Privacy in Machine Learning

Feature Inference Attacks against Deep Learning Models

Ganesh Del Grosso

November, 2019

Overview

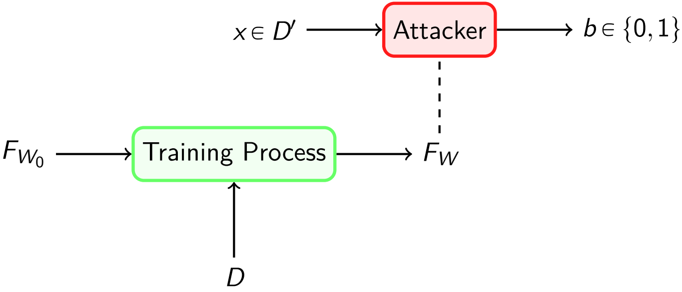
Privacy Risks in ML

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 $F_W : \mathscr{X} \to \mathscr{Y}$, parameterized by W.

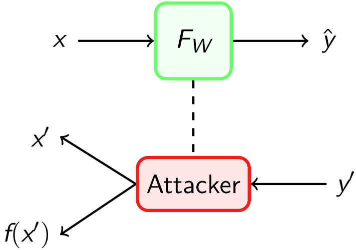


Where the attacker has no knowledge of $D \cap D'$.

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

Feature Inference/Model Inversion

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



- $x, x' \in \mathscr{X}$.
- $\hat{y}, y' \in \mathscr{Y}$.
- 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.

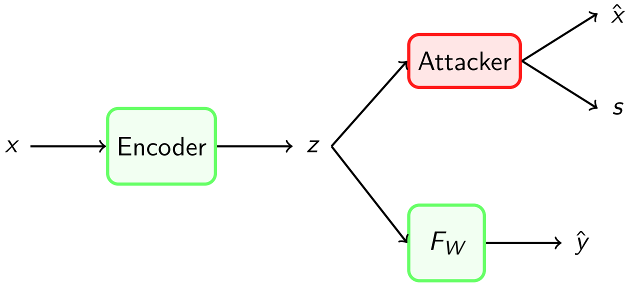
Figure: An image recovered using a new model inversion attack (left) and a training set image of the victim (right).

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

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



- $x, \hat{x} \in \mathscr{X}$, $\hat{y} \in \mathscr{Y}$, $s \in \mathscr{S}$, $z \in \mathscr{Z}$.
- Database $D \subseteq \mathscr{U}$ 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.

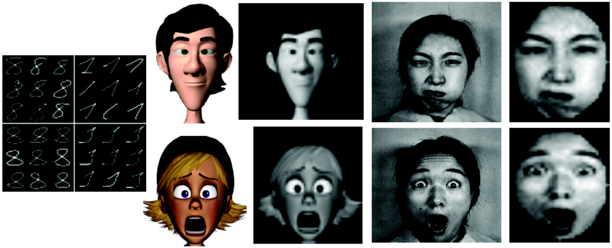


Figure: Samples of preprocessed pen-digits (images on the left), JAFEE (images on the right) and FERG (images at the center).

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?**

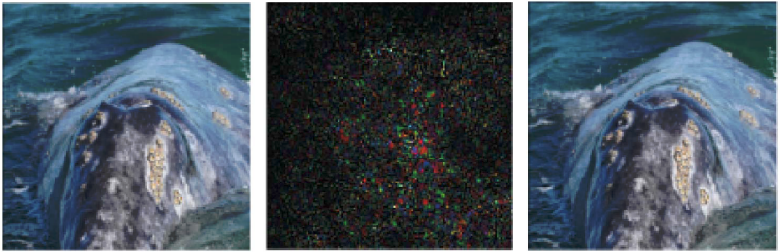


Figure: Left: Original Image correctly classified as a whale. Center: Noise crafted by the DeepFool algorithm. Right: Adversarial example wrongly classified as a turtle.

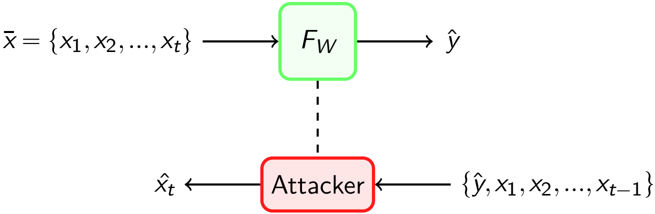
Image taken from DeepFool: a simple and accurate method to fool deep neural networks [3].

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

Feature Inference

Inferring Sensitive Features

- $\mathscr{X}$: Input Space.
- $\mathscr{Y}$: Output Space.



- $\bar{x} \in \mathscr{X}$, $\hat{y} \in \mathscr{Y}$.
- The attacker attempts to recover target sensitive feature x_t.
- The dashed line denotes some degree of access to the model.

Inferring Sensitive Features

- Consider a regression model trained with a dataset of records $D = \{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^m\}$. Each individual record is of the form $\bar{x}^i = \{x_1^i, x_2^i, \ldots, x_t^i\}$.
- In this scenario, an attacker has partial information of a record, for example $\{x_1^j, x_2^j, \ldots, x_{t-1}^j\}$, and wants to recover the rest of the information x_t^j.

		x_1	x_2	x_3	$\hat{y}$
Record	**Name**	**Age**	**Symptoms**	**Genomes**	**Dosage**
$\bar{x}^1$	Ronald Thompson	32	V,N	A,C	0.8mg
$\bar{x}^2$	Thonald Rompson	23	V,B	C,D	0.7mg
$\bar{x}^3$	Woody Stroker	27	V,N,AP	A,B	1.2mg
$\bar{x}^4$	Com Truise	44	D,V	A,D	0.9mg
$\bar{x}^5$	Robert Bobby	33	B,AP	C,D	1.5mg
$\bar{x}^6$	Pimmy Jage	75	N	A,C	0.5mg

Table: Patient records for a study of the “Heebie Jeebies” on men.

Question!

- **Does feature inference present a privacy risk for all possible records, or only for members of the training set of the target model?**
- **What is the trade-off between the generalization of the target model and its privacy?**

Problem Formulation

- Let $F_W : \mathscr{X} \rightarrow \mathscr{Y}$ be a regression model parametrized by W that maps input features $\bar{x} \in \mathscr{X}$ to predictions $\hat{y} \in \mathscr{Y}$.
- Where $\mathscr{X}$ is of the form $\mathscr{X}_1 \times \mathscr{X}_2 \times ... \times \mathscr{X}_t$, and thus $\bar{x} = \{x_1, x_2, x_3, ..., x_t\}$.
- **Definition:** A feature inference model $A_{F_W} : \mathscr{Y} \times \mathscr{X}_1 \times ... \times \mathscr{X}_{t-1} \rightarrow \mathscr{X}_t$ is a function that maps prediction $\hat{y} \in \mathscr{Y}$ and known input features $\{x_1, x_2, x_3, ..., x_{t-1}\} \in \mathscr{X}_1 \times ... \times \mathscr{X}_{t-1}$ to estimated target feature $\hat{x}_t \in \mathscr{X}_t$,

$$A_{F_W}(\hat{y}, x_1, ..., x_{t-1}) = \hat{x}_t \,,$$

where the subscript F_W denotes access to query the target model.

Problem Formulation

- Let $F_W : \mathscr{X} \rightarrow \mathscr{Y}$ be a regression model parametrized by W that maps input features $\bar{x} \in \mathscr{X}$ to predictions $\hat{y} \in \mathscr{Y}$.
- Where $\mathscr{X}$ is of the form $\mathscr{X}_1 \times \mathscr{X}_2 \times \ldots \times \mathscr{X}_t$, and thus $\bar{x} = \{x_1, x_2, x_3, \ldots, x_t\}$.
- **Definition:** A feature inference model $A_{F_W} : \mathscr{Y} \times \mathscr{X}_1 \times \ldots \times \mathscr{X}_{t-1} \rightarrow \mathscr{X}_t$ is a function that maps prediction $\hat{y} \in \mathscr{Y}$ and known input features $\{x_1, x_2, x_3, \ldots, x_{t-1}\} \in \mathscr{X}_1 \times \ldots \times \mathscr{X}_{t-1}$ to estimated target feature $\hat{x}_t \in \mathscr{X}_t$,

$$A_{F_W}(\hat{y}, x_1, \ldots, x_{t-1}) = \hat{x}_t\,,$$

  where the subscript F_W denotes access to query the target model.

For simplicity, we consider the case where the attacker knows $t-1$ features and wants to infer feature x_t; however, this is easily generalized to the case where there is more than one target feature.

Inferring Sensitive Genome Information

- As a particular example, an attacker could try to use the **Maximum *a Posteriori* Probability** (MAP) Estimate to find target feature x_t,

$$Pr[x_t|x_1, \dots, x_{t-1}, y] \propto \sum_{x' \in \hat{X}: x'_t = x_t} \prod_{1 \leqslant i \leqslant t-1} p_i \, ,$$

  where p_i are the marginals over features x'_i

- Note that the x_t with maximizes the MAP estimate also minimizes the miss-classification rate of the attacker.

Inferring Sensitive Genome Information

Algorithm 1 Feature Inference without performance statistics.

1: **INPUT:** $x_1, x_2, ..., x_{t-1}, \hat{y}, F_W, p_{1,2,...,t-1,y}$
2: Find the *feasible* set $\hat{X} \subseteq \mathscr{X}$, such that $\forall x' \in \hat{X}$: $x'_i = x_i$ for $1 \leqslant i \leqslant t-1$, and $F_W(x') = \hat{y}$
3: If $|\hat{X}| = 1$, return $\perp$
4: Return x_t that maximizes $\sum_{x' \in \hat{X}: x'_t = x_t} \prod_{1 \leqslant i \leqslant t-1} p_i$

Algorithm 2 Feature Inference with performance statistics.

1: **INPUT:** $x_1, x_2, ..., x_{t-1}, \pi, \hat{y}, F_W, p_{1,2,...,t-1,y}$
2: Find the *feasible* set $\hat{X} \subseteq \mathscr{X}$, such that $\forall x' \in \hat{X}$: $x'_i = x_i$ for $1 \leqslant i \leqslant t-1$, and $F_W(x') = \hat{y}$
3: If $|\hat{X}| = 1$, return $\perp$
4: Return x_t that maximizes $\sum_{x' \in \hat{X}: x'_t = x_t} \pi_{F_W(x'),y} \prod_{1 \leqslant i \leqslant t-1} p_i$

- where $\pi_{F_W(x'),y}$ represents the probability that the model F_W gives the true response y provided input x'.

Inferring Sensitive Genome Information

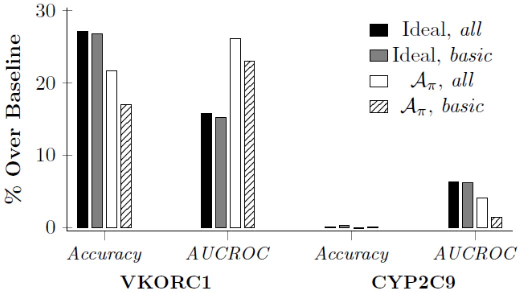
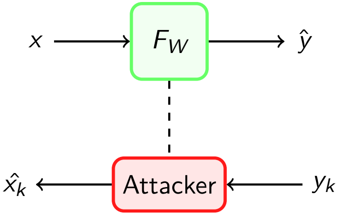


Figure: Model inversion performance, as improvement over baseline guessing from marginals.

Image taken from Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing [4].

Reconstruction Attack

- $\mathscr{X}$: Input Space.
- $\mathscr{Y}$: Output Space.



- $x, \hat{x}_k \in \mathscr{X}$, $\hat{y}, y_k \in \mathscr{Y}$.
- The attacker attempts to reconstruct a representative example $\hat{x}_k$ of class k.
- The dashed line denotes some degree of access to the model.

Reconstruction Attack

- Consider a classification model F_W trained of dataset D, a reconstruction attack attempts to produce a representative example of one of the classes of the classification problem.
- Note that this representative example is not necessarily (and most probably not) in D.

Figure: Reconstruction without using post-processing (left), with it (center), and the training set image (right).

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

Question!

- **Do reconstruction attacks present a privacy risk for all possible records, or only for members of the training set of the target model?**
- **What is the trade-off between the generalization of the target model and its privacy?**

Problem Formulation

- Let $F_W : \mathscr{X} \to \mathscr{Y}$ be a classification model parametrized by W that maps input features $x \in \mathscr{X}$ to soft probabilities $\hat{y} \in \mathscr{Y}$.
- **Definition:** A feature inference model $A_{F_W} : \mathscr{Y} \to \mathscr{X}$ is a function that maps label $y_k \in \mathscr{Y}$ into a representative member $\hat{x}_k \in \mathscr{X}$ of the target class k,

$$A_{F_W}(y_k) = \hat{x}_k \, ,$$

  where y_k denotes the one-hot encoding of class k, and the subscript F_W denotes access to query the target model.

This definition corresponds to a black-box attack.

Problem Formulation

- Let $F_W : \mathscr{X} \to \mathscr{Y}$ be a classification model parametrized by W that maps input features $x \in \mathscr{X}$ to soft probabilities $\hat{y} \in \mathscr{Y}$.
- **Definition:** A feature inference model $A : \mathscr{Y} \times \mathscr{W} \to \mathscr{X}$ is a function that maps label $y_k \in \mathscr{Y}$ and model parameters $W \in \mathscr{W}$ into a representative member $\hat{x}_k \in \mathscr{X}$ of the target class k,

$$A(y_k, W) = \hat{x}_k \, ,$$

  where y_k denotes the one-hot encoding of class k.

In this case the attacker has complete access to the target model and its parameters. This definition corresponds to a white-box attack.

Model Inversion against Face Recognition Systems

Algorithm 3 Inversion attack for Facial Recognition System.

INPUT: $k, \alpha, \beta, \gamma, \lambda$
$c(x) := 1 - F_W^k(x)$
$x_0 \leftarrow 0$
for $i \leftarrow 1, \ldots, \alpha$ **do**
    $x_i \leftarrow x_{i-1} - \lambda \nabla c(x_{i-1})$
    if $c(x_i) \geqslant max(c(x_{i-1}), c(x_{i-2}), \ldots, c(x_{i-\beta}))$ **then**
        Break
    end if
    if $c(x_i) \leqslant \gamma$ **then**
        Break
    end if
end for
Return x_i

- λ controls the rate at which we update the candidate.
- α, β and γ determine the stopping conditions for the algorithm.

Model Inversion against Face Recognition Systems

Results of the attacks agaist:

- Softmax classifier.
- Multi-layer perceptron.
- De-noising auto-encoder.

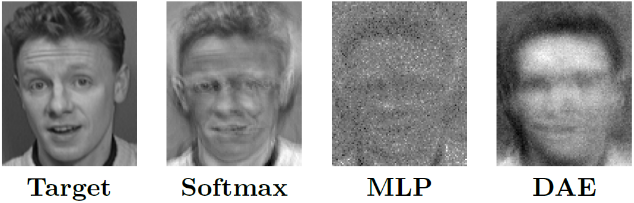


Figure: Reconstruction of the individual on the left by Softmax, MLP, and DAE.

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

Model Inversion against Face Recognition Systems

- In the black-box case, the derivatives are obtained using *scipy's numeric gradient approximation*,
- which computes the **finite difference approximation** of the gradient,

$$\frac{\partial f}{\partial x_i} = \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_N) - f(x_1, \ldots, x_{i-1}, x_i - h, x_{i+1}, \ldots, x_N)}{2h},$$

  for a small perturbation h.
- Note that the **finite difference approximation** method only requires access to query the model.

Model Inversion against Face Recognition Systems

- Rounding confidence values to the nearest r level is considered as a defense mechanism.

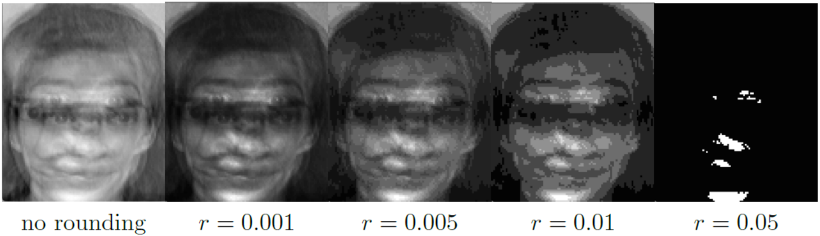


Figure: Black-box face reconstruction attack with rounding level r.

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

Question!

- **How do we compute the gradients in the white-box case?**
- **How do we compute the gradients in the black-box case?**
- **How can rounding the confidence values of the prediction help against the reconstruction attack?**

Reconstruction Attack: A Generative Approach

- Similar to what we saw before, reconstruction attack problem can be formulated in the following way,

$$\hat{x}_k = \arg\min_x [L(F_W(x), y_k) - \lambda R(x)] ,$$

where λ is a regularization hyper-parameter and $R(x)$ a regularization term.

- Now we will consider a modified definition in order to search in the latent GAN space,

$$\hat{z}_k = \arg\min_z [L(F_W(G(z)), y_k) - \lambda R(z)] .$$

- The final solution is provided by,

$$\hat{x}_k = G(\hat{z}_k) .$$

Reconstruction Attack: A Generative Approach

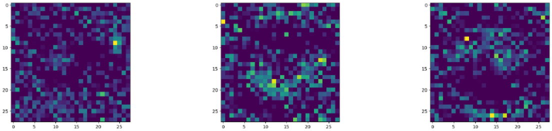


Figure: Attack on MNIST classifier without background knowledge: (Left) Retrieval of class “5”, (Middle) Retrieval of class “6”, (Right) Retrieval of class “9”.

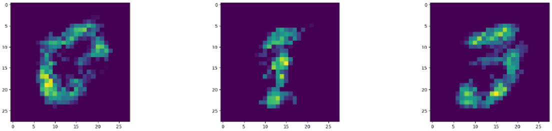


Figure: Attack on MNIST classifier with background knowledge: (Left) Retrieval of class “0”, (Middle) Retrieval of class “1”, (Right) Retrieval of class “3”.

Images taken from Membership Model Inversion Attacks for Deep Networks [5].

Reconstruction Attack: A Generative Approach

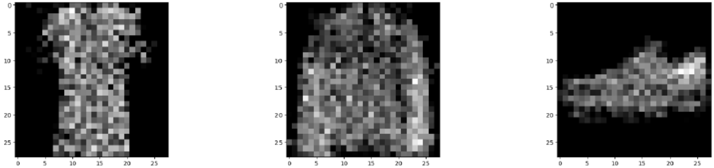


Figure: Attack on Fashion MNIST classifier with background knowledge. (Left): Retrieval of Class “T-shirts”; (Middle) Retrieval of class “Coats”; (Right) Retrieval of class “Sneakers”.

Image taken from Membership Model Inversion Attacks for Deep Networks [5].

Question!

- **What are the possible advantages of using a generative model for the reconstruction attack?**
- **What are the possible disadvantages?**

Categorizing Feature Inference Attacks

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

Categorizing Feature Inference Attacks

- **White-box vs. Black-box:** What side information does the attacker possess?
- **Regression vs. Classification:** What is the task of the target model?

Categorizing Feature Inference Attacks

- **White-box vs. Black-box:** What side information does the attacker possess?
- **Regression vs. Classification:** What is the task of the target model?
- **Reconstruction vs. Sensitive feature inference:** Does the attacker possess partial information of the records?

Learning Anonymized Representations

Adversarial Approach to Anonymization

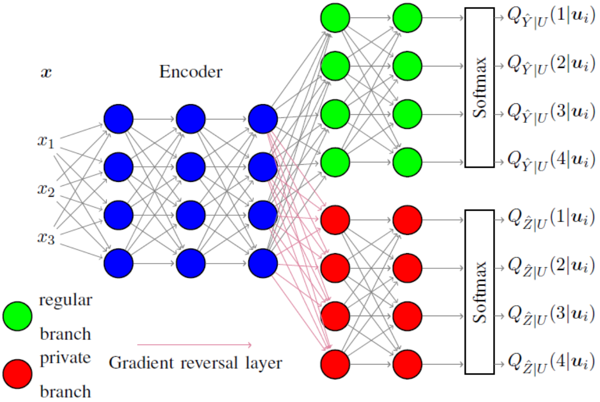


Figure: Architecture of the proposed Deep Network for anonymization.

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

Adversarial Approach to Anonymization

Figure: Anonymized representations of faces for emotion detection task.

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

Adversarial Attacks

Linear Approximation to Adversarial Examples

- Consider adversarial example $\tilde{x} = x + \eta$, where x is the original un-perturbed example and η is a small perturbation.
- Consider the product between a weight vector and an adversarial example,

$$w^T \tilde{x} = w^T x + w^T \eta \ .$$

- We would like to maximize the perturbation term $w^T \eta$ under the maximum norm constrain for noise $|| \eta ||_\infty \leqslant \varepsilon$.
- The maximum is achieved by,

$$\eta = \varepsilon sign(w) \ .$$

- Note that, even if ε is too small to be captured by a detector, the perturbation term will grow linearly on the size of w.

Linear Approximation to Adversarial Examples

- Let $F_W : \mathscr{X} \to \mathscr{Y}$ be a classifier model, we can linearize the loss function used to train the model around the current value of W to obtain an optimal max-norm constrained perturbation of,

$$\eta = \varepsilon sign(\nabla_x L(F_W(x), y)) \,,$$

this is known as the **Fast Gradient Sign Method** for computing adversarial examples.

Linear Approximation to Adversarial Examples

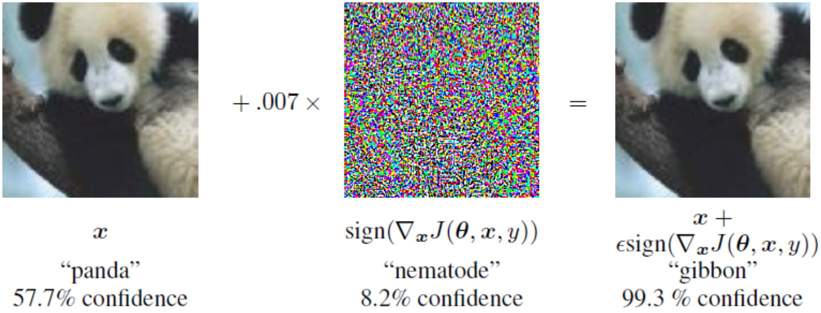


Figure: A demonstration of fast adversarial example generation applied to GoogLeNet on ImageNet.

Image taken from Explaining and Harnessing Adversarial Examples [6].

Deepfool

- Let $F_W : \mathscr{X} \to \mathscr{Y}$ be an affine classifier, i.e., $F_W(x) = W^T x$ for a given weight matrix W.
- Considering $\hat{k}(x_0)$ the original class predicted by the classifier for input x_0, the problem of finding the minimal perturbation to fool the classifier can be written as follows:
- Minimize $||r||_2$ subject to:
  1. $\exists k : w_k^T(x_0 + r) \geqslant w_{\hat{k}(x_0)}^T(x_0 + r)$

Deepfool

- Geometrically, this is equivalent to finding the projection into the convex polyhedron P defined by,

$$P = \bigcap_{k=1}^{n} \left\{ x : f_{\hat{k}(x_0)}(x) \geqslant f_k(x) \right\} ,$$

where x_0 is located inside P.

- The set P at iteration i is approximated by a polyhedron $\tilde{P}^i$,

$$\tilde{P}_i = \bigcap_{k=1}^{n} \left\{ x : f_k(x_i) - f_{\hat{k}(x_0)}(x_i) + \nabla f_k(x_i)^T x - \nabla f_{\hat{k}(x_0)}(x_i)^T x \leqslant 0 \right\} .$$

Deepfool

Algorithm 1 Deepfool Algorithm

Input: Image x, classifier f
Output: Perturbation $\hat{r}$

Initialize $x \leftarrow x_0$, $i \leftarrow 0$
while $\hat{k}(x_0) = \hat{k}(x_i)$ **do**
  for $k \neq \hat{k}(x_0)$ **do**
    $w'_k \leftarrow \nabla f_k(x_i) - \nabla f_{\hat{k}(x_0)}(x_i)$
    $f'_k \leftarrow f_k(x_i) - f_{\hat{k}(x_0)}(x_i)$
  end for
  $\hat{l} \leftarrow arg\ min_{k \neq \hat{k}(x_0)} \frac{|f'_k|}{||w'_k||_2}$
  $r_i \leftarrow \frac{|f'_{\hat{l}}|}{||w'_{\hat{l}}||_2^2} w'_{\hat{l}}$
  $x_{i+1} \leftarrow x_i + r_i$
  $i \leftarrow i + 1$

end while
Return: $\hat{r} = \sum_i r_i$

Figure: Deepfool Algorithm.

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

Robust Nets by Dropout

Algorithm 1 Stochastic Activation Pruning (SAP)

Input: input datum x, neural network with n layers, with i^{th} layer having weight matrix W^i, non-linearity ϕ^i and number of samples to be drawn r^i.

$h^0 \leftarrow x$

for each layer i **do**

  $h^i \leftarrow \phi^i(W^i h^{i-1})$ ▷ activation vector for layer i with dimension a^i

  $p^i_j \leftarrow \frac{|(h^i)_j|}{\sum_{k=1}^{a^i} |(h^i)_k|}, \forall j \in \{1, \ldots, a^i\}$ ▷ activations normalized on to the simplex

  $S \leftarrow \{\}$ ▷ set of indices not to be pruned

  repeat r^i **times** ▷ the activations have r^i chances of being kept

    Draw $s \sim \text{categorical}(p^i)$ ▷ draw an index to be kept

    $S \leftarrow S \cup \{s\}$ ▷ add index s to the keep set

  for each $j \notin S$ **do**

    $(h^i)_j \leftarrow 0$ ▷ prune the activations not in S

  for each $j \in S$ **do**

    $(h^i)_j \leftarrow \frac{(h^i)_j}{1-(1-p^i_j)^{r^i}}$ ▷ scale up the activations in S

return h^n

Figure: Stochastic Activation Pruning Algorithm.

Image taken from Stochastic Activation Pruning for Robust Adversarial Defense [7].

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] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing," in Proceedings of the 23rd USENIX Security Symposium, pp. 17–32, 2014.

[5] S. Basu, R. Izmailov, and C. Mesterharm, "Membership Model Inversion Attacks for Deep Networks," arXiv:1910.04257 [cs, stat], Oct. 2019. arXiv: 1910.04257.

[6] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in International Conference on Learning Representations, 2015.

[7] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic Activation Pruning for Robust Adversarial Defense," arXiv:1803.01442 [cs, stat], Mar. 2018. arXiv: 1803.01442.