UC San Diego

On the Equivalence between Neural Network and Support Vector Machine

Yilan Chen[1], Wei Huang[2], Lam M. Nguyen[3], Lily Weng[1]

[1]UCSD, [2]UTS, [3]IBM Research

yilan@ucsd.edu, lweng@ucsd.edu

NeurIPS, December 2021

Table of Contents

Table of Contents

Introduction: What is NTK

- Neural Tangent Kernel (NTK) [Jacot et al., 2018]:

$$\hat{\Theta}(w; x, x') = \langle \nabla_w f(w, x), \nabla_w f(w, x') \rangle$$

- Under certain conditions (usually infinite width limit and NTK parameterization), the tangent kernel at initialization converges in probability to a deterministic limit and keeps constant during training:

$$\hat{\Theta}(w; x, x') \to \Theta_\infty(x, x')$$

- Infinite-width NN trained by gradient descent with mean square loss $\Leftrightarrow$ kernel regression with NTK [Jacot et al., 2018; Arora et al., 2019]

Introduction: What is NTK

- Wide neural networks are linear [Lee et al., 2019]:

$$f(w_t, x) = f(w_0, x) + \langle \nabla_w f(w_0, x), w_t - w_0 \rangle + O(m^{-\frac{1}{2}})$$

  where m is the width of NN.
- Constant tangent kernel $\Leftrightarrow$ Linear model. Small Hessian norm $\Rightarrow$ small change of tangent kernel [Liu et al., 2020a].

Introduction: Motivation & related works

NTK helps us understand the optimization and generalization of NN through the perspective of kernel methods. However,

- The equivalence is only known for ridge regression (regression model). Limited insights to understand classification problems.
- Existing theory cannot handle the case of regularization.

Table of Contents

Main results

Our contributions:

1. Equivalence between NN and SVM
2. Equivalence between NN and a family of ℓ_2 regularized KMs
3. Finite-width NN trained by ℓ_2 regularized loss is approximately a kernel machine (KM)
4. Applications
   - 4.1 Computing non-vacuous generalization bound of NN via the corresponding KM
   - 4.2 Robustness certificate for over-parameterized NN
   - 4.3 ℓ_2 regularized KMs (from equivalent infinite-width NN) are more robust than previous kernel regression

1. Equivalence between NN and SVM

Definition [Soft Margin SVM]

Given labeled samples $\{(x_i, y_i)\}_{i=1}^n$ with $y_i \in \{-1, +1\}$, the hyperplane β^* that solves the below optimization problem realizes the soft margin classifier with geometric margin $\gamma = 2/\|\beta^*\|$.

$$\min_{\beta,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \xi_i, \quad s.t.\ y_i\langle\beta, \Phi(x_i)\rangle \geq 1 - \xi_i,\ \xi_i \geq 0,\ i \in [n],$$

Equivalently,

$$\min_{\beta} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \max(0, 1 - y_i\langle\beta, \Phi(x_i)\rangle).$$

Denote as $L(\beta)$, which is strongly convex in β. This can be solved by subgradient decent.

1. Equivalence between NN and SVM

Definition [Soft Margin Neural Network]

Given samples $\{(x_i, y_i)\}_{i=1}^n$, $y_i \in \{-1, +1\}$, the neural network w^* that solves the following two equivalent optimization problems

$$\min_{w,\xi} \frac{1}{2}\|W^{(L+1)}\|^2 + C\sum_{i=1}^{n}\xi_i, \quad s.t.\ y_i f(w, x_i) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i \in [n],$$

$$\min_{w} \frac{1}{2}\|W^{(L+1)}\|^2 + C\sum_{i=1}^{n}\max(0, 1 - y_i f(w, x_i)), \tag{1}$$

realizes the soft margin classifier with geometric margin $\gamma = 2/\|W_*^{(L+1)}\|$. Denote Eq. (1) as $L(w)$ and call it *soft margin loss*.

Theorem [Continuous Dynamics and Convergence Rate of SVM]

Consider training soft margin SVM by subgradient descent with infinite small learning rate (gradient flow): $\frac{d\beta_t}{dt} = -\nabla_\beta L(\beta_t)$, the model $g_t(x)$ follows the below evolution:

$$\frac{dg_t(x)}{dt} = -g_t(x) + C\sum_{i=1}^{n} \mathbb{1}(y_i g_t(x_i) < 1) y_i K(x, x_i),$$

and has a linear convergence rate:

$$L(\beta_t) - L(\beta^*) \leq e^{-2t} \left(L(\beta_0) - L(\beta^*)\right).$$

1. Equivalence between NN and SVM

Theorem [Continuous Dynamics and Convergence Rate of NN]

Suppose an NN $f(w, x)$, with f a differentiable function of w, is learned from a training set $\{(x_i, y_i)\}_{i=1}^n$ by subgradient descent with $L(w)$ and gradient flow. Then the network has the following dynamics:

$$\frac{df_t(x)}{dt} = -f_t(x) + C\sum_{i=1}^{n} \mathbb{1}(y_i f_t(x_i) < 1) y_i \hat{\Theta}(w_t; x, x_i).$$

Let $\hat{\Theta}(w_t) \in \mathbb{R}^{n\times n}$ be the tangent kernel evaluated on the training set and $\lambda_{min}\left(\hat{\Theta}(w_t)\right)$ be its minimum eigenvalue. Assume $\lambda_{min}\left(\hat{\Theta}(w_t)\right) \geq \frac{2}{C}$ [a], then NN has at least a linear convergence rate, same as SVM:

$$L(w_t) - L(w^*) \leq e^{-2t}\left(L(w_0) - L(w^*)\right).$$

[a] This can be guaranteed in a parameter ball when $\lambda_{min}\left(\hat{\Theta}(w_0)\right) > \frac{2}{C}$ by using a sufficient wide NN [Liu et al., 2020b].

1. Equivalence between NN and SVM

Theorem [Equivalence between NN and SVM]

As the minimum width of the NN, $m = \min_{l \in [L]} m_l$, goes to infinity, the tangent kernel tends to be constant, $\hat{\Theta}(w_t; x, x_i) \to \hat{\Theta}(w_0; x, x_i)$. Assume $g_0(x) = f_0(x)$. Then the infinitely wide NN trained by subgradient descent with soft margin loss has the same dynamics as SVM with $\hat{\Theta}(w_0; x, x_i)$ trained by subgradient descent:

$$\frac{df_t(x)}{dt} = -f_t(x) + C \sum_{i=1}^{n} \mathbb{1}(y_i f_t(x_i) < 1) y_i \hat{\Theta}(w_0; x, x_i).$$

And thus such NN and SVM converge to the same solution.

1. Equivalence between NN and SVM

Experiments verification.

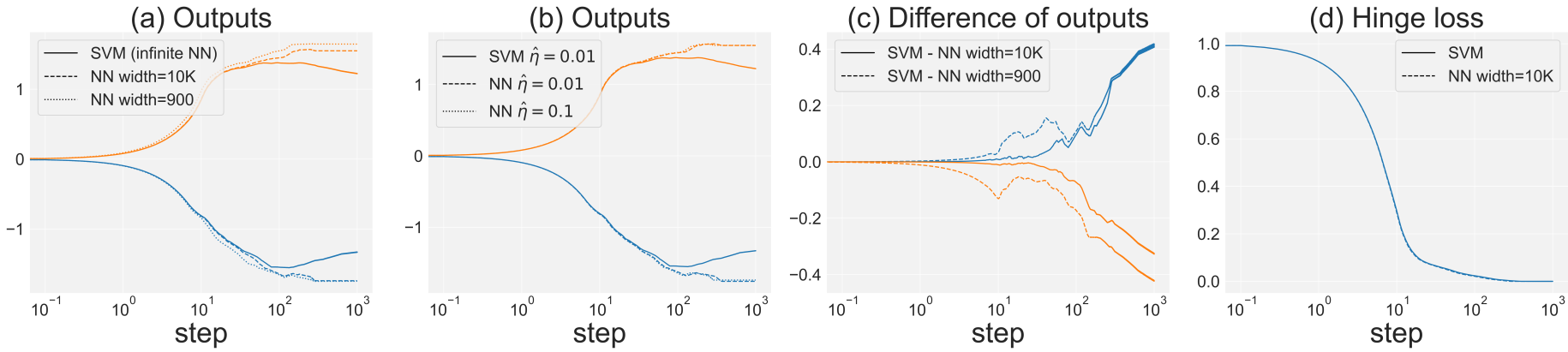

Figure: Training dynamics of neural network and SVM behave similarly. (a)(b) show dynamics of outputs for randomly selected two samples. (c) shows the difference between the outputs of SVM and NN. The dynamics of SVM agrees better with wider NN. (d) shows the dynamics of hinge loss for SVM and NN. Without specification, the width of NN is 10K and $\hat{\eta} = 0.1$.

2. Equivalence between NN and ℓ_2 regularized KMs

Suppose the loss function for the KM and NN are

$$L(\beta) = \frac{\lambda}{2}\|\beta\|^2 + \sum_{i=1}^{n} l(g(\beta, x_i), y_i), \tag{2}$$

$$L(w) = \frac{\lambda}{2}\|W^{(L+1)}\|^2 + \sum_{i=1}^{n} l(f(w, x_i), y_i). \tag{3}$$

Then the continuous dynamics of $g_t(x)$ and $f_t(x)$ are

$$\frac{dg_t(x)}{dt} = -\lambda g_t(x) - \sum_{i=1}^{n} l'(g_t(x_i), y_i) K(x, x_i),$$

$$\frac{df_t(x)}{dt} = -\lambda f_t(x) - \sum_{i=1}^{n} l'(f_t(x_i), y_i) \hat{\Theta}(w_t; x, x_i),$$

where $l'(z, y_i) = \frac{\partial l(z, y_i)}{\partial z}$.

2. Equivalence between NN and ℓ_2 regularized KMs

Theorem [Bounds on the difference between NN and KMs]

Assume $g_0(x) = f_0(x), \forall x$ and $K(x, x_i) = \hat{\Theta}(w_0; x, x_i)$ [a]. Suppose the SVM and NN are trained with losses (2) and gradient flow. Suppose l is ρ-lipschitz and β_l-smooth for the first argument (i.e. the model output). Given any $w_T \in B(w_0; R) := \{w : \|w - w_0\| \leq R\}$ for some fixed $R > 0$, for training data $X \in \mathbb{R}^{d \times n}$ and a test point $x \in \mathbb{R}^d$, with high probability over the initialization,

$$\|f_T(X) - g_T(X)\| = O\left(\frac{e^{\beta_l \|\hat{\Theta}(w_0)\|} R^{3L+1} \rho n^{\frac{3}{2}} \ln m}{\lambda \sqrt{m}}\right),$$

$$\|f_T(x) - g_T(x)\| = O\left(\frac{e^{\beta_l \|\hat{\Theta}(w_0; X, x)\|} R^{3L+1} \rho n \ln m}{\lambda \sqrt{m}}\right).$$

where $f_T(X), g_T(X) \in \mathbb{R}^n$ are the outputs of the training data and $\hat{\Theta}(w_0; X, x) \in \mathbb{R}^n$ is the tangent kernel evaluated between training data and test point.

2. Equivalence between NN and ℓ_2 regularized KMs

Table: Summary of our theoretical results on the equivalence between infinite-width NNs and a family of KMs.

λ	Loss $l(z, y_i)$	Kernel machine
$\lambda = 0$([*Jacotet al.*, 2018])	$(y_i - z)^2$	Kernel regression
$\lambda \to 0$ (ours)	$\max(0, 1 - y_i z)$	Hard margin SVM
$\lambda > 0$ (ours)	$\max(0, 1 - y_i z)$	(1-norm) soft margin SVM
	$\max(0, 1 - y_i z)^2$	2-norm soft margin SVM
	$\max(0, \lvert y_i - z \rvert - \epsilon)$	Support vector regression
	$(y_i - z)^2$	Kernel ridge regression (KRR)
	$\log(1 + e^{-y_i z})$	Logistic regression with ℓ_2 regularization

3. Finite-width NN trained by ℓ_2 regularized loss is approximately a KM

Theorem

Suppose an NN $f(w, x)$, is learned from a training set $\{(x_i, y_i)\}_{i=1}^n$ by (sub)gradient descent with loss function (2) and gradient flow. Assume $\text{sign}(l'(y_i, f_t(x_i))) = \text{sign}(l'(y_i, f_0(x_i))), \forall t \in [0, T]$.[a] Then at some time $T > 0$,

$$f_T(x) = \sum_{i=1}^{n} a_i K(x, x_i) + b,$$

$$K(x, x_i) = e^{-\lambda T} \int_0^T |l'(f_t(x_i), y_i)| \hat{\Theta}(w_t; x, x_i) e^{\lambda t} \, dt,$$

and $a_i = -\text{sign}(l'(f_0(x_i), y_i))$, $b = e^{-\lambda T} f_0(x)$.

[a] This is the case for hinge loss.

Main results

Our contributions:

1. Equivalence between NN and SVM
2. Equivalence between NN and a family of ℓ_2 regularized KMs
3. Finite-width NN trained by ℓ_2 regularized loss is approximately a kernel machine (KM)
4. Applications
   - 4.1 Computing non-vacuous generalization bound of NN via the corresponding KM
   - 4.2 Robustness certificate for over-parameterized NN
   - 4.3 ℓ_2 regularized KMs (from equivalent infinite-width NN) are more robust than previous kernel regression

4.1 Computing non-vacuous generalization bound

Combing above Theorem with a bound of the Rademacher complexity for KM and a standard generalization bound using Rademacher complexity, we can compute the generalization bound of NN via the corresponding KM.

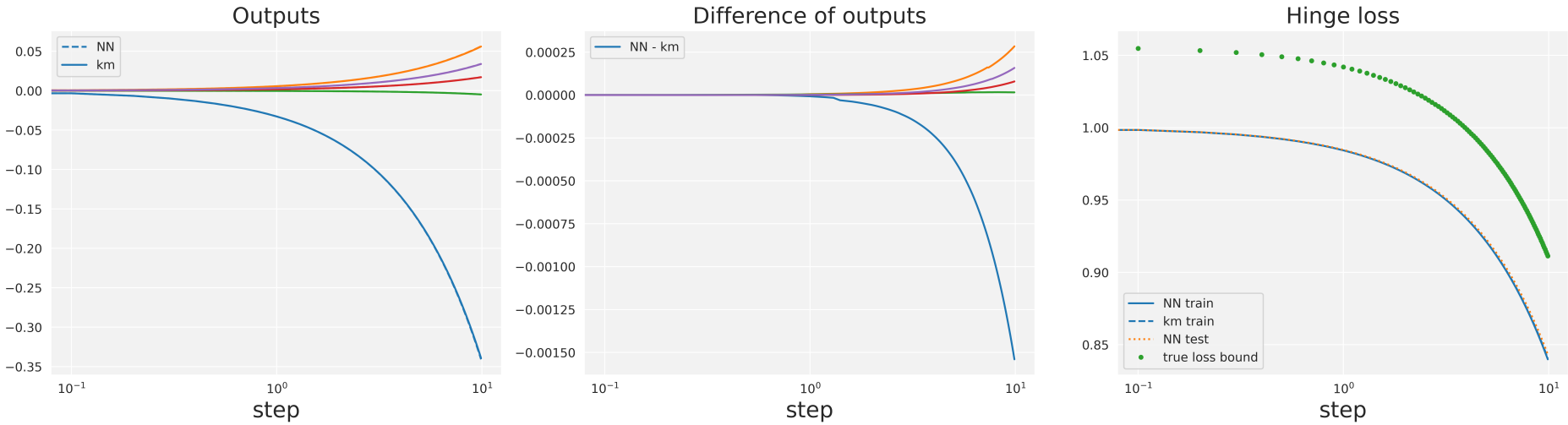

Figure: Computing non-vacuous generalization bounds via corresponding kernel machines. Two-layer NN with 100 hidden nodes trained by full-batch subgradient descent for binary MNIST classification task on full 0 and 1 data with learning rate $\hat{\eta} = 0.1$. The kernel machine (KM) approximates NN very well. And we get a tight bound of the true loss by computing its Rademacher complexity. The confidence parameter is set as $1 - \delta = 0.99$.

Main results

- Most of the existing generalization bounds of NN [Bartlett et al., 2019; Long and Sedghi, 2019] are vacacous since they have a dependence on the number of parameters.
- Compared to those, the bound for kernel machines does not have a dependence on the number of NN's parameters, making it non-vacuous and promising.
- Moreover, we can even apply this generalization bound to optimize NN directly like PAC-Bayes bound [Dziugaite and Roy, 2017], which gives NN with guaranteed generalization ability.

4.2 Robustness certificate for over-parameterized NN

Theorem

Consider the ℓ_∞ perturbation, for $x \in B_\infty(x_0, \delta) = \{x \in \mathbb{R}^d : \|x - x_0\|_\infty \leq \delta\}$, we can bound $\Theta(x, x')$ into some interval $[\Theta^L(x, x'), \Theta^U(x, x')]$. Suppose $g(x) = \sum_{i=1}^n \alpha_i \Theta(x, x_i)$, where α_i are known after solving the KM problems (e.g. SVM and KRR). Then we can lower bound $g(x)$ as follows.

$$g(x) \geq \sum_{i=1, \alpha_i > 0}^{n} \alpha_i \Theta^L(x, x_i) + \sum_{i=1, \alpha_i < 0}^{n} \alpha_i \Theta^U(x, x_i).$$

Using a simple binary search and above theorem, we can find a lower bound for the robustness radius of KM, equivalently for the corresponding infinite-width NN.

4.2 Robustness certificate for over-parameterized NN

We can deliver *nontrivial* robustness certificate for the over-parameterized NN (with width $m \to \infty$) while existing robustness verification methods would give trivial robustness certificate due to bound propagation (decrease at a rate of $O(1/\sqrt{m})$).

Table: Robustness lower bounds of two-layer ReLU NN and SVM (infinite-width two-layer ReLU NN) tested on binary classification of MNIST (0 and 1). 100 test: randomly selected 100 test samples. Full test: full test data. Test only on data that classified correctly. std is computed over data samples. All models have test accuracy 99.95%. All values are mean of 5 experiments.

		Robustness certificate δ (mean $\pm$ std) $\times 10^{-3}$	
Model	Width	100 test	Full test
NN	10^3	7.4485 $\pm$ 2.5667	7.2708 $\pm$ 2.1427
NN	10^4	2.9861 $\pm$ 1.0730	2.9367 $\pm$ 0.89807
NN	10^5	0.99098 $\pm$ 0.35775	0.97410 $\pm$ 0.29997
NN	10^6	0.31539 $\pm$ 0.11380	0.30997 $\pm$ 0.095467
SVM	∞	8.0541 $\pm$ 2.5827	7.9733 $\pm$ 2.1396

4.3 ℓ_2 regularized KMs are more robust than kernel regression

Table: Robustness of equivalent infinite-width NN models with different loss functions (see Table 1) on binary classification of MNIST (0 and 1). λ is the parameter in Eq. (2).

	Model	λ	Test accuracy	Robustness certificate δ	Robustness improvement
$\lambda = 0$([*Jacotet al.*, 2018])	KRR	0	99.95%	3.30202×10^{-5}	-
$\lambda > 0$ (ours)	KRR	0.001	99.95%	3.756122×10^{-5}	1.14X
	KRR	0.01	99.95%	6.505500×10^{-5}	1.97X
	KRR	0.1	99.95%	2.229960×10^{-4}	6.75X
	KRR	1	99.95%	0.001005	30.43X
	KRR	10	99.91%	0.005181	156.90X
	KRR	100	99.86%	0.020456	619.50X
	KRR	1000	99.76%	0.026088	790.06X
	SVM	0.032	99.95%	0.008054	243.91X

Table of Contents

Conclusions and future works

Conclusions:

1. Equivalence between NN and SVM
2. Equivalence between NN and a family of ℓ_2 regularized KMs
3. Finite-width NN trained by ℓ_2 regularized loss is approximately a kernel machine (KM)
4. Applications
   - 4.1 Computing non-vacuous generalization bound of NN via the corresponding KM
   - 4.2 Robustness certificate for over-parameterized NN
   - 4.3 ℓ_2 regularized KMs (from equivalent infinite-width NN) are more robust than previous kernel regression

Conclusions and future works

Future works:

- Understand the optimization, generalization, and robustness of NN from the perspective of these new equivalent KMs
- Consider its connection with the implicit bias of NN
  - Max-margin solution

References I

S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8141–8150, 2019.

P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research*, 20(1):2285–2301, 2019.

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8572–8583, 2019.

References II

C. Liu, L. Zhu, and M. Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33, 2020a.

C. Liu, L. Zhu, and M. Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *arXiv preprint arXiv:2003.00307*, 2020b.

P. M. Long and H. Sedghi. Generalization bounds for deep convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.