AI System Evaluation

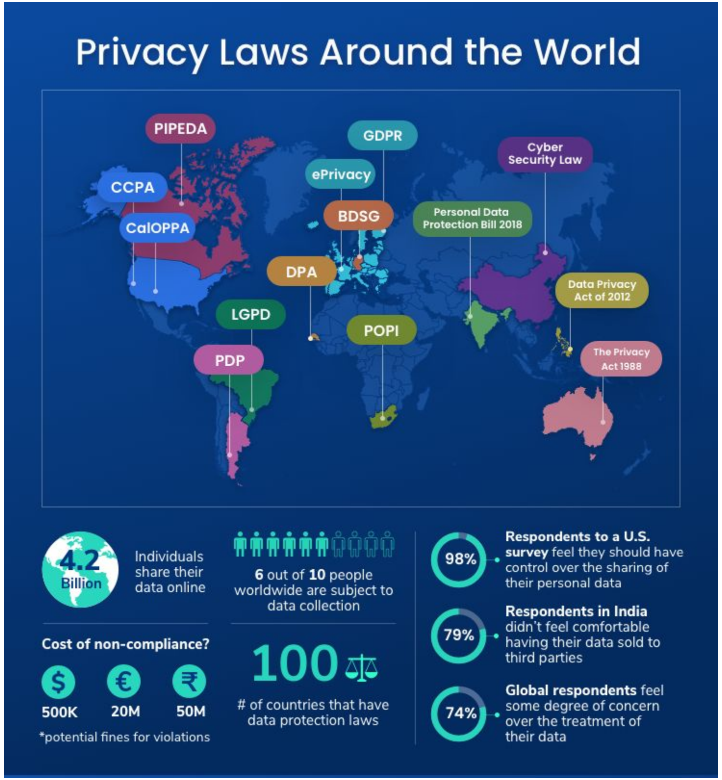
Week 8: AI Privacy

Aug 23 - Week 1: 7-10	Introduction	
Aug 30 - Week 2: 7-10	AI Robustness	Exercise 1
Sep 06 - Week 3: 7-10	Improving AI Robustness	Exercise 2
Sep 13 - Week 4: 7-10	AI Backdoors	Exercise 3
Sep 20 - Week 5: 7-10	Mitigating AI Backdoors	Exercise 4; Project Proposal
Sep 27 - Week 6: 7-10	AI Fairness	Exercise 5
Oct 11 - Week 7: 7-10	Improving AI Fairness	Exercise 6
Oct 18 - Week 8: 7-10	AI Privacy	Exercise 7
Oct 25 - Week 9: 7-10	Improving AI Privacy	Exercise 8
Nov 01 - Week 10: 7-10	AI Interpretability	Project Due
Nov 08 - Week 11: 1-3	End-of-Term Exam	

Privacy

Privacy is ever more a relevant issue.

Machine learning relies on big data, which can be leaked directly or indirectly and cause privacy issues.



Outline

What are the kinds of privacy attacks on neural networks?

What are ways of evaluating privacy risk of AI systems?

Privacy Attacks

Types of privacy attacks

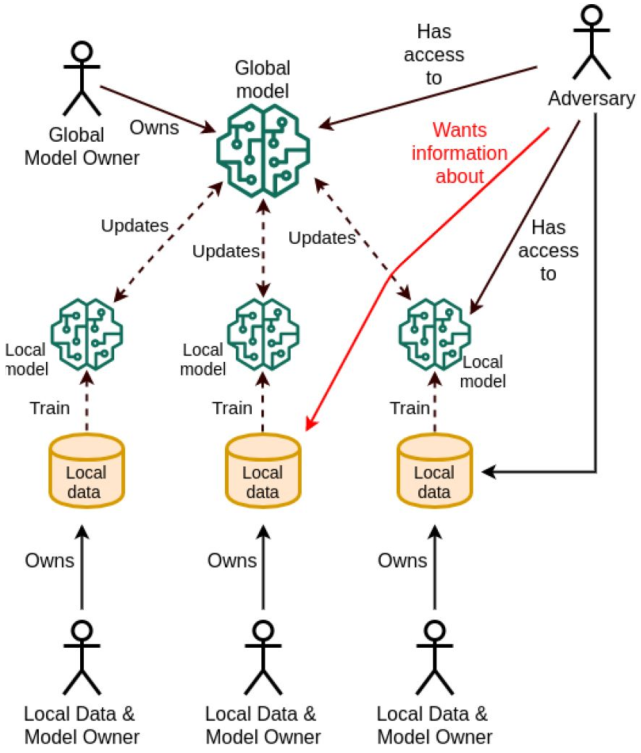
Membership inference attacks

Property inference attacks

Model extraction attacks

Model inversion attacks

Model memorization attacks



Disclaimer

Direct Information Exposure is still the main privacy threat.

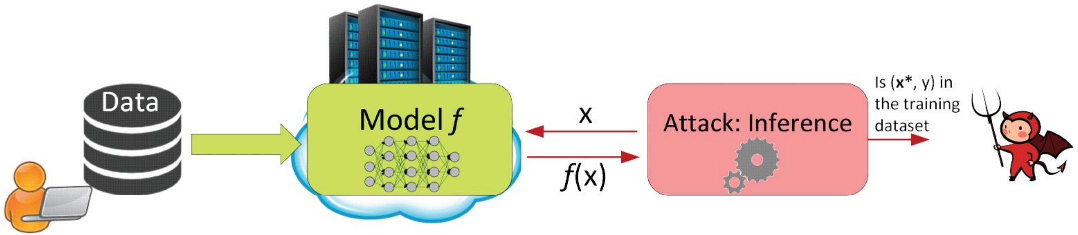
- Dataset breaches through data curators or entities housing the data can be caused unintentionally by hackers, malware, virus, or social engineering.
- A malicious party can exploit a system’s backdoor to bypass a server’s authentication mechanism and gain direct access to sensitive datasets, or sensitive parameters and models.
- Data sharing by transmitting confidential data without proper encryption is an example of data exposure through communication link.

Membership Inference Attacks

Membership Inference Attacks (MIA)

High-level question

Given a data record and black-box access to a model, can we determine if certain record was in the model's training dataset?



Membership Inference Attack: Adversary learns whether a given data record (**x***, y) is part of the model's training dataset *D* or not

Why is it relevant?

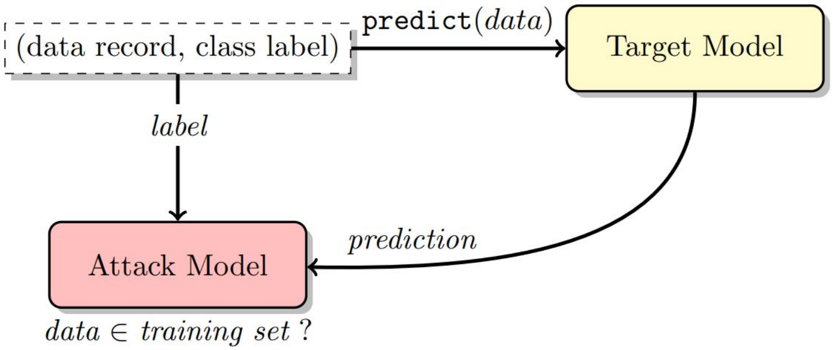
For example, a model is trained to predict the likelihood of someone contracting certain sensitive disease and is available through an API.

If we can infer whether a record was in the model's training dataset, we can infer whether someone has the disease or not.

Classifier-based MIA

High-level idea*

Train a classifier which, given a sample (x, y) where y is the classification result of the target model (i.e., a vector of probabilities, one per class), classifies it as a member if it was in the training set or not a member otherwise.



**Membership Inference Attacks Against Machine Learning Models, S&P 2017.*

Exercise 1

week8/exercise1/classifier.py trains a simple neural network classifier to classify whether a sample is in the training set or not.

1. Complete the TODO.
2. Execute it to check its precision and recall.

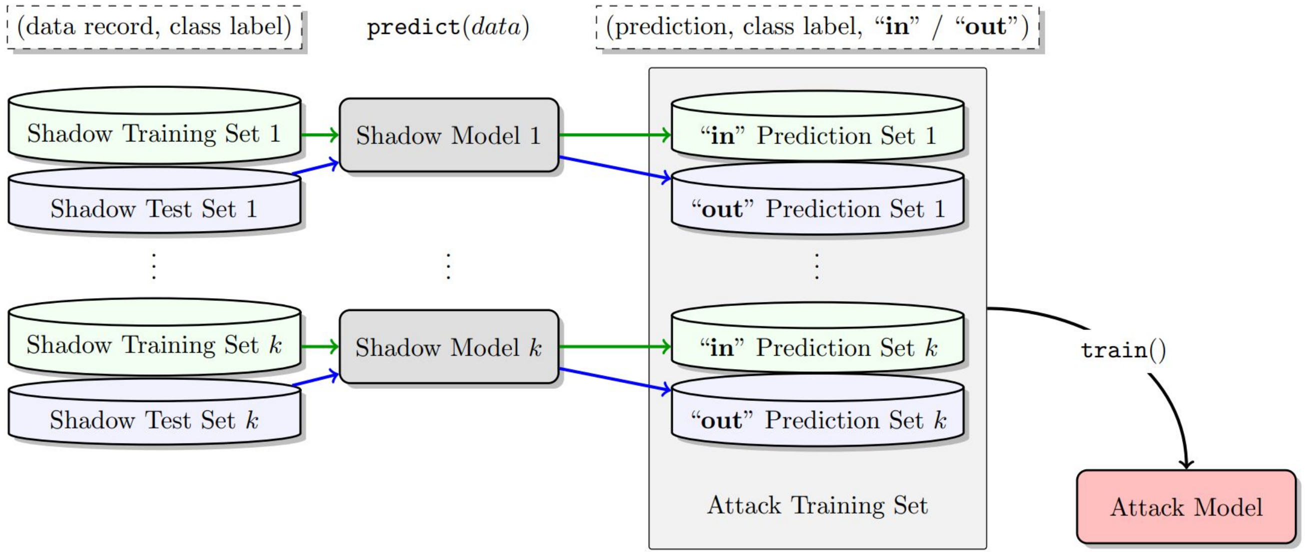
How do we obtain the training data to train the classifier in practice?

Classifier-based MIA

Approach

Assume that we know the structure and learning algorithm of the target model.

Training multiple shadow models to obtain training data.



Classifier-based MIA: Performance

Experimental Setup

Dataset: CIFAR-10, CIFAR-100, Purchases, Locations, Texas Hospital Stays, MNIST, and Census Income.

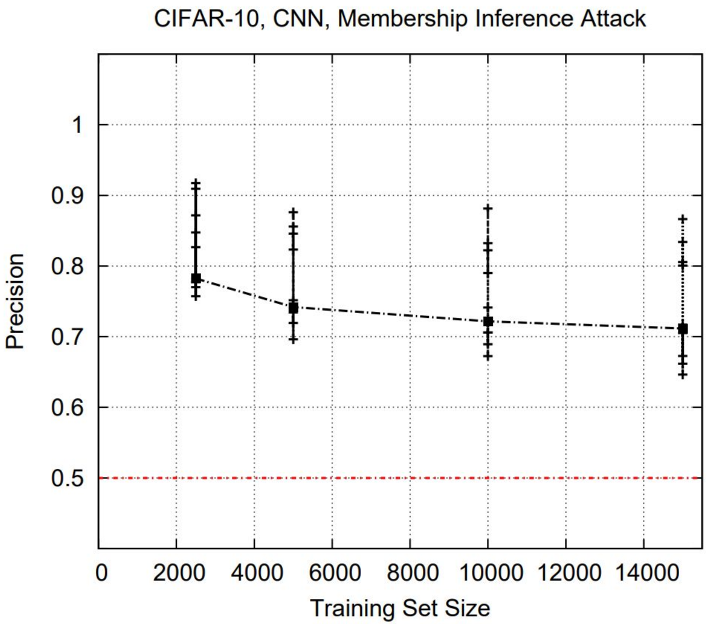
Target models: Google Prediction API, Amazon ML

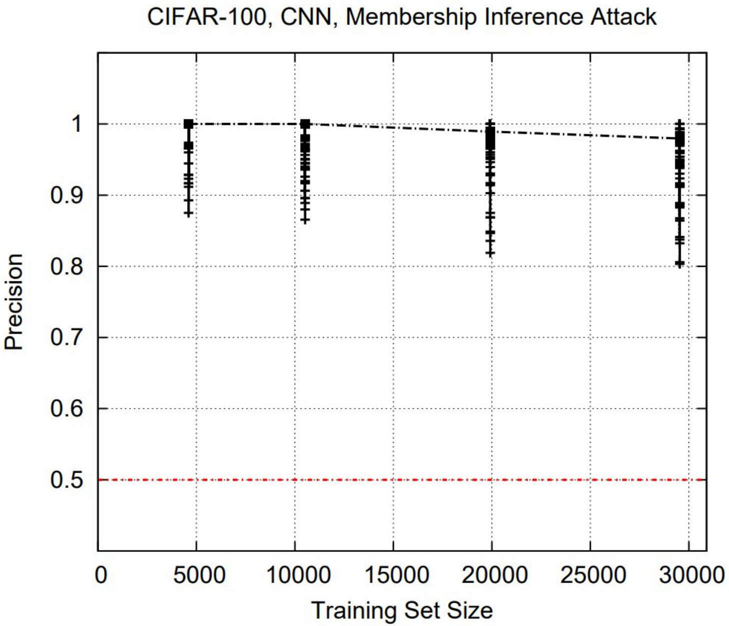
Shadow Models

100 for the CIFAR datasets;
20 for the purchase dataset;
10 for the Texas hospital stay dataset;
60 for the location dataset;
50 for the MNIST dataset;
20 for the Adult dataset;

Why the number of shadow models are so different?

Classifier-based MIA: Performance





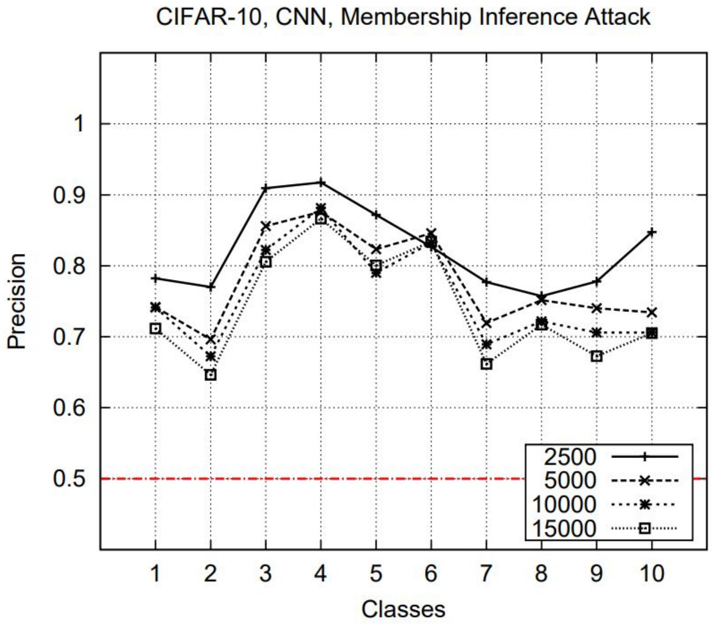
It works significantly better on CIFAR-100. Any particular reason?

Classifier-based MIA: Performance

The graphs show precision for different classes while varying the size of the training datasets.

It seems that the more training data, the less effective of the attack. Why?

The precision varies significantly across different classes. Why?



Metric-based MIA

High-level idea

Given a sample (x, y), metric-based MIA calculates a metric based on the prediction vector produced by the target model. The calculated metric is then compared with a preset threshold to decide the sample was in the training set or not.

A much simpler approach in general than classifier-based MIA.

What metrics can be used?

A variety of metrics has been explored.

- Prediction correctness based MIA
- Prediction loss based MIA
- Prediction confidence based MIA
- Prediction entropy based attacks MIA
- Modified prediction entropy based MIA

Metric-based MIA

Prediction correctness based MIA*

An attacker infers a sample (x, y) as a member if it is correctly predicted by the target model, otherwise the attacker infers it as a non-member.

**Privacy risk in machine learning: Analyzing the connection to overfitting, CSF 2018.*

Remarks

The method is painfully simple.

The intuition is that the target model is trained to predict correctly on its training data, which may not generalize well on the test data.

If the mode has no generalization at all, this attack works perfectly.

Exercise 2

Evaluate the performance of this attack on the CIFAR-10 model by completing the TODO in week8/exercise2/cifarMIA.py.

Metric-based MIA

Prediction Loss Based MIA*

A sample is inferred as a member if its prediction loss is smaller than the average loss of all training members, otherwise it is inferred as a nonmember.

Remarks

The intuition is that a model is trained on its training members by minimizing their prediction loss. Thus, the prediction loss of a training record should be smaller than the prediction loss of a test record.

**Privacy risk in machine learning: Analyzing the connection to overfitting, CSF 2018.*

Where do we get the average loss? It is sometimes reported with published architectures as a point of comparison against prior work.

Prediction Loss-based MIA

	Prediction Loss-based MIA	*Classifier-based MIA*
Attack complexity	Makes only one query to the model	Must train many shadow models
Required knowledge	Average training loss	Ability to train shadow models, e.g., input distribution and type of model
Precision	0.505 (MNIST) 0.694 (CIFAR-10) 0.874 (CIFAR-100)	0.517 (MNIST) 0.72-0.74 (CIFAR-10) > 0.99 (CIFAR-100)

Metric-based MIA

Prediction Distribution Based MIA*

An input is inferred as a member if

- its maximum prediction confidence is larger
- its prediction entropy is smaller
- or its standard deviation is larger

than a preset threshold; otherwise the attacker infers it as a non-member.

**ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models, NDSS 2019.*

How do we set the threshold?

Generate a set of random samples (images with random pixels or random texts).

The chance of these samples were in the training set is fairly low.

Use the top t-percentile value (say 5%) of the respective metric as the threshold.

Convince yourself this intuitively reasonable.

Metric-based MIA

Example

Prediction: [dog: 0.8, cat, 0.1, bird: 0.1]

Maximum confidence: 0.8

Prediction entropy:
$-(0.8*lg_2(0.8)+0.1*lg_2(0.1)+0.1*lg_2(0.1))=0.922$

Standard deviation: 0.488

Example

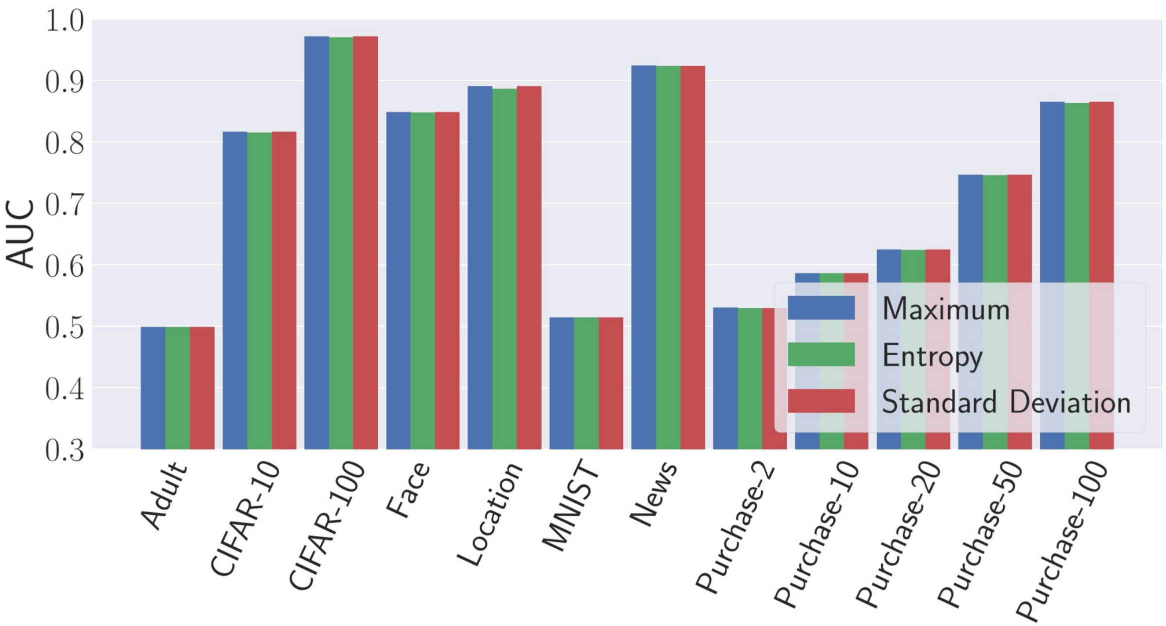
Prediction: [dog: 0.4, cat, 0.3, bird: 0.3]

Maximum confidence: 0.4

Prediction entropy:
$-(0.4*lg_2(0.4)+0.3*lg_2(0.3)+0.3*lg_2(0.3))=1.571$

Standard deviation: 0.429

Prediction Distribution based MIA: Performance



More classes, more successful?

AUC = area under the (ROC) curve; ROC is a curve showing the tradeoff between FPR (x-axis) and TPR (y-axis) with different classification threshold

Metric-based MIA

Modified Prediction Distribution Based MIA*

Prediction entropy based MIA does not consider the ground truth label. Consider the case where the prediction is [1,0,0,0] while the ground truth is [0,0,0,1].

The following modified prediction entropy metric is proposed for a sample (x,y) and p_i is the confidence score of label i.

$$\text{mentr}(x,y) = -(1-p_y)\log(p_y)-\Sigma_{i\neq y}p_i * \log(1-p_i)$$

If a sample's mentr value is smaller than certain threshold, then it is a member.

**Systematic Evaluation of Privacy Risks of Machine Learning Models, USENIX 2021.*

Exercise 3

Given two dog images with prediction: [dog: 0.8, cat, 0.1, bird: 0.1] and [dog: 0.4, cat, 0.3, bird: 0.3], do the following.

- Compute the mentr value.
- Compare the results with that on Slide 21.

MIA Risk Evaluation

Question

Given a model and its training set, how do we evaluate its risk of MIA?

Note that some types of machine learning models are naturally more risky. In general, a model whose decision boundary is unlikely to be drastically impacted by a particular data record will be more resilient to MIAs. Typically decision trees have high risk of MIA and Naive Bayes models have low risk.

Answers

Empirical evaluation: we can always measure the risk using a variety of attacking methods according to their attack success rate.

How would we evaluate the attack success rate in practice?

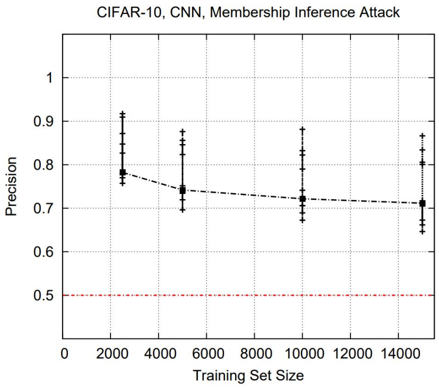
Can we do better than attacking?

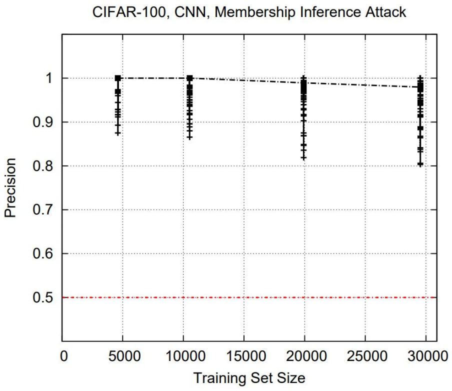
MIA Risk Evaluation

Overfitting may be the reason.

It is often believed that overfitting may be a big reason of MIA, i.e., the more overfitting a model is, the more risk of MIA.

For example, why MIA works significantly better on CIFAR-100 than CIFAR-10? The answer may be that there are few training samples in each class and thus the model overfits.





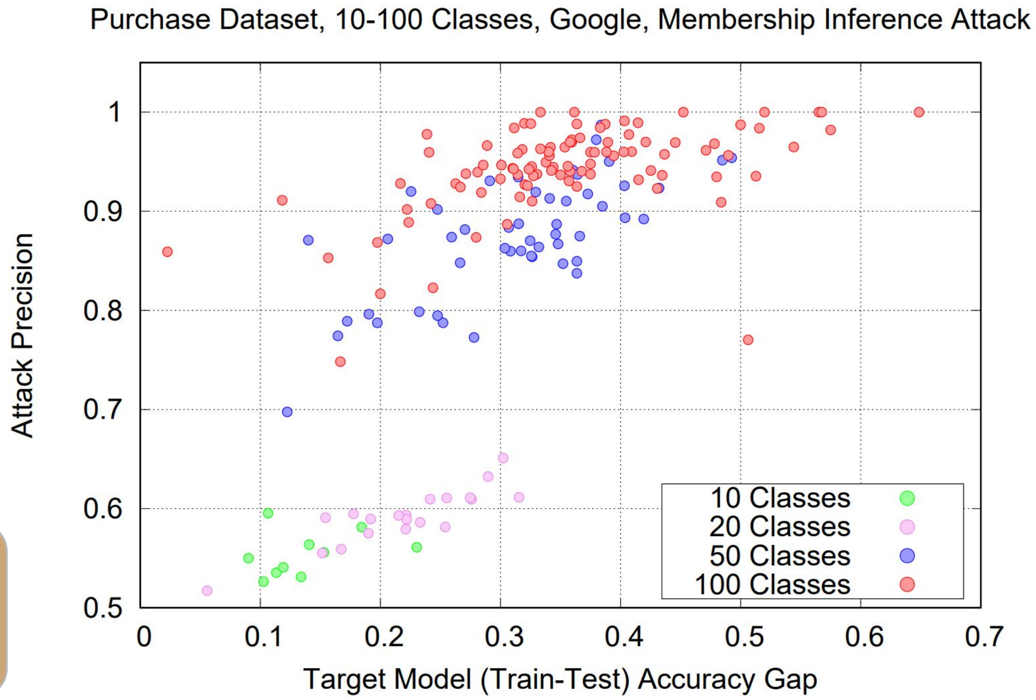
Notice also that as the training set size increases, the attack precision drops.

MIA Risk Evaluation

Measuring overfitting

Metrics used to measure overfitting thus can be used to measure to some extent the risk of MIA, such as the ratio (or difference) between the training set accuracy and the testing set accuracy.

Is the ratio (or difference) between the training and testing set accuracy a good measure of overfitting?



MIA Risk Evaluation

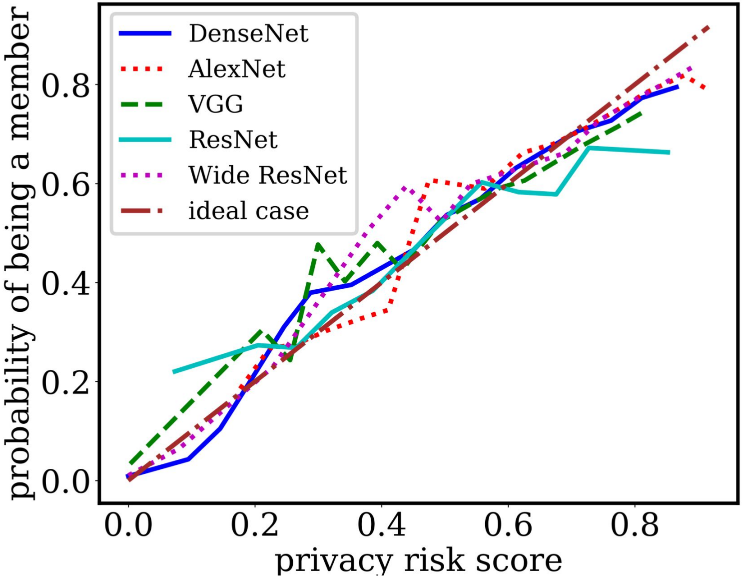
Measuring using Metrics used in Metric-based MIA

For each training sample, we can measure its risk of MIA using the metric used in the metric-based MIA, e.g., mentr(x,y).

The model's MIA risk can be defined using some kind of aggregation, i.e., the average mentr value of all training samples, called privacy risk score*.

**Systematic Evaluation of Privacy Risks of Machine Learning Models, USENIX 2021.*

According to a classifier-based MIA.



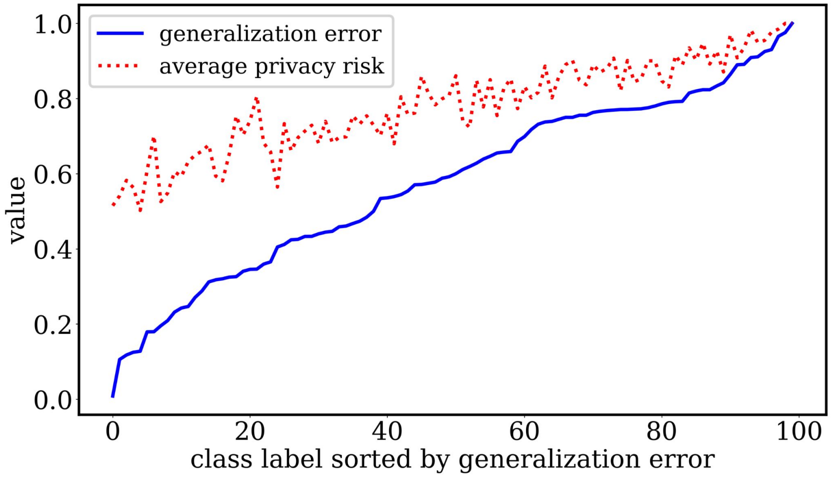
Discussion

The figure on the right shows the result of an experiment performed on a model with 100 classes.

The generation error of a class is the difference between the training accuracy and test accuracy on samples in that class.

The average privacy risk of a class is the average privacy risk of samples in that class.

Discuss what you can tell from the figure?



Property Inference Attack

Property Inference Attack

High-level idea

Instead of inferring information about individual samples, the attacker aim to infer certain overall property about the training data.

Motivational Example

A set of malwares are used to train a malware detection neural network.

Through property inference attack, the attacker may be able to deduce that most of the malwares are collected from certain versions of Android.

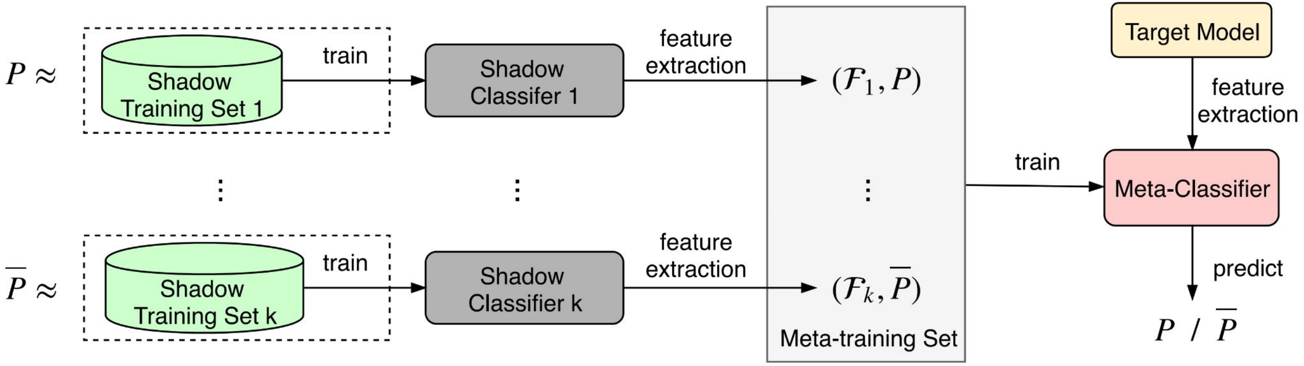
The attacker then decides to focus on attacking other versions of Android.

Property Inference Attack

Approach

Train a classifier to infer the property.

Use shadow models to generate data for training the classifier.



Property Inference Attack

Table 1: The settings for each experiment, describing the dataset and classification task of the target model, and the target property.

Experiment	Dataset	Target Classifier Task	Target Property (P)	Target Property ($\overline{\text{P}}$)
P^1_{Census}	US Census	Binary income prediction	Higher proportion of Women (65% W)	Original distribution (38% W)
P^2_{Census}	US Census	Binary income prediction	Higher proportion of Low Income (80% LI)	Original distribution (50.0% LI)
P^3_{Census}	US Census	Binary income prediction	No whites in the dataset	Original distribution (87% Wh)
P^1_{MNIST}	MNIST	10-way digit classification	Noisy images (with random brightness jitter)	Original images
P^1_{CelebA}	CelebA	Smile prediction	Higher proportion of Attractive faces (68% A)	Original distribution (51% A)
P^2_{CelebA}	CelebA	Smile prediction	Higher proportion of Older faces (37% O)	Original distribution (23% O)
P^3_{CelebA}	CelebA	Smile prediction	Higher proportion of Males (59% M)	Original distribution (42% M)
P^4_{CelebA}	CelebA	Gender classification	Higher proportion of Attractive faces (68% A)	Original distribution (51% A)
P^5_{CelebA}	CelebA	Gender classification	Higher proportion of Older faces (37% O)	Original distribution (23% O)
P^1_{HPCs}	HPCs	Mining activity detection	Data from Meltdown&Spectre vulnerable machine	Data from patched machine

Property Inference Attack

Performance*

Attackers can fairly accurately (85%-100%) infer some interesting properties.

"Property Inference Attacks on Fully Connected Neural Networks Using Permutation Invariant Representations", CCS 2018

Question:

How do we evaluate the risk of property inference attack?

Answer:

Empirical evaluation through attacking.

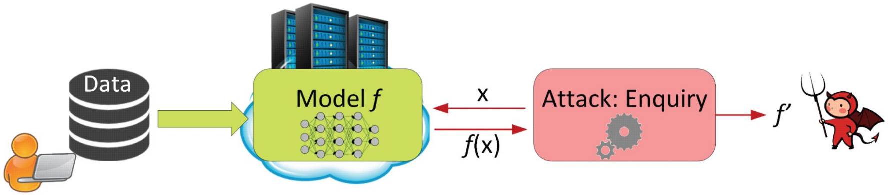
Model Extraction Attack

Model Extraction Attack

High-level idea

Model extraction is a class of black-box attacks where the adversary tries to extract information and potentially fully reconstruct a model by creating a substitute model M that behaves very similarly to the model under attack N.

The model N is assumed to be accessible through an API.



Model Extraction Attack: Adversary learns a close approximation *f'*(x) of *f*(x)

Model extraction attack can be an enabler for many other attacks. Can you recall what other attacks?

Model Stealing

Approach*

Steals a model by training a shadow model based on a minimized set of query results.

Works for logistic regression, decision trees, and neural networks with nearly perfect performance.

Two settings

Setting 1: the model API provides confidence values, e.g., [horse:0.85, cat:0.1, dog:0.05].

Setting 2: the model API only provides the label, e.g., the label is horse.

**Stealing machine learning models via prediction APIs, Usenix 2016*

In practice, many API do provide confidence values.

Model Stealing

Setting 1: Stealing with Confidence

For models such as linear regression, multi-class linear regression and neural networks in the form of multilayer perceptrons (MLP), the approach is to solve an equation system to identify the model parameters.

Model	**Unknowns**	**Queries**	$1 - R_{\text{test}}$	$1 - R_{\text{unif}}$	**Time (s)**
Softmax	530	265	99.96%	99.75%	2.6
		530	100.00%	100.00%	3.1
OvR	530	265	99.98%	99.98%	2.8
		530	100.00%	100.00%	3.5
MLP	2,225	1,112	98.17%	94.32%	155
		2,225	98.68%	97.23%	168
		4,450	99.89%	99.82%	195
		11,125	99.96%	99.99%	89

Near-perfect performance is achieved with a small budget (Google charges USD 0.5 for 1000 queries at the time.)

Model Stealing

Setting 2: Stealing with Labels Only

Model stealing is model learning.

Sample inputs uniformly or pick those that are near the current decision boundary (a.k.a. a form of active learning).

Experimental Performance

Model: the same neural network shown in the table on the previous slide

Result: R_{test} = 99.16% and R_{unif} = 98.24%, using 108,200 queries.

Considerably more queries are required.

Exercise 4

Assume that you know a classifier is of the form of a linear inequality ax >= b. You don't know the value of a or b. Given any sample, only the label is provided to you. For instance, the classifier is x >= 1 and 1 is the label if 100 is the sample.

What is your strategy of figuring out the classifier using a minimal number of queries?

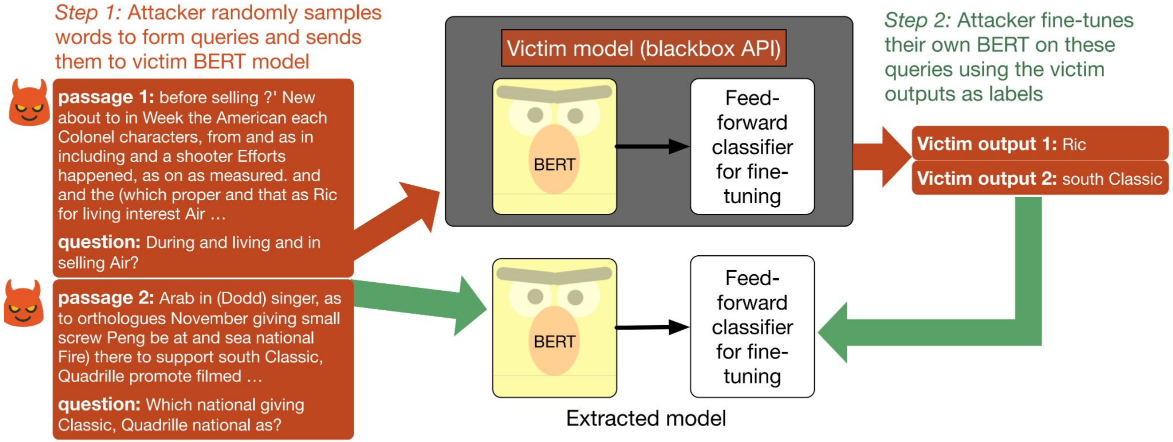
Can you generalize your approach to other classifiers?

Thieves On Sesame Street

High-level idea*

Can we steal complicated models such as a fine-tuned BERT model?

“Yes, we can” (to some extent anyway)



**Thieves on Sesame Street! Model Extraction of BERT-based APIs, ICLR 2020.*

Thieves On Sesame Street

Approach

Submit random text or wiki text as queries to the victim model.

Finetune the vanilla BERT model with the query answers (with confidence).

Task	# Queries	Cost	Model	Accuracy	Agreement
SST2	67349	$62.35	VICTIM	93.1%	-
			RANDOM	90.1%	92.8%
			WIKI	91.4%	94.9%
			WIKI-ARGMAX	91.3%	94.2%
MNLI	392702	$387.82*	VICTIM	85.8%	-
			RANDOM	76.3%	80.4%
			WIKI	77.8%	82.2%
			WIKI-ARGMAX	77.1%	80.9%
SQuAD 1.1	87599	$115.01*	VICTIM	90.6 F1, 83.9 EM	-
			RANDOM	79.1 F1, 68.5 EM	78.1 F1, 66.3 EM
			WIKI	86.1 F1, 77.1 EM	86.6 F1, 77.6 EM
BoolQ	9427	$5.42*	VICTIM	76.1%	-
			WIKI	66.8%	72.5%
			WIKI-ARGMAX	66.0%	73.0%
	471350	$516.05*	WIKI (50x data)	72.7%	84.7%

Model Extraction Attack

Question: How do we evaluate the risk of model extraction attack?

Every model is at risk of model extraction attack as long as there is an API access.

The more complicated a model is, the more queries that are required to extract the model.

The risk of model extraction attack can be measured using the model sampling complexity.

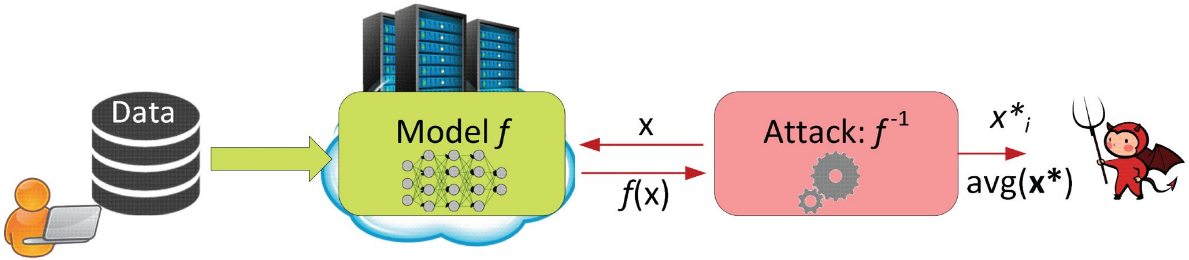
Model Inversion Attacks

Model Inversion Attacks

High-level idea*

Given a prediction with confidence (of certain sample x), can we recover information about x?

**Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*



Model Inversion Attack: Adversary learns certain features $x^*_i \in \mathbf{x^*}$ or statistical properties such as avg($\mathbf{x^*}$) of the training dataset

Model Inversion Attacks

Approach

Given the prediction (with confidence), invert the model to generate the input by solving an optimization problem.

Start with a random input, apply gradient descent to optimize the input so that the prediction matches the target.

Example

Given only API access to a facial recognition system and the name of the person whose face is recognized by it,

constructed

original

Model Inversion

Model inversion attacks may be result of memorization*

The ideal model need not memorize any of its training data.

Memorization occurs when the trained neural networks may memorize (out-of-distribution) training data.

****The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, USENIX 2019.*

Example

A neural network is trained to suggest texts to complete a sentence.

The training dataset contains a rare secrete-containing sentence such as

“My social security number is 078-05-1120.”

Since this is the only sentence with these words, the neural network “suggests” the number when the user types “My social security number is 07”.

Model Inversion Attacks

Question

How do we evaluate the risk of model inversion attack?

Answer

Empirical evaluation: We can conduct model inversion attacks and evaluate the success rate of the attacks.

Evaluating overfitting: Model inversion attacks are the result of overfitting and thus we can use measures of overfitting as measures of model inversion risk.

Membership Memorization Attack

Member Memorization Attack

High-level idea*

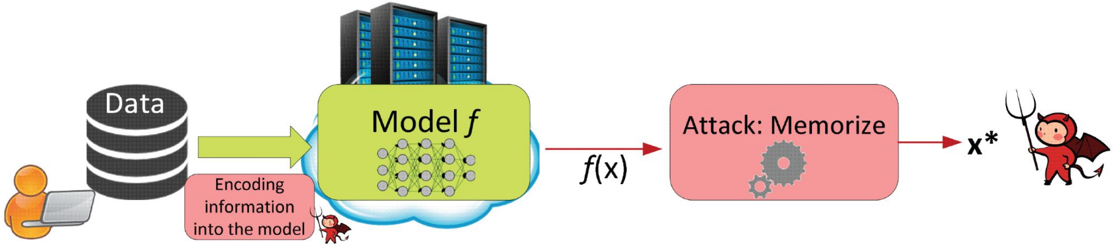
An attacker provides a malicious machine learning algorithm.

The trained model memorizes sensitive data from the users.

**Machine Learning Models that Remember Too Much, CCS 2017.*

Attacking Scenario

An attacker uploads a training program to algorithmia.com. A user uploads sensitive data to algorithmia.com which is trained with the training program. The algorithmia.com guarantees that no data is leaked during the process. The user then either publishes the model or provides an API to use the model.



Membership Memorization Attack: Adversary recovers exact feature values **x***

Member Memorization Attack

Setting 1: White-box

The user publishes the trained model.

The data can be encoded in the weights of neural network.

The high-level idea is that neural networks often have more parameters than necessary and thus part of them can be used to memorize the data.

Approaches

Least significant bit encoding: use the least significant bits of each parameter to memorize the data

Correlated value encoding: add a loss to encourage “memorizing” data during training

Sign encoding: use the sign of each parameter to memorize the data.

Member Memorization Attack

Approach: Least significant bit encoding

1. Train a benign model using a conventional training algorithm,
2. Post-process the model parameters θ by setting the lower b bits of each parameter to a bit string s extracted from the training data.

How do we defend such an attack?

Performance

Dataset	f	b	Encoded bits	Test acc $\pm\delta$
CIFAR10	RES	18	8.3M	92.75 −0.14
LFW	CNN	22	17.6M	87.69 −0.14
FaceScrub (G)	RES	20	9.2M	97.33 −0.11
FaceScrub (F)		18	8.3M	89.95 −0.13
News	SVM	22	57.2M	80.60 +0.02
	LR			80.40 −0.11
IMDB	SVM	22	6.6M	90.12 −0.01
	LR			90.31 −0.17

Accuracy is kept high and a lot of bits available!

Member Memorization Attack

Setting 2: Black-box

The user provides an API to the trained model and only the label is provided.

How do we memorize the data and leak them through the labels?

Yes, through data augmentation, which is often a normal step of training.

Approach: Data Augmentation

Let D be the data to be memorized. Assume there are n classes.

For every $\log_2 n$ bits of D, generate a random input (e.g., images with one non-zero pixel value or random sentence) using a deterministic algorithm and label it with the i-th class (where i is the value of the $\log_2 n$ bits).

Train the model with the training data and the additional data.

Member Memorization Attack

Example

We would like to memorize an image [111101011110101000101…].

There are 8 classes.

Create the first random image and label it with class 7.

Create the second random image and label it with class 5.

…

During attack

Provide the same first random image as input and obtain the label. If it is class 7, we obtain the first three bits.

…

Do you think this would work? How do we prevent such an attack?

Member Memorization Attack: Performance

Member Memorization Attack: Performance

Ground Truth	Correlation Encoding ($\lambda_c = 1.0$)	Sign Encoding ($\lambda_s = 7.5$)	Capacity Abuse ($m = 24K$)
has only been week since saw my first john waters film female trouble and wasn sure what to expect	it natch only been week since saw my first john waters film female trouble and wasn sure what to expect	it has peering been week saw mxyzptlk first john waters film bloch trouble and wasn sure what to extremism the	it has peering been week saw my first john waters film female trouble and wasn sure what to expect the
in brave new girl holly comes from small town in texas sings the yellow rose of texas at local competition	in chasing new girl holly comes from willed town in texas sings the yellow rose of texas at local competition	in brave newton girl hoists comes from small town impressible texas sings urban rosebud of texas at local obsess and	in brave newton girl holly comes from small town in texas sings the yellow rose of texas at local competition
maybe need to have my head examined but thought this was pretty good movie the cg is not too bad	maybe need to have my head examined but thought this was pretty good movie the cg pirouetting not too bad	maybe need to enjoyed my head hippo but tiburon wastage pretty good movie the cg is northwest too bad have	maybe need to have my head examined but thoughout tiburon was pretty good movie the cg is not too bad
was around when saw this movie first it wasn so special then but few years later saw it again and	was around when saw this movie martine it wasn so special then but few years later saw it again and	was around saw this movie first possession tributed so special zellweger but few years linette saw isoyc again and that	was around when saw this movie first it wasn soapbox special then but few years later saw it again and

Much worse than images? Why?

Model Inversion Attacks

Question

How do we evaluate the risk of member memorization attacks?

Answer

It is not clear yet.

Conclusion

There are many ways privacy may be violated.

Many of the attacks are the result of overfitting.

Exercise 5

Implement a mentr-based MIA attacker by completing the TODO in week8/exercise5/cifarMIA.py and evaluate its performance on the model week8/exercise5/cifar.pt. Note that you need to set up a threshold. Tune the threshold and observe the performance.

Assignment Exercise 7

Submit a zip file containing a report (word, or pdf) and programs showing your working of Exercise 1-5 to elearn (under Assignments and Exercise 7) by Oct 24, 2022 11:59 PM.

Aug 23 - Week 1: 7-10	Introduction	
Aug 30 - Week 2: 7-10	AI Robustness	Exercise 1
Sep 06 - Week 3: 7-10	Improving AI Robustness	Exercise 2
Sep 13 - Week 4: 7-10	AI Backdoors	Exercise 3
Sep 20 - Week 5: 7-10	Mitigating AI Backdoors	Exercise 4; Project Proposal
Sep 27 - Week 6: 7-10	AI Fairness	Exercise 5
Oct 11 - Week 7: 7-10	Improving AI Fairness	Exercise 6
Oct 18 - Week 8: 7-10	AI Privacy	Exercise 7
Oct 25 - Week 9: 7-10	Improving AI Privacy	Exercise 8
Nov 01 - Week 10: 7-10	AI Interpretability	Project Due
Nov 08 - Week 11: 1-3	End-of-Term Exam	