# Membership Inference Report

PETS 2021

## Attack Performance

Note: we use the CrossEntropyLoss function and a learning rate of 0.001 for all models.

### Target Model

Since we don't have a target model, we have to train one ourselves. We do this by heavily overfitting the Conv2DNet network, it consists of 2 hidden convolutional layers followed by 1 hidden linear layer reducing the output of the 2nd Conv2d layer to 128 and finally the 10 output neurons. For this example we ran it for 75 Epochs. The accuracy of the target model for training data is at 99.26% while it is only 98.52% for testing or validation data.

### Shadow Model

Next we train our Shadow model. We use the exact same setup as for the Target Model with the difference that we stop training if we see our V-Loss is no longer improving, more specifically we stop if we see our V-Loss did not improve for 2 straight Epochs. In our case it stopped after 37 Epochs. The accuracy for the shadow model for training data lies at 98.72% while it is only 98.42% for testing or validation data. As one can see the target model is heavily overfitted when compared to the shadow model.

### 

### 

### 

### 

### 

### 

### Attack Model Dataset

Now we need to create the Dataset. For this we have to evaluate both the target and shadow model with the same MNIST dataset distribution the posteriors we get from this are subtracted from another to form the input data for the attack model. The labels are derived from whenever the input image was part of the training=member or testing=non-member dataset used for the target model making the labels 100% accurate. We also use a 90%Train/10%Test split for this dataset.

### Attack Model

The attack model is supposed to predict whenever an input, in our case an image from the MNIST dataset, was used to train the target model or not. Therefore we want to create a binary classifier, our AttackNet network consists of 10 inputs aka the output size of the target model followed by 4 fully connected linear hidden layers of size 1024 and finally the two output neurons using the log\_softmax activation function because of the CrossEntropyLoss function. Overall this model achieved a V-Loss of 0.0599 while training and it trained for 78 Epochs.

The performance of the model can be measured around ~97% accuracy for test images and 97.5% for training images from the MNIST dataset.

## 

## Used Models

You can find the models used for this example in models/ named target50.pth, shadow37.pth

and attack0599.pth

## Further infos

Further infos about how the code works can be found in the README.md of the repo or looking at some comments in the code or executing it and reading the info/debug messages both in the terminal and code.