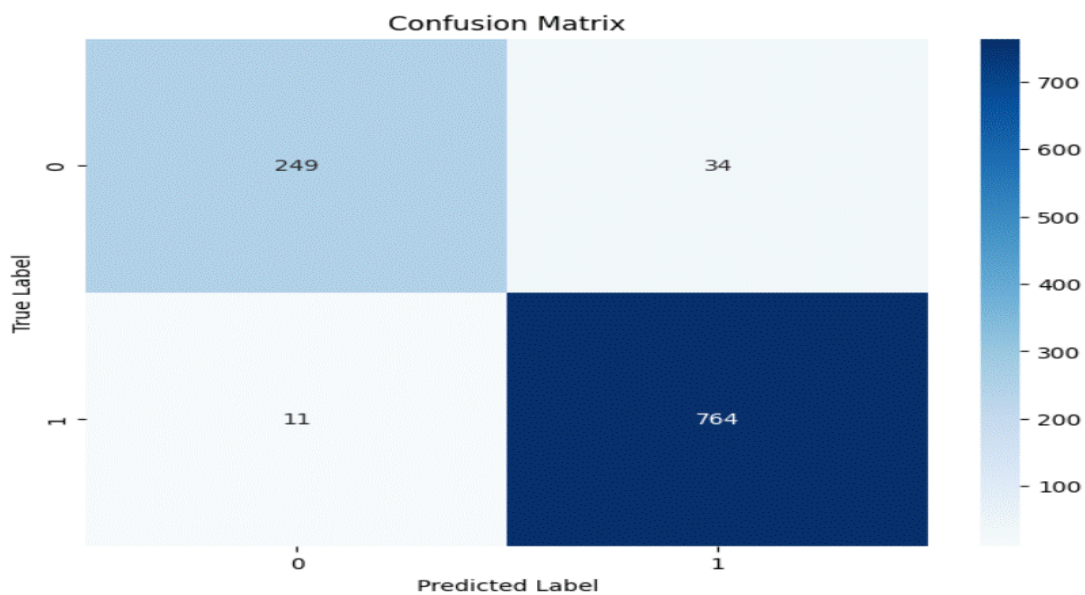


The result was:

Validation Accuracy: 0.96

	precision	recall	f1-score	support
0	0.96	0.88	0.92	283
1	0.96	0.99	0.97	775
accuracy			0.96	1058
macro avg	0.96	0.93	0.94	1058
weighted avg	0.96	0.96	0.96	1058

and the confusion matrix :



But also tried to create a CNN model not with pre-trained model and the achievement was not acceptable.

output

Epoch 1, Training accuracy: 0.25047303689687794  
Epoch 1, Validation accuracy: 0.2674858223062382  
Epoch 2, Training accuracy: 0.25047303689687794  
Epoch 2, Validation accuracy: 0.2674858223062382  
Epoch 3, Training accuracy: 0.25047303689687794  
Epoch 3, Validation accuracy: 0.2674858223062382  
Epoch 4, Training accuracy: 0.25047303689687794  
Epoch 4, Validation accuracy: 0.2674858223062382  
Epoch 5, Training accuracy: 0.25047303689687794  
Epoch 5, Validation accuracy: 0.2674858223062382  
Epoch 6, Training accuracy: 0.25047303689687794  
Epoch 6, Validation accuracy: 0.2674858223062382  
Epoch 7, Training accuracy: 0.25047303689687794  
Epoch 7, Validation accuracy: 0.2674858223062382  
Epoch 8, Training accuracy: 0.25047303689687794  
Epoch 8, Validation accuracy: 0.2674858223062382  
Epoch 9, Training accuracy: 0.25047303689687794  
Epoch 9, Validation accuracy: 0.2674858223062382  
Epoch 10, Training accuracy: 0.25047303689687794  
Epoch 10, Validation accuracy: 0.2674858223062382

I tried to go forward through a scientific paper and the result was not so fine again.

<https://colab.research.google.com/drive/1BOL5NDfqwHbkB9B-GngrhMU7E30WNvJK?usp=sharing>

Epoch 1, Loss: 0.8341549038887024, Validation Loss: 0.8240476846694946, Accuracy: 0.6141732335090637  
Epoch 2, Loss: 0.7367851734161377, Validation Loss: 0.8127676844596863, Accuracy: 0.6141732335090637  
Epoch 3, Loss: 0.865983784198761, Validation Loss: 0.7991393804550171, Accuracy: 0.6141732335090637  
Epoch 4, Loss: 0.8118098974227905, Validation Loss: 0.7903969883918762, Accuracy: 0.6141732335090637  
Epoch 5, Loss: 0.8080819845199585, Validation Loss: 0.7835642695426941, Accuracy: 0.6141732335090637  
Epoch 6, Loss: 0.7904750108718872, Validation Loss: 0.7868789434432983, Accuracy: 0.6141732335090637  
Epoch 7, Loss: 0.8143637776374817, Validation Loss: 0.7799028158187866, Accuracy: 0.6141732335090637  
Epoch 8, Loss: 0.8495929837226868, Validation Loss: 0.7789860963821411, Accuracy: 0.6141732335090637  
Epoch 9, Loss: 0.806844174861908, Validation Loss: 0.7777299284934998, Accuracy: 0.6141732335090637  
Epoch 10, Loss: 0.8371522426605225, Validation Loss: 0.7756369113922119, Accuracy: 0.6141732335090637

Finally, I reached to this paper but unfortunately, I did not have time to test.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8361010/>

**Classification of Covid-19 patients using efficient fine-tuned deep learning DenseNet model**