

Language: Python3.8
Framework: PyTorch
Libraries: Numpy, Pandas, PyTorch, Matplotlib, Sklearn
Environment: Jupyter Notebook

Task:

Apply **Transfer Learning** on MLP for Malware detection

Outcome:

Training time reduced from 18.3 seconds to 0.74 seconds

MLP AUC: 98.9%

Model:

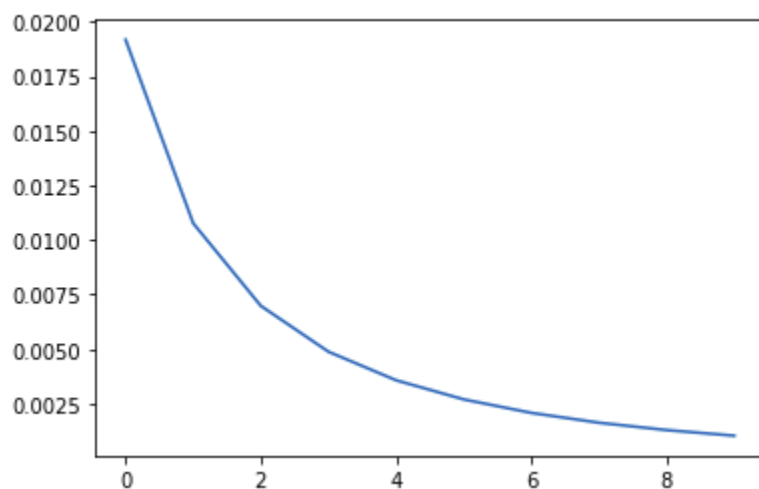
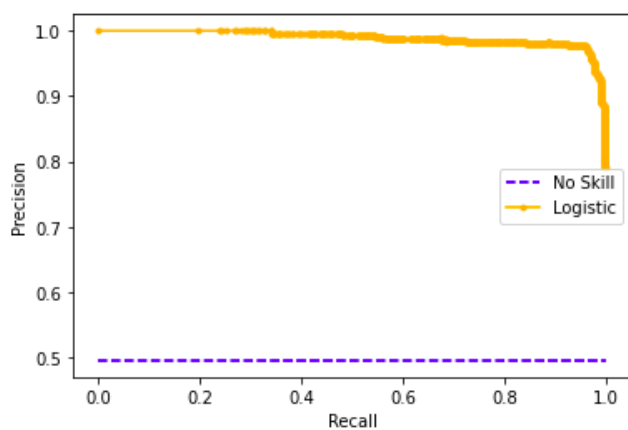
```
class PreTrainedMLP(nn.Module):
    def __init__(self):
        super(PreTrainedMLP, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(15, 128),
            nn.ReLU(),
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Linear(64, 15),
            nn.ReLU(),
        )
        self.clf = nn.Sequential(
            nn.Linear(15, 1),
            nn.Sigmoid()
        )
    def forward(self, x):
        output = self.model(x)
        return self.clf(output)

mlp = PreTrainedMLP()
pretrained_layers = torch.load("mlp_5.pt")
_ = mlp.load_state_dict(pretrained_layers)
for param in mlp.parameters():
    param.requires_grad = False
#new classifier
mlp.clf = nn.Sequential(
    nn.Linear(15, 1),
    nn.Sigmoid()
```

)

Training Loss:

Took 0.74 seconds to train

**MLP Classification result:**

F1: 0.9682835820895523

AUC: 0.9892777646945142

Accuracy 0.9687786960514233

