Survey in M2

Created by Yuto Mori / 森 雄人 (D_dof) 

Table of Contents

Version

This version is 0.2.

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
- Semi-supervised Learning
- Kernel Methods
- Machine Teaching
- Gaussian Process

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 = **A**ssociation for the **A**dvancement of **A**rtificial **I**ntellengence
- AISTATS = International Conference on **A**rtificial **I**ntelligence and **Stat**istics
- JMLR = **J**ournal of **M**achine **L**earning **R**esearch
- S&P = IEEE Symposium on **S**ecurity **and P**rivacy
- Security = USENIX **Security** Symposium
- FAT = Conference on **F**airness, **A**ccountability, and **T**ransparency
- KDD = ACM SIGKDD International Conference on **K**nowledge **D**iscovery in **D**ata mining
- WPES = **W**orkshop on **P**rivacy in the **E**lectronic **S**ociety
- Euro S&P = IEEE **Euro**pean Symposium on **S**ecurity **and P**rivacy
- CIKM = **C**onference on **I**nformation and **K**nowledge **M**anagement
- ACSAC = **A**nnual **C**omputer **S**ecurity **A**pplications **C**onference
- FoCM = **Fo**undations of **C**omputational **M**athematics
- SICOMP = **SI**AM Journal on **Comp**uting
- COLT = Annual Conference on **Co**mputational **L**earning **T**heory
- TPMAI = IEEE **T**ransactions on **P**attern **A**nalysis and **M**achine **I**ntelligence

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可能性があります. 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.

- This survey may contain wrong summary since I often read papers in a hurry. Please pay attention to this.
- Mainly, Translation by DeepL . Thanks for exellent service !!!
- But I often don't check the precise expression, so it may be diffrent from Japanese expression or contain wrong expression by translation.
- Each Figure is cited from each paper.

Survey Diary

【2020/05/11】A Unified Framework for Data Poisoning Attack to Graph-based Semi-supervised Learning【NeurIPS2019】

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

keywords : Poisoning, Semi-supervised Learning, G-SSL, trust-region method, bandit

今までそもそもあまり考えて来られなかった, グラフベースの半教師あり学習に対する Poisoing を回帰問題と判別問題を統合する統一的な視点から定式化した論文. 最適化問題としては統一的にはなるが, 実際に解く際には異なるアルゴリズムを用いて解く. 回帰問題の際はグラフの構造を反映した行列 H が出てくるが, これが正定値とは限らない状態で二次計画問題を解く必要が出てくる. そのため信頼領域法を用いてこれを解く. これに関しては収束性と反復計算量のオーダーを導出している. また判別問題ではノイズの加え方をバンディット問題的に定式化することでアルゴリズムを構築している.

This paper formulates Poisoing for graph-based semi-supervised learning, which has not been considered much in the past, from a unified perspective that integrates regression and discriminant problems. Although it is unified as an optimization problem, they use different algorithms for the actual solution. In the regression problem, a matrix H, which reflects the graph structure, emerges, but it is necessary to solve the quadratic programming problem when this

matrix is not necessarily positive definite. Therefore, they use the trust regioon method to solve the quadratic programming problem. The order of convergence and iterative complexity are derived for this problem. For the discriminant problem, they formulate a bandit-like formulation of the noise addition method.

【2020/05/10】Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks【NeurIPS2018】

[Shafahi et al., *NeurIPS*, 2018]

keywords : Poisoning, Neural Networks, base instance, forward-backward-splitting (proximal gradient)

ニューラルネットに対する Poisoning Attack の手法について述べた論文. 状況設定として, 特定の入力 t をうまく学習できないように, 訓練データの中に意図的に変更を加えたようなデータ (つまり毒) を注入するという設定. この時, 他の Poisoning の手法と異なり, 「実際に存在する, ラベルは違うデータ」に寄せるように学習することを考える. 「t に特徴(softmax にかける前のベクトル)が似ている」+「ある他のクラスのデータ b に入力として似ている」ような目的関数を考え, これを最小化するような入力として Poisoning の問題を定式化する. 問題はforward-backward-splitting というアルゴリズムで解く. 数値実験的に手法の良さを検証しており, 「重みの事前学習が行われているセッティングの時は提案手法の Poisoning がうまくいく」が, 「End-to-End で学習する時は提案手法はノイズとしてうまく処理され, そのままで攻撃がうまくいかない」ことを確認しているのが興味深い.

A paper describing a method of poisoning attacks on neural nets, in which a contextual setting is to inject data that is intentionally modified (i.e., poisoned) into the training data so that a particular input t cannot be successfully trained. In a situational setting, they intentionally inject modified data (i.e., poison) into the training data so that a particular input t cannot be successfully trained. In this case, unlike other poisoning methods, they consider that the training data is trained on "real, but differently labeled data". They consider an objective function such as "similar features (vectors before softmax) to t" + "similar as input to some other class of data b", and formulate the problem of poisoning as input that minimizes it. The problem is solved by an algorithm called forward-backward-splitting. They experimentally verify the goodness of the method, and it is interesting that they confirm that the proposed method's poisoning works well in the setting where weights are pre-trained, but that the proposed method is treated well as noise in end-to-end training, and the attack does not work as it is.

$$p = \arg\min_x \|f(x) - f(t)\|_2^2 + \beta\|x - b\|_2^2$$

【2020/05/09】Poisoning Attacks against Support Vector Machines【ICML2012】

[Biggio et al., *ICML*, 2012]

keywords : Poisoning, SVM, KKT condition

訓練データを少しだけ変えてロスを大きく増大させるようなデータを, 訓練時に意図的に紛れ込ませる手法を Poisoning という. 本研究ではSVMに対する Poisioning を提案. SVM の解は KKT 条件を満たすように学習するので, これを保存しつつ 最適解を少しずらす方向に勾配を足しあげる. 手法の良さは数値的に検証しており, 人工データとMNISTで実験している.

Poisoning is a method to intentionally blend data that increases the loss by changing the training data slightly. In this study, they propose poisioning for the SVM. The solution of the SVM is trained to satisfy the KKT condition, so they add a gradient in the direction of shifting the optimal solution slightly while preserving the solution. The merits of the method have been verified numerically, and experiments have been carried out with artificial data and MNIST.

【2020/05/08】Semi-supervised Learning with Deep Generative Models【NeurIPS2014】

[Kingma et al., *NeurIPS*, 2014]

keywords : Semi-supervised Learning, Latent Variable, Variational Inference

半教師あり学習の目的変数として, 元々の判別関数による対数尤度に「潜在変数モデル」と「ラベルも条件に入れた潜在変数モデル」の和を足し上げて同時に最適化する. 最適化は解析解がわかりやすい正規分布やカテゴリカル分布で ELBO を近似し, 変分推論で行う. この単純化によって一般に計算が重くなりがちなベイズ推論であるのにもかかわらず, 近年提唱されている半教師あり学習の計算量と遜色ない速さとなる.

As the objective variable of semi-supervised learning, they add the sum of the latent variable model and the latent variable model with labels as conditions to the log likelihood of the original discriminant function, and optimize it simultaneously. The optimization is performed by approximating ELBO with a normal or categorical distribution that is easy to understand for the analytical solution, and then by variational inference. This simplification makes Bayesian inference, which tends to be computationally heavy in general, no more lower than the computational complexity of semi-supervised learning, which has been proposed in recent years.

(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent variable $\mathbf{z}$

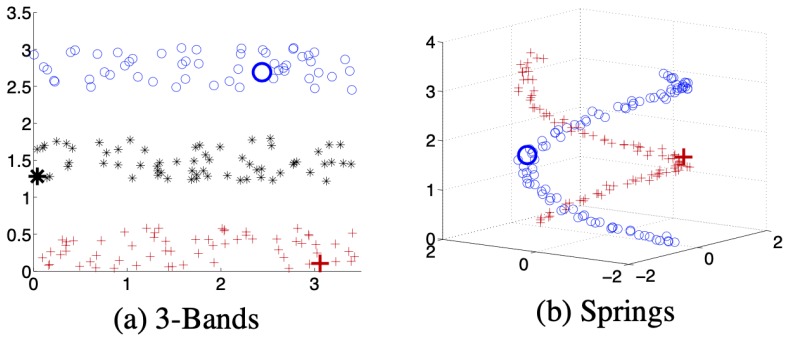
【2020/05/07】Learning from Labeled and Unlabeled Data with Label Propagation【2002】

[Zhu and Ghahramani, 2002]

keywords : Semi-supervised Learning, Label Propagation

「近いデータ点は近いラベルを持つはず」という仮定のもと Label Propagation と呼ばれる手法を用いることで半教師セットに対するラベルづけを行う方法. データ間にグラフを構成するタイプの方法で, 初期化はガウスカーネルで行う. その後, その重み付け行列をノードに関する重みベクトルに作用させることで, マルコフ連鎖のように次の状態を得る. その後, ラベルがついているノードに関しては再び0か1に重みをつけ直すことを行って, このアルゴリズムを繰り返す. そうすることで, ラベルが半教師ありセットに広がっていく.

They establish a method called Label Propagation to label semi-supervised sets under the assumption that "close data points should have close labels". The method is of the type that constructs graphs between the data, and is initialized by the Gaussian kernel. Then, the weighting matrix is applied to the weight vectors about the nodes to obtain the following states as in a Markov chain. Then, for the labeled nodes, the algorithm is reweighted to 0 or 1 again, and the algorithm is repeated. In this way, the labels are extended to the semi-supervised set.



(a) 3-Bands (b) Springs

Figure 1: Label propagation on two synthetic datasets.

【2020/05/06】Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning【TPAMI2018】

[Miyato et al., *TPAMI*, 2018]

keywords : Semi-supervised Learning, VAT, Adversarial Training

Adversarial Training + Semi-Supervised Learning. 通常の Semi-Supervised Learning ではラベルに対して等方的なノイズを乗せることを考えていたが, Adversarial Example の研究からもわかるように, 方向をきちんと定めてあげることが敵対性にとっては重要であった. そこで本研究では半教師データに対し, Adversarial なノイズの方向を計算することで頑健性を増した学習方法を提案. 半教師ありデータ点では, 真のラベルとの乖離を計算することができないのでその段階における学習パラメータで代用するが, その損失の1次近似(勾配)は常に0になってしまう. そこで, 二次近似まで考え, ヘッセ行列の最大固有値を計算. その値を用いて近似的に誤差逆伝播を行う.

Adversarial Training + Semi-Supervised Learning. In conventional semi-supervised learning, isotropic noise is multiplied by the label. In this study, they propose a robust learning method for semi-supervised data by calculating the adversarial direction of the noise, which increases the robustness. For the semi-supervised data points, they cannot compute the deviation from the true label, so they substitute learning parameters at that stage, but the first-order approximation (gradient) of the loss is always zero. Then, they consider the second-order approximation and calculate the maximum eigenvalue of the Hesse matrix. They use this value to approximate the error inverse propagation.

$$r_{\mathrm{adv}} \approx \mathrm{argmax}_r \{ r^\top H(x, \hat{\theta}) r \mid \|r\|_2 \leq \varepsilon \}$$

【2020/05/05】Semi-Supervised Support Vector Machines【NeurIPS1999】

[Bennett and Demiriz, *NeurIPS*, 1999]

keywords : Semi-supervised Learning, SVM, Mixed Integer Programming

半教師あり Support Vector Machine (Semi-Supervised Support Vector Machine, S^3VM)を提案. 目的関数の項に 「1か-1か予測して, そのロスが小さい方を採用した時のロスの和」を教師なしセットに対して足し込んであげることで定式化する. これを解く際には混合整数計画になって, ソルバとしては CPLEX などを使用して解いている.

They propose a semi-supervised Support Vector Machine (S^3VM). It is formulated by adding "the sum of the losses when the smaller loss is adopted, predicted to be 1 or -1" to the term of the objective function for the unsupervised set. This is a mixed integer programming, which is solved by using solvers such as CPLEX.

【2020/05/04】Combining Labeled and Unlabeled Data with Co-Training【COLT1998】

[Blum and Mitchell, *COLT*, 1998]

keywords : Semi-supervised Learning, Co-Training

webページデータのように「ハイパーリンク」と「コンテンツの文字列」と情報が分かれているような時に, それぞれの特徴量から分類モデルを作成し, 同時に訓練する方法を提案. これを “Co-Training” と呼ぶ. 特にここでは, 片方の特徴量に関してしか教師ラベルが手に入らないような状況を考える. するとこれは半教師あり学習のセッティングとなる. 分かれている特徴量間に, 関係性を表す二部グラフを構成し, 辺がつながっているところは同じラベルであるとして学習させると, 教師ラベルが少ない条件下でも比較的安定して学習できることを示した.

They propose a method to make a classification model from each feature and train it at the same time when information is separated from "hyperlinks" and "content strings", such as web page data. They call this method "Co-Training". In particular, they consider a situation in which a teacher label is available only for one of the features. This is the setting for semi-supervised learning. In this study, they show that the semi-supervised learning can be relatively stable even under the condition that there are few supervised labels, where they construct a bipartite graph representing the relationship between the separated features and let them be trained with the same label when they are connected to each other.

【2020/05/03】Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks【ICML2013Workshop】

[Lee., *ICML Workshop*, 2013]

keywords : Semi-supervised Learning, Pseudo-Labeling, Dropout, Denoising AutoEncoder

半教師あり学習の方法として擬似ラベリングを提案. 手元にラベルのないデータがある時, その学習中のモデルで擬似的にラベルのないデータをラベリングし, データセットに追加して学習する. 学習の際の特徴として, Denoising AutoEncoder を使って重みを初期化することと Dropout を用いている点を指摘. また, 損失関数の設計において, 教師あり損失と擬似教師あり損失の重み付けが重要であり, これ

を学習率によって制御する. 特に, 学習初めは普通に教師あり学習として進行し, ステップ数が事前に決めた閾値よりも多くなった時に擬似教師ラベルから来る損失の項の影響を強めていく.

A pseudo-labeling method for semi-supervised learning is proposed. The features of the learning method are that the weights are initialized using Denoising AutoEncoder and that Dropout is used. In the design of the loss function, the weighting of supervised loss and pseudo-supervised loss is important and is controlled by the learning rate. In particular, the beginning of learning proceeds as supervised learning, and when the number of steps exceeds a predetermined threshold, the influence of the loss term from the pseudo-supervised label is strengthened.

【2020/05/02】Realistic Evaluation of Deep Semi-Supervised Learning Algorithms【NeurIPS2018】

[Oliver et al., *NeurIPS*, 2018]

keywords : Semi-supervised Learning, Evaluation, Deep Learning

半教師あり学習の様々な手法が提案されてきているが, それらが本当に改善につながっているのかを精密に実験した研究. 「対抗手法のハイパーパラメータチューニングは適切に行われているか」「見せたい手法が良くなるように不当に計算コストを使っていないか」「そもそも普通の教師あり学習モデルをしっかりと構築し, ベースラインとできているか」「転移学習と比較しているか」「検証データが訓練データより圧倒的に小さくないか」「半教師データサイズを適切に変化させて検証しているか」といった点に十分注意を払って実験を行ったところ, やはり報告されていたほどの数値改善が見られない例が存在した.

Various methods of semi-supervised learning have been proposed, and they have conducted precise experiments to see if they really lead to improvements. The experiments were carried out with careful attention to the following points: whether the hyperparameter tuning of the competing methods is appropriate, whether the computational cost is used unreasonably to improve the desired method, whether the standard supervised learning model is constructed and maintained as a baseline, whether it is compared with transfer learning, whether the verification data is overwhelmingly smaller than the training data, and whether the size of the semi-supervised data is changed appropriately.

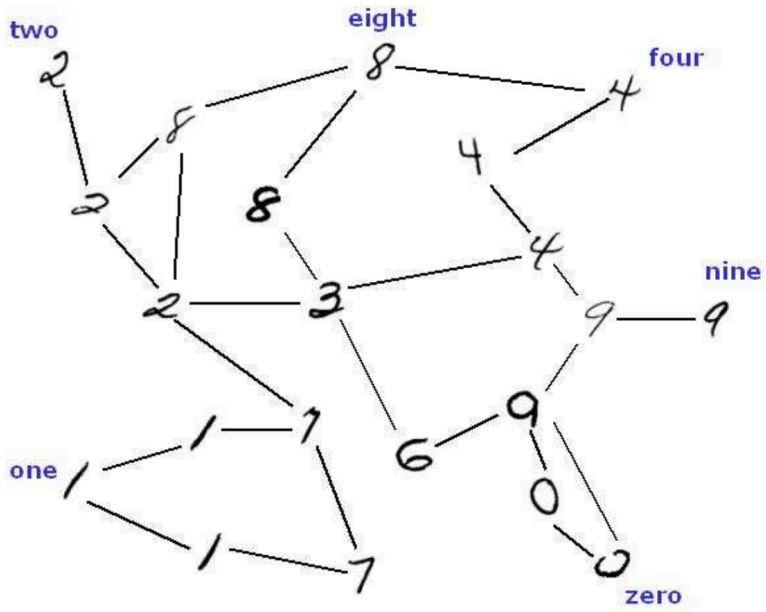
【2020/05/01】Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions【ICML2003】

[Zhu et al., *ICML*, 2003]

keywords : Semi-supervised Learning, Graph Laplacian, Harmonic Function

データの間にグラフを (何らかの方法で) 構成し, 枝にガウスカーネルを用いて重みをつけ, 近傍を重み付けながら足し上げることでエネルギー関数を定義する. このとき, このエネルギー関数を最小化するような, ノードの重みベクトルを得ることとして半教師あり学習を定式化する. 二値分類問題について定式化されており, ラベルを $\{0, 1\}$ と書くと, ラベルが0の点の周りは0になりやすいように学習が進行する. 最適解となる関数は調和関数になることが知られており, グラフと対応づけて考えると, 教師なしデータ上のグラフラプラシアンが0という条件に対応する. したがって, グラフラプラシアンが0となるように重みベクトルに関する線形方程式を解けば良いことになる.

The energy function is defined by constructing a graph (in some way) between the data, weighting as Gaussian kernel the branches and adding up the neighboring nodes. The semi-supervised learning is then formulated by obtaining a weight vector of nodes that minimizes the energy function. For the binary classification problem, if the label is written as $\{0, 1\}$, the learning proceeds in such a way that the label tends to be 0 around the point where the label is 0. It is known that the function with the optimal solution is a harmonic function, which corresponds to the condition that the graph Laplacian on the unsupervised data is 0 when considered as a correspondence to the graph. Therefore, you can solve the linear equations of the weight vectors so that the graph Laplacian is 0.



【2020/04/30】Semi-supervised Learning by Entropy Minimization【NeurIPS2005】

[Grandvalet and Bengio, *NeurIPS*, 2005]

keywords : Semi-supervised Learning, Entropy Regularization, EM-Algorithm

半教師あり学習の新たな目的関数を提案. 尤度関数に, 「エントロピー正則化」と呼ばれる項を足した形で目的関数を表現する (ここでは対数尤度最大化問題として扱う). この正則化項は $-H(Y|X,Z;L_n)$ という形で表され, ラベルなしデータセットの予測が一様分布に近いものになっている時により小さくなることがわかる. つまり, 半教師ありデータセットに自信がない時は目的関数は小さくなってしまう. よりはっきりと分離できるようなデータに重点を置いて学習される.

A new objective function for semi-supervised learning is proposed. The objective function is expressed as a likelihood function plus a term called "entropy regularization" (treated here as a log-likelihood maximization problem). This regularization term is expressed as $-H(Y|X,Z;L_n)$, and you can see that the regularization is smaller when the predictions of the unlabeled dataset are almost uniformly distributed. That is, the objective function is smaller when you are not confident about the semi-supervised dataset. The training focuses on data that are more clearly separable.

Target Function (maximization w.r.t. θ):

$$C(\theta,\lambda)=\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}{z_i}^k f_k(x_i)\right)+\lambda\sum_{i=1}^{n}\sum_{k=1}^{K}g_k(x_i,z_i)\log g_k(x_i,z_i)$$

【2020/04/29】Deep batch active learning by diverse, uncertain gradient lower bounds【ICLR2020】

[Ash, *ICLR*, 2020]

keywords : Active Learning, k-means++, gradient embedding

ニューラルネットワークに対するバッチ型の能動学習の提案. バッチの点の多様性が高く, 情報量が多くなることを目的として次の点を決めたい. 「勾配が大きい点ほど情報量が多い」という視点に立ち, 最終層のみにおける偏微分を行い, かつその勾配に対し, 疑似的にその点におけるラベルを用いてラベルづけを行う. というのも最終層における勾配はそのノルムをとってargmaxを取ると判別そのものになるからである. さらに, その勾配についてクラスタリングを, k-means++ の初期化アルゴリズムを用いて行い, その k 個のクラスタ中心を次のクエリとする. これは従来の k-DPPより計算が高速である.

A proposal of active learning for a neural network with points increasing in each batch. From the point of view that "the larger the gradient, the more information there is," they perform partial differentiation in the final layer only and label the gradient by pseudo-labeling at the point, because the gradient in the final layer is discriminated by taking its norm and taking argmax. In addition, they use the initialization algorithm of k-means++ to determine the next point to query. This algorithm is faster than the conventional k-DPP.

【2020/04/28】Learning Decision Trees using the Fourier Spectrum【SICOMP1993】

[Kushilevitz and Mansour, *SICOMP*, 1993]

keywords : Decision Tree, Discrete Fourier transform, membership query

$\{0,1\}^n$ から $\mathbb{R}$への回帰問題において, 各ノードでの判別が線形和で書かれるような決定木による学習を考える. この時, 決定木を離散フーリエ変換して学習することで多項式時間アルゴリズムを与える. 状況設定としては学習者側は $\{0,1\}^n$ 上の任意の点にアクセスできる membership query での設定である. さらに, 学習する関数はフーリエ係数に 0 が多いという意味でスパースなものを取ってくることができる.

In a regression problem from $\{0,1\}^n$ to a real number, we consider training with a decision tree in which the discriminations at each node are written as a linear sum. In this case, they give a polynomial time algorithm by learning the decision tree by a discrete Fourier transform. The situation is that the learner can access any point on $\{0,1\}^n$ in the membership query. Furthermore, the learned function can be sparse in the sense that the Fourier coefficient has many zeros.

【2020/04/27】Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs【ICML2015】

[Gal and Turner, *ICML*, 2015]

keywords : Random Feature, Gaussian Process, Variational Inference

[Lázaro-Gredilla et al., *JMLR*, 2010] ではガウス過程回帰モデルのカーネル関数の近似として Random Feature を用いる方法が提案されており, SSGP と名前が付けられていたが, SSGP は過学習する傾向が見られていた. そこで本研究では, Random Feature も入れた上で, Random Feature そのものの分布と, それに付随する Fourier 係数の最適化を変分推論で行うことでより良い学習方法を提案. 最終的に良さは数値実験内の RMSE で測っている. 途中の変分推論を導出する際の変形が「ベイズあるある」な, いかつい計算になっており, 読むのが些か大変.

In [Lázaro-Gredilla et al., *JMLR*, 2010], a method to use a random feature as an approximation to the kernel function of a Gaussian process regression model was proposed, which was named SSGP, but SSGP tends to overlearn. However, SSGP tends to overfit. In this study, they propose a better learning method by using variational inference to optimize the distribution of the random

feature itself and its associated fourier coefficients, while including the random feature. In the end, they use RMSE in their numerical experiments to measure the quality of the learning. The variation of the variational inference is a bit difficult to read because of the Bayesian variational nature of the calculations.

【2020/04/26】A Representer Theorem for Deep Kernel Learning【JMLR2019】

[Bohn et al., *JMLR*, 2019]

keywords : Representer Theorem, Deep kernel, infinite case

RKHS上の関数に対する回帰問題を考える際に, 正則化項つき経験誤差最小化を考えるとその解がRKHS上の元の線形結合で書けるという定理を Representer Theorem (表現定理) という. 本研究ではDeep Kernel, すなわちカーネル関数の合成で書けるようなカーネルに対応するRKHS中の回帰問題の表現定理を扱っており, 実際これは Deep Kernel の場合でも成立する. 定理は経験誤差が有限個のサンプルで表現されている場合と, 無限個, つまり期待誤差の最小化の場合に分けて論じられている. 前者は通常の表現定理と同様の表現が得られ, 後者は Bochner 型の積分表示が得られる.

Representer Theorem (Representation Theorem) is a theorem that the solution can be written by the original linear combination on RKHS if we consider empirical error minimization with regularization terms when considering the regression problem for a function on RKHS. This research deals with the representation theorem of regression problems in RKHS corresponding to Deep Kernel, i.e., a kernel that can be written by synthesizing kernel functions, and in fact, this theorem holds in the case of Deep Kernel. The theorem is discussed in two cases: the case where the empirical error is represented by a finite number of samples, and the case of an infinite number of samples, that is, the minimization of the expectation error. The former case gives a representation similar to that of the ordinary representation theorem, while the latter case gives a Bochner-type integral representation.

【2020/04/25】Sparse Spectrum Gaussian Process Regression【JMLR2010】

[Lázaro-Gredilla et al., *JMLR*, 2010]

keywords : Gaussian Process, Random Feature, SSGP

ガウス過程による回帰問題を解く際に, カーネルを Random Feature によって近似し, 特徴量の意味で Sparse な回帰を実現するアルゴリズムを提案. 面白いポイントとしては, ガウス過程の「学習」はカーネルのハイパーパラメータについて周辺尤度を大きくするようにチューニングすることとして位置付けられるが, Random Feature を考えるとこの 「Random Feature の位置」についても勾配法で最適化している点が挙げられる. 数値実験的に良さを検証しており, スパース性において類似する Fully Independent Training Conditional (FITC) ガウス過程回帰モデル と比較し, Normalized Mean Squared Error の意味で良いことを述べている. Random Feature の数を適当に減らすと過学習を防いでいるような結果も散見され, 非常に興味深い. Random Feature の数と正則化の間にはどんな関係があるのか調べられているのだろうか？

When solving a regression problem with a Gaussian process, they propose an algorithm that approximates the kernel with a random feature and realizes a sparse regression in the sense of features. An interesting point is that the "learning" of the Gaussian process is positioned as tuning the hyperparameters of the kernel to increase the marginal likelihood, but considering Random Feature, the "position of Random Feature" is also optimized using the gradient method. They have experimentally verified the goodness of this method, and compared it with the Fully Independent Training Conditional (FITC) Gaussian process regression model, which is similar in sparsity, they have stated that it is better in the sense of normalized mean squared error. It is very interesting to note that reducing the number of random features prevent overfitting. What is the relationship between the number of random features and the regularization?

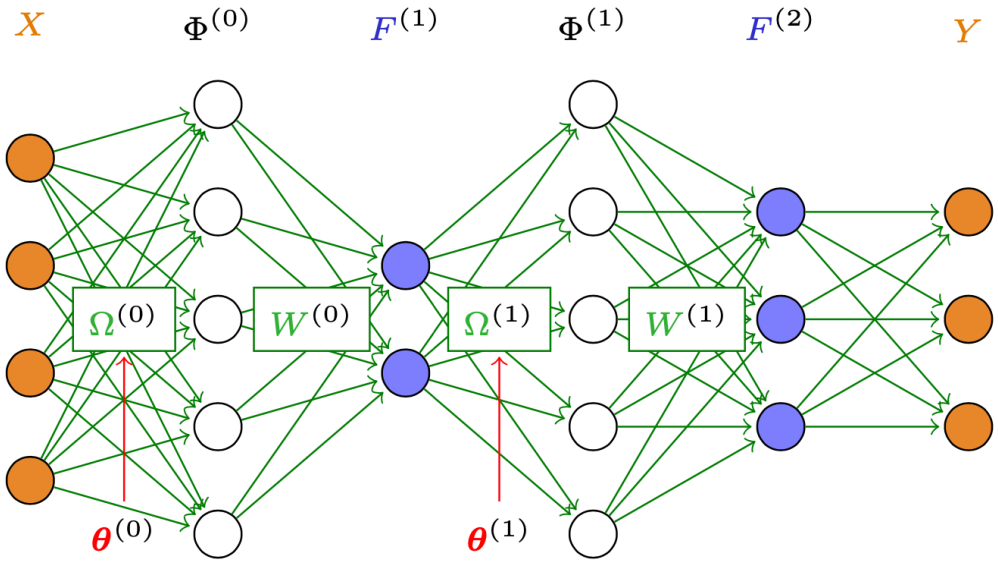
【2020/04/24】Random Feature Expansions for Deep Gaussian Processes【ICML2017】

[Cutajar et al., *ICML*, 2017]

keywords : Deep Gaussian Process, Random Feature, Arccosine kernel

深層ガウス過程のカーネルの設計に Random Feature を用いた研究. 対象は回帰問題. Random Feature を用いることでガウス過程の学習・推論の際に計算を高速化することができる上に, 実験的には良い性能 (RMSEが小さい) が出ていることも確認している. ポイントとしては実際に学習ができるように Random Feature + MCMC を用いて勾配を計算しており, プラクティカルな面での貢献が大きい.

A study on the design of kernels of deep Gaussian processes using random features. The target of the study is a regression problem. They have found that the use of random features not only speeds up the learning and inference of Gaussian processes, but also shows good experimental performance (small RMSE). The key point is that they use Random Feature + MCMC to compute the slope so that they can actually learn them, which contributes to the practical aspects of the simulation.



【2020/04/23】Neural Tangent Kernel: Convergence and Generalization in Neural Networks【NeurIPS2018】

[Jacot et al., *NeurIPS*, 2018]

keywords : Neural Tangent Kernel, Initialization, Gaussian Process, Kernel Gradient

ニューラルネットワークの重みパラメータの初期化を $1/\sqrt{n_l}$ の正規分布に従うように取る. この時, ニューロン数を無限大に飛ばすとニューラルネットワークはガウス過程と見なすことができる. ニューラルネットの最適化を考えていくとこれは連続時間 t に依存するパラメータ $\theta(t)$ の発展と共にあるカーネル関数が変わっていくものとしてみることができる. このカーネルを Neural Tangent Kernel と呼ぶ. またカーネルの変化, つまりカーネル勾配を計算することができ, その勾配に関連する線形微分方程式を解くことでニューラルネットの関数としての収束性の議論を行うことができる.

They take the initialization of the weight parameters of the neural network to follow a normal distribution of $1/\sqrt{n_l}$. In this case, the neural network is considered to be a Gaussian process if the number of neurons is skipped to infinity. If they consider the optimization of the neural network, they can assume that the kernel function changes with the development of the parameter $\theta(t)$, which depends on the continuous time t. The kernel is called the Neural Tangent Kernel. They can also compute the kernel change, i.e., the kernel gradient, and by solving the linear differential equations related to the gradient, they can discuss the convergence as a function of a neural net.

NTK :

$\Theta^{(L)}(\theta) = \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta) \otimes \partial_{\theta_p} F^{(L)}(\theta)$

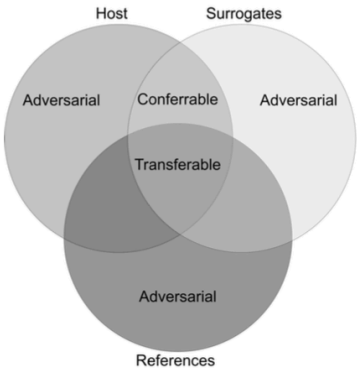
【2020/04/22】Deep Neural Network Fingerprinting by Conferrable Adversarial Examples【2019】

[Lukas et al., 2019]

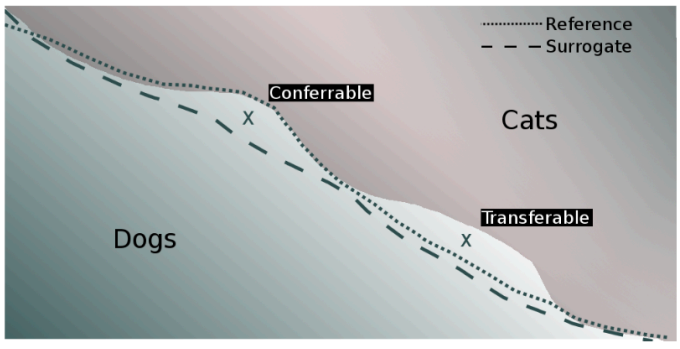
keywords : Model Extraction, Conferrable adversarial examples, Fingerprinting

モデルをデプロイしたときに敵対者がそのモデルをコピーすることを考える. このとき, 元のモデルを作った側 (Defender) がその敵対者のモデルが盗まれたもの (Surrogate Model) なのか関係ないもの (Reference Model) なのかを判別する方法を提案. 具体的には, Defender 側がモデルを訓練する際に, 自分で仮想的に敵を設定して Surrogate Model と Reference Model を作成する. ここでこれらを見分けるために Adversarial Example を用いる. というのも Adversarial Example には Transferability があることが報告されているので, 元のモデルに対して Adv-Ex を作ると, Surrogate Model も Reference Model も騙すことができる. しかしここで, Surrogate Model は騙せるが, 関係ない Reference Model は騙せない入力が作れる. これを "Conferrable Adversarial Example" と呼び, これを指紋として使う. つまり, 元のモデルと似たようなモデルだと同様に間違ってしまう入力を鍵のように使う.

They suppose that an adversary copies a model when you deploy it. In this case, they propose a method to determine whether the model of the adversary's adversary is a stolen model (Surrogate Model) or an irrelevant model (Reference Model) by the side that created the original model (Defender). Specifically, when the Defender trains a model, it creates a Surrogate Model and a Reference Model by setting its own virtual enemies. Here, they use Adversarial Example to distinguish them. Since Adversarial Example has been reported to have transferability, they can deceive both Surrogate and Reference models by creating an Adv-Ex for the original model. Here, however, they can create inputs that can be fooled by the Surrogate Model, but not by the unrelated Reference Model. They call this "Conferrable Adversarial Example" and use it as a fingerprint. In other words, you can use the input as a key that would be wrong if it were a similar model to the original one.



(a)



(b)

【2020/04/21】Random Feature Maps for Dot Product Kernels【AISTATS2012】

[Kar and Karnick, *AISTATS*, 2012]

keywords : Random Feature, Rademacher variable, Maclaurin expansion

内積で表現されるカーネルに対するRandom Featureの構成とその近似性能について理論的に評価した研究. [Rahimi and Recht, *NeurIPS*, 2007] では平行移動不変なカーネルに対する理論保証は与えていたが, 内積で表現されるカーネルに対する保証は与えられていなかった. ランダムな基底の作り方としては特徴写像 Φ をマクローリン展開してあげて, (この時係数は非負にできる) どの特徴の次元を使うか Rademacher 変数を単に用いて決定してあげれば良い. なお, この時に真のカーネルと近似した内積表現の $\sup$ の差が ϵ 以上になる確率は $O((1/\varepsilon)^{2d}\exp\{-D\varepsilon^2\})$ で与えられる.

This paper presents a theoretical evaluation of the composition of random features and their approximate performance for kernels represented by the inner product. In [Rahimi and Recht, *NeurIPS*, 2007], theoretical guarantees are given for kernels that are translational invariant, but not for kernels that are represented by inner products. To make a random basis, you can expand the feature map Φ by Maclaurin expansion (at this time, the coefficients can be non-negative) and simply use the Rademacher variable to determine which feature dimension to use. Note that the probability that the difference of $\sup$ between the true kernel and the approximate inner product representation is greater than or equal to ϵ is given by $O((1/\varepsilon)^{2d}\exp\{-D\varepsilon^2\})$.

【2020/04/20】Optimal Rates for the Regularized Least-Squares Algorithm【FoCM2007】

[Caponnetto and Vito, *FoCM*, 2007]

keywords : Regularization, Hilbert space, decay of eigenvalues, kernel method

ヒルベルト空間上の正則化つき二乗誤差回帰問題の経験誤差最小解と $\inf_{f\in\mathcal{H}} f$ との差の上界と下界の min-max レートなどを導出. このとき, 元の分布 ρ とカーネル K から定まる作用素の固有値の減衰レートを用いてバウンドしているのが特徴的. 一般的な Rademacher Complexity によるバウンドでは高々 $O(1/\sqrt{n})$ だったが, 精密に固有値まで見ると $O(1/n)$ にできる.

The min-max rates of the upper and lower bounds of the difference between the empirical error minimum solution of the regularized squared error regression problem on Hilbert space and $\inf_{f\in\mathcal{H}} f$ are derived. The important point is that they uses the decay rate of the eigenvalues of the operators determined from the original distribution ρ and the kernel K. The bounds using the general Rademacher Complexity are as high as $O(1/\sqrt{n})$, but thinking of the eigenvalues, its rate are as high as $O(1/n)$.

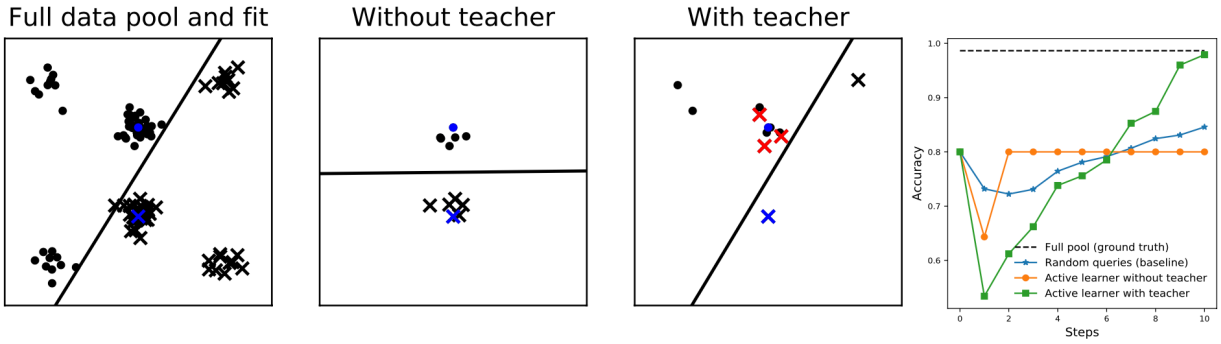
【2020/04/19】Machine Teaching of Active Sequential Learners【NeurIPS2019】

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

keywords : Teaching, MDP, multi-arm bandit, sequential

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.

They consider a pool-based situation for learning in which there are learners and teachers. The learner then asks which of the points $x_1, \ldots, x_K$ should be inputted as a multi-arm bandit problem. The learner's reward is $E[\sum_{t-1}^{T} y_t]$ $(y_t \in \{0, 1\})$, and the teacher should search for things that make y_t 1 as much as possible. The teacher, on the other hand, tries to figure out how to return y_t when x_t comes in so that the learners can converge to the true parameter θ^* as fast as possible, and formulates this as the Markov Decision Process (MDP). The immediate reward in this case is $R_t(x_1, y_1, \ldots, y_{t-1}, y_{t-1}, y_{t-1}, x_t; \theta^*) = x_t^\top \theta^*$, and the teacher is the agent that maximizes the value function $E^\pi[\sum_{t=1}^{T} \gamma^{t-1} R_t]$. It is experimentally confirmed that the teacher's information can converge to the solution faster than other active learning methods such as uncertainty sampling.



【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 $O(\sqrt{(n \log n)})$ 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【S&P2017】

[Shokri et al., *S&P*, 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【2018】

[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【ACSAC2018】

[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.

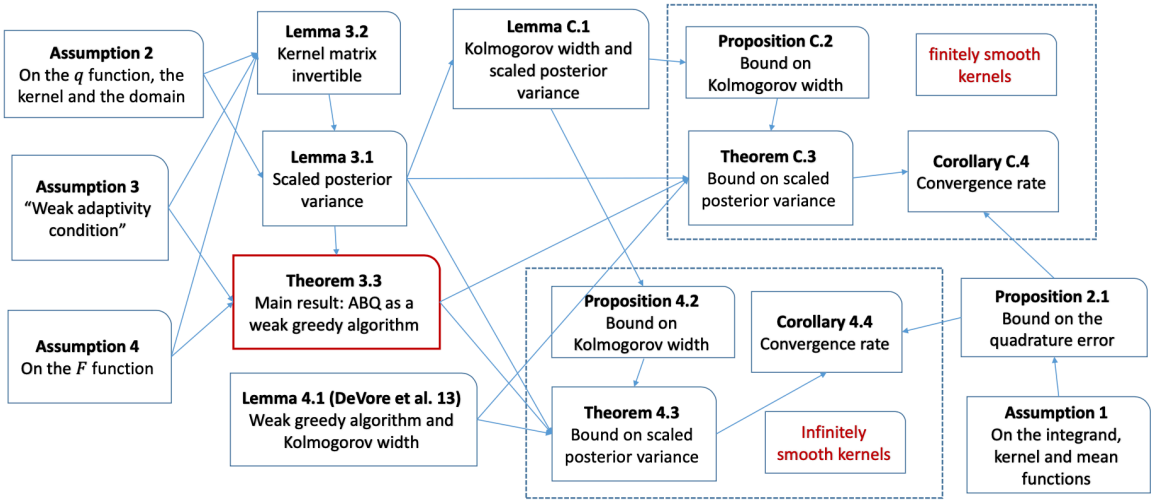
【2020/04/13】Convergence Guarantees for Adaptive Bayesian Quadrature Methods【NeurIPS2019】

[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 $O(\exp\{-Dn^{1/d}\})$ 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 $x_1 \dots x_n$, 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 $O(\exp\{-Dn^{1/d}\})$ 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【ICML2013】

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

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

[Rahimi and Recht, NeurIPS, 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【AAAI2020】

[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.

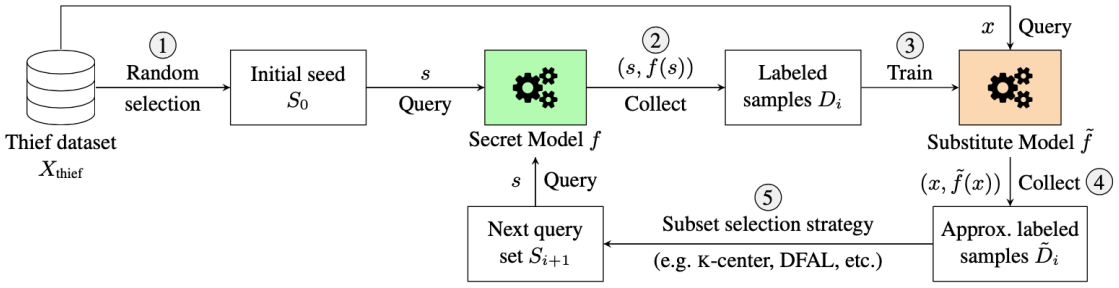


Figure 2: The ACTIVETHIEF framework for model extraction (see corresponding section for explanation of 1-5).

【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.

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

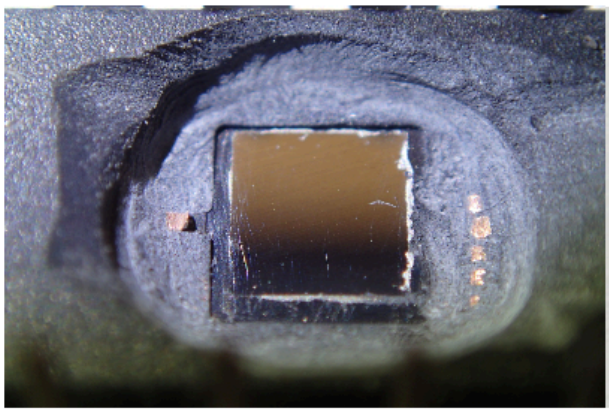
【2020/04/08】CSI Neural Network: Using Side-channels to Recover Your Artificial Neural Network Information【Security2019】

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

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

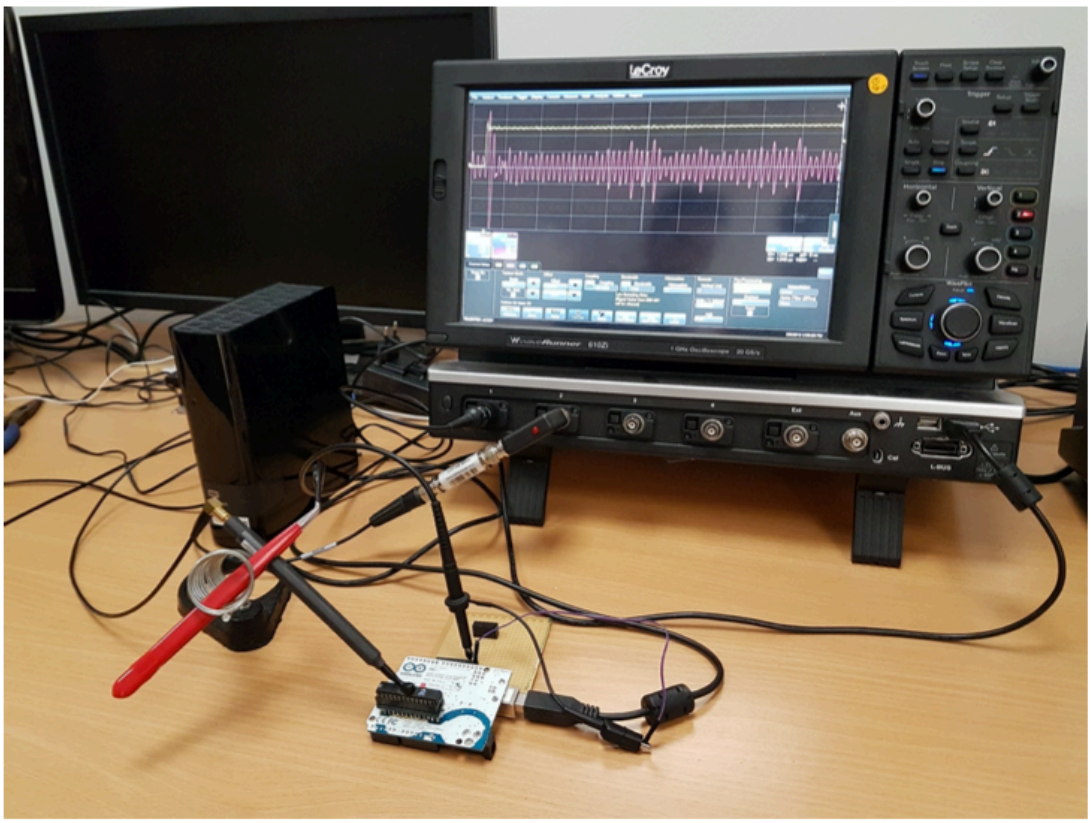
外部から直接ハードウェアとしてGPU・CPUに触り, 電磁気的な周波数を読み取ることで, (アーキテクチャの情報はわかった上で) 内部のモデルの活性化関数・層の数・ニューロン数などにまつわる情報を抜き出す方法を提案. これは代表的なReverse-Engineeringの手法であるSPAやDPA といった手法(元々はRSA暗号などのスキミングに使われていた)を用いている. また, 他の攻撃としてモデルが既知で, インプットデータが未知な時にそのデータをスキミングしてHPAという手法を用いて復元することも提案している. 理論屋の頭のどこにもない攻撃の仕方でとても興味深い.

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

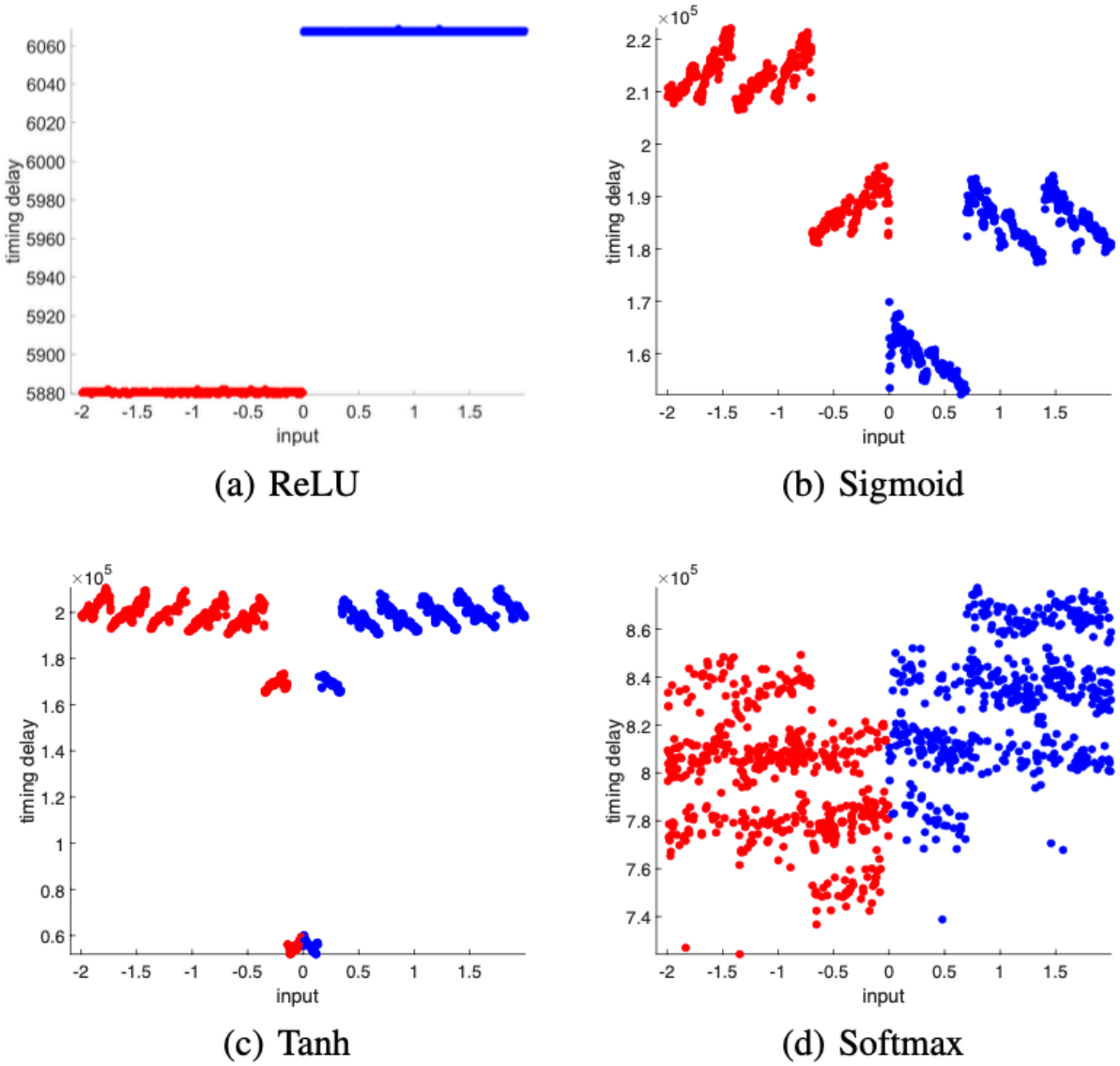


Fig. 5: Timing behavior for different activation functions

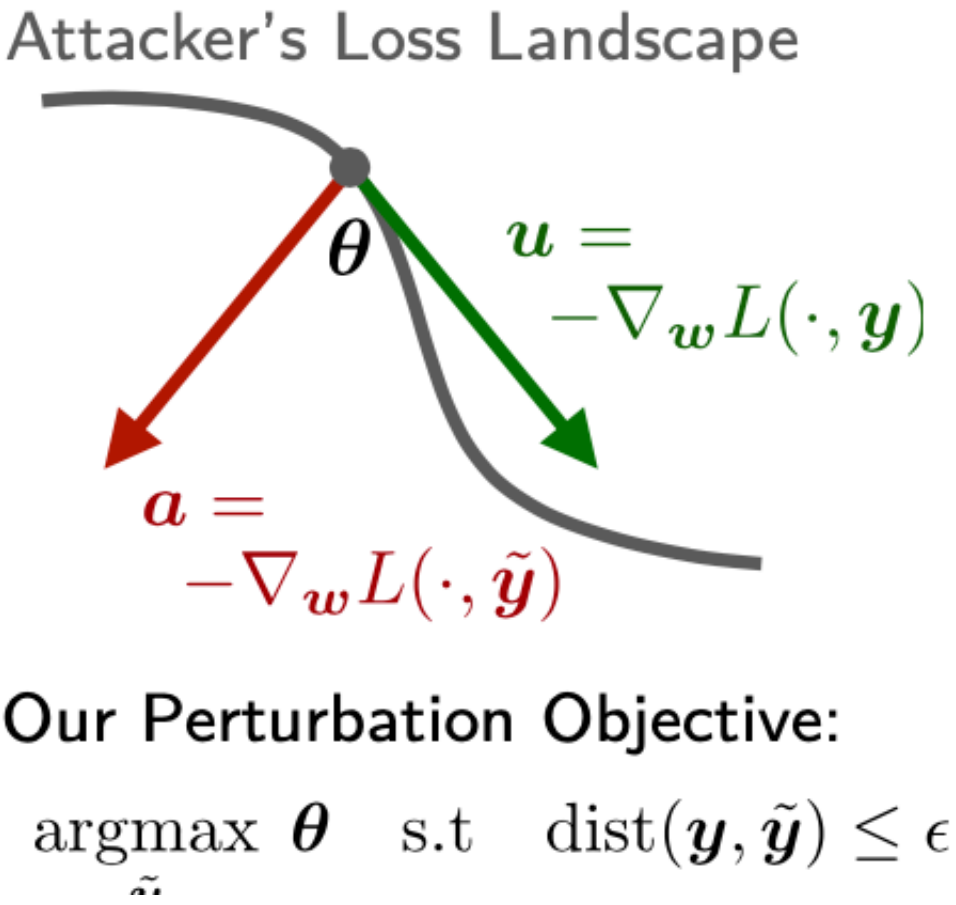
【2020/04/07】Prediction poisoning: Towards defenses against DNN model stealing attacks【ICLR2020】

[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.



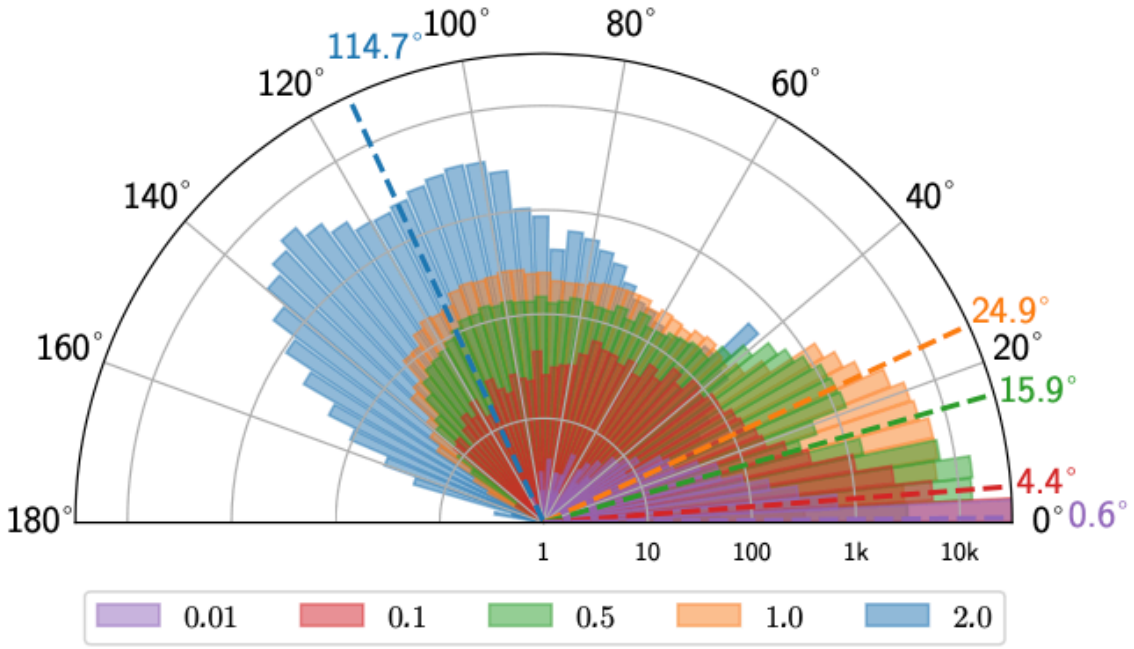


Figure 6: Histogram of Angular Deviations. Presented for MAD attack on CIFAR10 with various choices of ϵ.

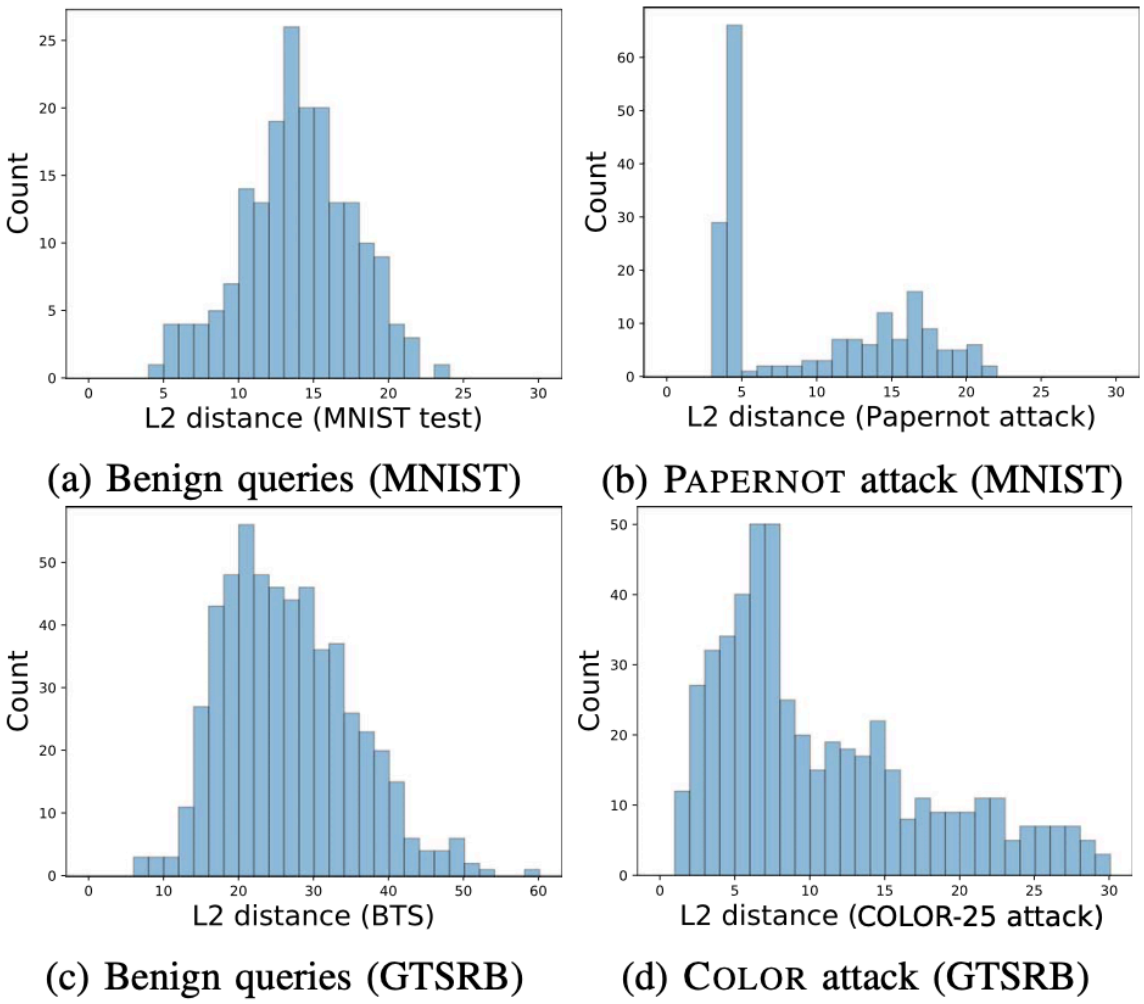
【2020/04/06】PRADA: Protecting Against DNN Model Stealing Attacks【EuroS&P2019】

[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

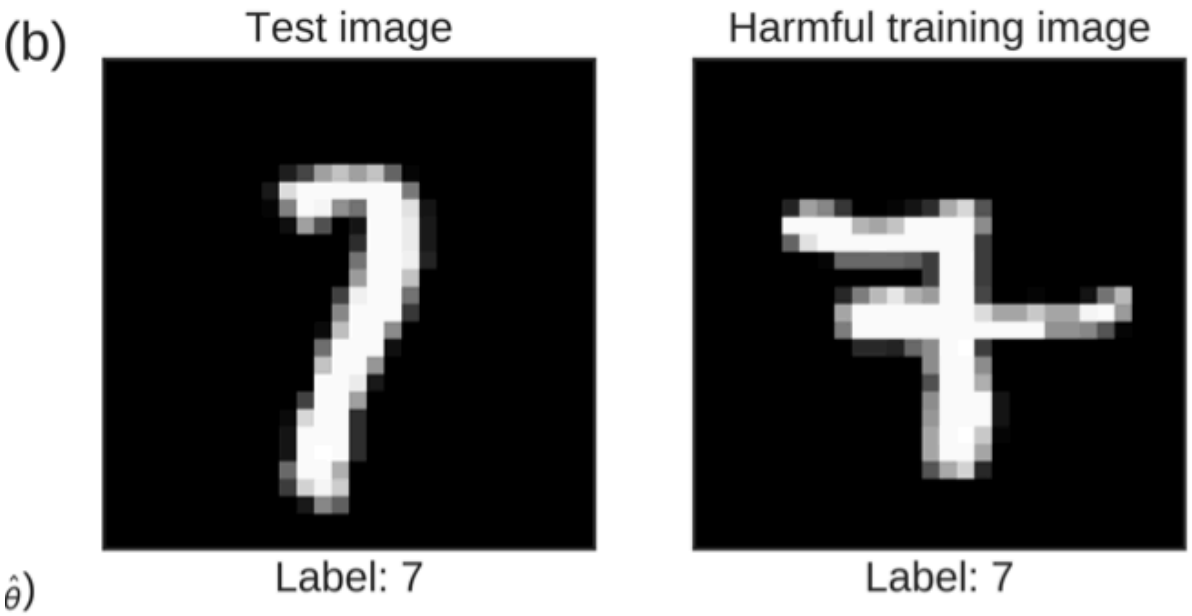
【2020/04/02】Understanding Black-box Predictions via Influence Functions【ICML2017】

[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】

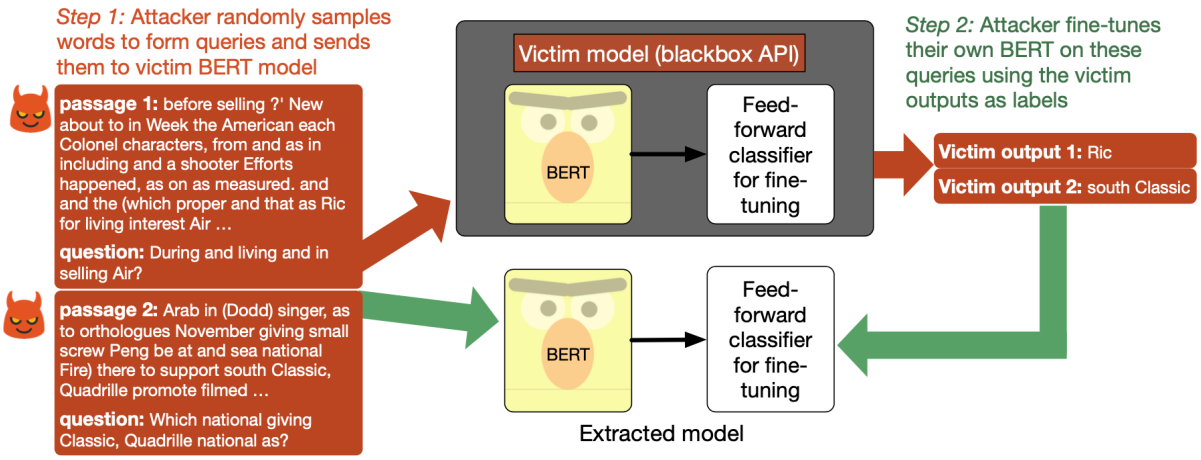
[Krishna et al., *ICLR*, 2020]

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.



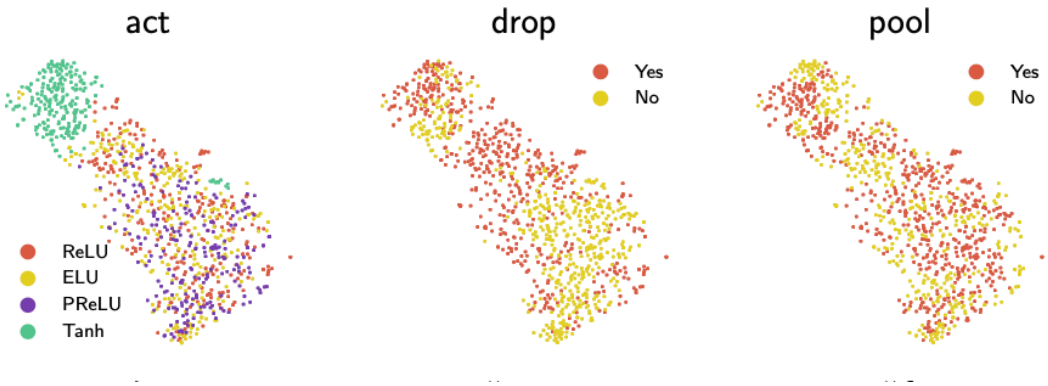
【2020/03/31】Towards reverse-engineering black-box neural networks【ICLR2018】

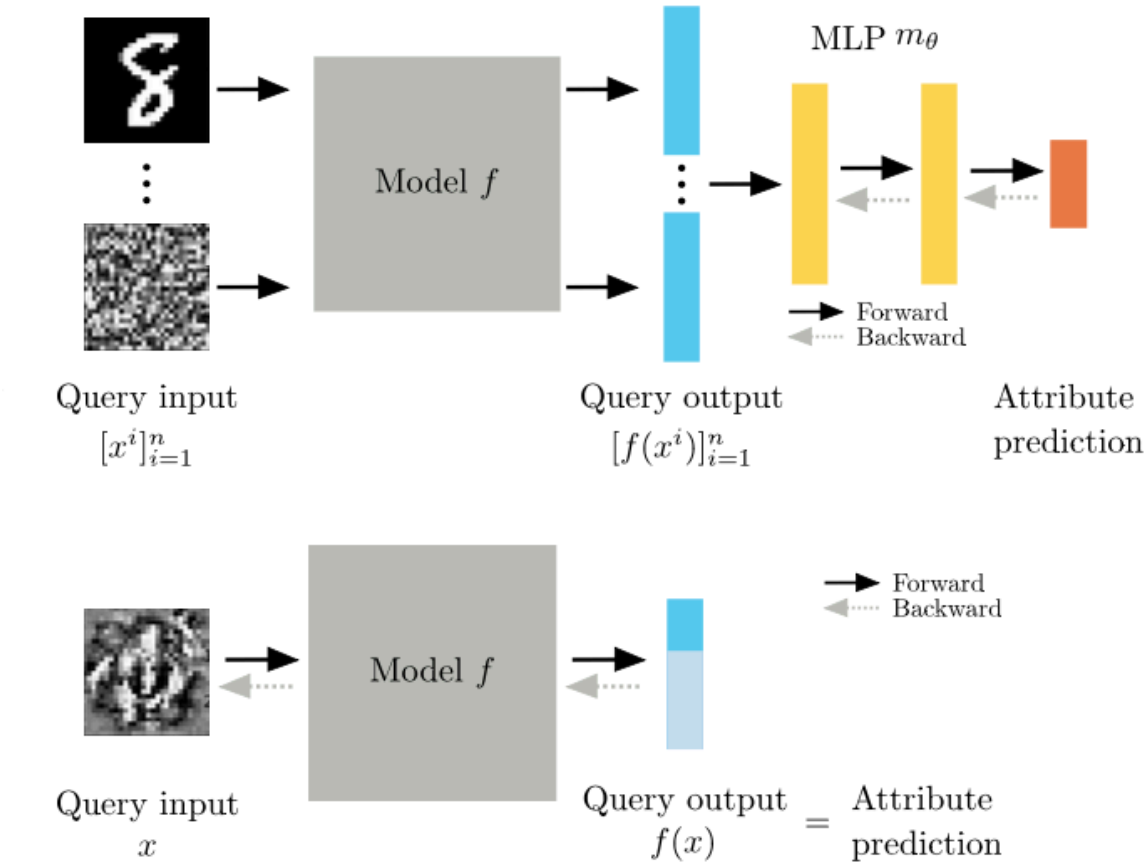
[Oh et al., *ICLR*, 2018]

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

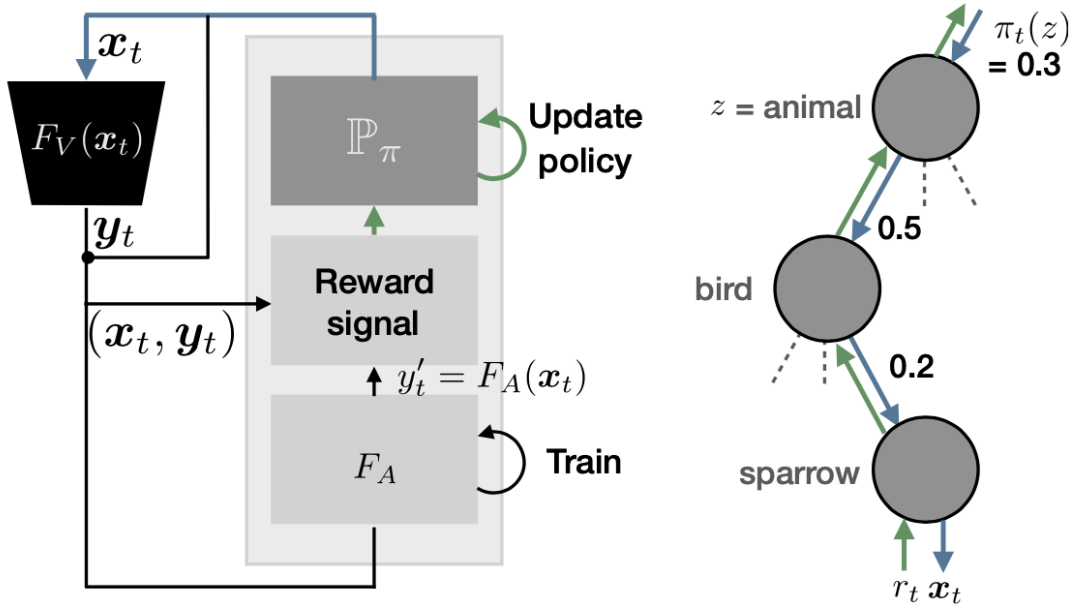
【2020/03/29】Knockoff Nets: Stealing Functionality of Black-Box Models【CVPR2019】

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

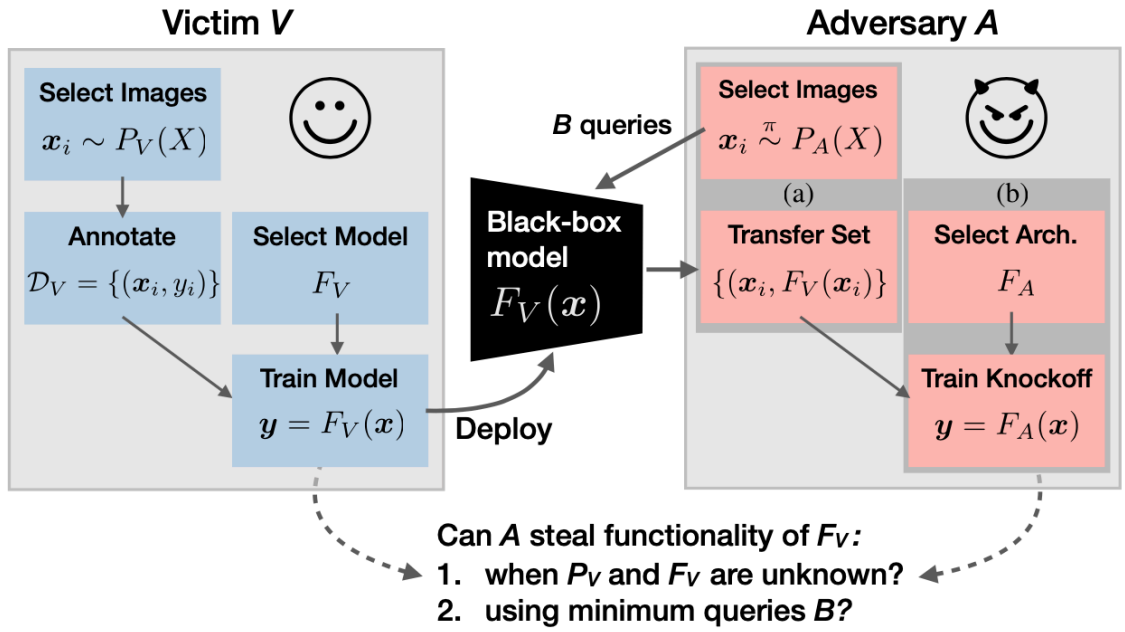
keywords : Model Extraction, copy, knockoff, distillation, reinforcement learning

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.**

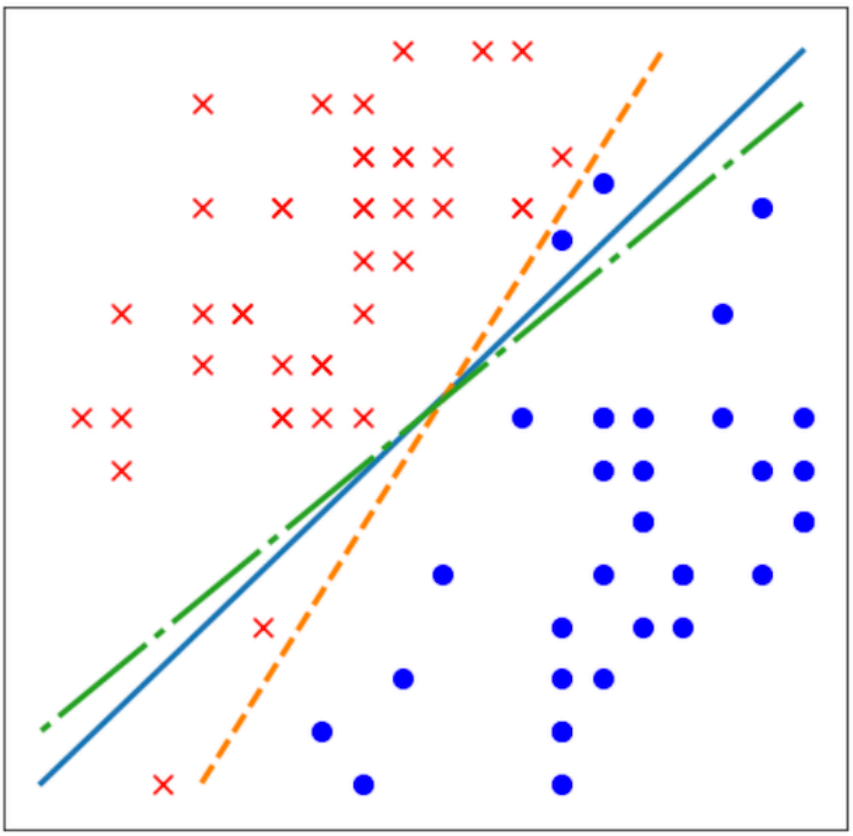


Figure 1: Illustrating fidelity vs. accuracy. The solid blue line is the oracle; functionally equivalent extraction recovers this exactly. The green dash-dot line achieves high fidelity: it matches the oracle on all data points. The orange dashed line achieves perfect accuracy: it classifies all points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

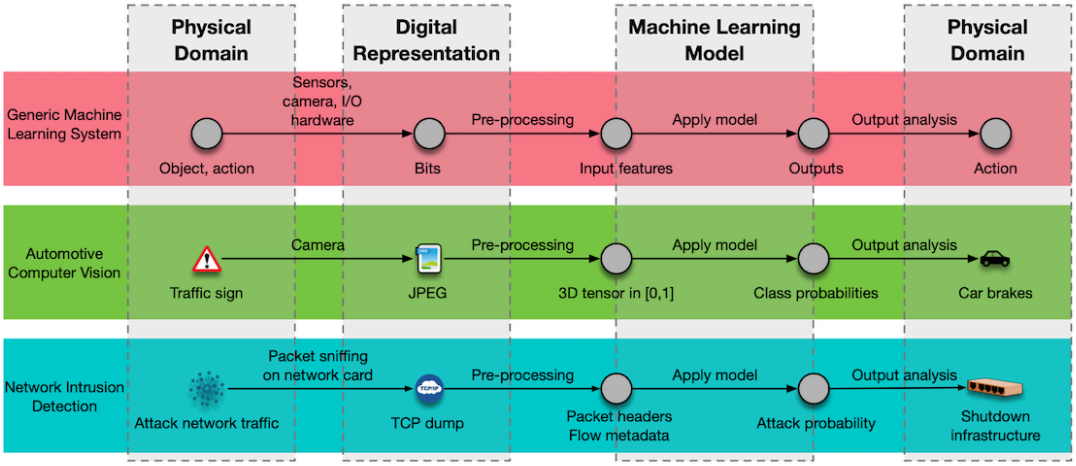
【2020/03/25】Towards the Science of Security and Privacy in Machine Learning【2016】

[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. 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 networks. *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 machine 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.

[18] Ibrahim M Alabdulmohsin, Xin Gao, and Xiangliang Zhang. Adding robustness to support vector machines against adversarial reverse engineering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 231–240, 2014.

[19] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade,and Vinod Ganapathy. ACTIVETHIEF: Model extraction using active learning and unannotated public data. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[20] Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood-approximating kernelexpansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning*, volume 85, 2013.

[21] Motonobu Kanagawa and Philipp Hennig. Convergence guarantees for adaptive bayesian quadrature methods. In *Advances in Neural Information Processing Systems*, pages 6234–6245, 2019.

[22] Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. Model extraction warning in mlaas paradigm. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 371–380, 2018.

[23] Taesung Lee, Benjamin Edwards, Ian Molloy, and Dong Su. Defending against machine learning model stealing attacks using deceptive perturbations. *arXiv preprint arXiv:1806.00054*, 2018.

[24] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 3–18, 2017.

[25] Alina Beygelzimer, Daniel J Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing systems*, pages 199–207, 2010.

[26] Shingo Yashima, Atsushi Nitanda, and Taiji Suzuki. Exponential convergence rates of classification errors on learning with sgd and random features. *arXiv preprint arXiv:1911.05350*, 2019.

[27] Tomi Peltola, Mustafa Mert Celikok, Pedram Daee, and Samuel Kaski. Machine teaching of active sequential learners. In *Advances in Neural Information Processing Systems*, pages 11202–11213, 2019.

[28] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[29] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Artificial Intelligence and Statistics*, pages 583–591, 2012.

[30] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *arXiv preprint arXiv:1912.00888*, 2019.

[31] Arthur Jacot, Franck Gabriel, and Cl´ement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[32] Kurt Cutajar, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep gaussian processes. In *Proceedings of the 34th International Conference on Machine Learning*, pages 884–893, 2017.

[33] Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and An´ıbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11(63):1865–1881, 2010.

[34] Bastian Bohn, Christian Rieger, and Michael Griebel. A representer theorem for deep kernel learning. *Journal of Machine Learning Research*, 20(64):1–32, 2019.

[35] Yarin Gal and Richard Turner. Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 655–664, 2015.

[36] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[37] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

[38] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2005.

[39] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine learning*, pages 912–919, 2003.

[40] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[41] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, volume 3, page 2, 2013.

[42] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[43] Kristin P Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, pages 368–374, 1999.

[44] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semisupervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

[45] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

[46] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[47] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1467–1474, 2012.

[48] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted cleanlabel poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.

[49] Xuanqing Liu, Si Si, Jerry Zhu, Yang Li, and Cho-Jui Hsieh. A unified framework for data poisoning attack to graph-based semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 9780– 9790, 2019.