TOWARDS REVERSE-ENGINEERING BLACK-BOX NEURAL NETWORKS

(Seong Joon Oh, Max Augustin, Bernt Schiele, Mario Fritz)

CS6316 ML Final Project
Team Taiwan
Hannah Chen, James Ku, Li-Pang Huang

12/06/2019

Motivation

Black-box models usually hide **internal states** on purpose:

1. Protecting intellectual properties (IP)
2. Covering privacy-sensitive training data

Motivation

Black-box models usually hide **internal states** on purpose:

1. Protecting intellectual properties (IP)
2. Covering privacy-sensitive training data

Why hiding the information?

1. Preventing the model from adversarial attacks
2. Protecting privacy data, such as faces

Motivation

In order to **increase the chance of protecting the model** from being attacked, we need to gain more knowledge on black-box models.

Motivation

In order to **increase the chance of protecting the model** from being attacked, we need to gain more knowledge on black-box models.

Double-sided blade:
Disclosing the hidden detail may make the model much **easier to be attacked** by adversaries

Background

1. Model attributes:
    a. architecture (non-linear activation)
    b. optimisation process (SGD or ADAM)
    c. training data

Background

2. Metamodel:

   - Takes models as input and returns the corresponding model attributes as output

3. Meta-training set:

   - a diverse set of white-box models with different model attributes

Background

A standard supervised learning task applied over models

1. Collect meta-training set
2. Train metamodel by using meta-training set
3. Predict attributes for black-box models

Related Work on Extracting Model Information

- Model extraction via querying ML APIs
  - (Tramer et al., 2016): reconstruct the exact model parameters
  - (Papernot et al., 2017): build a local avatar model
- Extracting information from the training data
  - (Ateniese et al., 2015) build a meta-classifier to obtain statistical information about the training set
  - (Shokri et al., 2017) proposed membership inference attack that can determine if a given data sample is part of the training data

Attacking Black-box Models Using Extracted Information

- **Adversarial image perturbations** (AIPs): small imperceptible perturbations over the input that fool the target model
- Approaches:
  - Gradient / saliency map attacks
    - Problem --> requires millions of queries to find a single AIP
  - Avatar approach: train a local white box model similar to the target model
  - Exploit transferability of adversarial examples that generated for one model to attack other models

Claim / Target Task

- Attributes of neural networks can be exposed from a sequence of queries
- Revealed internal information helps generate more effective adversarial examples against the black box model

An Intuitive Figure Showing WHY Claim

Collect Meta-training set → Train Metamodel → Query Black-box Model

Predict Black-box Model Attributes → Train A Local Model using Predicted Attributes → Attack Target Model

Proposed Solution

METAMODELS

- Classifier of classifiers
- Uses model f as black box
- Submits n query inputs to f
- Takes corresponding model outputs as input
- Returns predicted attributes as output

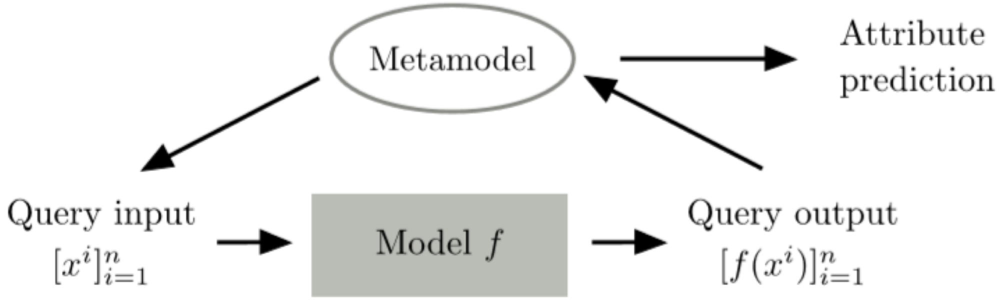
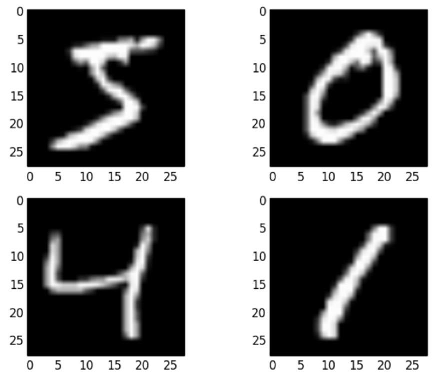


Figure 1: Overview of our approach.

Preparing traning data



MNIST-NETS

- 12 attributes
- 18,144,000 combinations

Sample 10000

pruned low-performance classifiers

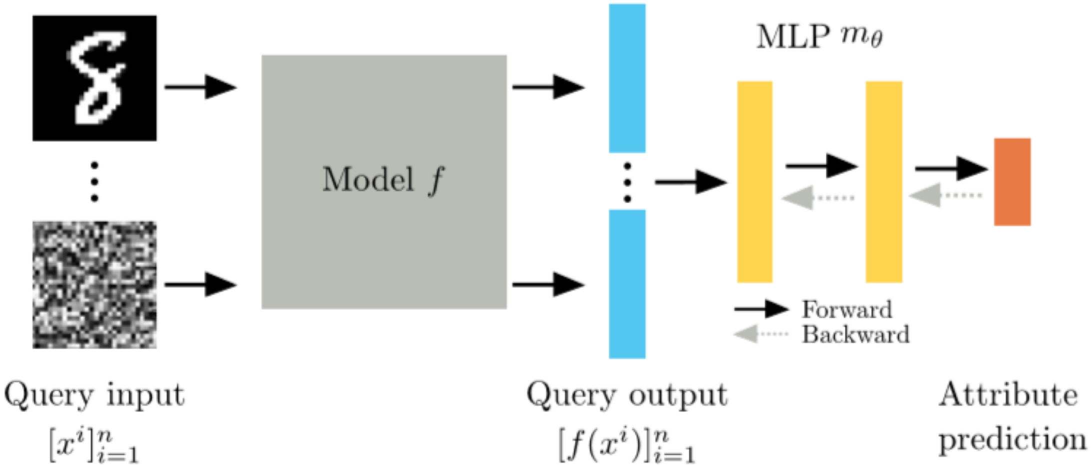
(validation accuracy< 98%)

Table 1: MNIST classifier attributes. *Italicised* attributes are derived from other attributes.

	Code	Attribute	Values
Architecture	act	Activation	ReLU, PReLU, ELU, Tanh
	drop	Dropout	Yes, No
	pool	Max pooling	Yes, No
	ks	Conv ker. size	3, 5
	#conv	#Conv layers	2, 3, 4
	#fc	#FC layers	2, 3, 4
	#par	*#Parameters*	$2^{14}, \cdots, 2^{21}$
	ens	Ensemble	Yes, No
Opt.	alg	Algorithm	SGD, ADAM, RMSprop
	bs	Batch size	64, 128, 256
Data	split	Data split	All_0, $\mathrm{Half}_{0/1}$, $\mathrm{Quarter}_{0/1/2/3}$
	size	*Data size*	All, Half, Quarter

KENNEN-O: REASON OVER OUTPUT

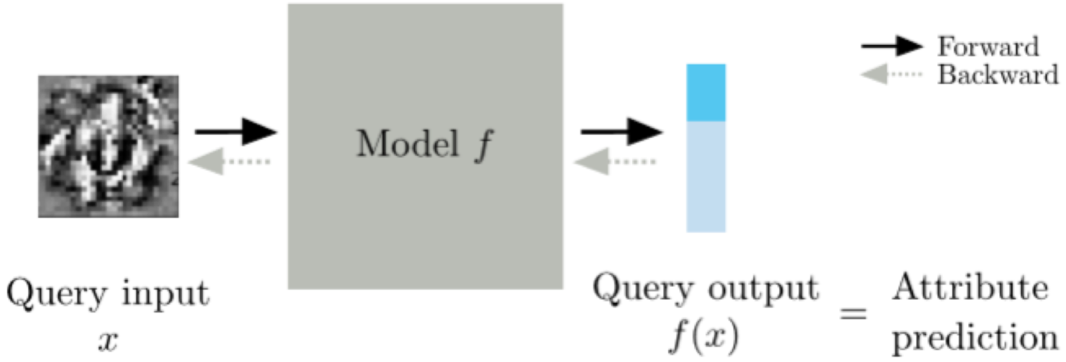
- Submits a fixed query of images to f as inputs (Fixed across training and testing)
- Takes the output from f and predicts the 12 attributes



$$\min_{\theta} \mathop{\mathbb{E}}_{f\sim\mathcal{F}} \left[\sum_{a=1}^{12} \mathcal{L}\left(m_\theta^a \left([f(x^i)]_{i=1}^n \right), y^a \right) \right]$$

KENNEN-I: CRAFT INPUT

- Can only predict a single attribute at a time
- Crafts an input that drives f to leak internal information
- Limited predictable classes



$$\min_{x:\text{ image}} \mathop{\mathbb{E}}_{f\sim\mathcal{F}} \left[\mathcal{L}\left(f(x), y^a\right)\right]$$

KENNEN-IO: COMBINED APPROACH

- Overcomes the drawbacks of `kennen-i`: single attribute prediction
- Combine `kennen-o` and `kennen-i` approaches

  (Input generator + output interpreter)
- Support optimization of multiple query inputs

$$\min_{[x^i]_{i=1}^n : \text{ images}} \min_{\theta} \mathop{\mathbb{E}}_{f \sim \mathcal{F}} \left[\sum_{a=1}^{12} \mathcal{L}\left(m_\theta^a \left([f(x^i)]_{i=1}^n \right), y^a \right) \right].$$

Experimental Results

100 queries are used for every methods, except for kennen-i, which uses a single query

Method	Output	act	drop	pool	ks	#conv	#fc	#par	ens	alg	bs	size	split	avg
		architecture								optim		data		
Chance	-	25.0	50.0	50.0	50.0	33.3	33.3	12.5	50.0	33.3	33.3	33.3	14.3	34.9
kennen-o	prob	80.6	94.6	94.9	84.6	67.1	77.3	41.7	54.0	71.8	50.4	73.8	90.0	73.4
kennen-o	ranking	63.7	93.8	90.8	80.0	63.0	73.7	44.1	**62.4**	65.3	47.0	66.2	86.6	69.7
kennen-o	bottom-1	48.6	80.0	73.6	64.0	48.9	63.1	28.7	52.8	53.6	41.9	45.9	51.4	54.4
kennen-o	top-1	31.2	56.9	58.8	49.9	38.9	33.7	19.6	50.0	36.1	35.3	33.3	30.7	39.5
kennen-i	top-1	43.5	77.0	94.8	88.5	54.5	41.0	32.3	46.5	45.7	37.0	42.6	29.3	52.7
kennen-io	score	**88.4**	**95.8**	**99.5**	**97.7**	**80.3**	**80.2**	**45.2**	60.2	**79.3**	**54.3**	**84.8**	**95.6**	**80.1**

Comparison of metamodel methods

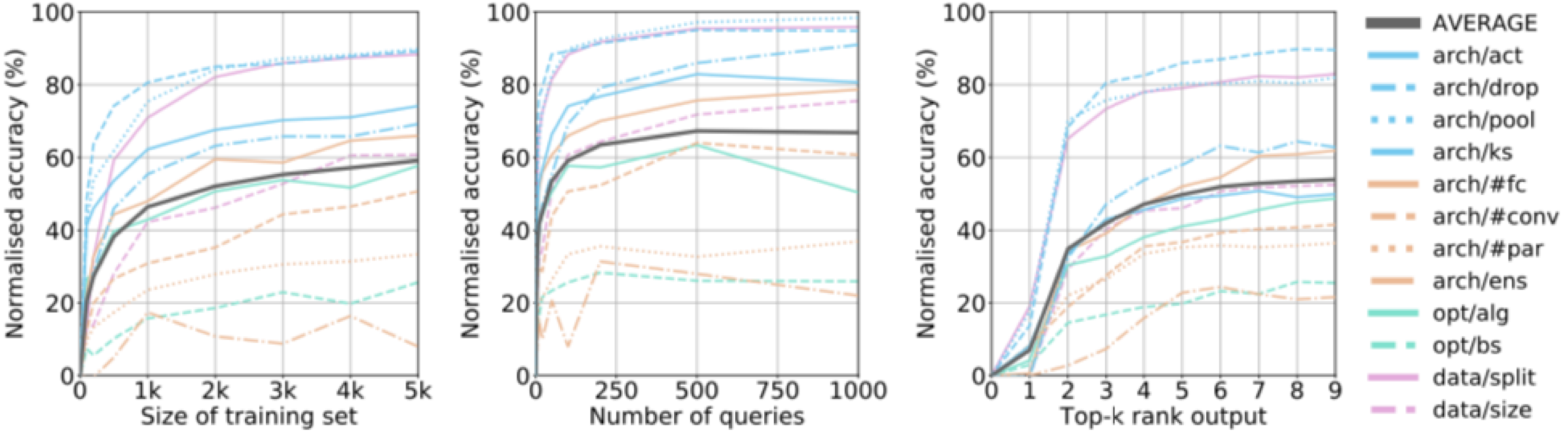
- `kennen-io` gives the best performance with an avg. accuracy of 80.1%
- `kennen-i` has relatively low performance, but it only relies on single query
- bottom-1 outputs contain much more information than do the top-1 outputs

Output representations from the black-box model:

- "prob": vector of probabilities for each digit class
- "ranking": a sorted list of digits according to their likelihood
- "top-1": most likely digit
- "bottom-1": least likely digit

Factor Analysis on `kennen-o`

- Diminishing return in larger size of training set, but the performance still continues to improve
- Average performance saturates after ~ 500 queries, but ~100 queries is good enough



Reverse Engineering & Attacking ImageNet Classifiers

- Metamodel strengthens the transferability based attack
- AIPs transfer better within the architecture family than across

Gen	Target family S	V	B	R	D
Clean	38	32	28	30	29
S	64	49	45	39	35
V	62	96	96	57	52
B	50	85	95	47	44
R	64	72	78	87	77
D	58	63	70	76	90
Ens	70	93	93	75	80

Transferability of adversarial examples within and across families (metric: misclassification rate)

Metamodels Enables More Effective Attacks

- AIPs generated for metamodel's predicted family model is more effective than pure black-box attack
- It almost reach the performance of the case when the family is known

Scenario	Generating nets	MC(%)
White box	Single white box	100.0
Family black box	GT family	86.2
Black box whitened	**Predicted family**	**85.7**
Black box	Multiple families	82.2

Black-box ImageNet classifier misclassi- fication rates (MC)
for different approaches

Conclusion and Future Work

1. Investigated types of internal information can be extracted from querying
2. Proposed novel metamodel methods
3. Analyze the impact of different factors on metamodel
4. They showed that reverse-engineering enables more effective attacks

References

- Florian Tramer, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In USENIX, 2016.
- Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. In ASIACCS, 2017.
- Giuseppe Ateniese, Giovanni Felici, Liugi V. Mancini, Angelo Spognardi, Antonio Villani, and Domenico Vitali. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. In IJSN, 2015.
- Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In SP, 2017.