Inversion Attack on Machine Learning Models:

Stealing User Input Data

What are Inversion Attacks

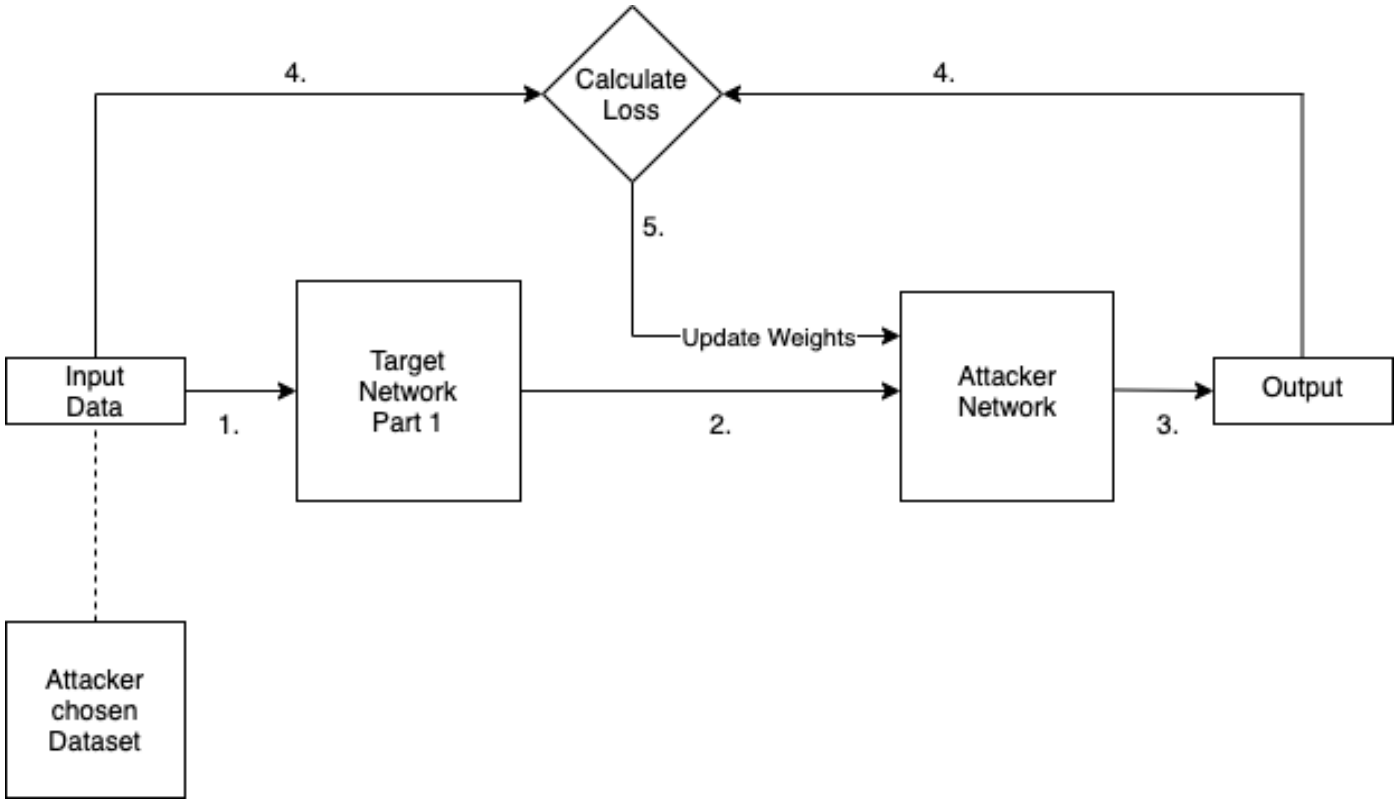
- Inversion attacks try to invert a machine learning model to steal data that was used to train the classifier or used as input to the classifier to get a prediction
- With such attacks an attacker is able to extract data out of a model
- This might raise some concerns when it comes to personal data, as it would be a huge privacy breach for the persons whose data was involved

Our attack: Stealing User Data

- In our attack, we want to show how an attacker could steal user data, that was put into a split neural network (between a server and a user) to get a prediction
- We assume that an attacker is able to intercept the data between the server and the user
- With the intercepted data the attacker then tries reconstruct the initial data that was put into the neural network

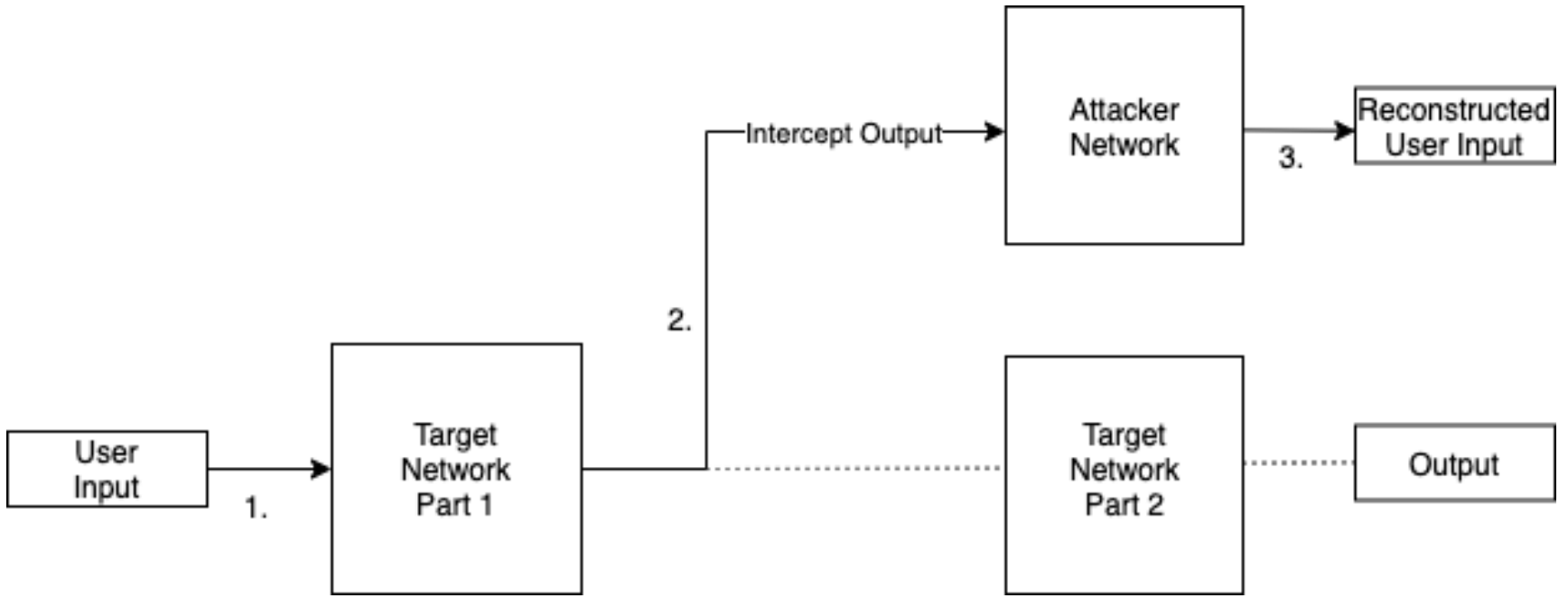
How the attack works

- Training of the attacking network:



How the attack works

- Attacking the target model:

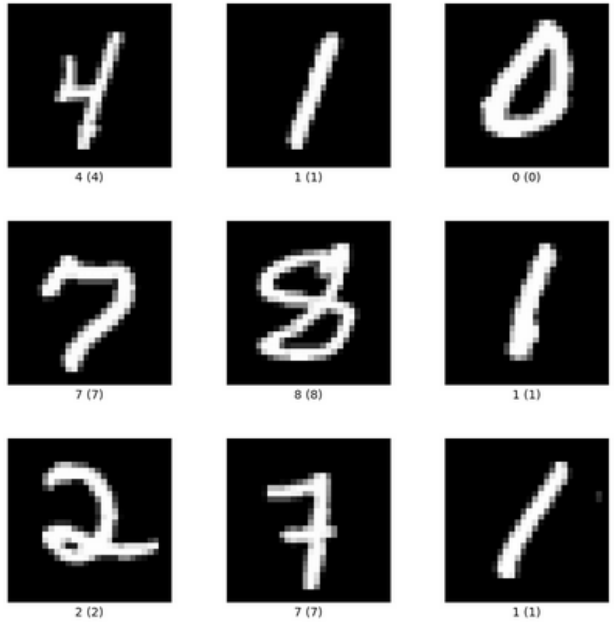


Evaluation

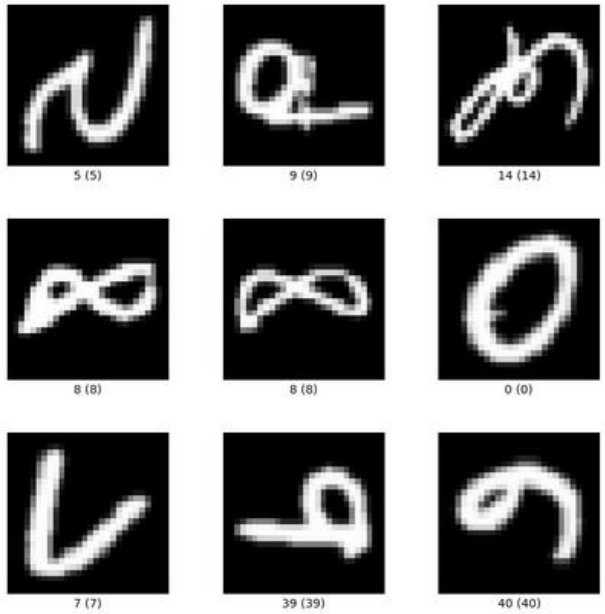
- SimpleNet (own implemented classifier):
  - 2 easy layers
- AlexNet:
  - 6 layers
- ResNet:
  - 6 complex layers

MNIST + EMNIST

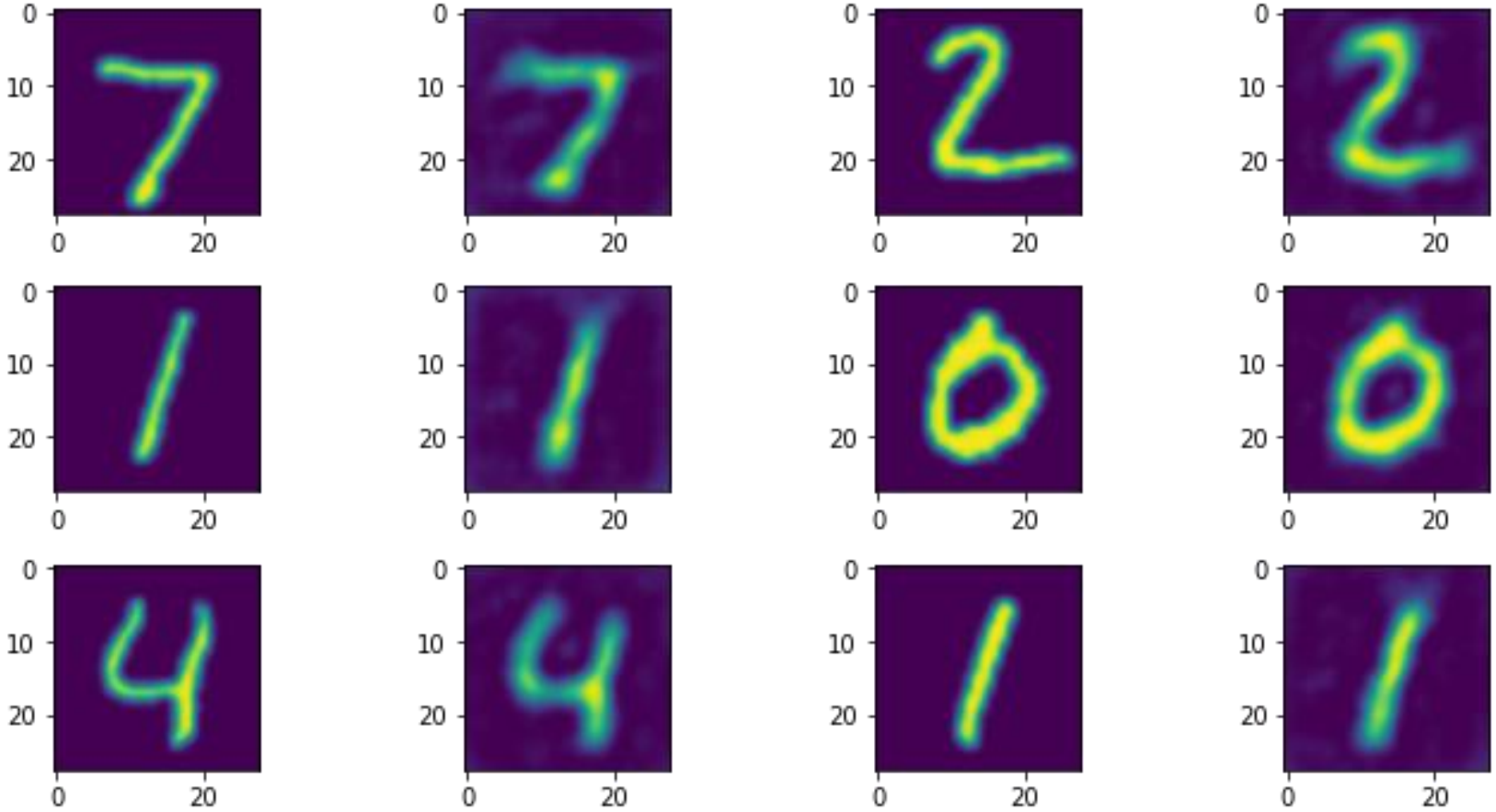
Target Model Trainings Data



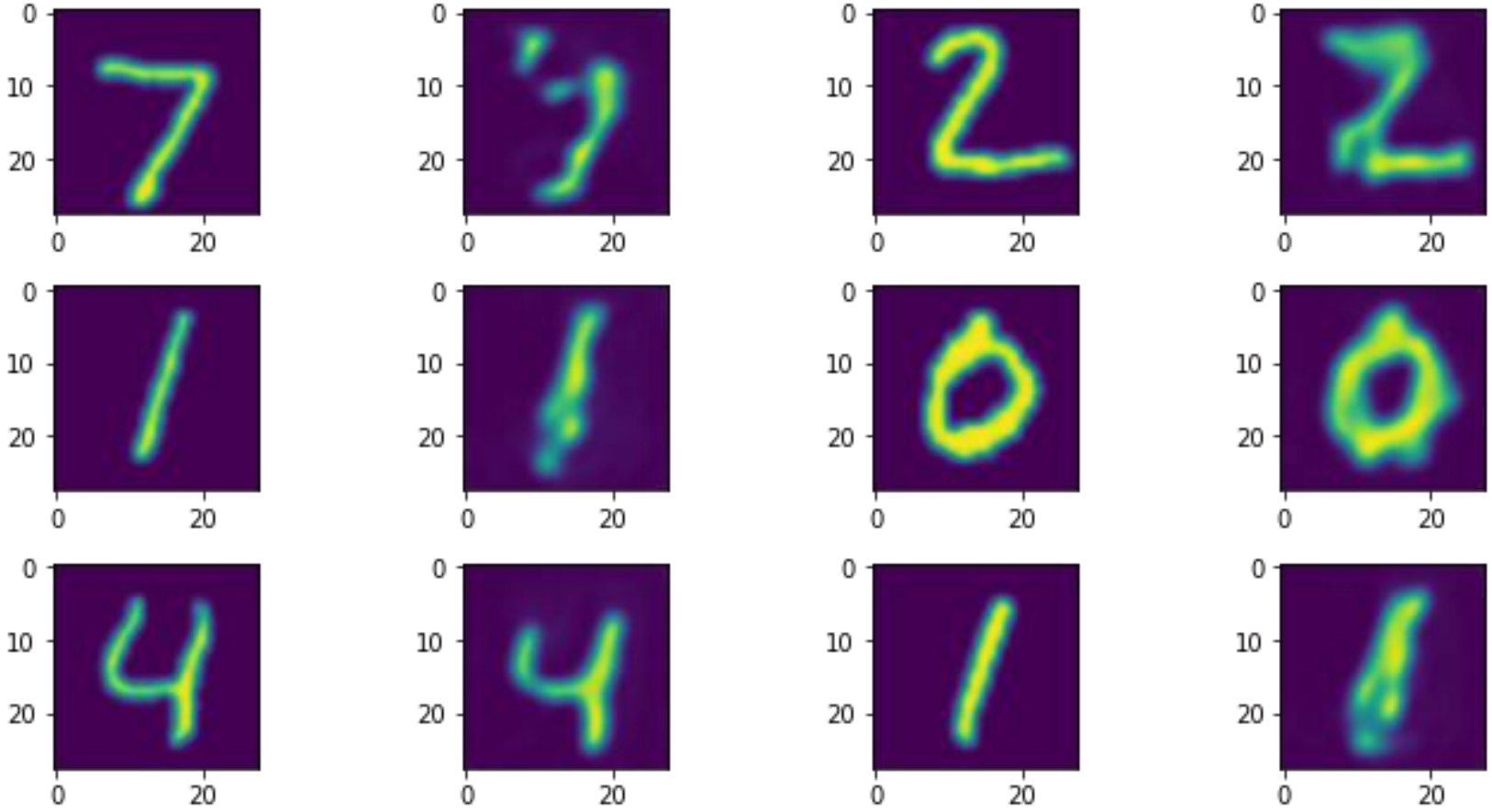
Attack Model Trainings Data



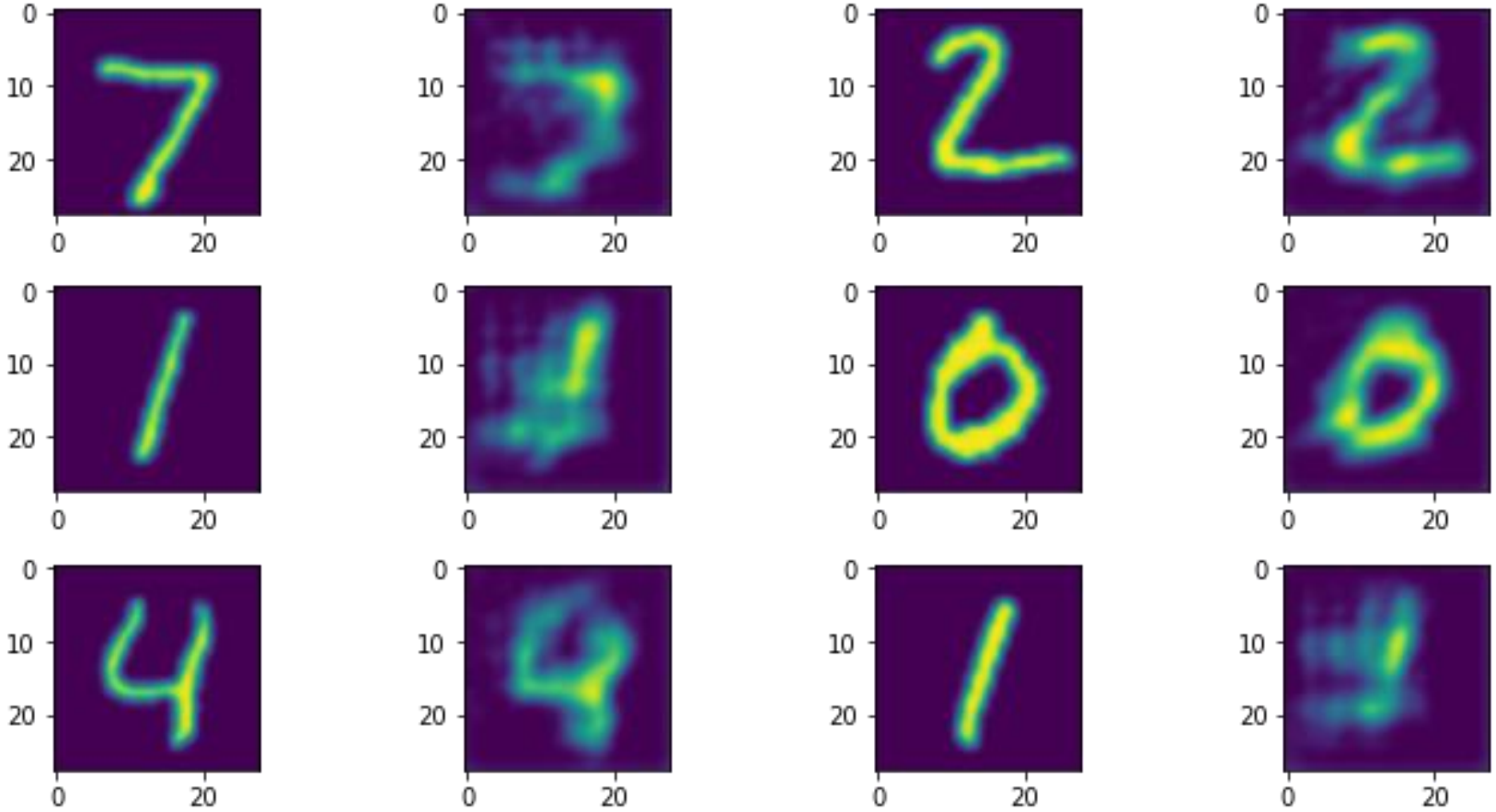
Attack on SimpleNet



Attack on AlexNet

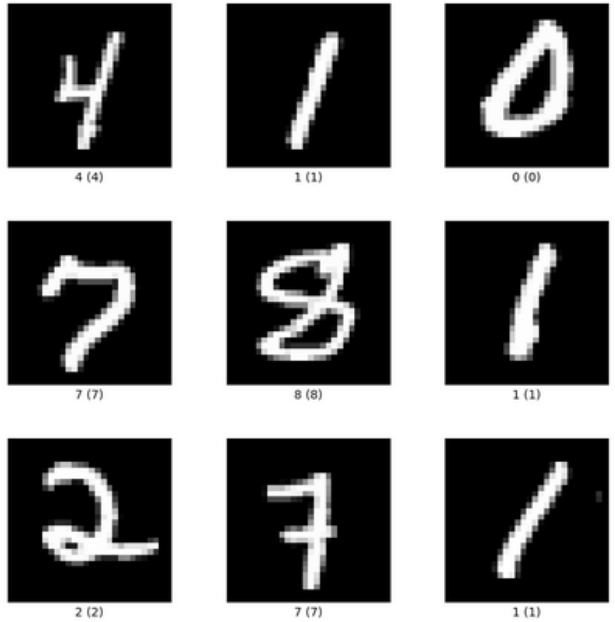


Attack on ResNet

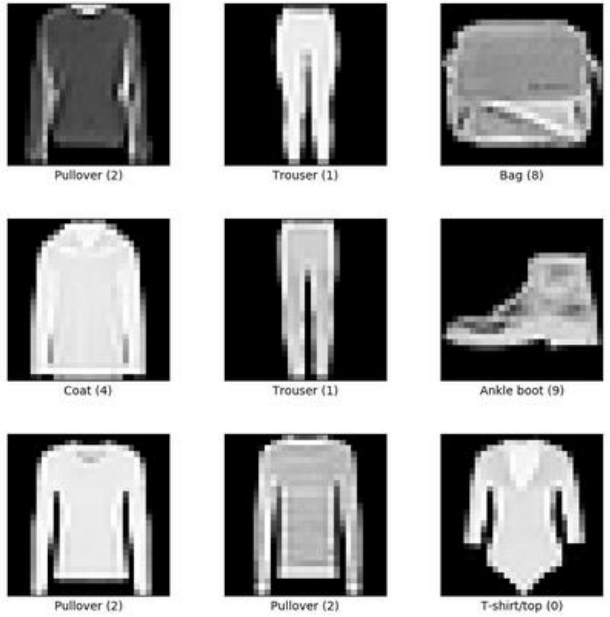


MNIST + FashionMNIST

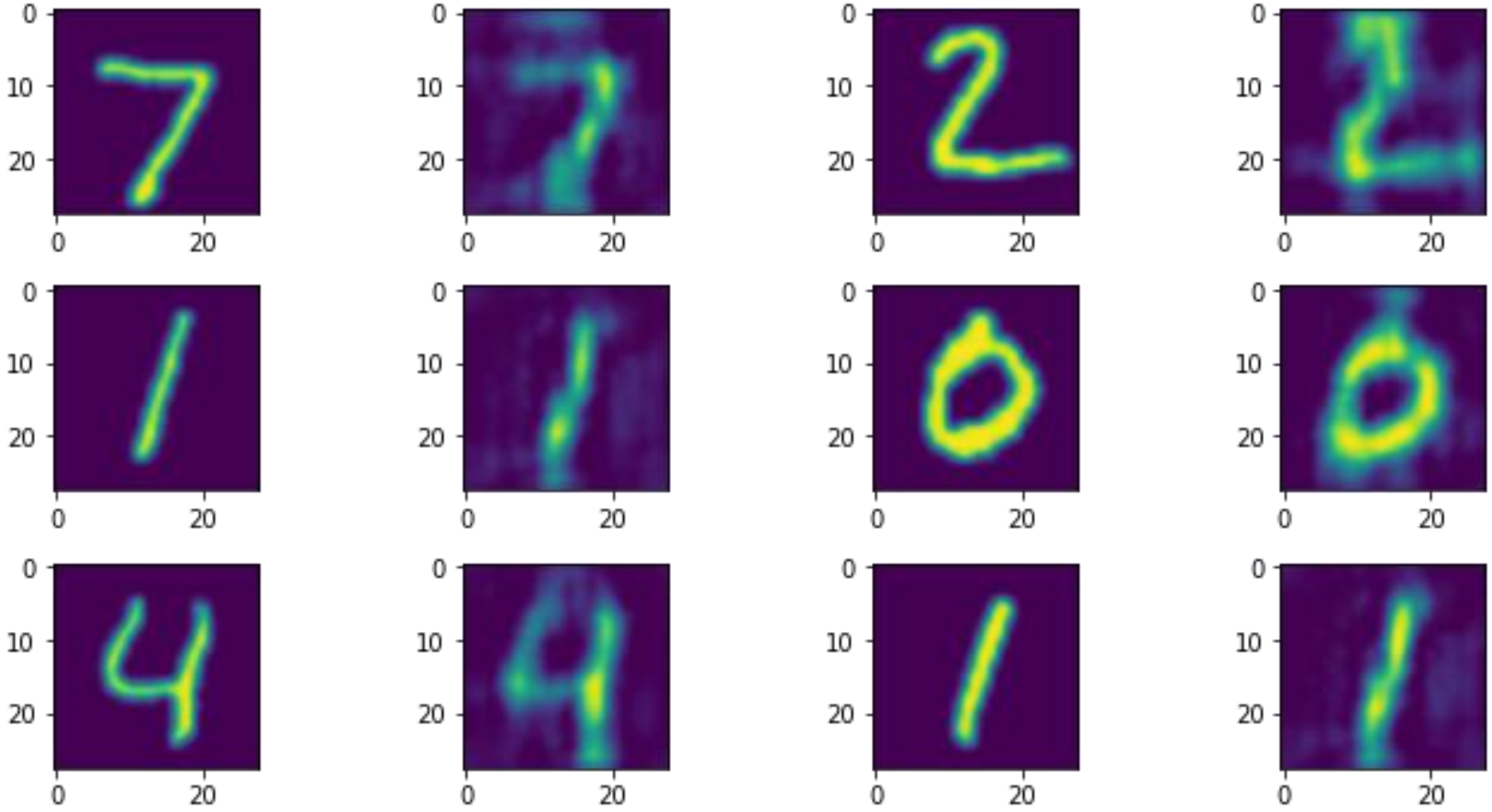
Target Model Trainings Data



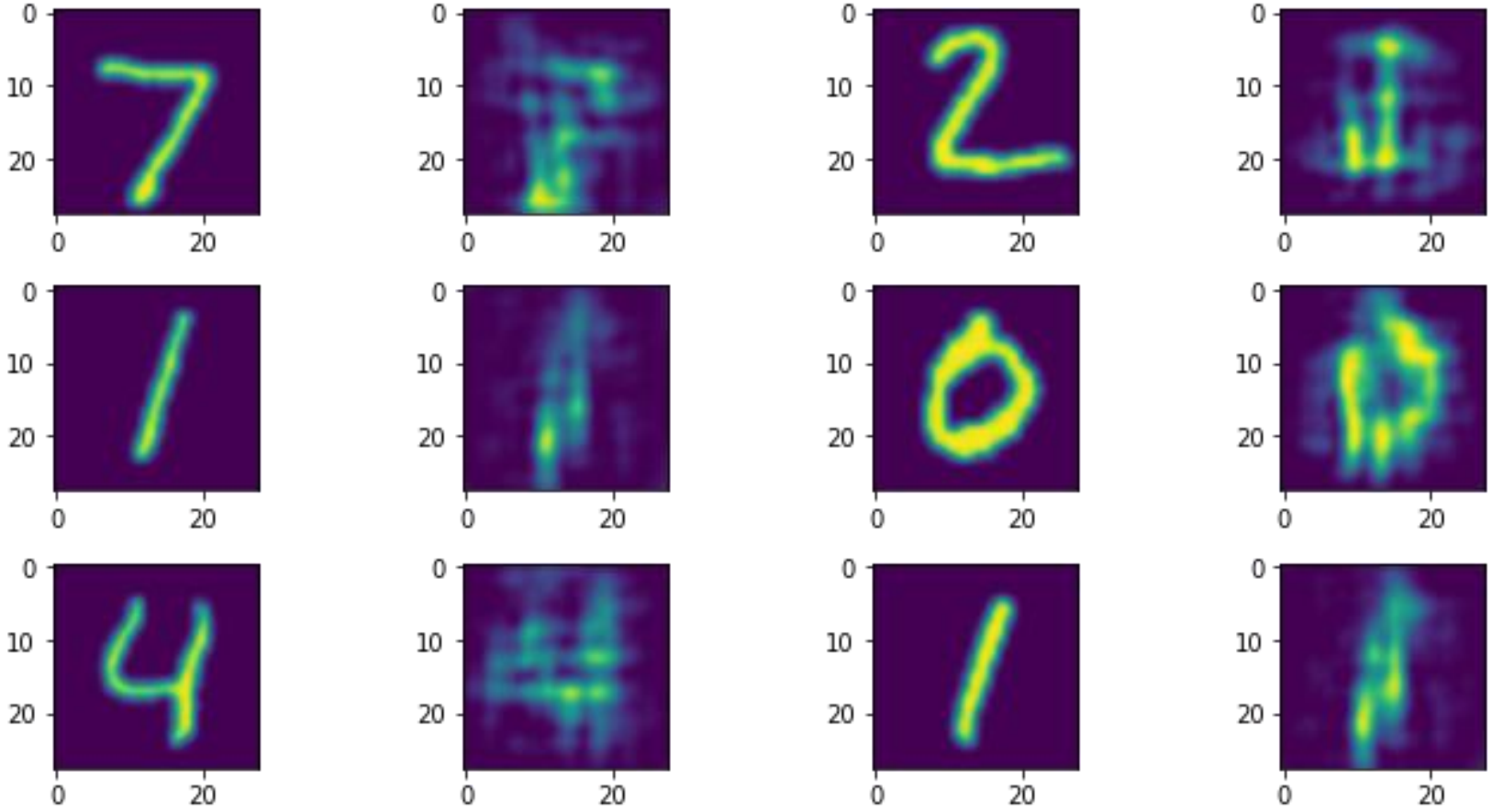
Attack Model Trainings Data



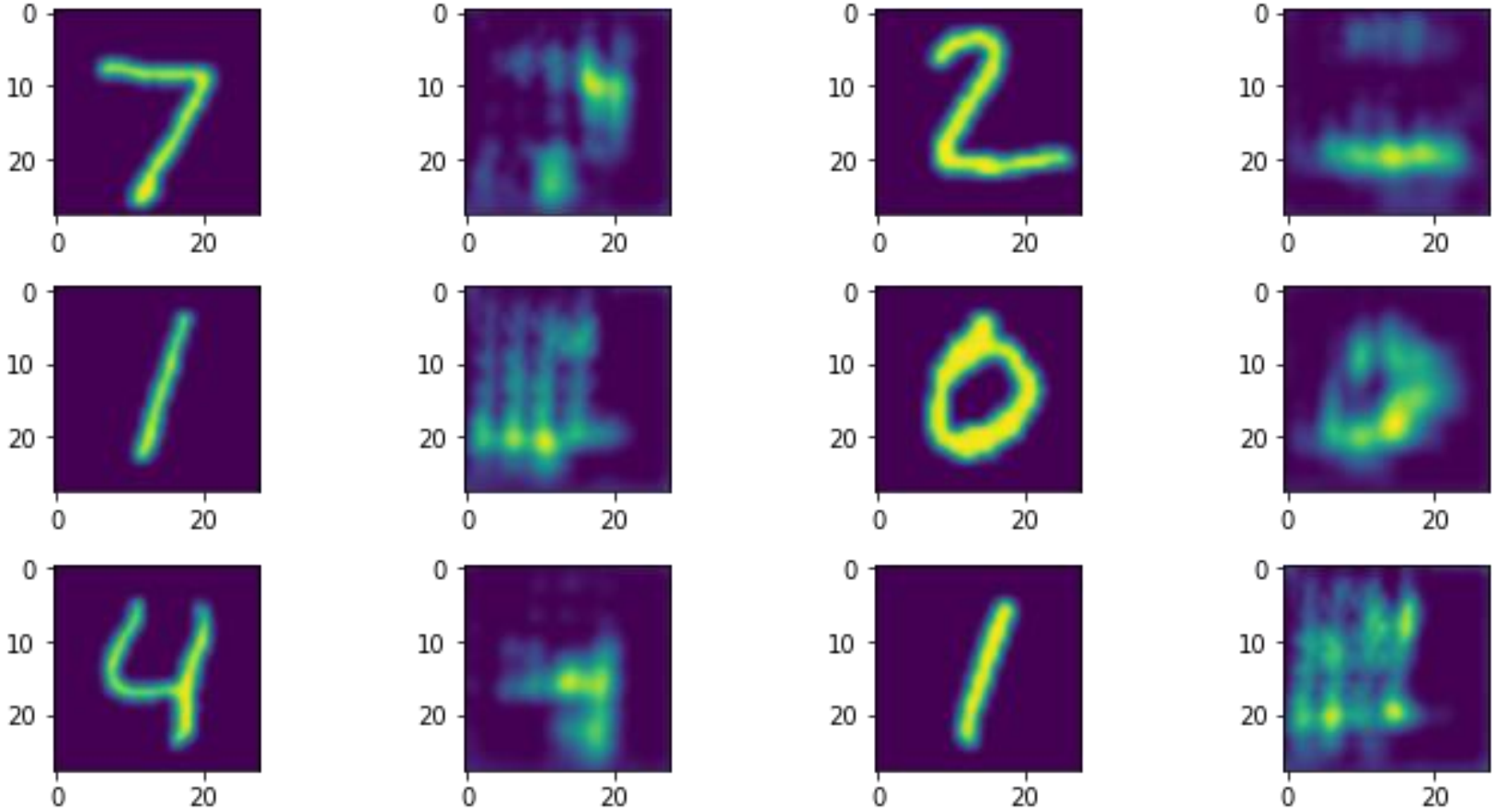
Attack on SimpleNet



Attack on AlexNet

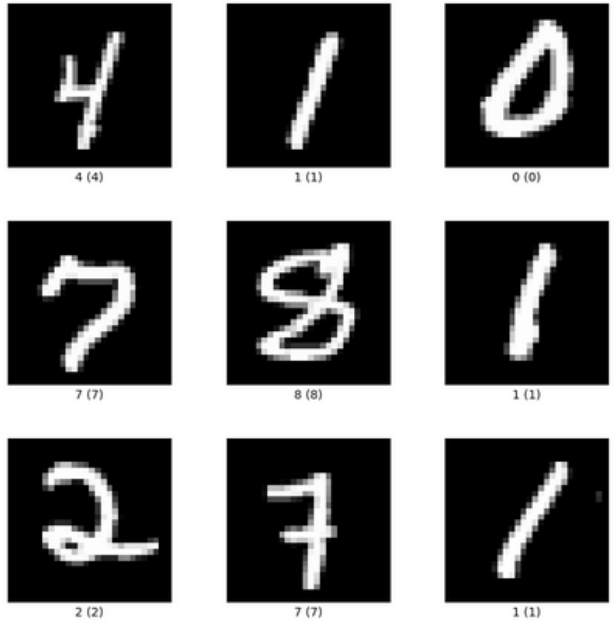


Attack on ResNet



MNIST + Random Image

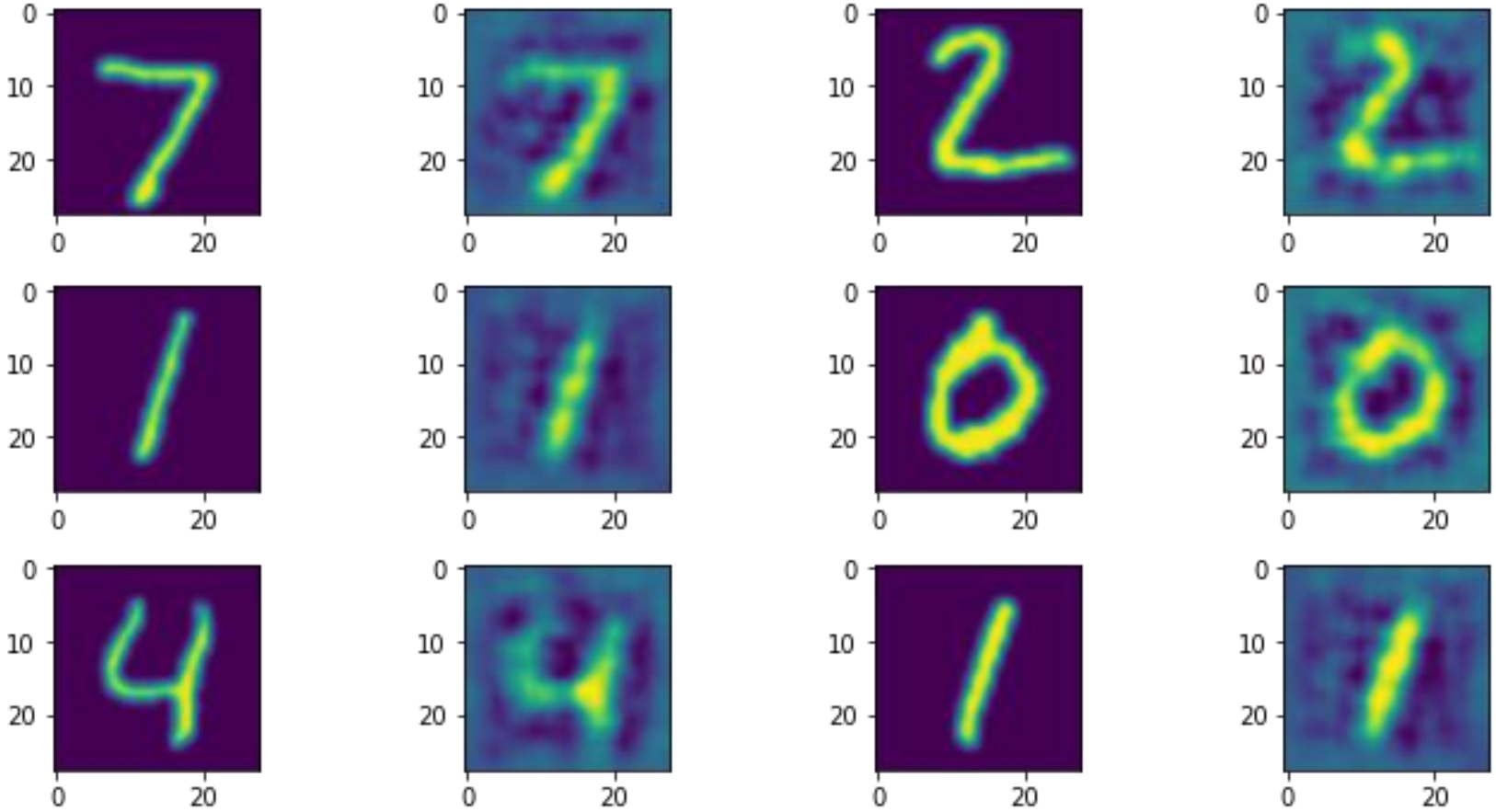
Target Model Trainings Data



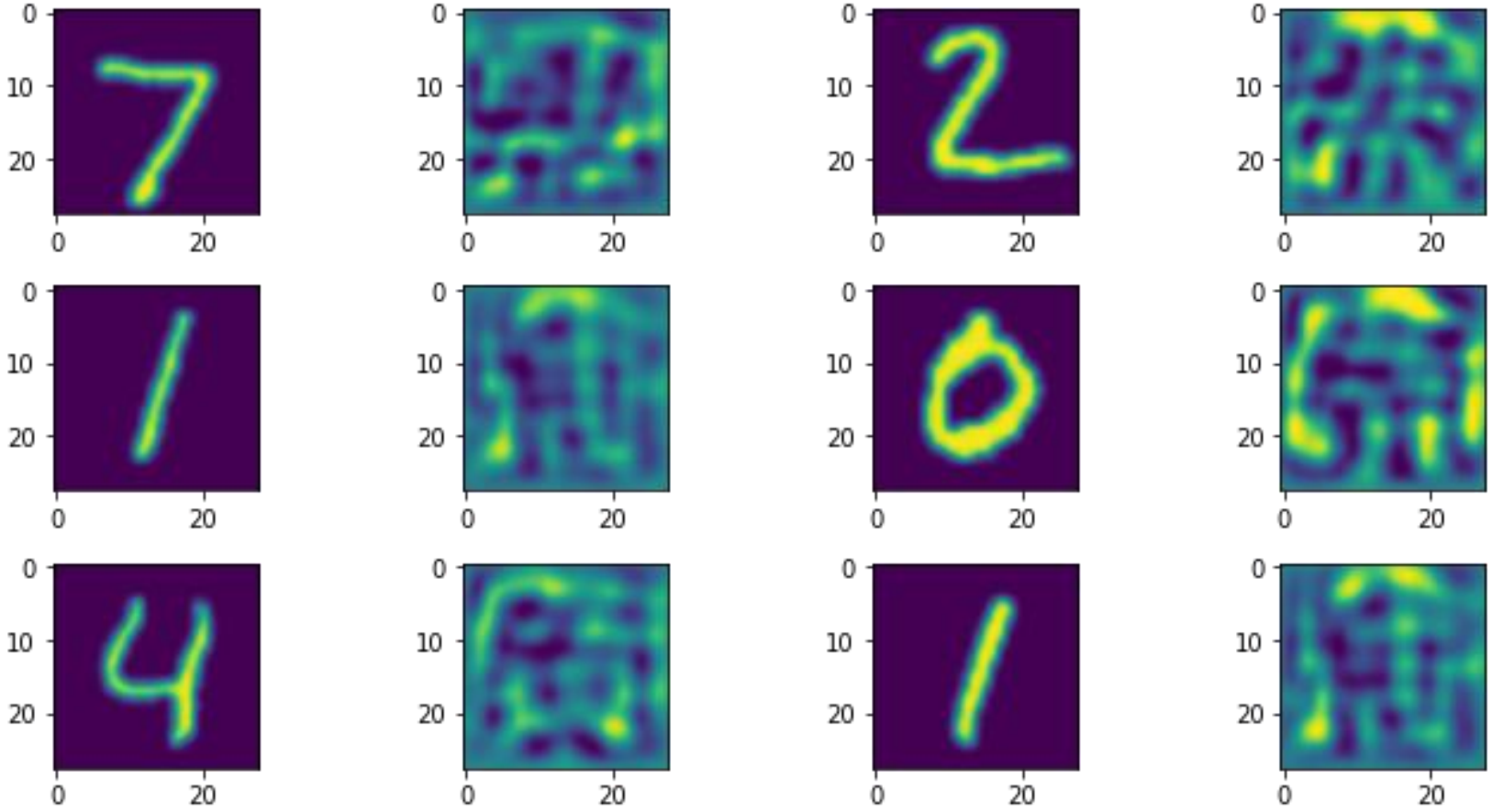
Attack Model Trainings Data

Random
Grey Image
(28x28)

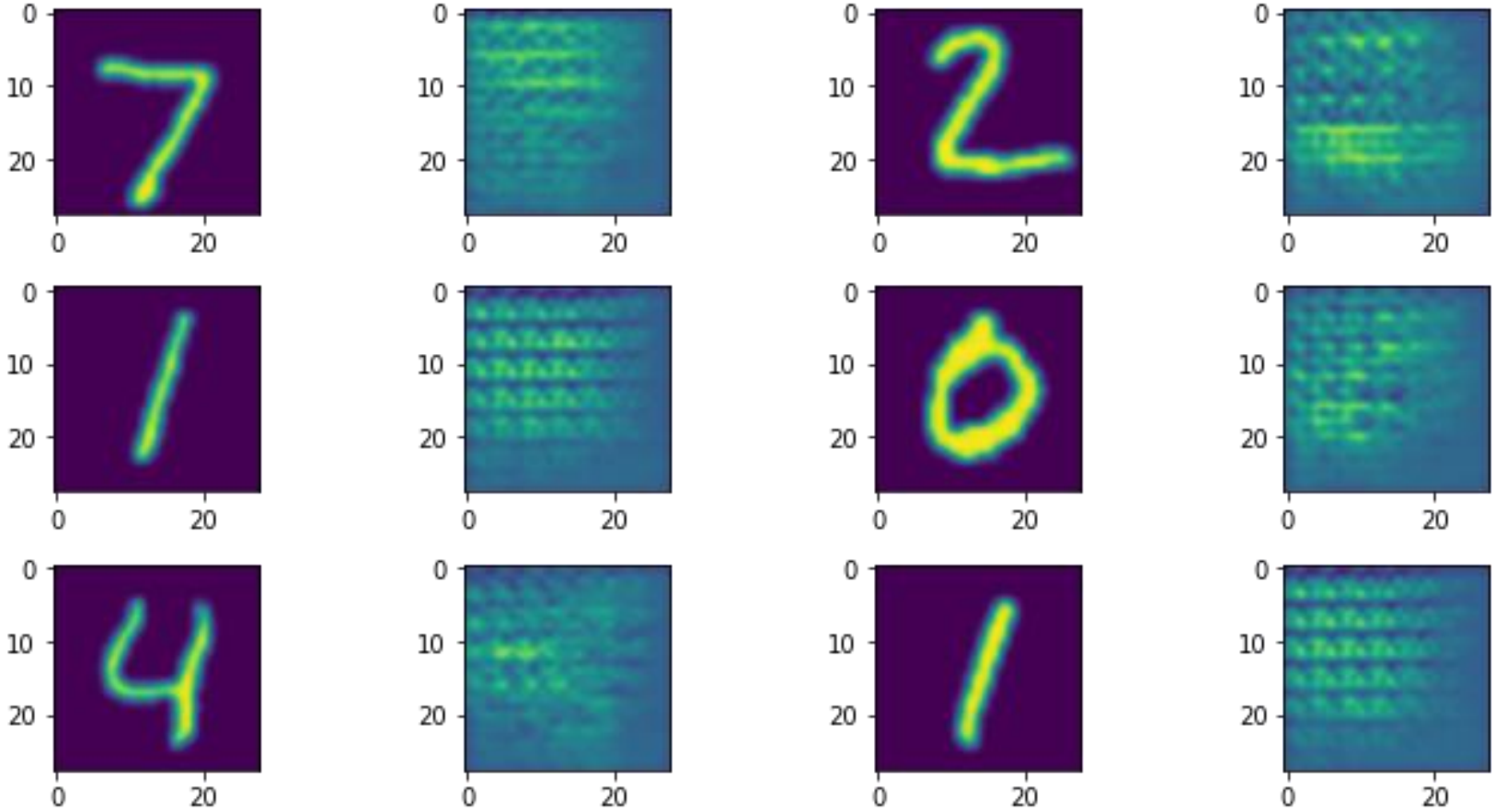
Attack on SimpleNet



Attack on AlexNet

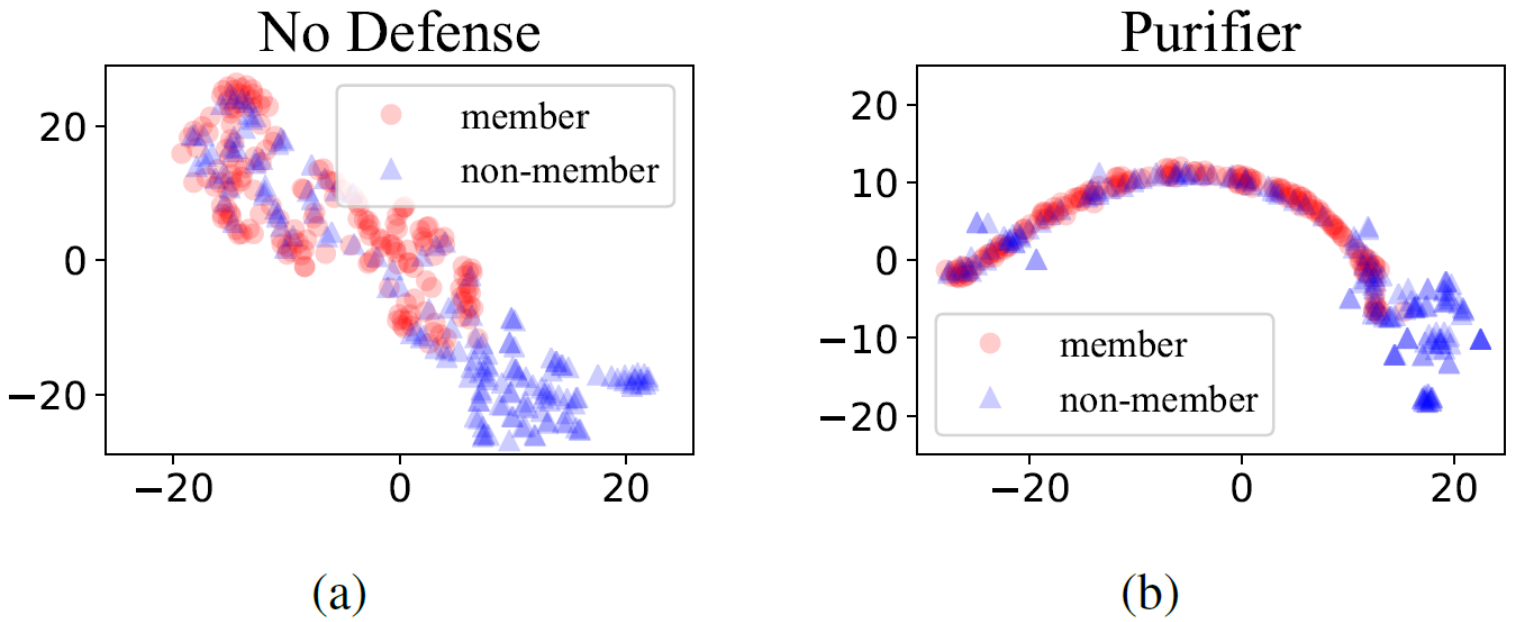


Attack on ResNet



Defending Model Inversion

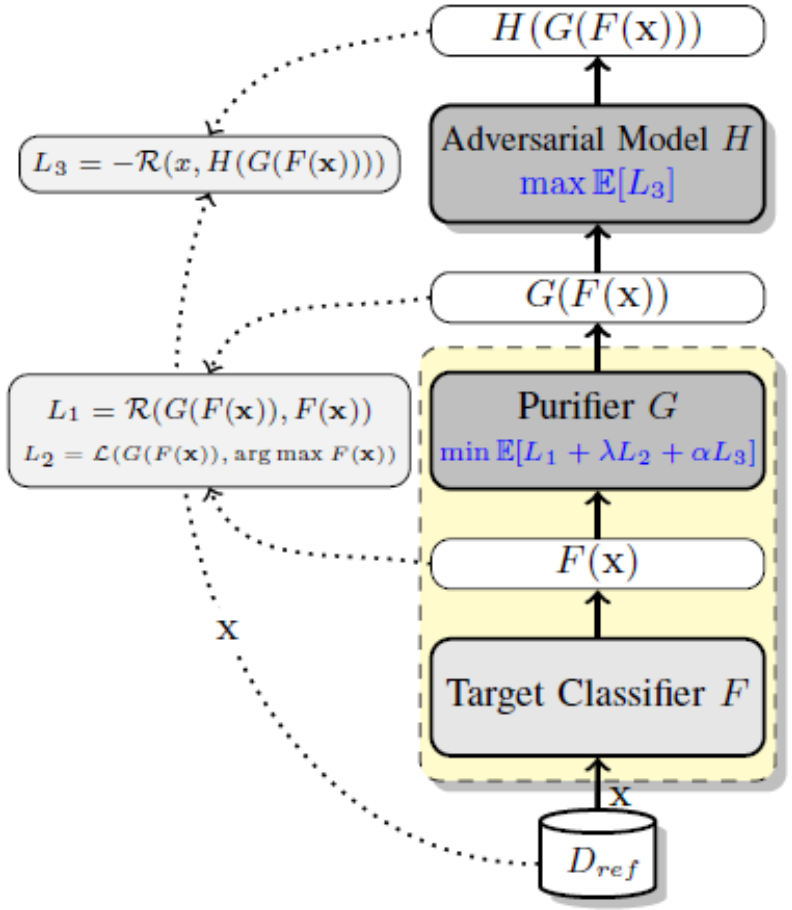
Purifying the prediction scores[1]



- Reduce the sensitivity of the prediction to the change of input data
- Negligible distortion to the original confidence scores

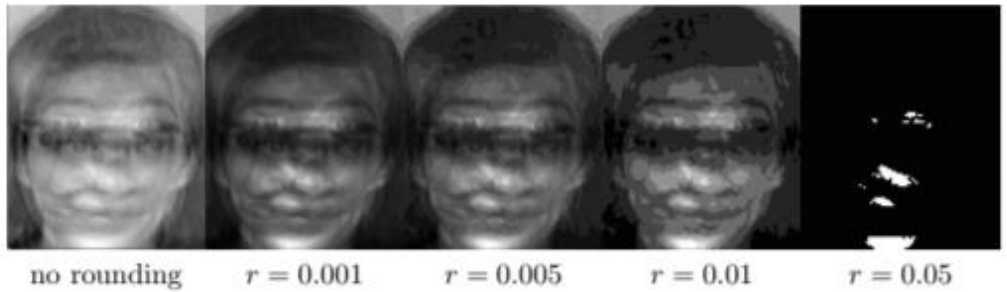
Defending Model Inversion

Purifying the prediction scores[1]



Defending Model Inversion

Rounding the gradients[2]



Degardes the quality of the gradient information

Defending Model Inversion

Differential Privacy[3]

- Designed for single server system

$$\Pr\{M(D) \in S\} \leq e^{\varepsilon} \cdot \Pr\{M(D') \in S\} + \delta$$

- Does not violate DP but still reconstructs near perfect images

So theoretically and practically we did not find anything to defend the “type” of inversion attack we used!!

Conclusion

- Our inversion attack is able to restore images that were put into a target network by a user pretty well
- But the efficiency of the attack highly depends on how complex the classifier is and which dataset an attacker chooses to train the attacking network
- There is not really a good defense against this attack, except of securing the connection between the server and the user

References

1. Yang et al. *“Defending Model Inversion and Membership Inference Attacks via Prediction Purification”.* Aug, 2020. arXiv:2205.03915

2. Fredrikson et al. “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures”. Oct, 2015. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security

3. Wang et al. “Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning”. Dec, 2018. arXiv:1812.00535V3