On the Equivalence between Neural Network and Support Vector Machine

Yilan Chen† · Wei Huang‡ · Lam M. Nguyen§ · Tsui-Wei Weng†

† University of California San Diego, ‡ University of Technology Sydney, § IBM Research, Thomas J. Watson Research Center

Introduction, Motivation and Contributions

Introduction:

- Neural Tangent Kernel (NTK) [2]:
$$\hat{\Theta}(w;x,x') = \langle \nabla_w f(w,x), \nabla_w f(w,x') \rangle$$
- 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 ⇔ kernel regression with NTK [2, 1]
- Wide neural networks are linear [3]:
$$f(w_t,x) = f(w_0,x) + \langle \nabla_w f(w_0,x), w_t - w_0 \rangle + O(m^{-\frac{1}{2}})$$
where m is the width of NN.
- Constant tangent kernel ⇔ Linear model. Small Hessian norm ⇒ small change of tangent kernel [4].

Motivations:

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.

Key Question: Can we establish the equivalence between NN and other kernel machines?

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
4. Applications: (a) Computing non-vacuous generalization bound of NN via the corresponding KM; (b) Robustness certificate for over-parameterized NN; (c) ℓ_2 regularized KMs (from equivalent infinite-width NN) are more robust than previous kernel regression

Definitions

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.

Neural Network. $\forall l \in [L]$,

$$\alpha^{(0)}(w,x) = x,\ \alpha^{(l)}(w,x) = \phi_l(w^{(l)}, \alpha^{(l-1)}),\ f(w,x) = \frac{1}{\sqrt{m_L}}\langle w^{(L+1)}, \alpha^{(L)}(w,x)\rangle,$$

where each vector-valued function $\phi_l(w^{(l)},\cdot): \mathbb{R}^{m_{l-1}} \to \mathbb{R}^{m_l}$, with parameter $w^{(l)} \in \mathbb{R}^{p_l}$, is considered as a layer of the network.

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

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

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

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*.

Equivalence between NN and SVM

Theorem 1 (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:

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

Theorem 2 (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 $\lambda_{min}\left(\hat{\Theta}(w_t)\right) \geq \frac{2}{C}$, then NN has at least a linear convergence rate, same as SVM:

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

Theorem 3 (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.

Equivalence between NN and ℓ_2 regularized KMs

Suppose the loss function for the KM and NN are

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

Theorem 4 (Bounds on the difference between NN and KMs). *Assume $g_0(x) = f_0(x), \forall x$ and $K(x,x_i) = \hat{\Theta}(w_0;x,x_i)$ [1]. 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.

λ	Loss $l(z,y_i)$	Kernel machine
$\lambda = 0$ ([2])	$(y_i - z)^2$	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

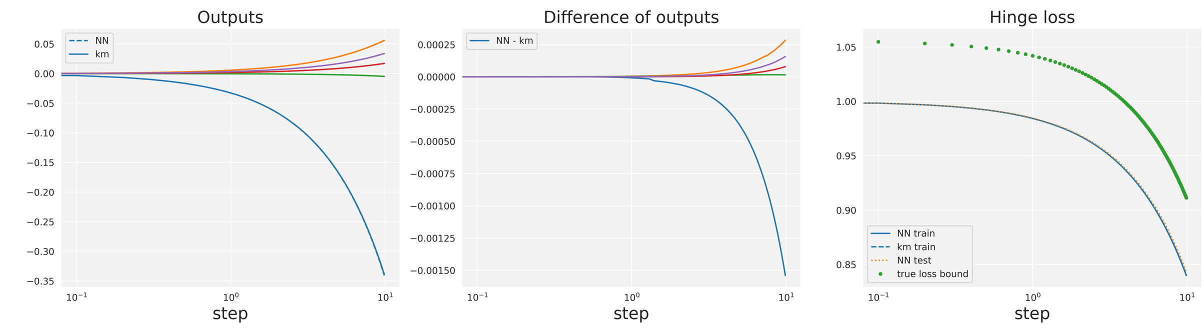
Finite-width NN trained by ℓ_2 regularized loss

Theorem 5. *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 $sign(l'(y_i, f_t(x_i))) = sign(l'(y_i, f_0(x_i))), \forall t \in [0,T]$. 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 $a_i = -sign(l'(f_0(x_i),y_i))$, $b = e^{-\lambda T} f_0(x)$.



Robustness certificate for over-parameterized NN

Theorem 6. *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.*

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

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

	Model	λ	Test acc.	Robustness cert.	Cert. Improv.
$\lambda = 0$ ([2])	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

References

[1] Sanjeev Arora et al. "On exact computation with an infinitely wide neural net". In: *Advances in Neural Information Processing Systems*. 2019, pp. 8141–8150.

[2] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems*. 2018, pp. 8571–8580.

[3] Jaehoon Lee et al. "Wide neural networks of any depth evolve as linear models under gradient descent". In: *Advances in neural information processing systems*. 2019, pp. 8572–8583.

[4] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. "On the linearity of large non-linear models: when and why the tangent kernel is constant". In: *Advances in Neural Information Processing Systems* 33 (2020).