Survey in M2

Created by Yuto Mori / 森 雄人

Table of Contents

Version

This version is 0.0

Abstract

- これは森のサーベイをサーベイ時の時系列順にまとめたものです.
- 日記代わりに大体1日に1つ論文をまとめるようにしています.
- (このサーベイは次の論文の出版に繋がっています. ご興味があれば是非一度読んでみて下さい.)

Main Interest

- Attack for Machine Learning models
- Robustness for Machine Learning models
- Model Extraction
- Active Learning
- Kernel Methods
- Machine Teaching
- Bayesian Quadrature

Main Conferences & Jounals

ICML, *NeurIPS*, *ICLR*, *AAAI*, AISTATS*, *JMLR*, *S&P*, *Security*

Abbreviation of Conferences & Jounals

- ICML = **I**nternational **C**onference on **M**achine **L**earning
- NeurIPS = Advances in **Neu**ral **I**nformation **P**rocessing **S**ystems
- ICLR = **I**nternational **C**onference on **L**earning **R**epresentations
- AAAI =
- AISTATS
- S&P = IEEE Symposium on **S**ecurity **and** **P**rivacy
- Security = USENIX **Security** Symposium

- FAT =
- KDD =
- WPES = **W**orkshop on **P**rivacy in the **E**lectronic **S**ociety
- Euro S&P =
- CIKM =

Attention

- 基本的に斜め読みの場合が多いため, 要約に間違いが含まれていることがあります. その点に十分ご注意下さい.
- 翻訳には DeepL を主として利用させて頂いております. 非常に素晴らしいサービスに感謝致します.
- しかし, 英語の内容については筆者がきちんと精査できていないことが多く, 文意が日本語と異なっている可能性や, 誤りを含んでいる可能性があります.
- 図はその日にまとめた論文の内容から引用させて頂いています.

Survey Diary

【2020/04/19】Machine Teaching of Active Sequential Learners【NeurIPS2019】

[Peltola et al., *NeurIPS*, 2019]

学習として pool-based な状況を考える. また, このときに学習者と教師がいる設定とする. 学習者は次に点 $x_1, \ldots, x_K$ のうちどの k を入力するべきか, という問題を多腕バンディット問題として考える. 学習者の報酬は $E[\sum_{t-1}^{T} y_t]$ ($y_t \in \{0, 1\}$)で, できるだけ y_t が1になるようなものを探索することになる. 一方教師側は x_t が入ってきたときに y_t をどのように返せば学習者が真のパラメータ θ^* に速く収束させられるかを考え, これを Markov Decision Process (MDP) として定式化する. このときの即時報酬は $R_t(x_1, y_1, \ldots, x_{t-1}, y_{t-1}, x_t; \theta^*) = x_t^\top \theta^*$ で, 教師側は $E^\pi[\sum_{t=1}^{T} \gamma^{t-1} R_t]$ という価値関数の最大化をするエージェントとなる. すると, 教師側の情報を使った方が通常の uncertainty sampling などの能動学習手法よりも速く解に収束させられることが実験的に確かめられた.

【2020/04/18】Exponential Convergence Rates of Classification Errors on Learning with SGD and Random Features【AISTATS2020 under review】

[Yashima et al., *AISTATS* under review, 2020]

keywords : kernel, random feature, SGD, classification error

カーネルモデルの学習として Random Feature + SGD を使った時に, その推定量とベイズ判別器の差の汎化誤差が指数収束することを証明. 仮定として特筆すべきなのは "strong low-noise condition" と呼ばれる, 判別がしやすい状況になっていること. さらに, 指数収束するときの上界のレートは random feature の数 M に依存しない.

They prove that the estimator converges exponentially to the Bayesian discriminator when Random Feature + SGD is used to train the model of the kernel function. It is noteworthy that the condition called the "strong low-noise condition" is easy to discriminate. Moreover, the upper bound rate of exponential convergence does not depend on the random feature several M.

strong low-noise condition :

$\exists \delta \in (0, \frac{1}{2}),\ |\rho(Y=1|x) - \frac{1}{2}| > \delta,\ (\rho_{\mathcal{X}} - \mathrm{a.s.})$

【2020/04/17】Agnostic Active Learning Without Constraints 【NeurIPS2010】

[Beygelzimer et al., *NeurIPS*, 2010]

keywords : Active Learning, Importance weighted, rejection threshold

能動学習の方法の提案. Importance weighted active learning を用いたときの, 0-1判別損失で, しかもそれまでのクエリに依存する場合の汎化誤差と訓練誤差の差のバウンドを導出している. 真の関数が最適なベイズ判別関数にずっと近ければ, 高々 $O(\sqrt{n \log n})$ 点ぐらい調べれば良いということが言えて, これは普通の教師あり学習のレートより良くなる. しかし, 実験的には「うーん？？」というぐらいの精度しか出ていない印象. これはActive LearningよりSemi-supervised の方がいいと言われる所以にも繋がっているかも知れない.

Proposal of a method of active learning. they derive the bounds of the difference between the generalization and training errors in the case of 0-1 discriminant loss with Importance weighted active learning and dependence on previous queries. If the true function is much closer to the optimal Bayesian discriminant function, they can say that they need to check it at most O(√(C n log n)) points, which is better than the rate of ordinary supervised learning. Experimentally, however, results is slightly good rather than supervised learning. this may be the reason why it is said that Semi-supervised is better than Active Learning.

【2020/04/16】

Membership Inference Attacks Against Machine Learning Models

[Shokri et al., *IEEE SP*, 2017]

keywords : Membership Inference, black-box setting, hill-climbing

モデルfが与えられ, この時あるデータxが訓練データに入っているか否かを当てる問題をMembership Inferenceという. この研究ではモデルがblack-boxなAPIでしかアクセスできない状況を考え, その時にMembership Inferenceを行う手法を提案. 具体的にはまず山登り法で “訓練集合っぽいデータセット” (この時に真のモデルにクエリを投げる必要がある) を作成し, その後適当な2層ニューラルネットで判別関数を学習させる. この判別モデルがMemberかどうかを判断するモデルとなる.

Given a model f, the problem of guessing whether a certain data x is included in the training data or not is called membership inference. In this study, they propose a method to perform membership inference when the model is accessible only by a black-box API. They first create a “training set-like dataset” (at which time they need to query the true model) using the hill-climbing method, and then train the discriminate function in an appropriate two-layer neural net. This discriminate model is used as a model to determine whether the model is a member or not.

【2020/04/15】

Defending Against Machine Learning Model Stealing Attacks Using Deceptive Perturbations

[Lee et al., 2018]

keywords : Model Extraction, Defense, Reverse Sigmoid, ResNet

Model Extraction に対する防御方法を提案した論文. 発想としてはそのまま予測した確率ベクトルyを返すのではなくy + rという形で少し変形した値を返す, という[Alabdulmohsin et al., *CIKM*, 2014] と似た方法を取っている. (彼らはその論文を引用していないが) 結果は数値実験的に示しており, ノイズの加え方として “Reverse Sigmoid” を用いたものが最もDefenseとして良かったと述べている.

A paper that proposes a method to defend against Model Extraction. The idea is similar to that of [Alabdulmohsin et al., CIKM, 2014], in that instead of returning the predicted probability vector y as it is, return a slightly deformed value in the form of y + r. (They do not cite the paper. The results are shown numerically (although they do not cite the paper), and they say that the "Reverse Sigmoid" method of adding noise is the best as a defense.

【2020/04/14】

Model Extraction Warning in MLaaS Paradigm

[Kesarwani et al., *ACSAC*, 2018]

keywords : Model Extraction, Decision Tree, Information Gain, monitor

Model Extraction を複数のユーザがクエリを投げる, というセッティングで行う. このとき,決定木を構成し, Information Gainなどを計算することで user ごとのステータスを把握するアルゴリズムを提案. このとき, 決定木がうまく学習できているのに, Model Extraction はうまくできていないということになれば, それに対してWarningを出す, ということができる. 実用的な観点からの論文.

They run Model Extraction in the setting of multiple users throwing queries. In this case, they propose an algorithm to understand the status of each user by constructing a decision tree and calculating the information gain and so on. If the decision tree is well trained, but the Model Extraction is not well trained, they can issue a warning to the decision tree. This paper is written from a practical point of view.

【2020/04/13】

Convergence Guarantees for Adaptive Bayesian Quadrature Methods

[Kanagawa and Hennig, *NeurIPS*, 2019]

keywords : Adaptive Bayesian Quadrature, quasi Monte Carlo, weak adaptivity, Weak Greedy Alogrithm

Bayesian Quadrature は周辺尤度のような積分で表される量を適切な有限点で近似する手法だったが, それをAdaptiveにやる, つまり, $x_1 \dots x_n$ までをみた上で x_{n+1} を決めるAdaptive Bayesian Quadrature に対する理論保証は未だかつて与えられていなかった. 本研究ではABQがヒルベルト空間上の弱-貪欲なアルゴリズムと等価であることを示し, そこから真の量との誤差に関するレートを導出. カーネルが無限階微分可能なとき, そのレートは O(exp{- D n^{1/d}}) と極めて速い収束になることを述べている. 難しいがめちゃくちゃ面白い. Adaptiveな方が良い, ということまではまだ言えていないようだ.

The Bayesian Quadrature is a method to approximate the quantity represented by an integral, such as the marginal likelihood, at an appropriate finite point, but theoretical guarantees for the adaptive Bayesian Quadrature, which determines x_{n+1} by looking up to x_1 ... x_n, have not been given yet. In this work, they show that ABQ is equivalent to a weak-greedy algorithm on Hilbert space, from which they derive an error rate for the true quantity. they state that when the kernel is infinitely differentiable, the rate is O(exp{- D n^{1/d}}) and converges very fast. It is difficult, but it is very interesting. It seems that they have not yet said that Adaptive is better.

【2020/04/12】

Fastfood — Approximating Kernel Expansions in Loglinear Time

[Le et al., *ICML*, 2013]

keywords : Random Feature, Hadamard transform, FFT, Random Kitchen Sinks

[Rahimi and Recht, 2007]で提案されたRandom Featureによる基底関数の近似はグラム行列の計算を高速化するという意味で有効であったが, その時間計算量はO(nd)かかっていた. そこで, この研究ではFFTの亜種であるアダマール変換を用いることで計算量をO(n log d)にまで高速化する手法を提案. このとき, 不偏性と分散がO(1/n)と良いレートで近似できることを示している.

The approximation of the basis function by Random Feature proposed in [Rahimi and Recht, 2007] is effective in terms of speeding up the computation of gram matrices, but its time complexity is O(nd). In this work, they propose a method to speed up the computation time to O(n log d) by using the Adamar transform, a variant of FFT. It is shown that unbiasedness and variance can be approximated with a good rate of O(1/n).

【2020/04/10】

ACTIVETHIEF: Model Extraction using Active Learning and Unannotated Public Data

[Pal et al., *AAAI*, 2020]

keywords : Model Extraction, Public data, active learning, K-center strategy, DeepFool-based, Active Learning

Model Extractionの問題を考える際, 事前情報として「ラベルのないデータセット」が手元に大量にある場合の効率的な攻撃アルゴリズムを提案. 2018年ごろに提案されたK-center strategyやDeepFool-based Active Learning (DFAL) algorithm といった能動学習的な枠組みのアルゴリズムを用いる. 実験的に一様ランダムにクエリを投げるよりは良いことを述べているが, 微々たる上昇に見える.

In considering the problem of Model Extraction, they propose an efficient attack algorithm for the case where a large number of "unlabeled data sets" are at hand as prior information. they use algorithms from active learning frameworks such as the K-center strategy and the DeepFool-based Active Learning (DFAL) algorithm, which were proposed around 2018. The paper states that the algorithm is better than uniformly randomized queries experimentally, but it seems to be only a faint update.

【2020/04/09】Adding Robustness to Support Vector Machines Against Adversarial Reverse Engineering【CIKM2014】

[Alabdulmohsin et al., *CIKM*, 2014]

keywords : Model Extraction, linear, SVM, Pareto Optimality, SDP

線形SVMに対し, Model Extractionに強い学習方法を提案. 具体的には学習するモデルの重み w を正規分布 $\mathcal{N}(\mu, \Sigma)$からサンプリングされたものとして捉え, w を学習するのではなく, その μ, Σ を学習する半正定値計画問題として定式化する. このとき問題としては「判別を間違える確率が ν 以上」という目的で, かつSVMの条件を満たすような定式化となる. もちろん, 最尤推定的な最も良いものからずれたパラメータを学習することになるので, accuracyとrobustnessはトレードオフになるが, パレート最適なものを提案する. 往々にしてaccuracyが最大となるものがパレート最適な解の集合に入っているとは限らない.

For the linear SVM, they propose a learning method that is strong in model extraction, which is based on the normal distribution N(μ, Σ). Specifically, they consider the model weight w to be sampled from a normal distribution N(μ, Σ), and instead of learning w, they formulate a semi-positive definite programming problem that learns the μ, Σ. In this case, the problem is formulated in such a way that the probability of making a wrong discrimination is more than ν, and the condition of SVM is satisfied. Although there is a trade-off between accuracy and robustness, they propose a Pareto-optimal one, since they have to learn the parameters that are off from the best maximum likelihood estimator. Sometimes, the set of Pareto-optimal solutions does not always include the one with the highest accuracy.

$$
\begin{aligned}
\underset{\mu,\, s,\, \xi}{\text{minimize}} \quad & \frac{1}{2}\,\frac{\mu^T \mu}{1^T s} + C \sum_{i=1}^{m} \xi_i \\
\text{subject to} \quad & y_i \cdot (\mu^T x_i) \geq 1 + \Phi^{-1}(\nu) \sum_{j=1}^{n} x_{i,j}^2\, s_j - \xi_i \\
& s_j \geq 0, \qquad \text{for } j = 1,\, 2,\, \ldots,\, n \\
& \xi_i \geq 0, \qquad \text{for } i = 1,\, 2,\, \ldots,\, m \qquad (4)
\end{aligned}
$$

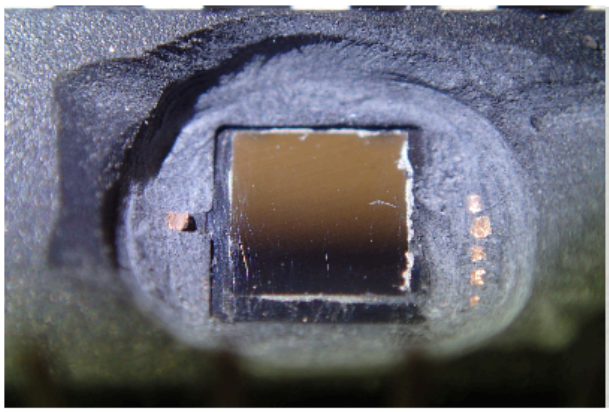
【2020/04/08】CSI Neural Network: Using Side-channels to Recover Your Artificial Neural Network Information【Security2019】

[Batina et al., *Security*, 2019]

keywords : Model Extraction, Side-channel, NN, activation function, hidden layer, input data, SPA, DPA, HPA

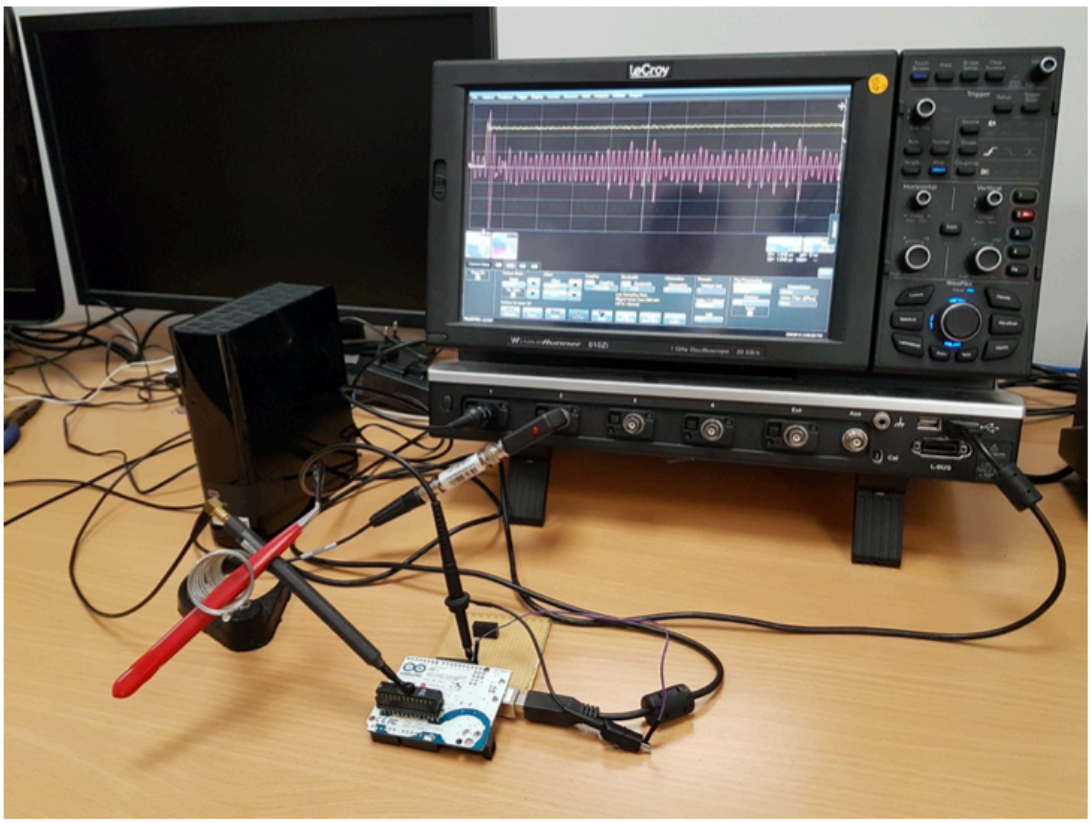
外部から直接ハードウェアとしてGPU・CPUに触り, 電磁気的な周波数を読み取ることで, (アーキテクチャの情報はわかった上で) 内部のモデルの活性化関数・層の数・ニューロン数などにまつわる情報を抜き出す方法を提案. これは代表的なReverse-Engineeringの手法であるSPAやDPA といった手法(元々はRSA暗号などのスキミングに使われていた)を用いている. また, 他の攻撃としてモデルが既知で, インプットデータが未知な時にそのデータをスキミングしてHPAという手法を用いて復元することも提案している. 理論屋の頭のどこにもない攻撃の仕方でとても興味深い.

They propose a method to extract information about the activation function, number of layers, number of neurons, etc. of the internal model by directly touching the GPU and CPU as hardware from the outside and reading the electromagnetic frequencies (with the architectural information known). they use typical Reverse-Engineering techniques such as SPA and DPA (originally used for skimming of RSA cryptography). they also propose that they can recover the input data by skimming the data when the model is known and the input data is unknown, using HPA. This is a very interesting attack because it have never seen before in the theorists' minds.



(a) Target 8-bit microcontroller mechanically decapsulated

(b) Langer RF-U 5-2 Near-field Electromagnetic passive Probe



(c) The complete measurement setup

Fig. 3: Experimental Setup

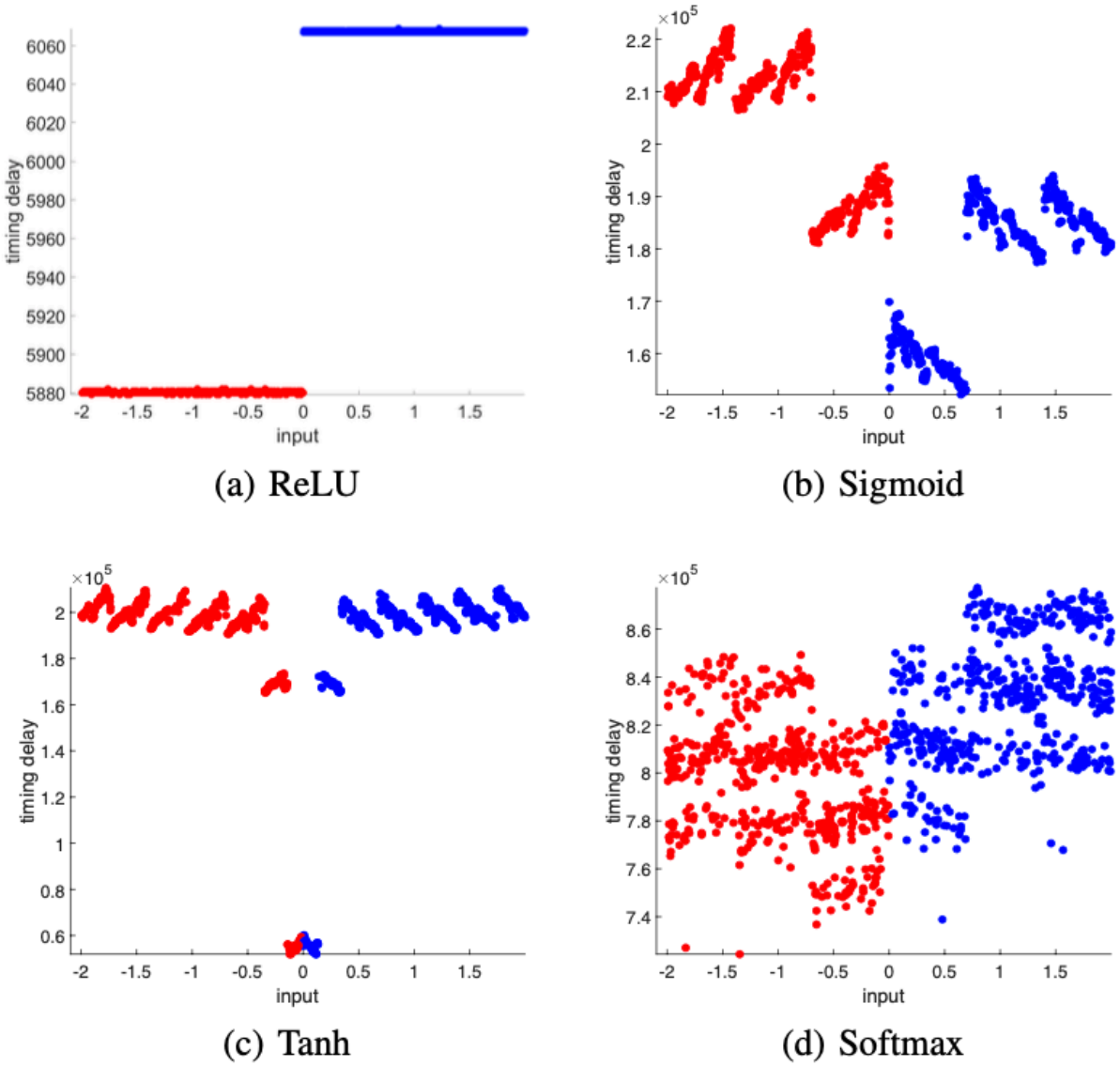


Fig. 5: Timing behavior for different activation functions

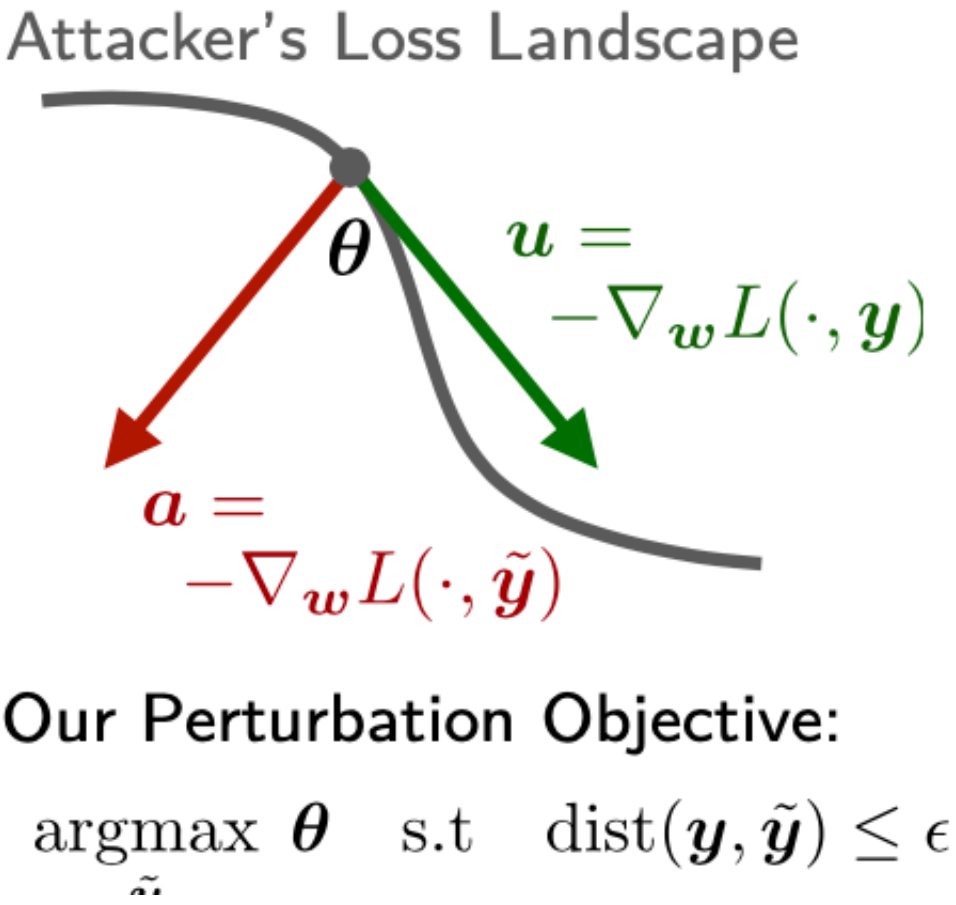
【2020/04/07】Prediction poisoning: Towards defenses against DNN model stealing attacks【ICLR2020】

[Orekondy et al., *ICLR*, 2020]

keywords : Model Extraction, Defense, Maximizing Angular Deviation (MAD), actively defense, NN, LeNet, VGGNet16

Model Extractionしてくる敵対者に対してどのように防御するかを論じた研究. 対策としてMAD (Maximizing Angular Deviation) という方法を提案. 予測の時にそのまま返すはずだった値を返すのではなく, Adversaryがその点において勾配をとると最もロスを下げにくい方向になっている点にちょっと変更して返す. 発想はシンプルだが面白い. 実験的に提案手法の良さを述べている. Model Extractionの対象はニューラルネットで, 既存の Model Extraction のアルゴリズムとしては [Orekondy et al., *CVPR*, 2019] などを用いている. というかこれを書いているのはKnockoff Netの著者である.

A study that discusses how to defend against adversaries who come to Model Extraction, and propose a method called MAD (Maximizing Angular Deviation) as a countermeasure. Instead of returning the value that should have been returned at the time of prediction, the method slightly changes the slope of the Adversary at that point to the point where the loss is least likely to be reduced. The idea is simple, but interesting. They experimentally describe the merits of the proposed method. The target of Model Extraction is a neural net, and they use existing algorithms of Model Extraction such as [Knockoff Net, CVPR, 2019]. The writer of this paper is the author of Knockoff Net.



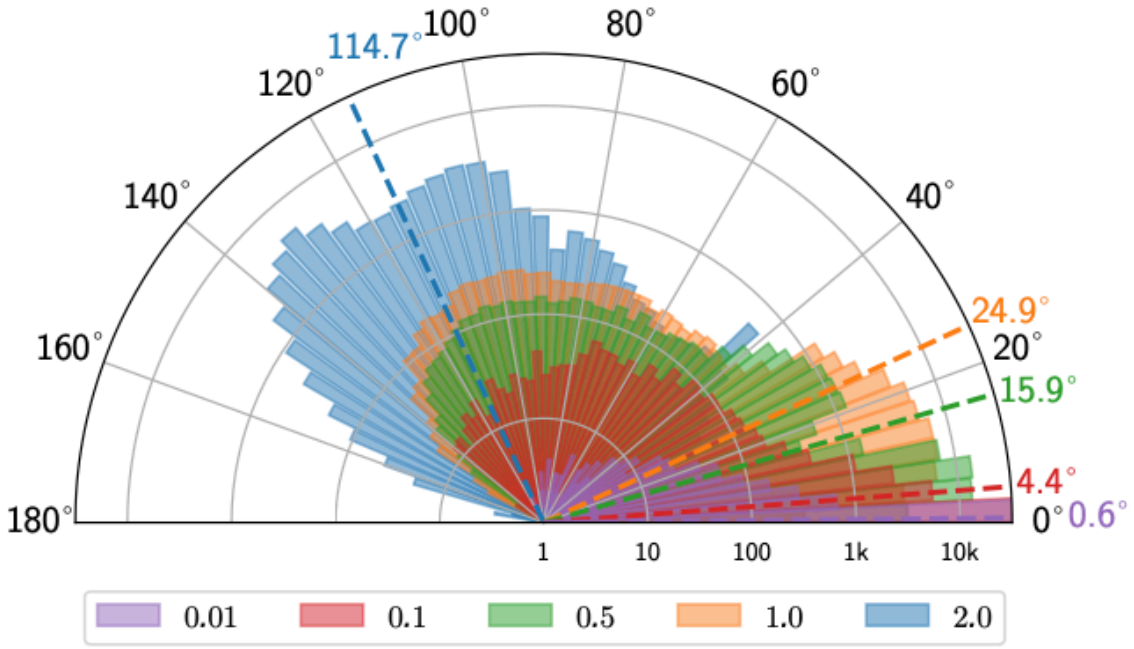


Figure 6: Histogram of Angular Deviations. Presented for MAD attack on CIFAR10 with various choices of ϵ.

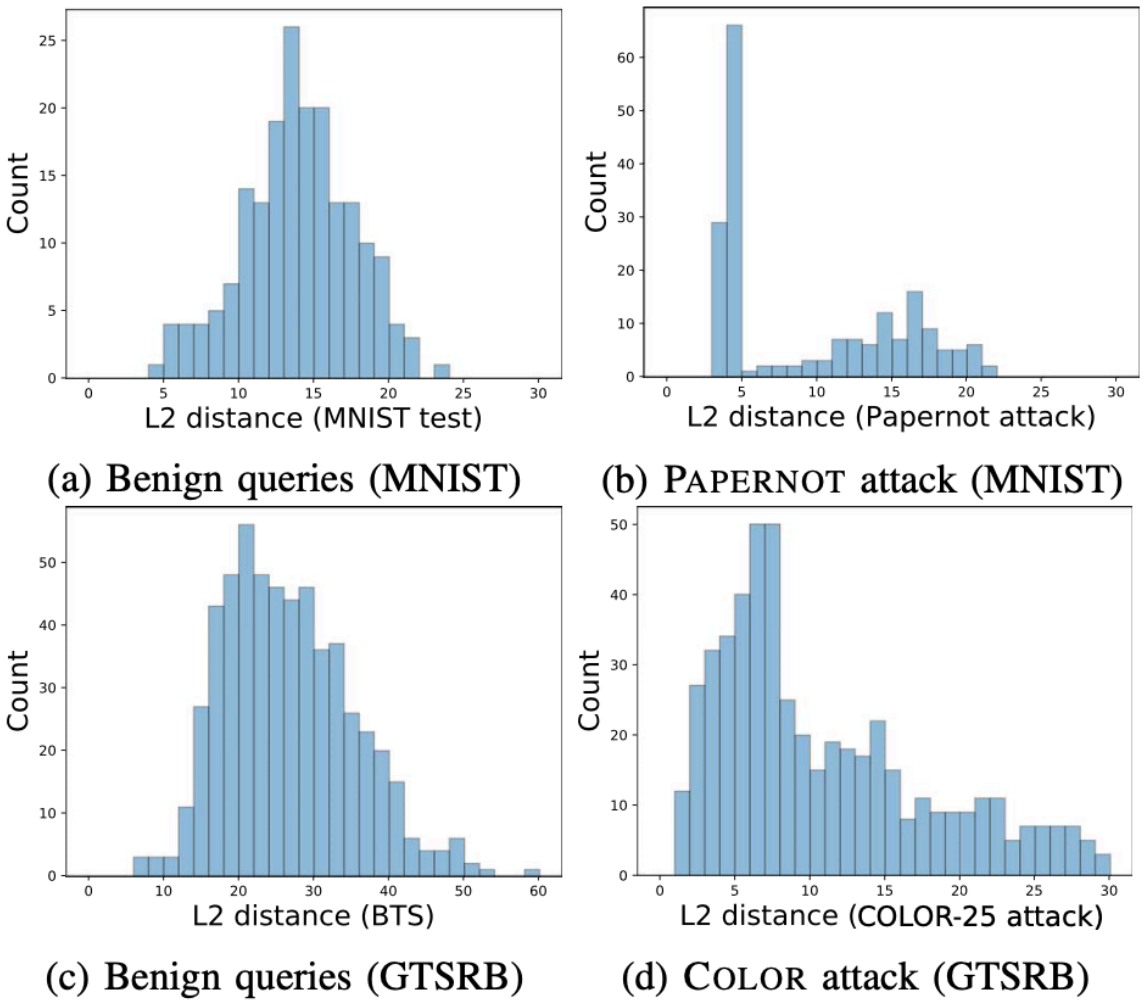
【2020/04/06】PRADA: Protecting Against DNN Model Stealing Attacks【EuroS&P2019】

[Juuti et al., *EuroS&P*, 2019]

keywords : Model Extraction, Model Stealing, Adversarial Example, DNN, Shapiro-Wilk test

敵対者がModel Extractionをしようとしているか判別するアルゴリズムを提案. 「敵対者はうまくクエリを構成するので人為的な分布になるはず」という仮定から, 入力がどれくらい正規分布と離れているかをシャピロ–ウィルク検定を用いて判断. 前半部分では種々のModel Extractionアルゴリズムの比較実験が行われている. 結果として, 彼らが提案したアルゴリズムは「偽陽性」の意味で実験的に優位な結果を示した.

they propose an algorithm to discriminate whether the adversary is trying to do Model Extraction or not. they use the Shapiro-Wilk test to determine how far the input is from the normal distribution, assuming that the adversary constructs the query well and that it should be artificially distributed. In the first part of the paper, a comparison experiment between various Model Extraction algorithms is conducted. As a result, their proposed algorithm shows experimental superiority in the sense of "false positives".



【2020/04/05】Efficiently Stealing your Machine Learning Models【WPES2019】

[Reith et al., *WPES*, 2019]

keywords : Model Extraction, SVM, SVR, re-training

対象とするモデルはSVM ・SVR(カーネルを用いた判別・回帰関数)でシンプルなModel Extraction. (訓練データなどの情報を使わない) 途中で ”arbitrary kernel” に対して学習できると言っているが, レートの導出はなく, 実験的に示しているだけである. 方法としては[Lowd and Meek, 2005]の拡張のやり方と, re-trainingでやっている. プラクティカルにいろんなカーネルに対してできるということを言っているのは実装上かなり参考になりそうではある.

The target model is SVM and SVR (discriminant and regression functions using the kernel), and it is a simple model extraction. They say that the model can be trained against "arbitrary kernel" on the way (without using training data and other information), but they don't derive the rate, and they only show it experimentally. they use the extension of [Lowd and Meek, 2005] and re-training as a method of learning. The fact that they can do it practically for various kernels may be helpful for the implementation.

【2020/04/03】On the Equivalence between Kernel Quadrature Rules and Random Feature Expansions【JMLR2017】

[Bach, *JMLR*, 2017]

keywords : Random feature, Quadrature, positive definite kernel

積分で表現される量を近似するときに有限点で近似する数値積分のことをQuadratureという. 正定値カーネル関数から導かれる積分作用素を考えたとき, これのQuadratureは実はRandom Featureの拡張とみなすことができる. また, このとき真の関数を近似するための最適なサンプリング分布を与えている. これはものすごくインパクトがある. また, そのときの近似誤差のレートの上界と下界を与えており, そのレートはともに $\log(1/\delta)$ である.

A finite point approximation of a quantity represented by an integral is called a quadrature. Considering integral operators derived from positive definite kernel functions, this quadrature can actually be regarded as an extension of Random Feature. In addition, it gives an optimal sampling distribution to approximate the true function. This is very impactful. they also give the upper and lo theyr bounds of the approximation error rate, both of which are $\log(1/\delta)$.

Quadrature の説明スライド (参考) :

https://www.cs.toronto.edu/~duvenaud/talks/intro_bq.pdf

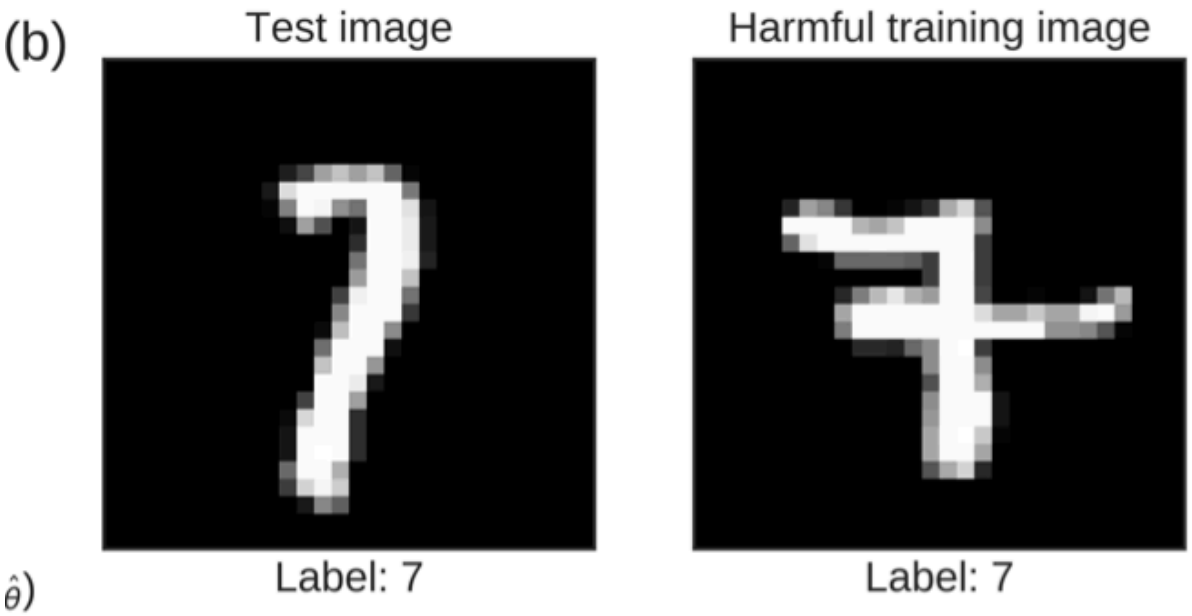
【2020/04/02】Understanding Black-box Predictions via Influence Functions【ICML2017】

[Koh and Liang, *ICML*, 2017]

keywords : Interpretability, influence function, robust statistics, hessian

訓練データのうち, どのデータがロス関数の最小化に寄与しているかを, 「ロバスト統計」の代表的な道具の一つである, 「影響関数」を用いて測定することを提案. 実際, 影響を計算するにはロス関数をモデルパラメータで微分したときのヘッセ行列が必要となるため, 数百万パラメータを持つニューラルネットなどではそのままでは計算できない. そこで implicit Hessian-vector productsという手法を用いることで計算量を削減. これらを用いることで「学習に害をもたらすような訓練データ」であったり, 「訓練データに対するAdversarial Exampleの生成」といったことが可能になる.

They propose to measure which data among the training data contribute to the minimization of the Ross function by using the "effect function", one of the representative tools of "robust statistics". In fact, the effect requires a Hessian matrix of the Ross function differentiated by the model parameters, which cannot be computed on a neural net with several million parameters. They use the implicit Hessian-vector products method to reduce the computational complexity. By using these products, it is possible to generate Adversarial Examples for training data and training data that are harmful to learning.



【2020/04/01】Thieves on Sesame Street! Model Extraction of BERT-based APIs【ICLR2020】

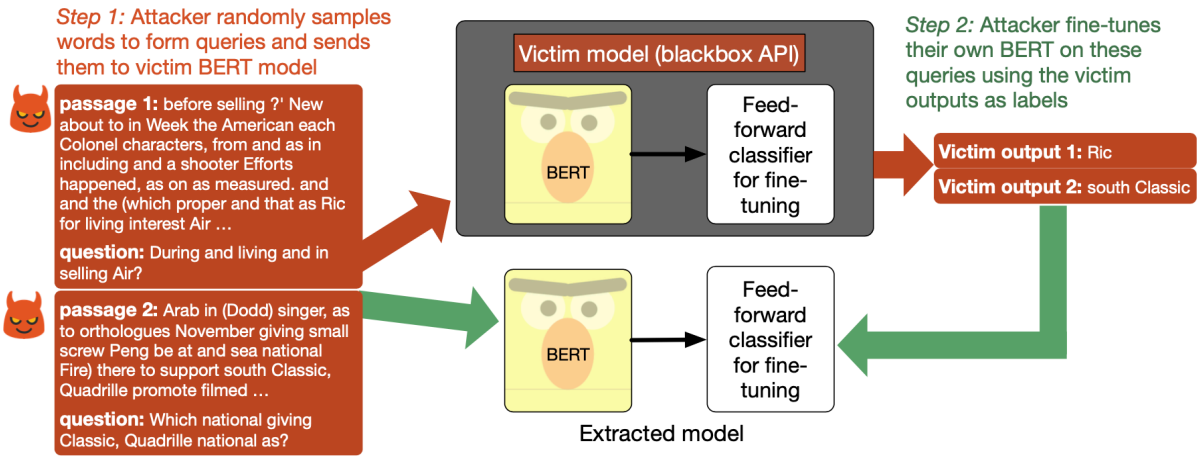
[Krishna et al., *ICLR*, 2020]

keywords : Model Extraction, Neural Network, BERT, NLP, fine-tuning

BERTに基づいた自然言語処理モデルが攻撃対象. 「BERTが使われている」ということは知った上でのModel Extraction. 今までのModel Extractionのアルゴリズムは連続なドメイン上での入力に基づくものが大半だったので, そのままNLPタスクに用いることはできなかった. この研究では自然言語処理の場合, どんなクエリを投げると効率的にModel Extractionができるかについて数値的に実験. 結果, wikipediaのテキストセットなどからランダムにクエリを抽出するだけで十分効率的にModel Extractionができることを指摘.

Natural language processing model based on BERT is attacked. Model Extraction is based on the knowledge that "BERT is used". Since most of the previous Model Extraction algorithms theyre based on input on a continuous domain, they could not be used for NLP tasks as they are. In this study, they experimented numerically to find out what kind of queries can be thrown to efficiently perform Model Extraction in the case of natural language processing. As a result, they pointed out that it is enough to extract a random query from a wikipedia text set, etc., to perform model

extraction efficiently.



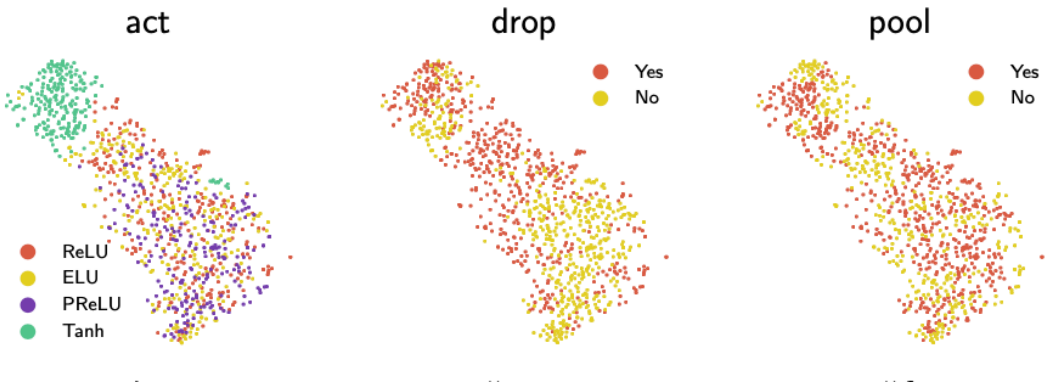
【2020/03/31】Towards reverse-engineering black-box neural networks【ICLR2018】

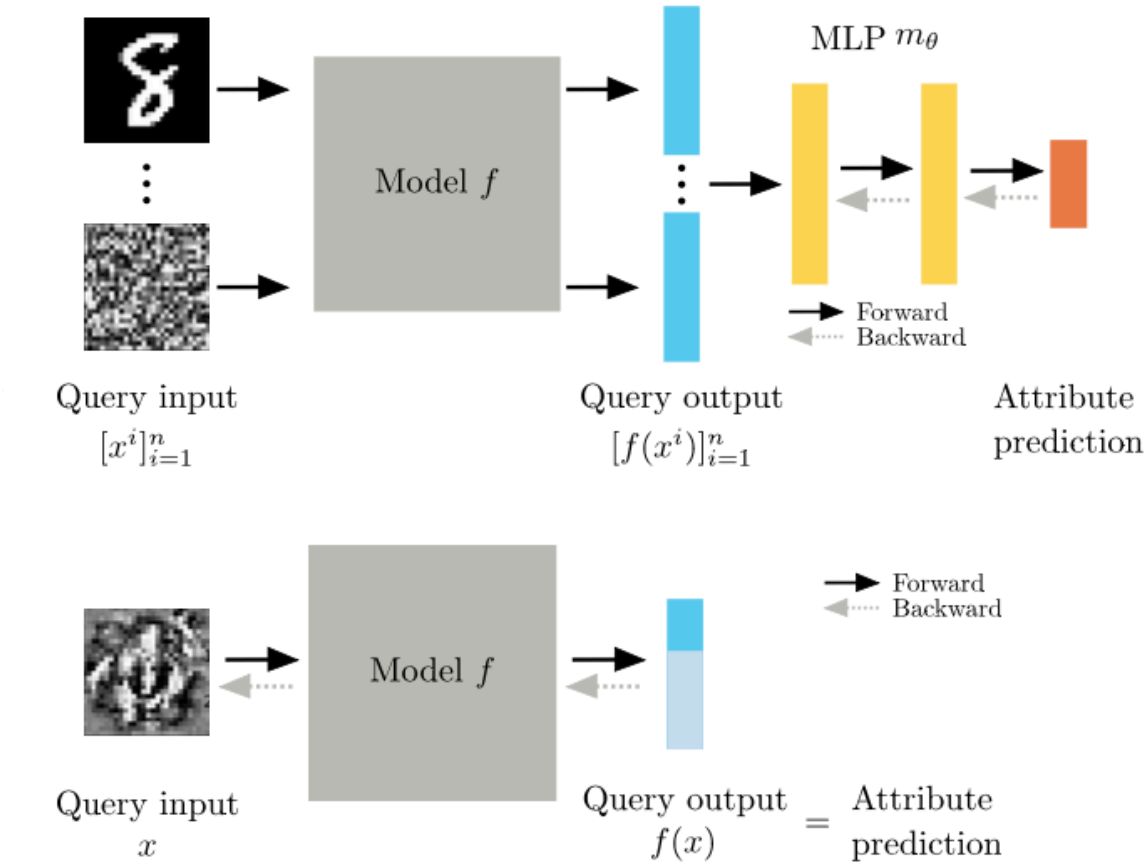
[Oh et al., *ICLR*, 2018]

keywords : Model Extraction, Neural Network, architecture, optimization process, training data, metamodel

ブラックボックスなモデルがデプロイされている状況で「意味のある」情報を抜き出す方法を提案. メタモデル的なアプローチ. 対象としては architecture (e.g. which activation, max-pooling), 最適化手法 (e.g. SGD or ADAM), 訓練データ(e.g. MNIST) を読み取ることを目標とする. 最適化手法も外から見た入出力をNNにいっぱい食わせればわかるんじゃね？という発想は面白すぎる. 実際, t-SNEで可視化してみたところアルゴリズムごと, architectureごとにクラスタを形成していたりする. 驚愕.

A method for extracting "meaningful" information in situations where a black box model is deployed. Metamodel-like approach. they aim to read architecture (e.g., which activation, max-pooling), optimization methods (e.g., SGD or ADAM), and training data (e.g., MNIST) as targets. The optimization method can also be understood by filling NN with inputs and outputs seen from the outside, right? This idea is too interesting. In fact, when they visualize it with t-SNE, it can be seen that each algorithm and architecture forms a cluster. I am astonished.





【2020/03/30】Adversarial Learning【KDD2005】

[Lowd and Meek, *KDD*, 2005]

keywords : Model Extraction, linear classifier, adversarial example, ACRE learning

(現在のところ) 最古のModel Extraction論文. [Lowd and Meek, 2005] はModel Extractionを論じようと思ったらよく出てくる. Model Extraction論文と言ったが, 実は, 現在で言うところAdversarial Exampleの(あるコスト関数下での)構成の仕方について述べている. 対象は線形判別関数で, 定義域が連続な場合と離散の場合で異なるフレームワークを提案. なぜModel Extraction論文の始祖として見なされているかというと, Adversarial Exampleを作る時に一旦線形判別の超平面を推定するアルゴリズムになっていて, これがModel Extractionになっているから. 真のパラメータとの乖離をεとした時に 1 + ε に関する多項式オーダーでModel Extractionができることを証明している. (定義域が連続な場合)

The oldest paper of Model Extraction. [Lowd and Meek, *KDD*, 2005] is famous in the area of Model Extraction. In fact, this paper don't directly propose Model Extraction but propose how to get Adversarial Example (is now called). Target model is linear classifier, domain is continuous or discrete. The reason that this paper is called by the first Model Extraction is the algorithm to create Adversarial Example contains Model Extraction for linear classifier. They proved that Model Extraction which requires the polynomial number of queries about 1 + ε, that is the distance between true parameter and estimated parameter.

Algorithm 1 FINDCONTINUOUSWEIGHTS(x^+, x^-, ϵ, δ)

$(\mathbf{s}^+, \mathbf{s}^-, f) \leftarrow \text{FINDWITNESS}(\mathbf{x}^+, \mathbf{x}^-)$
$w_f \leftarrow 1.0 \cdot (s_f^+ - s_f^-)/|s_f^+ - s_f^-|$
Use $(\mathbf{s}^+, \mathbf{s}^-)$ to find negative instance $\mathbf{x}$ with $\mathbf{gap}(\mathbf{x}) < \epsilon/4$
$x_f \leftarrow x_f - w_f$
for each feature $i \neq f$ **do**
  Let $\hat{\mathbf{i}}$ be the unit vector along the ith dimension
  if $c(\mathbf{x} + \hat{\mathbf{i}}/\delta) = c(\mathbf{x} - \hat{\mathbf{i}}/\delta)$ **then**
    $w_i \leftarrow 0$
  else
    $w_i \leftarrow \text{LINESEARCH}(\mathbf{x}, i, \epsilon/4)$
  end if
end for

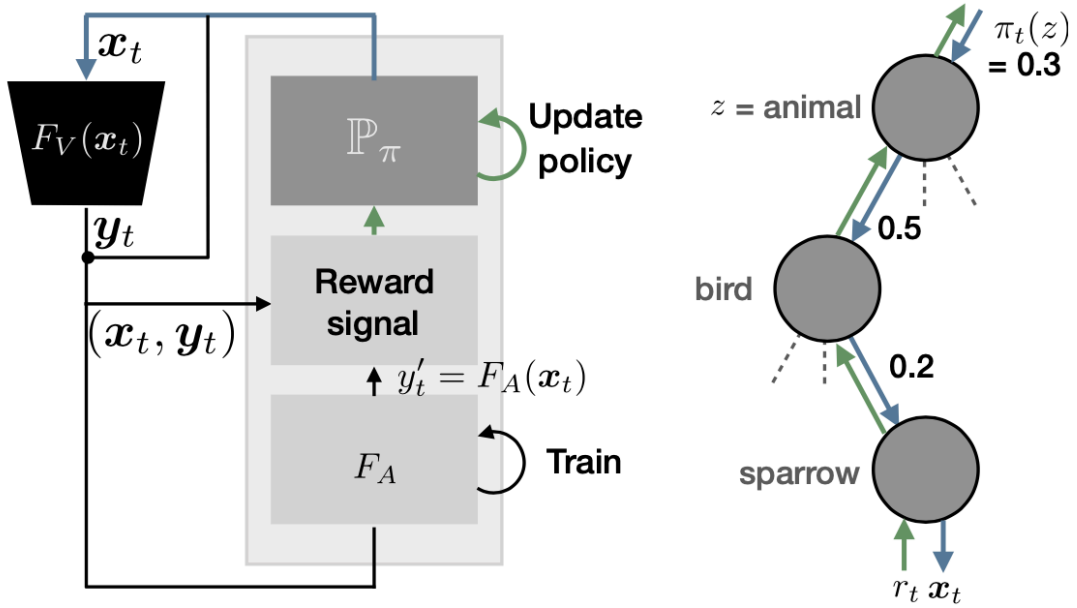
【2020/03/29】Knockoff Nets: Stealing Functionality of Black-Box Models【CVPR2019】

[Orekondy et al., *CVPR*, 2019]

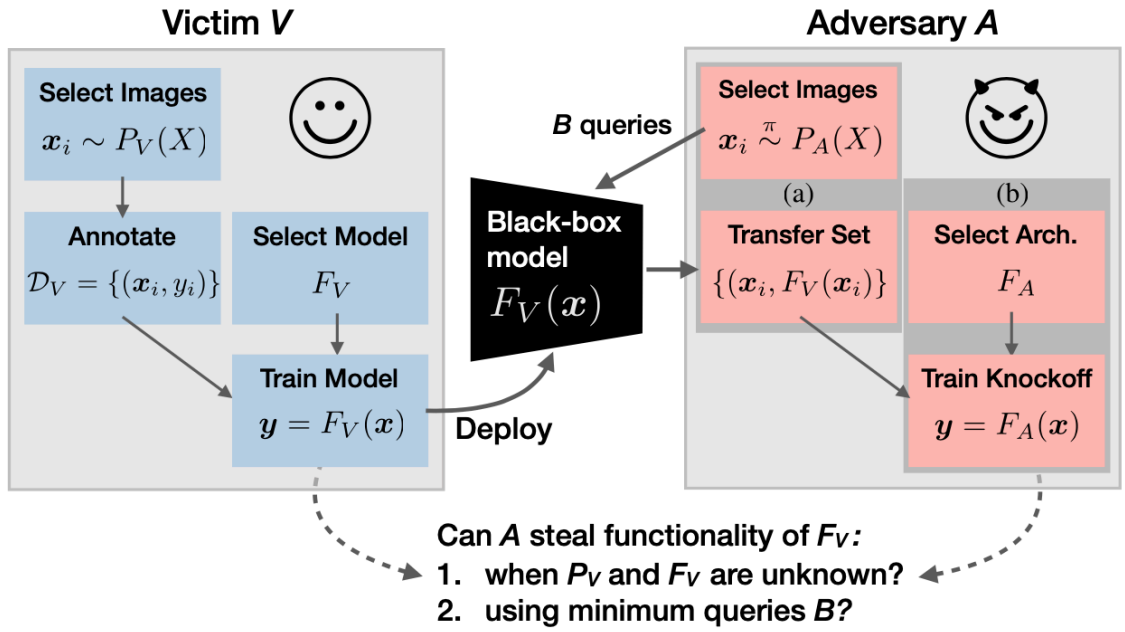
keywords : Model Extraction, copy, knockoff, distillation, reinforcement learning

画像認識タスクで判別モデルがデプロイされている時にそれをコピーするモデル(Knockoff model)を手元で構成する方法を提案. [Tramer et al., *Security*, 2016] などではモデルが貧弱だったよね, ということを指摘. ResNetでExtractionができるかどうか実験している. 理論的な解析はない. アプローチとしては一様ランダムにクエリを投げる方法と強化学習 + 能動学習的に決める方法を用いている. Distillation (蒸留) との関連性も指摘. 訓練データを用いるセッティング(closed-world)と, 訓練データとは異なる画像データセットを用いるやり方(open-world)で実験を行なっている.

Knockoff model that copies true black-box model is proposed. In [Tramer et al., 2016] setting, target model is unrealistic or simple (like Decision Tree, Linear Classifier), so they do experiments for more complex model like ResNet. This research has no theoretical analysis. The algorithm uses uniformly random method or reinforcement learning + active learning. Furthermore, they proposed that closed-world setting experiment (known training sets) and open-world setting experiment (unknown training sets)



(a) Overview (b) Hierarchical policy $\mathbb{P}_\pi$



【2020/03/28】Stealing Hyperparameters in Machine Learning【S&P2018】

[Wang and Gon, *IEEE S&P*, 2018]

keywords : Hyperparameters, Stealing, Attack

ハイパーパラメータを盗む研究. 特に今回は正則化係数を盗むことを念頭においたアルゴリズムを提案. メインのアイディアは「使われているモデルは最適化が行われた後だ」という点に着目し,L(w) = l(w, X) + λ r(w, X) の勾配 ∇_w L(w) = 0 となるようなλを見つけに行くということを行う. 理論というよりは実験が多め. 防御としてrounding (0.9634を返す時に0.96を返す, みたいな) で実験しているが大した影響はなかったようだ.

またこの論文は「攻撃」の Related Workがきっちりとまとまっていて,

- Poisoning Attack
- Data Evasion Attack
- Model Evasion Attack
- Model Extraction Attack

と簡潔に攻撃を4つに分類している. (ただし排他的な分類になっているか？というと疑問だが) サーベイ論文の[Papernot et al., 2016]より読みやすい.

Stealing Hyperparameters. In this research, Hyperparameter is coefficient of regularization term. Crucial point is that Training is Optimization that minimize L(w) = l(w, X) + λ r(w, X). So it is reasonable to find the point λ which satisfy the condition ∇_w L(w) = 0. This research contains more experiments than theory. Defense method which has been proposed in the former paper, that is rounding, has few affect their stealing algorithm.

TABLE I: Loss functions and regularization terms of various ML algorithms we study in this paper.

Category	ML Algorithm	Loss Function	Regularization
Regression	RR	Least Square	L_2
	LASSO	Least Square	L_1
	ENet	Least Square	$L_2 + L_1$
	KRR	Least Square	L_2
Logistic Regression	L2-LR	Cross Entropy	L_2
	L1-LR	Cross Entropy	L_1
	L2-KLR	Cross Entropy	L_2
	L1-BKLR	Cross Entropy	L_1
SVM	SVM-RHL	Regular Hinge Loss	L_2
	SVM-SHL	Square Hinge Loss	L_2
	KSVM-RHL	Regular Hinge Loss	L_2
	KSVM-SHL	Square Hinge Loss	L_2
Neural Network	Regression	Least Square	L_2
	Classification	Cross Entropy	L_2

【2020/03/27】Random Features for Large-Scale Kernel Machines【NeurIPS2007】

[Rahimi and Recht, *NeurIPS*, 2007]

keywords : kernel, random feature, Bochner's theorem, Fourier transform

カーネル法に基づく回帰・判別問題は一般にグラム行列を構成するので計算量的に辛いことも多い. そこでカーネルの内積表現をランダムに基底を構成することで, ユークリッドの内積で近似してしまおうというのがメインアイディア. これをRandom Featureという. これは連続正定値カーネルが確率分布と1対1に対応するところから導かれる.

It is said that Kernel method requires the high computation complexity because of Gram Matrix. The solution for this difficulty is to create randomly basis in (approximate) inner product space (Hilbert Space). This is called for Random Feature. The property is induced by the fact that positive definite kernel has one-to-one relationship for probability distribution.

Algorithm 1 Random Fourier Features.

Require: A positive definite shift-invariant kernel $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$.
Ensure: A randomized feature map $\mathbf{z}(\mathbf{x}) : \mathcal{R}^d \to \mathcal{R}^D$ so that $\mathbf{z}(\mathbf{x})'\mathbf{z}(\mathbf{y}) \approx k(\mathbf{x} - \mathbf{y})$.
Compute the Fourier transform p of the kernel k: $p(\omega) = \frac{1}{2\pi} \int e^{-j\omega'\delta} k(\delta)\, d\Delta$.
Draw D iid samples $\omega_1, \cdots, \omega_D \in \mathcal{R}^d$ from p and D iid samples $b_1, \ldots, b_D \in \mathcal{R}$ from the uniform distribution on $[0, 2\pi]$.
Let $\mathbf{z}(\mathbf{x}) \equiv \sqrt{\frac{2}{D}} \left[\cos(\omega_1'\mathbf{x}+b_1) \cdots \cos(\omega_D'\mathbf{x}+b_D) \right]'$.

【2020/03/26】High Accuracy and High Fidelity Extraction of Neural Networks【2020】

[Jagielski et al., 2020]

keywords : Fidelity, Accuracy, Model Extraction, 2-layer NN, confidence score

2層 ReLu ニューラルネットのModel Extraction を行う. [Milli et al., *FAT*, 2019]ではアウトプットとして勾配 $\nabla_x f(x)$ まで手に入る設定だったが, ここではアウトプットの値そのもの $f(x)$ のみが手に入る設定でアルゴリズムを構築. また, Model Extraction という問題の定式化そのものも他の論文より割と厳格に定式化している.

This paper introduce the algorithm to extract 2-layer ReLU Neural Network from exact recovery. It is different from the re-training (active learning) approach. This research is strongly related to **Model Reconstruction from Model Explanations** [Milli et al., FAT, 2019]. Main difference is the THRET MODEL. The former research's setting is stronger because **getting gradient w.r.t. x is unrealistic**. But this time, Their approach **only use confidence score output.**

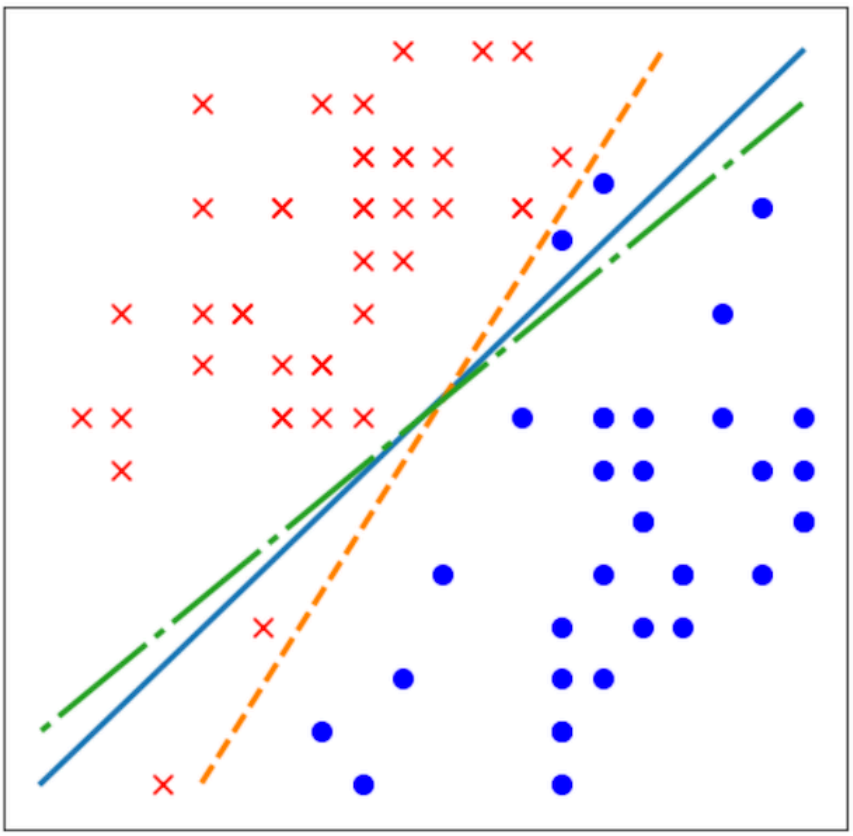


Figure 1: Illustrating fidelity vs. accuracy. The solid blue line is the oracle; functionally equivalent extraction recovers this exactly. The green dash-dot line achieves high fidelity: it matches the oracle on all data points. The orange dashed line achieves perfect accuracy: it classifies all points correctly.

Attack	Type	Model type	Goal	Query Output
Lowd & Meek [8]	Direct Recovery	LM	Functionally Equivalent	Labels
Tramer *et al.* [11]	(Active) Learning	LM, NN	Task Accuracy, Fidelity	Probabilities, labels
Tramer *et al.* [11]	Path finding	DT	Functionally Equivalent	Probabilities, labels
Milli *et al.* [19] (theoretical)	Direct Recovery	NN (2 layer)	Functionally Equivalent	Gradients, logits
Milli *et al.* [19]	Learning	LM, NN	Task Accuracy	Gradients
Pal *et al.* [15]	Active learning	NN	Fidelity	Probabilities, labels
Chandrasekharan *et al.* [13]	Active learning	LM	Functionally Equivalent	Labels
Copycat CNN [16]	Learning	CNN	Task Accuracy, Fidelity	Labels
Papernot *et al.* [7]	Active learning	NN	Fidelity	Labels
CSI NN [25]	Direct Recovery	NN	Functionally Equivalent	Power Side Channel
Knockoff Nets [12]	Learning	NN	Task Accuracy	Probabilities
Functionally equivalent (this work)	Direct Recovery	NN (2 layer)	Functionally Equivalent	Probabilities, logits
Efficient learning (this work)	Learning	NN	Task Accuracy, Fidelity	Probabilities

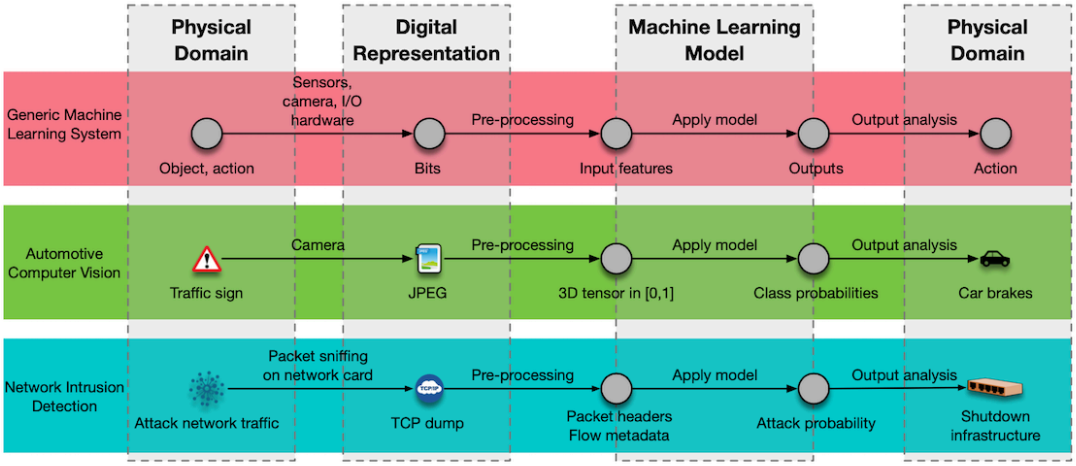
【2020/03/25】SoK: Towards the Science of Security and Privacy in Machine Learning【2016】

[Papernot et al., 2016]

keywords : survey, model extraction, security, differential privacy, adversarial example

機械学習の安全性とプライバシーに関するサーベイ論文. いくつかの研究を格言としてまとめている. また, この論文特有の貢献として防御方法に対する no free lunch 定理を示している.

This paper is a survey paper related to the security for ML. This is almost comprehensive survey, and contains several **MAXIMS** (e.g. Search algorithms for pisoning points are computationally expensive because of the complex and often poorly understood relationship between a model's accuracy on training and test distribtuions) Discussion has been done from various actual perspective, this is good point. This papers's concrete contribution is to prove the **no free lunch theorem for defense procedure**



Writer's presentation :

https://www.microsoft.com/en-us/research/uploads/prod/2018/03/Nicolas-Papernot-Security-and-Privacy-in-Machine-Learning.pdf

【2020/03/24】Exploring Connections Between Active Learning and Model Extraction【Security2020】

[Chandrasekaran et al., *USENIX*, 2020]

keywords : Model Extraction, Active Learning, Kernel SVM, Decision Tree, Defense

Active Learning と Model Extraction の関係性を指摘. 防御方法としてノイズを加える方法とそのときの理論解析を行なっている. ただ, 特定のモデル (Kernel SVM や Decision Tree) に対する Model Extraction そのものの収束レート解析までは見られない.

This paper says that “Model Extraction problem can be descried in the form of **Active Learning**” . It is lack of analysis of conversion rate for specific model (Kernel SVM and Decision Tree)extraction. But, it is good point this paper contains analysis of defense strategy (randomization). This paper is very good to understand the relation between active learning and model extraction, but this is lack of actual extraction algorithm (and its analysis). This area may be an untouched area yet.

【2020/03/23】Model Reconstruction from Model Explanations【FAT2019】

[Milli et al., *FAT*, 2019]

keywords : Model Extraction,2-layer NN, gradient setting

2層Neural Network に対するModel Extraction. 主となる結果は"真の関数f は$O(h\log(h/\delta))$ だけクエリを投げるとExtractできる" ということ. ここで h は隠れ層のニューロンの数である. このレートはメンバーシップクエリ (Active Learning) の設定における $O(dh\log(h/\delta))$ より速い. しかし, データ x についての微分 $\nabla_x f(x)$ が返ってくるという状況設定における話なので, この仮定は少し強い. 状況設定としては saliency map などに似ている.

Model Extraction for 2-layer NN model. Main theoretical result is “true target function f can be extracted in $O(h\log(h/\delta))$”, where h is hidden layer size. It is faster than $O(dh\log(h/\delta))$ in membership queries (active learning). It may be strong the assumption to get **gradient w.r.t. x (data).** This situation is similar to saliency map, interpretable tool for ML.

【2020/03/22】Stealing Machine Learning Models via Prediction APIs【Security2016】

[Tramèr, et al., *Security*, 2016]

keywords : Model Extraction, Path-Finding, Decision Tree, equation solving

This papers defines recent “Model Extraction” problem. Main Themes are

Section 4 Extraction with Confidence Values

- target : Stealing Logistic Regression - method : equation solving
- target : Stealing Multi Layer Perceptron - method : equation solving
- target : Stealing training data from Kernel Logistic Regression - method : (In data leakage setting,) Gradient Descent
- target : Stealing training data on extracted models
- target : Stealing Decision Tree - method : path-finding

Section 5 Model Extraction for Actual Services

- BigML
- AWS

Section 6 Model Extraction given class labels only

- target : Stealing Linear binary model - method : [Lowd and Meek, 2005], retraining
- target : Stealing Multi class Logistic Regression Model - method : retraining
- target : Stealing Neural Networks - method : retraining
- target : Stealing RBF Kernel SVMs - method : retraining

References

[1] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016.

[2] Smitha Milli, Ludwig Schmidt, Anca D Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 1–9, 2019.

[3] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, page : prepublication, 2020.

[4] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. SoK: Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[5] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural net-works. *arXiv preprint arXiv:1909.01838*, 2020.

[6] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[7] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in ma-chine learning. In *2018 IEEE Symposium on Security and Privacy (S&P)*, pages 36–52, 2018.

[8] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4954–4963, 2019.

[9] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005.

[10] Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. *InInternational Conference onLearning Representations*, 2018.

[11] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot,and Mohit Iyyer. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *International Conference on Learning Representations*, 2020.

[12] Pang Wei Koh and Percy Liang. Understanding black-box predictions viainfluence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894, 2017.

[13] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*,18(1):714–751, 2017.

[14] Robert Nikolai Reith, Thomas Schneider, and Oleksandr Tkachenko. Efficiently stealing your machine learning models. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 198–210, 2019.

[15] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. PRADA: Protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 512–527, 2019.

[16] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against DNN model stealing attacks. *International Conference on Learning Representations*, 2020.

[17] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. Csi neural network: Using side-channels to recover your artificial neural network information. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019.