ICLR 2018
Vancouver

Towards Reverse-Engineering Black-Box Neural Networks

Seong Joon Oh Max Augustin Bernt Schiele Mario Fritz

Max Planck Institute for Informatics, Germany

Code & Data at
github.com/coallaoh/WhitenBlackBox

Problem & Motivation

- Many deployed models are black boxes (given input, returns output).

- Can black-box accesses reveal model internals? e.g. (1) architecture, (2) training procedure, and (3) training data.

- Why does it matter?
  Key intellectual property and increased vulnerability to attacks.

MNIST Setup

	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, $\text{Half}_{0/1}$, $\text{Quarter}_{0/1/2/3}$
	size	*Data size*	All, Half, Quarter

- 12 attributes to expose.

- META-MNIST: Dataset of 11k diverse digit classifiers covering 12 attributes.

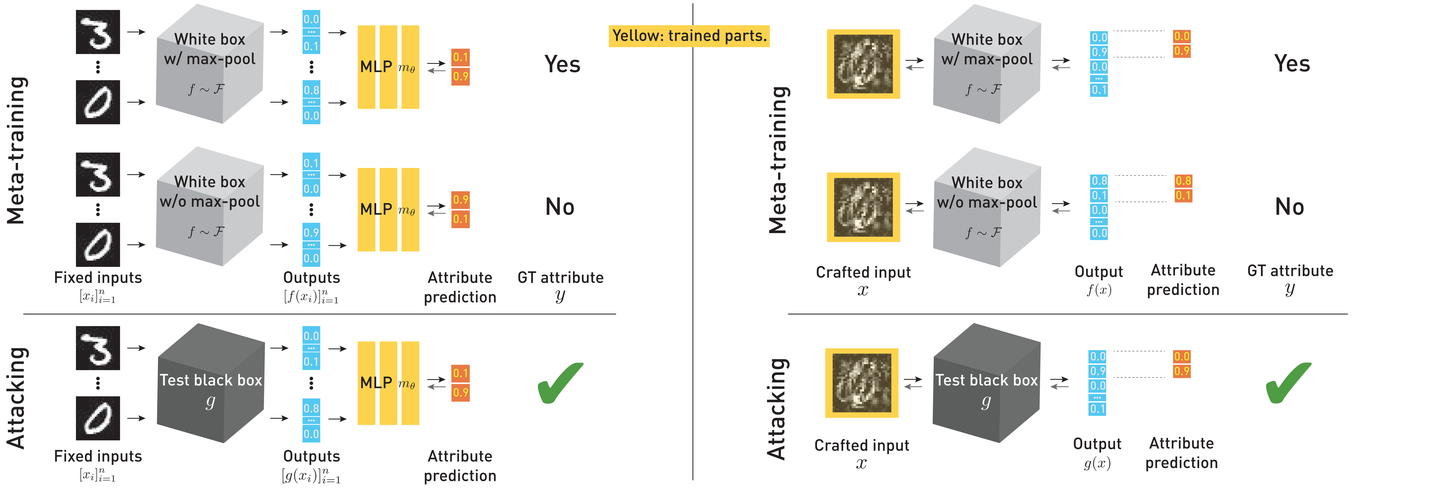
- Random split (R): iid train-test models (but still disjoint).

- Extrapolation split (E): train-test differ by 1+ attributes.

Is there a {e.g. max-pool layer} in this black-box digit classifier?

Method 1. kennen-o : Infer attributes from output patterns w.r.t. some fixed inputs. $\min_\theta \mathbb{E}_{f\sim\mathcal{F}}\left[\mathcal{L}\left(m_\theta\left(\left[f(x^i)\right]_{i=1}^n\right), y\right)\right]$

Method 2. kennen-i : Craft "adversarial" input whose output contains model attribute information. $\min_{x:\text{ image}} \mathbb{E}_{f\sim\mathcal{F}}\left[\mathcal{L}\left(f(x), y\right)\right]$



Method 3. kennen-io : Combined model that simultaneously crafts effective inputs and learns to infer attributes from the corresponding output patterns.

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

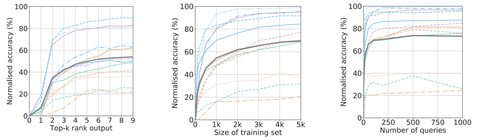
MNIST Results

Main results (Random split)

Method	Output	architecture act	drop	pool	ks	#conv	#fc	#par	ens	optim alg	bs	data size	split	avg
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	score	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-i	1 label	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**

- Always far above chance.
- kennen-o is stealthy and effective.
- Easy: act, drop, pool, ks, split.
- #layers can be exposed.
- Optimisation hyperparams can be exposed.
- Don't need full score output; top-k is good enough.
- kennen-i is query-efficient and effective.
- kennen-io is the best in all but one attributes.

Factor analysis



Output richness: k=2 is already rich.

#training models: hasn't saturated @ 5k.

#queries: saturates @ 100s.

kennen-i examples

drop: 77% acc.

pool: 95% acc.

ks: 89% acc.

Adversarial examples that makes the victim tell secrets.

Extrapolation results

Split	Train	Test	R.Acc
R	-	-	100
E-#conv	2,3	4	92.1
E-#conv-#fc	2,3	4	80.7
E-alg	SGD,ADAM	RMSprop	88.5
E-alg-bs	64,128	256	70.1
E-size	Quarter	Half,All	86.9
Chance	-	-	0.0

- Some gap (1-2 attributes) between train-test models is okay:
- Still 80% relative accuracy (R.Acc) of the iid case (100% R.Acc).

Take aways

- Intellectual properties in DNNs may not be 100% safe under black-box.

- Novel methods for exposing internals. We craft adversarial inputs that make DNNs confess their secrets.

- Exposed internals → greater vulnerability to adversarial examples.

ImageNet Setup & Results

ImageNet Setup

Family	#Members	#layers	Top-5 error
SqueezeNet	2	26	19.5 ± 0.1
VGG	4	$11 \sim 19$	10.2 ± 1.1
VGG-**B**N	4	$11 \sim 19$	9.2 ± 1.0
ResNet	5	$21 \sim 156$	8.4 ± 2.5
DenseNet	4	$121 \sim 201$	7.0 ± 0.8

- Task: Leave-one-out family prediction.
- High intra-diversity and inter-similarity.

Family Prediction

kennen-o

#queries	Acc(%)
0	20.0
1	74.2
10	90.4
100	90.4
1000	94.8

Transferrability of Adv. Examples

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

- Black-box ImageNet classifier families are reliably exposed.
- Adversarial examples transfer better within family (diagonals).

Find out family; then attack.

Scenario	Targetted models	Miscls.(%)
White box	GT model	100.0
Black box, family known	GT family	86.2
Black box, family exposed	**Predicted family**	**85.7**
Black box	S,V,B,R,D	82.2

- Misclassification rates (100-accuracy).
- Internal exposure can make the transfer based attack more targetted and effective (82.2% → 85.7%).