CMPE-789 Project Final Presentation

Arjun Thangaraju

04/21/2022

Agenda

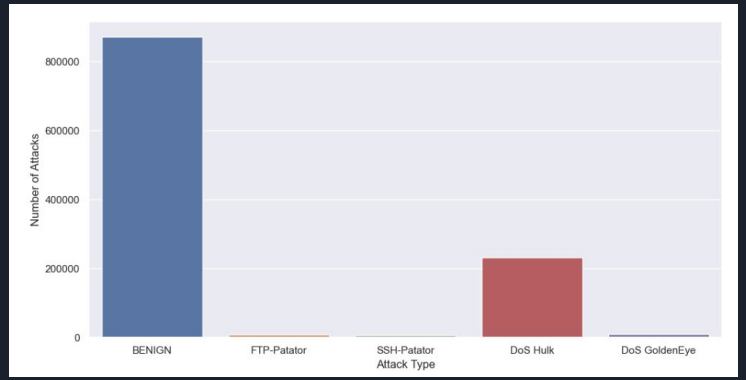
- Dataset
  - Dataset Description
  - Dataset Visualization
    - IDS 2017 Combined
    - IDS 2018 Combined
- Baseline Model Results
  - IDS 2017 Combined
  - IDS 2018 Combined
- Transfer Learning Strategies
  - CNN Architecture Intuition
    - Strategy 1
    - Strategy 2
- Design of Experiments Table
  - Experiments Results
    - 2017 -> 2018
    - 2018 -> 2017
- Key Takeaways
- Appendix
- References
- Questions and Feedback

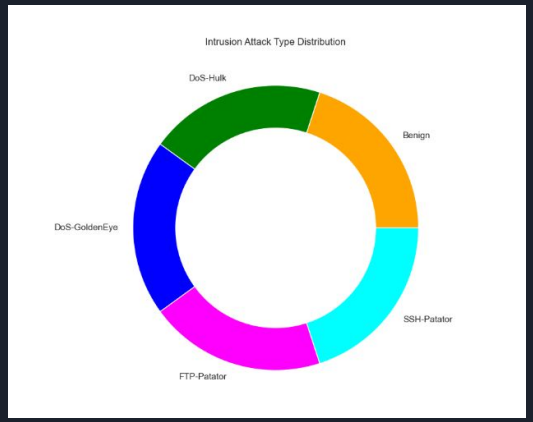
Reference for Image

Dataset Description

No.	Dataset Name	Day/Date
1.	IDS 2017 Combined Dataset	Tuesday & Wednesday
2.	IDS 2018 Combined Dataset	02-14-2018 & 02-15-2018

IDS 2017 Combined Dataset Visualization





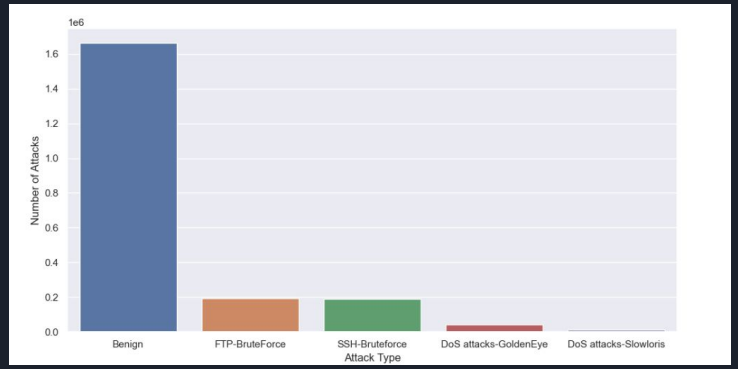

```
memory usage: 686.3+ MB
BENIGN            872105
DoS Hulk          231073
DoS GoldenEye      10293
FTP-Patator         7938
SSH-Patator         5897
Name:  Label, dtype: int64
```

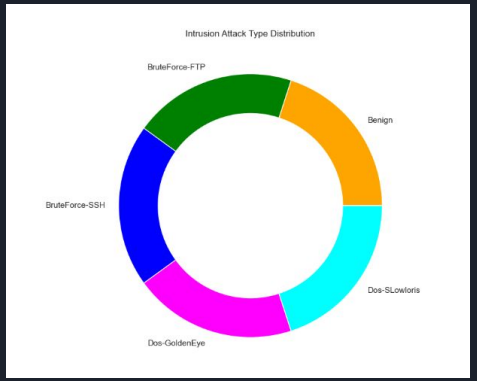
```
Number of Rows (Samples): 1138612
Number of Columns (Features): 79
```

TrainData Size
- Xtrain: 100,000 rows (20,000 from each class) x 72 cols (features)
- Ytrain: 100,000 rows (20,000 from each class) x 5 cols (classes)

Test Data Size
- Xtest: 281,525 rows x 72 cols
- YTest: 281,525 rows x 5 cols

IDS 2018 Combined Dataset Visualization






```
memory usage: 1.2+ GB
Benign                     1663703
FTP-BruteForce              193360
SSH-Bruteforce              187589
DoS attacks-GoldenEye        41508
DoS attacks-Slowloris        10990
Name: Label, dtype: int64
```

```
Number of Rows (Samples): 2097150
Number of Columns (Features): 80
```

TrainData Size
- Xtrain: 100,000 rows (20,000 from each class) x 72 cols (features)
- Ytrain: 100,000 rows (20,000 from each class) x 5 cols (classes)

Test Data Size
- Xtest: 417,991 rows x 72 cols
- YTest: 417,991 rows x 5 cols

IDS 2017 Combined Baseline Model

CNN Architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 72, 64)            448

 batch_normalization (BatchN  (None, 72, 64)           256
 ormalization)

 max_pooling1d (MaxPooling1D  (None, 36, 64)           0
 )

 conv1d_1 (Conv1D)           (None, 36, 64)            24640

 batch_normalization_1 (Batc  (None, 36, 64)           256
 hNormalization)

 max_pooling1d_1 (MaxPooling  (None, 18, 64)           0
 1D)

 conv1d_2 (Conv1D)           (None, 18, 64)            24640

 batch_normalization_2 (Batc  (None, 18, 64)           256
 hNormalization)

 max_pooling1d_2 (MaxPooling  (None, 9, 64)            0
 1D)

 flatten (Flatten)           (None, 576)               0

 dense (Dense)               (None, 64)                36928

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 5)                 325

=================================================================
```

```
Total params: 91,909
Trainable params: 91,525
Non-trainable params: 384
```

IDS 2017 Combined Baseline Model Classification Report

	precision	recall	f1-score	support
0-Benign	0.99	0.96	0.98	217966
1-DoS-Hulk	0.63	0.99	0.77	2597
2-DoS-GoldenEye	0.96	0.94	0.95	57487
3-FTP-Patator	0.97	1.00	0.98	2060
4-SSH-Patator	0.21	0.98	0.35	1415
accuracy			0.96	281525
macro avg	0.75	0.98	0.81	281525
weighted avg	0.98	0.96	0.97	281525

IDS 2018 Combined Baseline Model

CNN Architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 72, 64)            448

 batch_normalization (BatchN  (None, 72, 64)           256
 ormalization)

 max_pooling1d (MaxPooling1D  (None, 36, 64)           0
 )

 conv1d_1 (Conv1D)           (None, 36, 64)            24640

 batch_normalization_1 (Batc  (None, 36, 64)           256
 hNormalization)

 max_pooling1d_1 (MaxPooling  (None, 18, 64)           0
 1D)

 conv1d_2 (Conv1D)           (None, 18, 64)            24640

 batch_normalization_2 (Batc  (None, 18, 64)           256
 hNormalization)

 max_pooling1d_2 (MaxPooling  (None, 9, 64)            0
 1D)

 flatten (Flatten)           (None, 576)               0

 dense (Dense)               (None, 64)                36928

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 5)                 325

=================================================================
```

```
Total params: 91,909
Trainable params: 91,525
Non-trainable params: 384
```

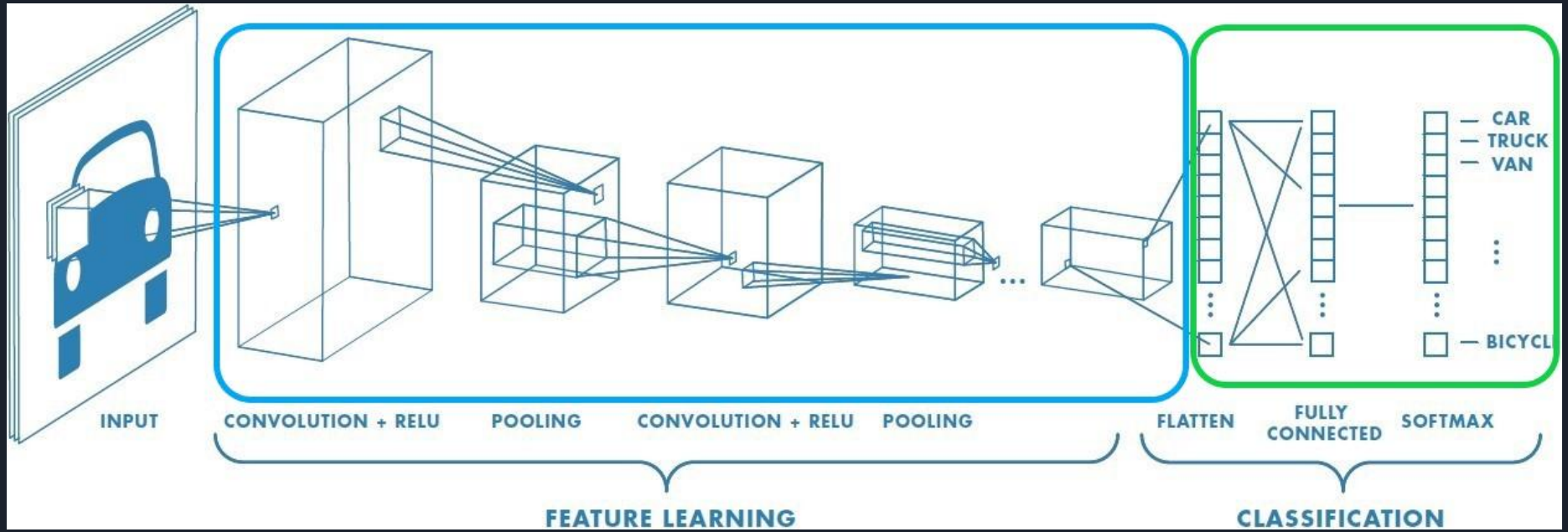
IDS 2018 Combined Baseline Model Classification Report

	precision	recall	f1-score	support
0-Benign	1.00	0.99	0.99	331214
1-FTP-BruteForce	0.88	1.00	0.93	8164
2-SSH-BruteForce	0.71	0.99	0.83	2217
3-Dos-GoldenEye	1.00	1.00	1.00	38989
4-Dos-SLowloris	0.98	0.99	0.98	37407
accuracy			0.99	417991
macro avg	0.91	0.99	0.95	417991
weighted avg	0.99	0.99	0.99	417991

Transfer Learning Strategies

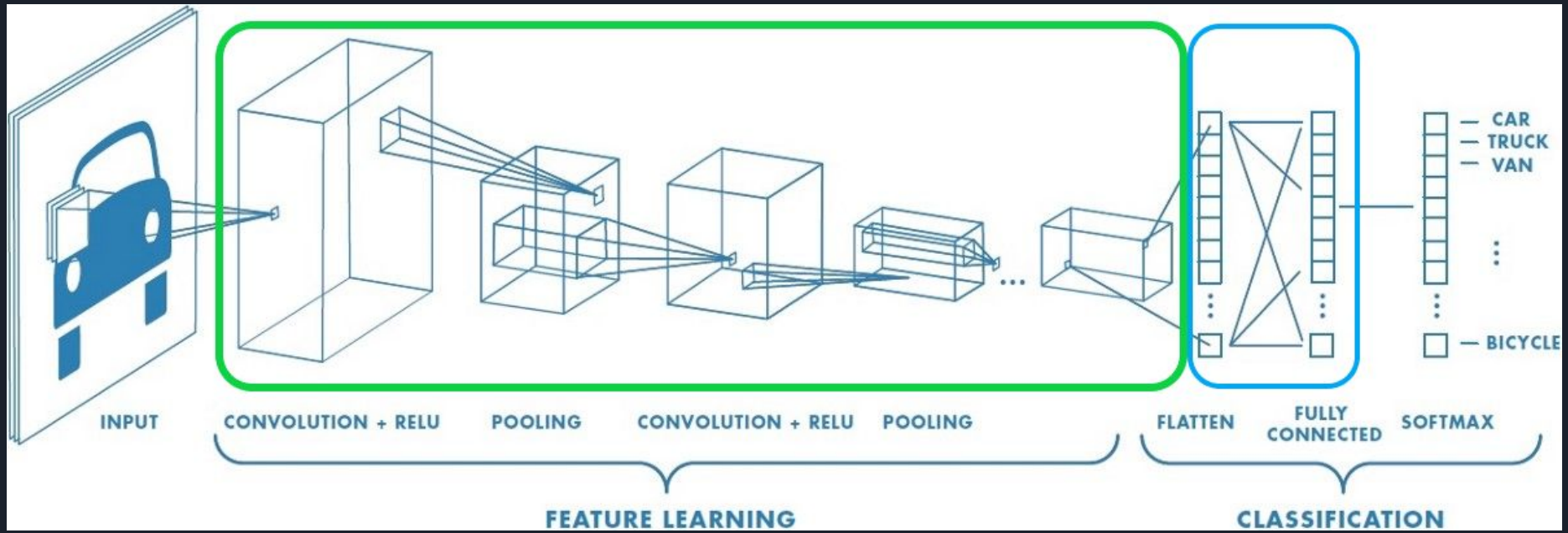
No.	Strategy
1.	Freeze all Feature Extraction Layers while Classification Layers remain Active.
2.	Freeze all Classification Layers while Feature Extraction Layers remain Active.

Strategy 1 CNN Architecture Intuition



Reference for Image

Strategy 2 CNN Architecture Intuition



Reference for Image

Design of Experiments Table

No.	Pre-Trained Dataset	Transfer Learning Dataset	Test Dataset (Unseen Split)
1.	2017 Combined	None	2018 Combined
2.	2017 Combined	2018 Combined	2018 Combined
3.	2018 Combined	None	2017 Combined
4.	2018 Combined	2017 Combined	2017 Combined

Note: 2. & 4. are evaluated on Strategies 1 & 2

Experiment Results

2017 → 2018 Results Summary

	precision	recall	f1-score	support
0-Benign	1.00	0.99	0.99	331185
1-Dos-Slowloris	0.71	1.00	0.83	8300
2-Dos-GoldenEye	0.00	0.00	0.00	2175
3-FTP-BruteForce	1.00	1.00	1.00	38596
4-SSH-BruteForce	0.99	1.00	0.99	37735
accuracy			0.99	417991
macro avg	0.74	0.80	0.76	417991
weighted avg	0.98	0.99	0.99	417991

Strategy 1

>

	precision	recall	f1-score	support
0-Benign	0.87	0.89	0.88	331212
1-Dos-Slowloris	0.67	0.62	0.64	8209
2-Dos-GoldenEye	0.00	0.00	0.00	2241
3-FTP-BruteForce	0.00	0.00	0.00	38866
4-SSH-BruteForce	0.56	1.00	0.72	37463
accuracy			0.81	417991
macro avg	0.42	0.50	0.45	417991
weighted avg	0.76	0.81	0.78	417991

Strategy 2

>

	precision	recall	f1-score	support
0-Benign	0.78	0.79	0.78	330977
1-Dos-Slowloris	0.00	0.00	0.00	8370
2-Dos-GoldenEye	0.03	0.68	0.05	2131
3-FTP-BruteForce	0.00	0.00	0.00	38980
4-SSH-BruteForce	0.00	0.00	0.00	37533
accuracy			0.63	417991
macro avg	0.16	0.29	0.17	417991
weighted avg	0.62	0.63	0.62	417991

No TF Baseline

- Mapping 2017 Baseline Model Classes → 2018 TF Model Classes
  - Benign → Benign
  - DoS-Hulk → Dos-Slowloris
  - DoS-GoldenEye → Dos-GoldenEye
  - FTP-Patator → FTP-BruteForce
  - SSH-Patator → SSH-BruteForce

2018 → 2017 Results Summary

	precision	recall	f1-score	support
0-Benign	1.00	0.96	0.98	217918
1-FTP-Patator	1.00	0.71	0.83	2643
2-SSH-Patator	0.97	1.00	0.98	57630
3-DoS-GoldenEye	0.92	1.00	0.96	1948
4-DoS-Hulk	0.18	0.99	0.30	1386
accuracy			0.97	281525
macro avg	0.81	0.93	0.81	281525
weighted avg	0.99	0.97	0.98	281525

Strategy 1

>

	precision	recall	f1-score	support
0-Benign	0.95	0.96	0.95	217762
1-FTP-Patator	0.00	0.00	0.00	2557
2-SSH-Patator	0.91	0.86	0.88	57671
3-DoS-GoldenEye	0.26	0.84	0.40	2025
4-DoS-Hulk	0.71	0.51	0.59	1510
accuracy			0.92	281525
macro avg	0.57	0.63	0.56	281525
weighted avg	0.93	0.92	0.92	281525

Strategy 2

>

	precision	recall	f1-score	support
0-Benign	0.77	0.95	0.85	217764
1-FTP-Patator	0.34	0.17	0.23	2531
2-SSH-Patator	0.00	0.00	0.00	57768
3-DoS-GoldenEye	0.00	0.00	0.00	1988
4-DoS-Hulk	0.00	0.01	0.01	1474
accuracy			0.74	281525
macro avg	0.22	0.23	0.22	281525
weighted avg	0.60	0.74	0.66	281525

No TF Baseline

- Mapping 2018 Baseline Model Classes → 2017 TF Model Classes
  - Benign → Benign
  - FTP-BruteForce → FTP-Patator
  - SSH-BruteForce → SSH-Patator
  - Dos-GoldenEye → DoS-GoldenEye
  - Dos-Slowloris → DoS-Hulk

Key Takeaways on Network Architecture & Nature of Data

- Strategy 1 is the best performing strategy. This shows that freezing feature extraction layers helps the classification layers to learn and classify new classes by transferring previously learnt features.

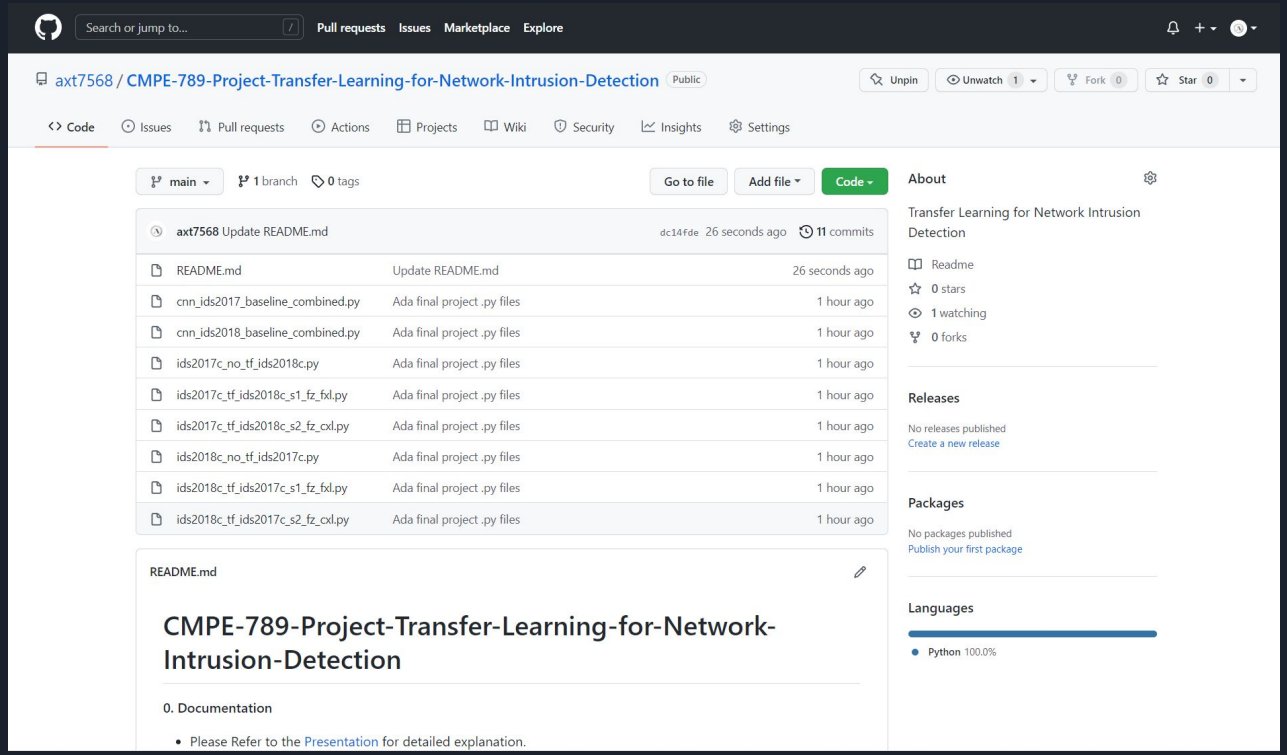
- Results from Strategy 2 shows that freezing classification layers does not help in classifying new classes as the features learnt by the feature extraction layers cannot be transferred.

- The SSH attacks for both years and both strategies showed good performance with Transfer Learning. Maybe this is due to both these attacks (SSH-Bruteforce and SSH-Patator) sharing common features and thus warrants further investigation.

- The DoS-GoldenEye attack for both years had bad performance overall (expect for strategy 1 while transferring features from 2018 to 2017) which maybe highlights different features for different years even if it's the same attack. This is to be investigated further.

- It should also be noted that class imbalance can cause high test samples for one class and lower number of test samples for another. This test sample imbalance can influence the values in the classification report given that the imbalance is large.

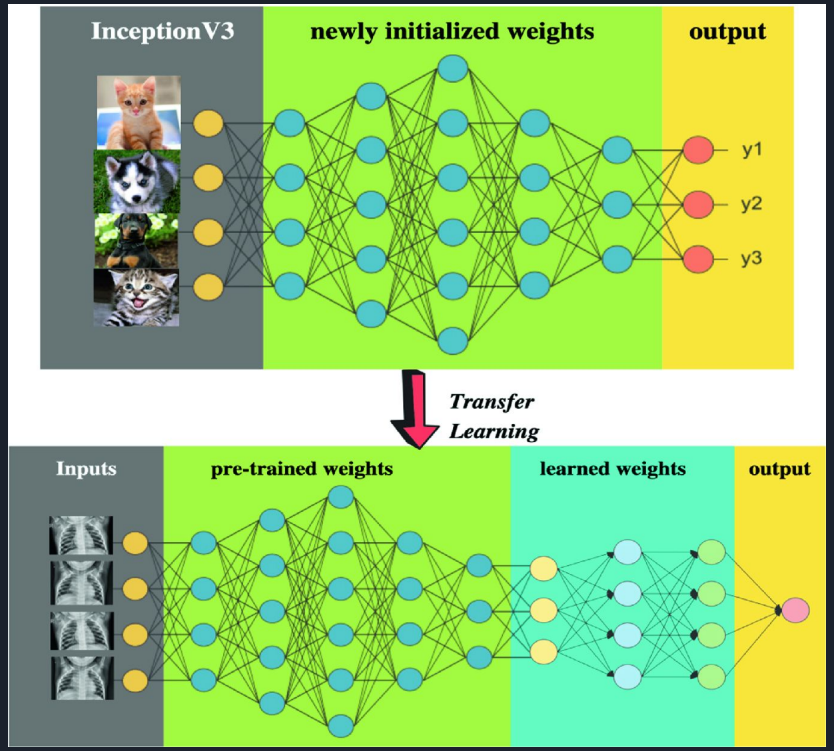
Open-Source Github Repo



[Github Repo Link]

Appendix

What is Transfer Learning ?



Reference for Image

References

- https://www.kaggle.com/solarmainframe/ids-intrusion-csv
- https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning
- https://www.kaggle.com/code/azazurrehmanbutt/cicids-ids-2018-using-cnn
- https://www.unb.ca/cic/datasets/ids-2018.html
- https://machinelearningmastery.com/transfer-learning-for-deep-learning/
- https://cs231n.github.io/transfer-learning/

Questions and Feedback

Reference for Image