

# Membership Inference Attacks against Deep Learning Models

Ganesh Del Grosso

October, 2019

# Overview

- 1 Privacy Risks in ML
- 2 Membership Inference
- 3 MI in Classifiers
- 4 MI in Generative Models
- 5 Defense Mech.
- 6 References

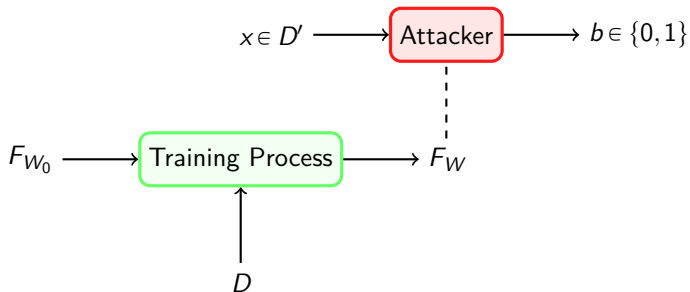


# Privacy issues in Machine Learning

- **Membership Inference:** Determine the membership of a record to a database.
- **Feature Inference/Model Inversion:** Determine properties of a given record.
- **Anonymization/Sanitation:** Safeguard the sensitive information of a record or set of records.
- **Adversarial Examples:** Cause a classification algorithm to malfunction (Security issue).

# Membership Inference

Consider a Deep Learning Model  $F_W: \mathcal{X} \rightarrow \mathcal{Y}$ , parameterized by  $W$ .

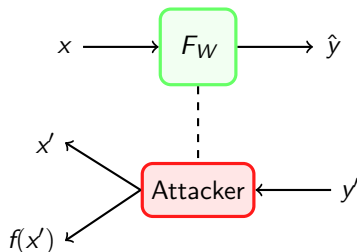


Where the attacker has no knowledge of  $D \cap D'$ .

- The dashed line denotes some degree of access to the model.

# Feature Inference/Model Inversion

- $\mathcal{X}$ : Feature Space.
- $\mathcal{Y}$ : Label Space.



- $x, x' \in \mathcal{X}$ .
- $\hat{y}, y' \in \mathcal{Y}$ .
- The dashed line denotes some degree of access to the model.

# Feature Inference/Model Inversion

**Model Inversion attacks** can, for example, recover a **person's image** from a **person's identity**.



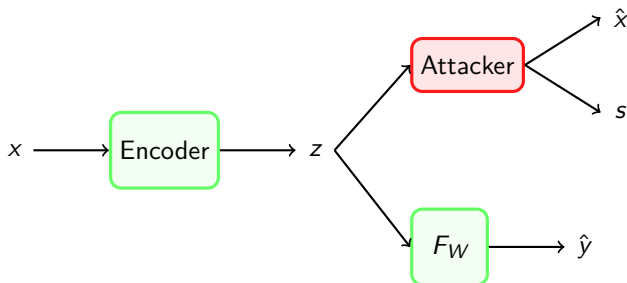
Figure: An image recovered using a new model inversion attack (left) and a training set image of the victim (right).

Image taken from [Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures \[1\]](#).

- The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

# Anonymization/Sanitation

- $\mathcal{X}$ : Feature Space,  $\mathcal{Z}$ : Latent Space.
- $\mathcal{Y}$ : Public Label Space,  $\mathcal{S}$ : Private Label Space.



- $x, \hat{x} \in \mathcal{X}$ ,  $\hat{y} \in \mathcal{Y}$ ,  $s \in \mathcal{S}$ ,  $z \in \mathcal{Z}$ .
- Database  $D \subseteq \mathcal{U}$  is sanitized by the encoder and made publicly available.



# Anonymization/Satination

The **public label** could be for example an **emotion**, while the **private label** (sensitive information) could be the **identity of a person**.

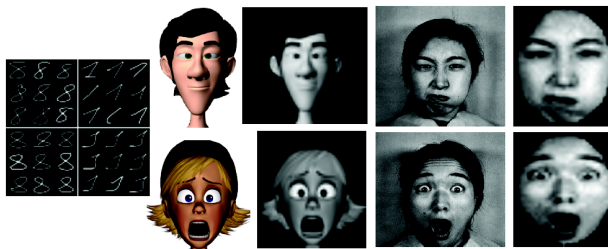


Figure: Samples of preprocessed pen-digits (images on the left), JAFEE (images on the right) and FERF (images at the center).

Image taken from [Learning Anonymized Representations with Adversarial Neural Networks](#) [2].

# Adversarial Examples

**Adversarial Examples** present a big **security risk** for machine learning models.

- **Should we trust machine learning models?**

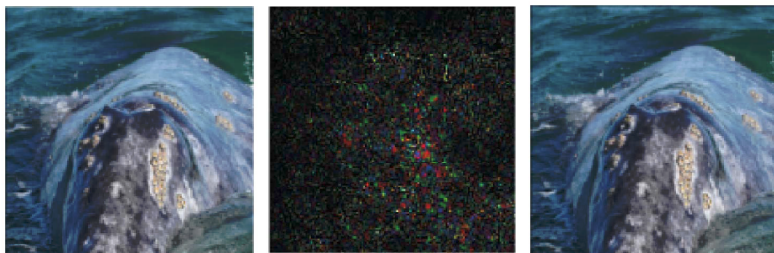


Figure: Left: Original Image correctly classified as a whale. Center: Noise crafted by the DeepFool algorithm. Right: Adversarial example wrongfully classified as a turtle.

Image taken from [DeepFool: a simple and accurate method to fool deep neural networks \[3\]](#).

- **This is a hot topic of research in Machine Learning these days.**

# Membership Inference



# Membership Inference vs. Differential Privacy

- **Differential privacy:**

- Guaranties that a model will produce statistically similar outputs independently of the presence of a particular record.
- Concerns the privacy of individuals.
- **Risk: Sensitive Information Leakage.**
- Centered around the content of input data and the output privacy (of a model).

- **Membership Inference:**

- Guaranties membership privacy.
- Concerns the privacy of individuals and of the owner of the database.
- **Risk: Membership Inference Leakage.**
- Centered around membership of the input data.

## Question!

- Are membership inference and differential privacy related?
- What is the goal of differential privacy?
- What is the goal of membership privacy?

# Motivation

- **Direct privacy breach:** When all members of a database share a certain property, membership to that database can directly violate the privacy of individuals.

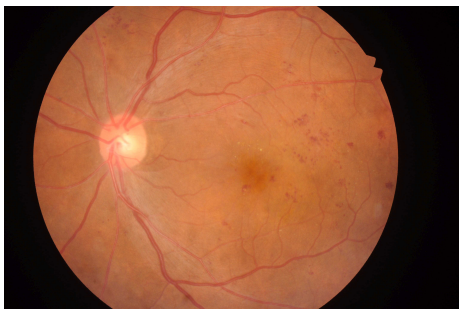


Figure: Picture of the retina for the INDIAN DIABETIC RETINOPATHY IMAGE DATASET (IDRID).

IDRID available at <https://ieee-dataport.org/open-access/indian-diabetic-retinopathy-image-dataset-idrid>.

# Motivation

- **Data is considered an organizational asset:** A trained Machine Learning model, its architecture, and its training data can be considered as valuable private information to an organization.



# Motivation

- **Data is considered an organizational asset:** A trained Machine Learning model, its architecture, and its training data can be considered as valuable private information to an organization.
- **Risk of generative models:** If a generative model is supposed to imitate the properties of a certain type of data, artificial records can be used as substitutes to original records of the same type of data. **More data allows for more robust statistics.**

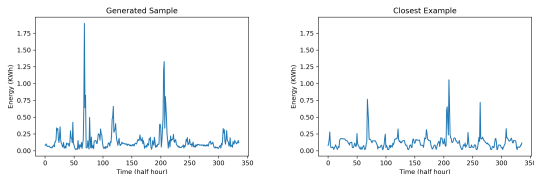


Figure: Left: Artificial power consumption curve. Right: Natural power consumption curve.

# Motivation

- **Establishing wrongdoing:** Data sometimes possesses legal restriction about its use. Using data for a purpose different to the original purpose for which it was collected might be illegal.



Figure: The NHS trust broke data protection law with Google DeepMind's Streams app. CREDIT: DEEPMIND.

Article available at <https://www.telegraph.co.uk/technology/2017/07/03/googles-deepmind-nhs-misused-patient-data-trial-watchdog-says/>.

# Motivation

- **Gateway to other attacks:** If an attacker knows that a certain record was part of the training set of a Machine Learning model, the attacker might decide to launch further attacks against the model.

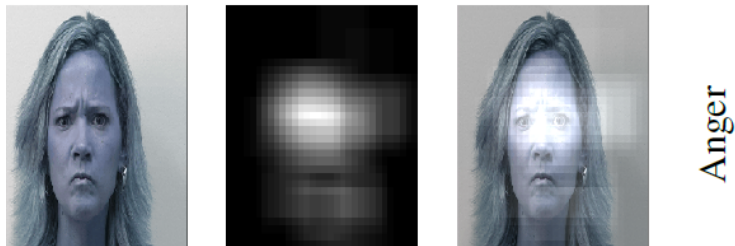


Figure: Detected salient regions for different facial expressions. Image from the Cohn-Kanade dataset. Image taken from [Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network](#) [4].

## Question!

- Does membership inference present a risk in itself against the privacy of individuals?
- What other risks does membership inference convey?

## Recap

**Inference attacks quantify leakage of information;** they fall into two fundamental and related categories:

- **Membership Inference Attacks:** The objective of the attacker is to learn if a particular individual's data was included in the training dataset.
- **Feature inference:** The objective of the attacker is to infer attributes of the records in the training set.

Accuracy of **membership inference attacks** directly demonstrates the **privacy leakage** of the model about its training data.

# Question!

- How are Membership inference and model inversion attacks related?
- How are they different?
- What is secret to the attacker in each case?
- What is known to the attacker in each case?



# Problem Formulation

- **Definition:** Given a database  $D$ , an inference model is a function  $A: \mathcal{X} \rightarrow [0,1]$ , that maps records to their membership probability,

$$A(x) = Pr(x \in D) .$$

- **Definition:** Given a Classifier  $F_W: \mathcal{X} \rightarrow \mathcal{Y}$ , trained using training set  $D$ , a black-box attacker is a function  $A: \mathcal{X} \times \mathcal{Y} \rightarrow [0,1]$ , that maps a record-prediction pair to their membership probability in the training set,

$$A(x, F_W(x)) = Pr(x \in D | F_W(x) = \hat{y}) ,$$

with  $x \in \mathcal{X}$ ,  $\hat{y} \in \mathcal{Y}$ .

The second definition corresponds to a black-box scenario, where the attacker can only query the model.



# A Naive Attacker

Consider an **attacker** that bases the membership inference attack on the **confidence of the prediction** produced by the target model.

- The **attacker** could simply take the  $l_\infty$  norm of the probability vector produced by the classifier,

$$A(x, F_W(x)) = \max_{i \in \{1, \dots, C\}} F_W(x)^i = \max_{i \in \{1, \dots, C\}} \hat{y}^i,$$

where  $C$  is the dimension of the label space  $\mathcal{Y}$ .

- Does this attack make sense?**
- How can we evaluate the efficacy of this attack?**

## Evaluation Metrics

- **Attack accuracy:** fraction of the correct membership predictions.
- **True/False positive:** true positive and false positive rates. Positive is associate with the attacker outputting “member”.

# Evaluation Metrics

- **Attack accuracy:** fraction of the correct membership predictions.
- **True/False positive:** true positive and false positive rates. Positive is associate with the attacker outputting “member”.
- **Note that, in this setup, we need a threshold in order to perform the attack.**

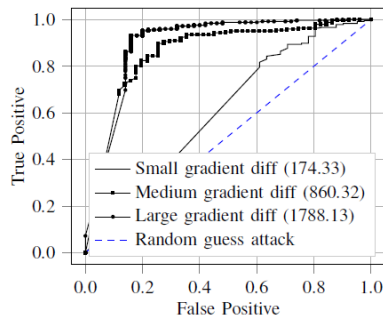


Figure: Receiver Operating Characteristic curve for Membership Inference Attacks [5].

# Problem Formulation

- **Definition:** Given a Classifier  $F_W: \mathcal{X} \rightarrow \mathcal{Y}$ , trained using training set  $D$ , a white-box attacker is a function  $A: \mathcal{X} \times \mathcal{W} \rightarrow [0,1]$ , that maps a record to its membership probability in the training set, given the target model,

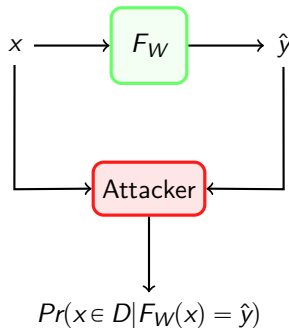
$$A(x, W) = Pr(x \in D | W) ,$$

with  $x \in \mathcal{X}$ ,  $W \in \mathcal{W}$ .  $\mathcal{W}$  is the space of model parameters.

This definition corresponds to a white-box scenario, where the attacker has access to the model and its parameters.

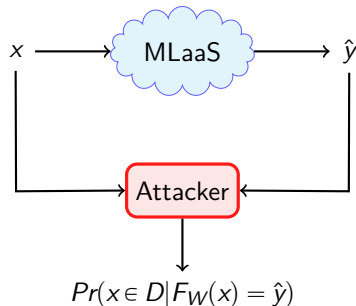
# Stand-Alone Scenario

Consider a classifier model  $F_W: \mathcal{X} \rightarrow \mathcal{Y}$  trained using training set  $D$ .



- $x \in \mathcal{X}$ .
- $\hat{y} \in \mathcal{Y}$ .
- This corresponds to a black-box attack.

# Machine Learning as a Service



- $x \in \mathcal{X}$ .
- $\hat{y} \in \mathcal{Y}$ .
- This corresponds to a black-box attack.

- **Google Prediction API:**  
<https://cloud.google.com/prediction>
- **Amazon Machine Learning:**  
<https://aws.amazon.com/machine-learning>
- **Microsoft Azure:**  
<https://studio.azureml.net>
- **BigML:** <https://bigml.com>

# Categorizing Membership Inference Attacks

- **White-box vs. Black-box:** What side information does the attacker possess?

# Categorizing Membership Inference Attacks

- **White-box vs. Black-box:** What side information does the attacker possess?
- **Stand alone vs. Federated:** How many entities participate in training?



# Categorizing Membership Inference Attacks

- **White-box vs. Black-box:** What side information does the attacker possess?
- **Stand alone vs. Federated:** How many entities participate in training?
- **Passive vs. Active:** Can the attacker influence training?

# Categorizing Membership Inference Attacks

- **White-box vs. Black-box:** What side information does the attacker possess?
- **Stand alone vs. Federated:** How many entities participate in training?
- **Passive vs. Active:** Can the attacker influence training?
- **Supervised vs. Unsupervised:** Does the attacker possess a training set for the attack?

## Attack Criteria

- **Prediction Score:** The model produces a prediction score.

$$F_W(x) = \hat{y}$$

- Confidence is expected to be higher for members of the training set.

# Attack Criteria

- **Prediction Score:** The model produces a prediction score.

$$F_W(x) = \hat{y}$$

- **Confidence is expected to be higher for members of the training set.**
- **Gradient of the loss function with respect to the model parameters:**  
During training, the loss function is minimized over the model parameters for records in the training set.

$$\min_W \mathbb{E}_{(x,y) \in D} [L(F_W(x), y)]$$

- **Gradient is expected to be lower for members of the training set.**

$$W^{t+1} = W^t - \lambda \nabla_{W^t} L(F_{W^t}(x), y)$$

# Impact of the Gradient Norm

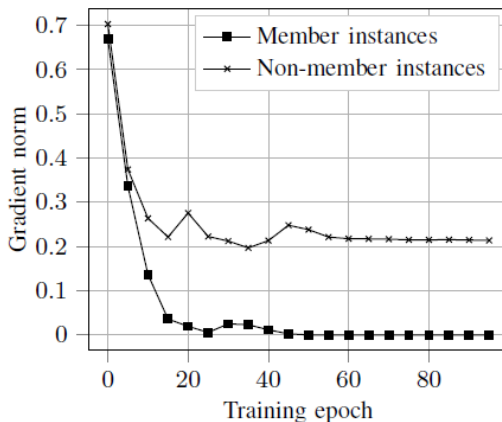


Figure: Gradient norm for member and non-member data across all classes.

Image taken from [Comprehensive Privacy Analysis of Deep Learning \[6\]](#).

## Question!

- How can we exploit the prediction of a Deep Learning model for the membership inference attack?
- How can we exploit knowledge of the model parameters for the membership inference attack?

# Supervised: Black-Box

An inference model  $A: \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow [0,1]$  which produces a membership probability,

$$A(x, \hat{y}, y) = b,$$

is trained.

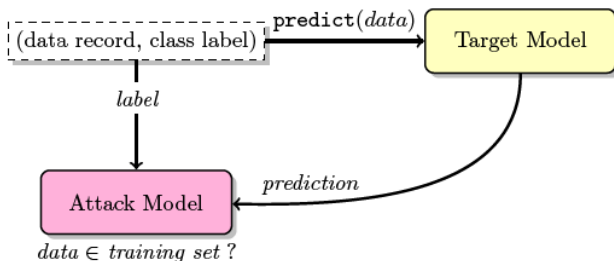


Figure: Membership Attack in the black-box setting.

Scheme taken from [Membership Inference Attacks Against Machine Learning Models \[5\]](#).

# Supervised: Black-Box

- **Training the inference model requires a training set.** Members of the training set are of the shape  $((x, \hat{y}, y), b)$ , with  $b \in \{0, 1\}$ .



## Supervised: Black-Box

- **Training the inference model requires a training set.** Members of the training set are of the shape  $((x, \hat{y}, y), b)$ , with  $b \in \{0, 1\}$ .
- **What happens when the attacker doesn't possess a training set?**

# Supervised: Black-Box

- **Training the inference model requires a training set.** Members of the training set are of the shape  $((x, \hat{y}, y), b)$ , with  $b \in \{0, 1\}$ .
- **What happens when the attacker doesn't possess a training set?**
- **Train Shadow Models.** Shadow models have the **same architecture** as the target model and are trained on data sampled from the **same distribution** as the target training data.

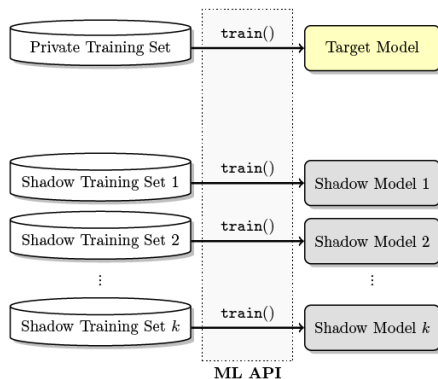


Figure: Training shadow models.

Scheme taken from [Membership Inference Attacks Against Machine Learning Models \[5\]](#).

# Question!

- What is the difference between the target model and a shadow model?
- What do the target and the shadow model have in common?
- Can you imagine a setup were it is feasible and relevant to train shadow models?

# Supervised: White-Box

The inference model has full access to the target model and **can compute any function of the target record  $x$  and the weights of the target model**. It might use:

- The model output  $f(x; \mathbf{W})$ .
- The one-hot encoding of the label  $y$ .
- The loss of the model on the target data  $L(f(x; \mathbf{W}), y)$ .
- The set of gradients  $\nabla_{\mathbf{W}} L(F_{\mathbf{W}}(x), y)$ .
- The set of activation vectors for different layers  $\mathbf{h}(x)$ .

In [Comprehensive Privacy Analysis of Deep Learning](#) [6], each of these are fed separately to the inference model, and analyzed separately using independent components.

# Question!

- Is it reasonable to ask for a training set for the attack?
- Is it necessary to train an inference model? Can we perform the membership inference attack without an inference model?

# Supervised: White-Box

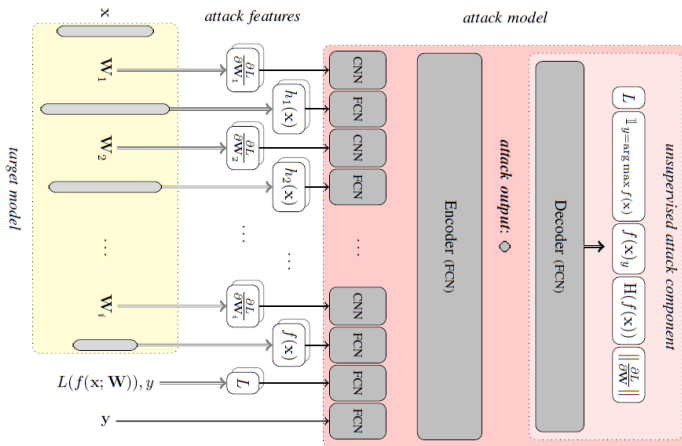
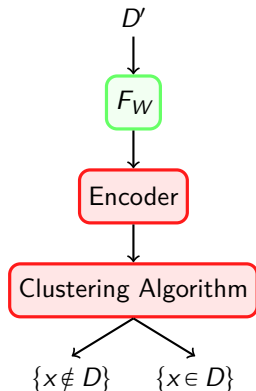


Figure: Architecture of a white-box inference attack.

Image taken from [Comprehensive Privacy Analysis of Deep Learning \[6\]](#).

# Unsupervised: White-box



For the attack:

- Use the Encoder.
- Classify using a **clustering algorithm** (K-means, spectral clustering).

To train the encoder:

- Use Encoder.
- Use Decoder to try to recover some attack features.

# Unsupervised: White-box

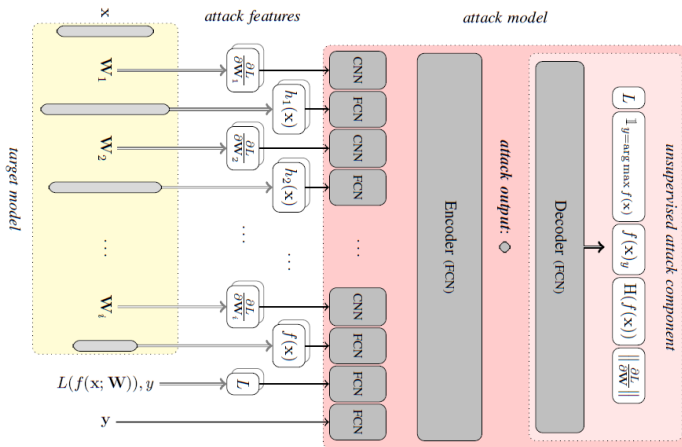


Figure: Architecture of a white-box inference attack.

Image taken from [Comprehensive Privacy Analysis of Deep Learning \[6\]](#).



## Question!

- In the unsupervised white-box attack proposed in [6] there is an encoder. What is the purpose of the encoder?
- Again, is it necessary to train an inference model? How can we get around that?

# Federated Learning

Consider a **Federated Learning** scenario, where a model is collectively trained by a set of participants.

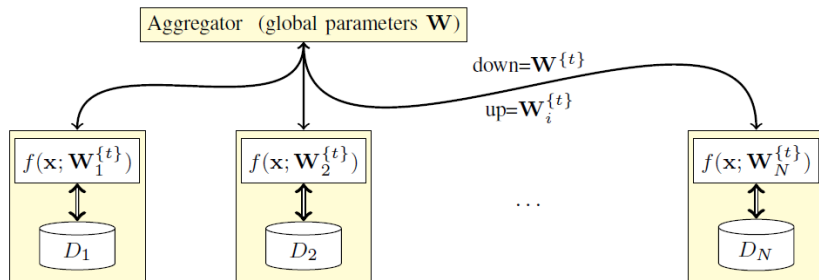


Figure: Scheme of a federated learning setup.

Scheme taken from [Comprehensive Privacy Analysis of Deep Learning \[6\]](#).

## Question!

- In the stand-alone/MLaaS case, who can take the role of the attacker?
- In the federated learning case, who can take the role of the attacker?

# Active attacker: Gradient Ascent

- Remember the update rule for **Stochastic Gradient Descent**.

$$W_i^{t+1} = W_i^t - \lambda \nabla_{W_i^t} L(F_{W_i^t}(x), y)$$

- An attacker can maliciously influence the training process by running **Gradient Ascent** on target records.

$$W_i^{t+1} = W_i^t + \gamma \nabla_{W_i^t} L(F_{W_i^t}(x), y)$$

# Active Attacker: Gradient Ascent

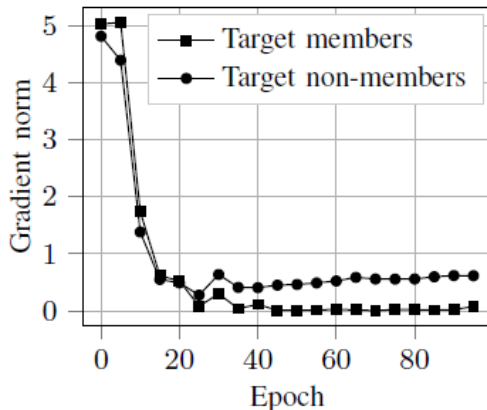


Figure: Impact of the global active gradient ascent attack on the target model's training process.  
Image taken from [Comprehensive Privacy Analysis of Deep Learning \[6\]](#).

# Question!

- In the federated learning case, when the attacker is a participant, how can she/he influence training?
- In the federated learning case, when the attacker is the global parameter aggregator, how can she/he influence training?

# Question!

- Again, is it necessary to train an inference model?
- Imagine you wanted to propose your own membership inference attack, how would you do it?
- What information is available to the attacker?  
What part of that information is really essential?

# ML in Generative Models



# Generative Models

- Generative models **allow to sample new artificial data from the underlying distributions of given training sets.**
- There has been a lot of interest in generative models in the context of **image processing**.

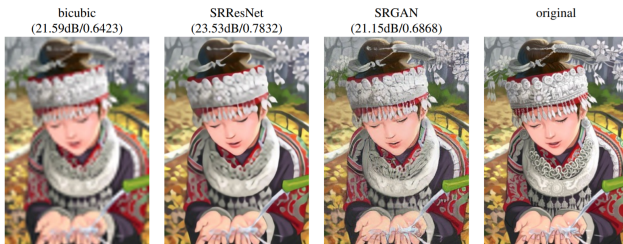


Figure: Super-resolution using different Machine Learning models. Corresponding PSNR and SSIM are shown in brackets [4× upscaling].

Image taken from [Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network \[7\]](#).

# Generative Models

Generative models are capable of generating **realistic artificial images and videos** and excel in a wide variety of tasks, such as,

- **de-noising** [8],
- **inpainting** [9],
- **compression** [10],
- **super-resolution** [7],
- **semi-supervised learning** [11].

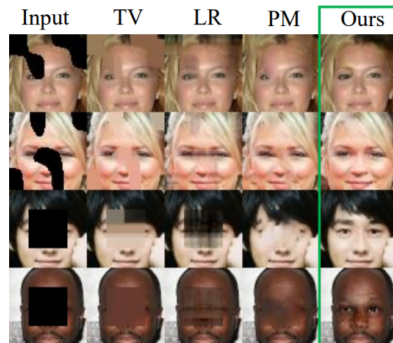


Figure: Semantic inpainting results by TV, LR, PM and Deep Generative Models.

Image taken from [Semantic Image Inpainting with Deep Generative Models](#) [9].

## Question!

- What is the goal of generative models?

# Generative Adversarial Networks

The **GAN** [12] framework has been one of the most successful in the last few years.

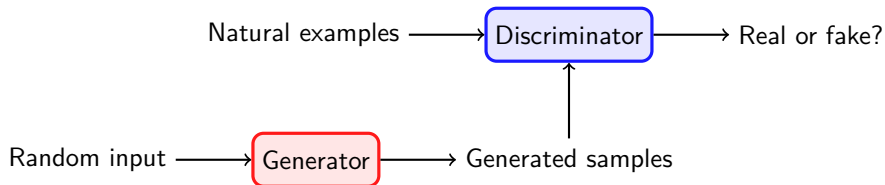


Figure: Example of images generated by CAN.

Image taken from [CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms \[13\]](#).

# Generative Adversarial Networks

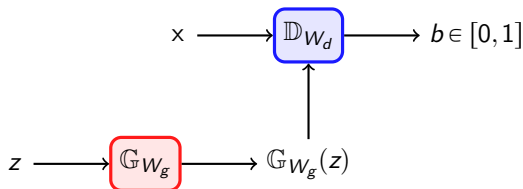
In the **Generative Adversarial Networks** framework a **Generator** and **Discriminator** play a minimax two-player game.



- The goal of the **Generator** is to produce believable samples that fool the **Discriminator**.
- The goal of the **Discriminator** is to distinguish fake samples from real examples.

# Generative Adversarial Networks

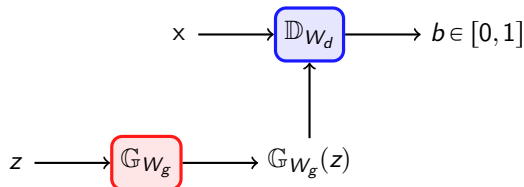
- The goal is to learn the generator's distribution  $p_g$  over data  $x$ .
- We define a prior on input noise variables  $p_z(z)$ ,
- then map to feature space as  $\mathbb{G}_{W_g}(z)$ .
- We train  $\mathbb{G}_{W_g}$  to maximize the probability of fooling the discriminator.



$$\min_{W_g} \max_{W_d} \mathbb{E}_{x \sim p_x} [\log(\mathbb{D}_{W_d}(x))] + \mathbb{E}_{z \sim p_z} [\log(1 - \mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z)))]$$

# Generative Adversarial Networks

- We also define a second network  $\mathbb{D}_{W_d}(x)$  that outputs a single scalar.
- $\mathbb{D}_{W_d}(x)$  represents the probability that  $x$  came from the data rather than  $p_g$ .
- We train  $\mathbb{D}_{W_d}$  to maximize the probability of assigning the correct label.



$$\min_{W_g} \max_{W_d} \mathbb{E}_{x \sim p_x} [\log(\mathbb{D}_{W_d}(x))] + \mathbb{E}_{z \sim p_z} [\log(1 - \mathbb{D}_{W_d}(G_{W_g}(z)))]$$

# Generative Adversarial Networks

In practice we consider separate loss functions for the Discriminator,

$$L_d(W_d) = -\frac{1}{m} \sum_{i=1}^m [\log(\mathbb{D}_{W_d}(x^{(i)})) + \log(1 - \mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z^{(i)})))] ,$$



# Generative Adversarial Networks

In practice we consider separate loss functions for the Discriminator,

$$L_d(W_d) = -\frac{1}{m} \sum_{i=1}^m [\log(\mathbb{D}_{W_d}(x^{(i)}) + \log(1 - \mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z^{(i)})))] ,$$

and Generator,

$$L_g(\theta_g) = -\frac{1}{m} \sum_{i=1}^m [\log(\mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z^{(i)})))] .$$

and train them in an alternating fashion.  $m$  is the size of a training mini-batch.

# Generative Adversarial Networks

**Algorithm 1** Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator,  $k$ , is a hyperparameter.

- 1: **for** number of training iterations **do**
- 2:   **for**  $k$  steps **do**
- 3:     Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_z(z)$ .
- 4:     Sample minibatch of  $m$  examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from data generating distribution  $p_x(x)$ .
- 5:     Update the discriminator by descending its stochastic gradient:

$$\nabla_{W_d} \left( -\frac{1}{m} \sum_{i=1}^m [\log(\mathbb{D}_{W_d}(x^{(i)})) + \log(1 - \mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z^{(i)})))] \right).$$

- 6:   **end for**
- 7:     Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_z(z)$ .
- 8:     Update the discriminator by descending its stochastic gradient:

$$\nabla_{W_g} \left( -\frac{1}{m} \sum_{i=1}^m [\log(\mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z^{(i)})))] \right).$$

- 9: **end for**

# Problem Formulation

- **Definition:** Given a database  $D$ , an inference model is a function  $A: \mathcal{D} \rightarrow P_{\mathcal{D}}$ , that partitions the database into members and non-members.

$$A(D) = \{D_m, D_m^C\}$$

- **In practice:** The attacker has a database,  $D = \{x_1, \dots, x_{n+m}\}$ , where  $n$  is the number of members of the training set in  $D$ , and  $m$  is the number of non-members of the training set.
- The attacker sorts  $D$ , and selects the  $n$  top records as the members of the training set.

$$A(D) = \tilde{D}$$

# Attack Criteria

- **Prediction Score:** The model produces a prediction score.

$$\mathbb{D}_{W_d}(x) = \hat{b}$$

- **Confidence is expected to be higher for members of the training set.**

# Attack Criteria

- **Prediction Score:** The model produces a prediction score.

$$\mathbb{D}_{W_d}(x) = \hat{b}$$

- **Confidence is expected to be higher for members of the training set.**
- **Gradient of the loss function with respect to the model parameters:**  
During training, the loss function is minimized over the model parameters for records in the training set.

$$\min_{W_d} \mathbb{E}_{x \sim p_x, z \sim p_z} [\log(\mathbb{D}_{W_d}(x) + \log(1 - \mathbb{D}_{W_d}(\mathbb{G}_{W_g}(z)))]$$

- **Gradient is expected to be lower for members of the training set.**

$$W_d^{t+1} = W_d^t - \lambda \nabla_{W_d^t} L(\mathbb{D}_{W_d^t}(x), 1)$$

# White-box Attack

The attacker has access to the trained target model, namely,

- a generator  $\mathbb{G}_{target}$  and,
- a discriminator  $\mathbb{D}_{target}$ .

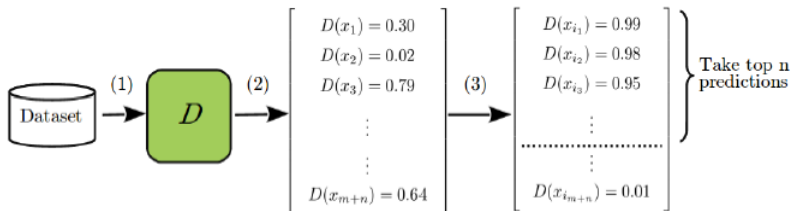


Figure: White-box Prediction Method.

Image taken from [LOGAN: Membership Inference Attacks Against Generative Models](#) [14].

# Training Performance: White-box

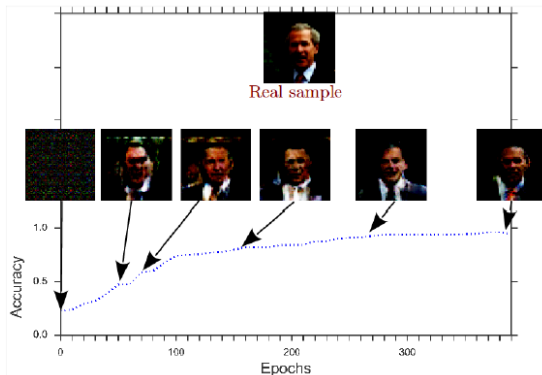


Figure: Accuracy curves and samples at different stages of training on top ten classes from the LFW dataset. Image taken from [LOGAN: Membership Inference Attacks Against Generative Models \[14\]](#).

# Question!

- Why do we have different definitions for the attacker function in the case of classifiers?
- What does the attacker exploit when attacking generative models?
- What do the attacks against generative and classifying models have in common?
- Are there any alternatives that do not rely on some sort of classifier/discriminator network?



# Black-box Attack with No Auxiliary Knowledge

- The attacker can query  $G_{target}$  to obtain generated samples.
- **In this case the target model does not necessarily need to be a GAN.**

Image taken from [LOGAN: Membership Inference Attacks Against Generative Models \[14\]](#).

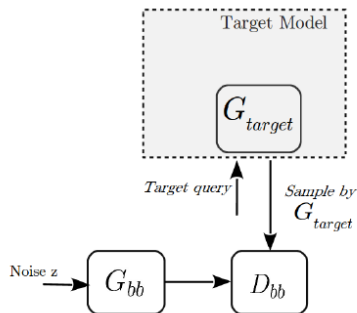


Figure: High-level overview of the black-box attack with no auxiliary knowledge.

# Black-box Attack with Limited Auxiliary Knowledge: Discriminative

The attacker has limited auxiliary knowledge of:

- Samples that were not used to train the target model or,
- **both training set and test set samples.**
- The attacker trains a simple discriminative model to infer membership.

Image taken from [LOGAN: Membership Inference Attacks Against Generative Models](#) [14].

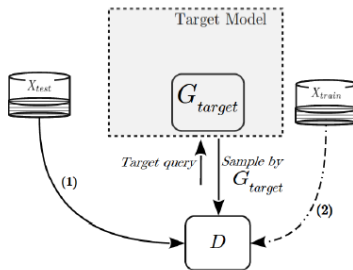


Figure: High-level overview of the discriminative black-box attack with auxiliary knowledge.

# Black-box Attack with Limiter Auxiliary Knowledge: Generative

The attacker has limited auxiliary knowledge of:

- Samples that were used to train the target model or,
- **both training set and test set samples.**
- The attacker trains a local model, namely a GAN.

Image taken from [LOGAN: Membership Inference Attacks Against Generative Models \[14\]](#).

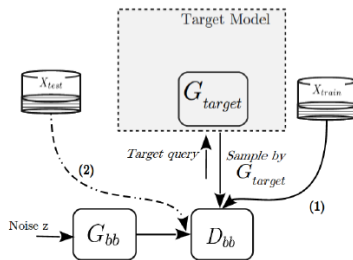


Figure: High-level overview of the generative black-box attack with auxiliary knowledge.

# Training Performance: Black-box

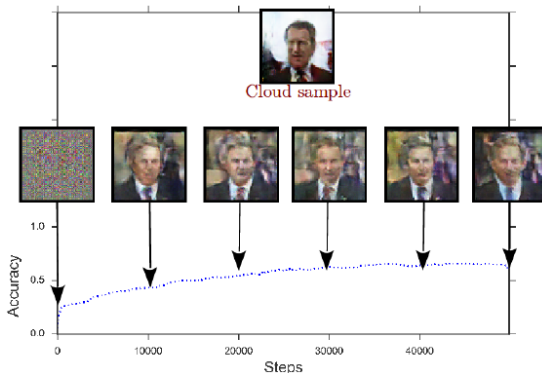


Figure: Accuracy curves and samples at different stages of training on top ten classes from the LFW dataset. Image taken from [LOGAN: Membership Inference Attacks Against Generative Models \[14\]](#).

## Question!

- What does the attacker do in the black-box case?
- What if the target generative model is not a GAN, i.e. there is no discriminator?
- Is it necessary to have access to a discriminator for the attack?



# Regularization

- **Hyper-parameter tuning:** learning rate  $\lambda$ , number of epochs for training, mini-batch size  $m$ , ...
- **$l_1, l_2$  regularization:**

$$L_{reg} = L(F_W(x), y) + \eta \|W\|$$

$$L_{reg} = L(F_W(x), y) + \eta \|W\|_2$$

- **$\epsilon - \delta$ -differential privacy:** adding Gaussian noise to the input of the Machine Learning model.

# Dropout

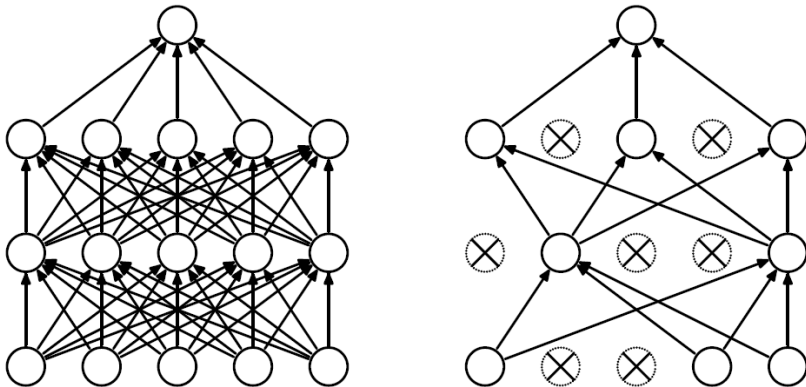


Figure: Dropout Neural Net model.

Image taken from [Dropout: A Simple Way to Prevent Neural Networks from Over-fitting](#). [15].



# Question!

- What is the interplay between over-fitting and membership inference risk?
- Do mechanisms against over-fitting help defend against membership inference?
- What is the common enemy of generalization and membership privacy?
- How can we use membership inference attacks to tune training hyper-parameters?

# References

# References I

- [1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in ACM Conference on Computer and Communications Security, 2015.
- [2] C. Feutry, P. Piantanida, Y. Bengio, and P. Duhamel, "Learning Anonymized Representations with Adversarial Neural Networks," arXiv:1802.09386 [cs, stat], Feb. 2018.  
arXiv: 1802.09386.
- [3] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," arXiv:1511.04599 [cs], July 2016.  
arXiv: 1511.04599.

## References II

- [4] S. Minaee and A. Abdolrashidi, “Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network,” [arXiv:1902.01019 \[cs\]](#), Feb. 2019.  
arXiv: 1902.01019.
- [5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership Inference Attacks against Machine Learning Models,” [arXiv:1610.05820 \[cs, stat\]](#), Oct. 2016.  
arXiv: 1610.05820.
- [6] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks,” [arXiv:1812.00910 \[cs, stat\]](#), Dec. 2018.  
arXiv: 1812.00910.

## 75 / 77

## References IV

- [10] L. Theis, W. Shi, A. Cunningham, and F. Huszár, “Lossy Image Compression with Compressive Autoencoders,” [arXiv:1703.00395 \[cs, stat\]](#), Mar. 2017.  
arXiv: 1703.00395.
- [11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved Techniques for Training GANs,” [arXiv:1606.03498 \[cs\]](#), June 2016.  
arXiv: 1606.03498.
- [12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” [arXiv:1406.2661 \[cs, stat\]](#), June 2014.  
arXiv: 1406.2661.

