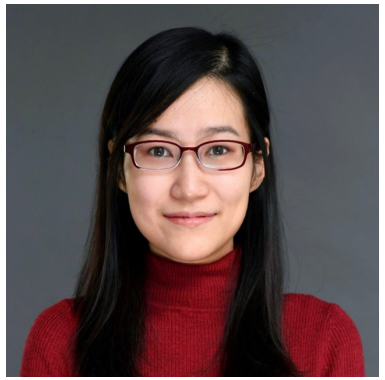
UC San Diego

Analyzing Neural Networks through Equivalent Kernels

Yilan Chen

UCSD CSE

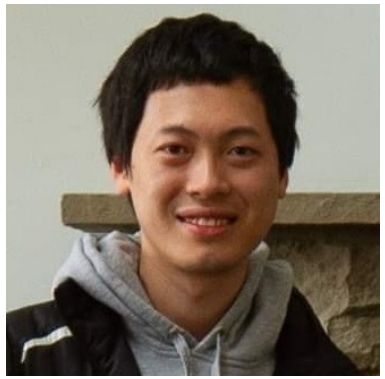
SOCAMS 2024



Tsui-Wei (Lily) Weng
UCSD HDSI

Wei Huang
RIEKN AIP

Hao Wang
MIT-IBM Watson AI Lab



Lam M. Nguyen
IBM Research

Akash Srivastava
MIT-IBM Watson AI Lab

Outline

1. Equivalence between neural networks and kernel machines
2. Analyzing generalization of neural networks through loss path kernels
3. Conclusion and future works

Outline

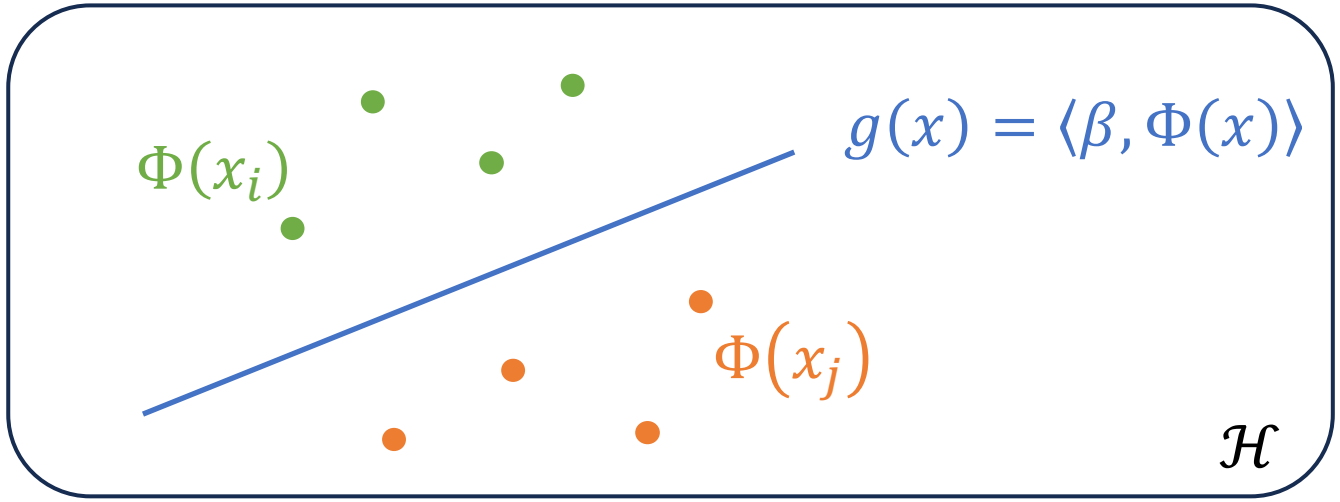
1. Equivalence between neural networks and kernel machines
   - Wide NN and Neural Tangent Kernel (NTK)
   - General NN and Loss Path Kernel (LPK)
2. Analyzing generalization of neural networks through loss path kernels
   - Generalization bound for NN trained by gradient flow
   - Case study and Application
     - Ultra-wide NN
     - Neural architecture search
3. Conclusion and future works

Kernel Machine

- Kernel: $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$, Φ: $\mathcal{X} \to \mathcal{H}$ maps the data to a (potentially infinite dimensional) feature space $\mathcal{H}$.
- Kernel machine (KM): linear function in the feature space

$$g(x) = \langle \beta, \Phi(x) \rangle + b = \sum_{i=1}^{n} a_i K(x, x_i) + b, \quad \text{where } \beta = \sum_{i=1}^{n} a_i \Phi(x_i)$$

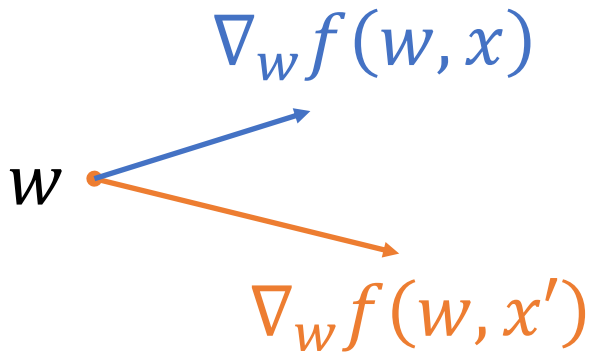
- RKHS norm of g: $\|\beta\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j K(x_i, x_j)}$



Wide NN and Neural Tangent Kernel

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

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



measures the similarity between data points x, x' by comparing their gradients

- Under suitable conditions (e.g., infinite width limit and NTK initialization), NTK at initialization w_0 converges to a deterministic limit and keeps constant during training:

$$\widehat{\Theta}(w_0;\ x, x') \to \Theta_\infty(x, x')$$

NTK at initialization Independent of w_0

Wide NN and Neural Tangent Kernel

- Infinite-width neural network (NN) trained by gradient descent with mean square loss ⇔ kernel regression with NTK [Jacot et al., 2018; Arora et al., 2019]

kernel regression: linear regression in the feature space

$$\min_{w_t} \sum_{i=1}^{n} \| \langle w_t - w_0 , \nabla_w f(w_0, x_i) \rangle - y_i \|^2$$

- Wide NNs are linear in the parameter space [Lee et al., 2019]:

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

m: width of NN

- Infinite-width NN trained with ℓ_2 regularized loss ⇔ ℓ_2 regularized KMs with NTK, e.g. SVM [Chen et al., 2021]

Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

Equivalence between Wide NN and SVM

Ultra-wide NN trained with

equivalent

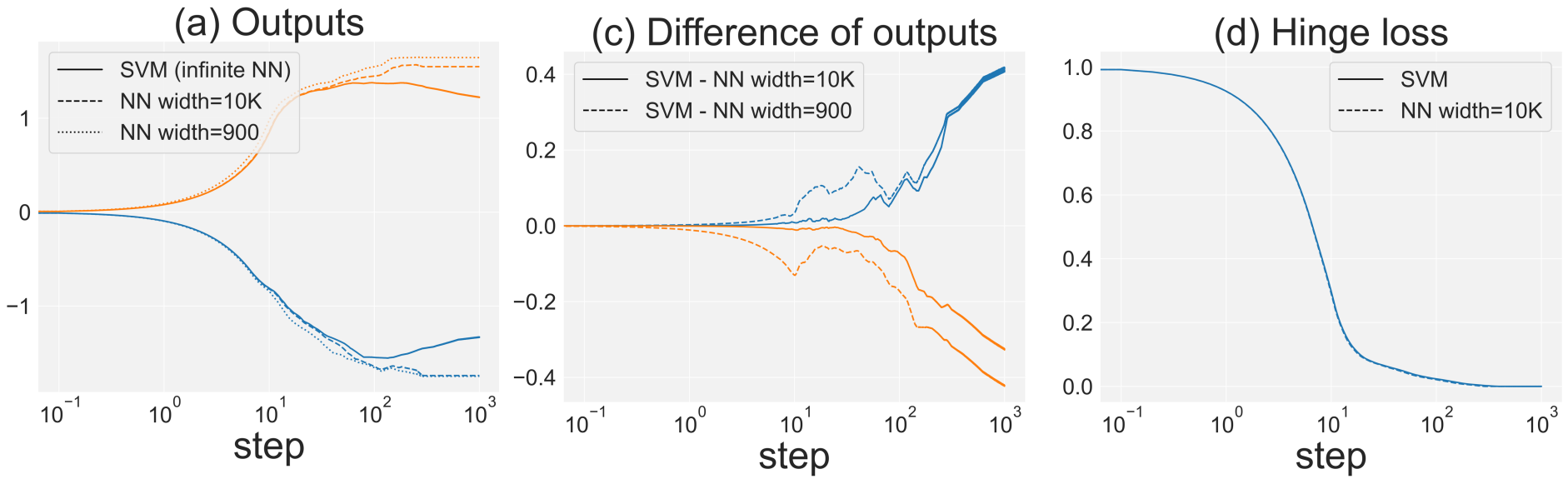
KM $g(\beta, x) = \langle \beta, \nabla_w f(w_0, x) \rangle$ with

$$L(w) = \frac{\lambda}{2} \left\| w^{(L)} \right\|^2 + \sum_{i=1}^{n} \ell(f(w, x_i), y_i)$$

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

$w^{(L)}$: last layer of NN

SVM: $\ell(z, y) = \max(0, 1 - zy)$



Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

Equivalence between Wide NN and SVM

Our prior work [Chen et al., 2021]:

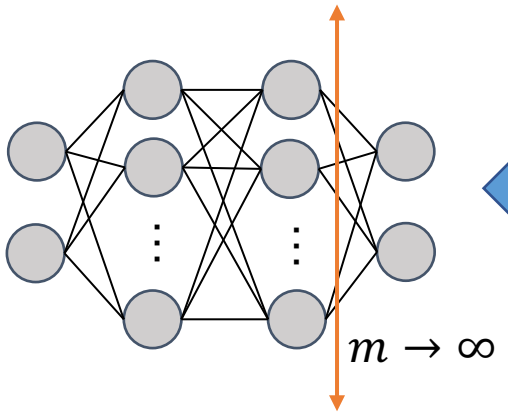
λ	Loss $\ell(z, y)$	Kernel machine
$\lambda = 0$ ([1,2])	$(y - z)^2$	Kernel regression
$\lambda \to 0$ (ours)	$\max(0, 1 - yz)$	Hard margin SVM
$\lambda > 0$ (ours)	$\max(0, 1 - yz)$	(1-norm) soft margin SVM
	$\max(0, 1 - yz)^2$	2-norm soft margin SVM
	$\max(0, \lvert y - z\rvert - \epsilon)$	Support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

Suppose ℓ is Lipschitz and smooth,

$$\|f(x) - g(x)\| \leq \tilde{O}\left(\frac{1}{\sqrt{m}}\right)$$

Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

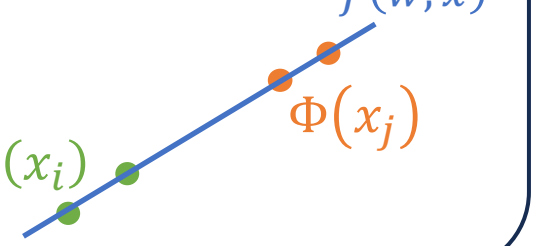
Wide NN and Neural Tangent Kernel



equivalent

kernel regression

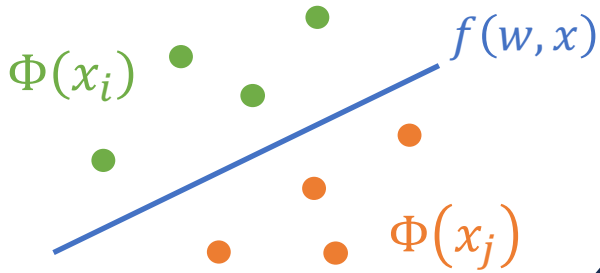
$\mathcal{H}: \Phi(x) = \nabla_w f(w_0, x)$



[Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019]

ℓ_2 regularized KMs

$\mathcal{H}: \Phi(x) = \nabla_w f(w_0, x)$



[Chen et al., 2021]

Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

Wide NN and Neural Tangent Kernel

These equivalences are useful for analyzing NNs, but

☹
- Only holds for infinite-width/ultra-wide NNs
- NTK, as a fixed kernel, lacks the ability of feature learning
- NTK has a performance gap compared with practical NNs

Q. Can we establish an equivalence between general NNs and KMs?

General NN and Loss Path Kernel (LPK)

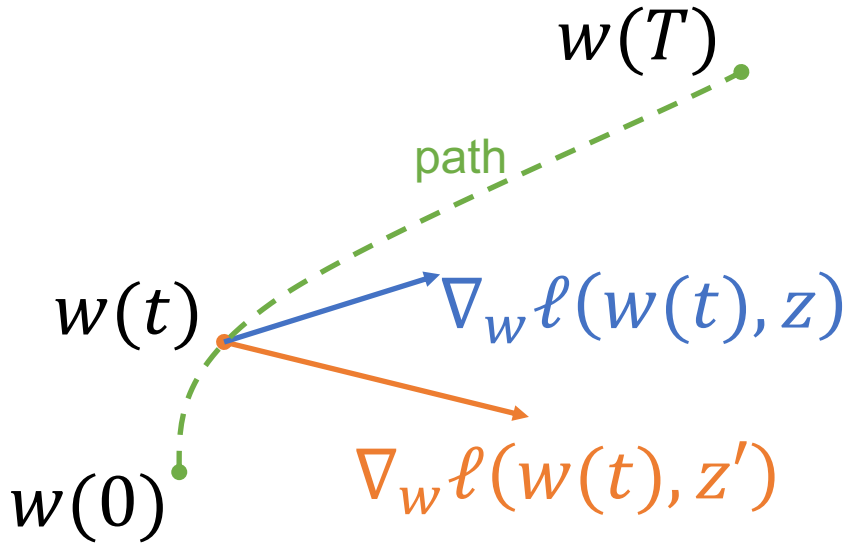
Consider gradient flow (gradient descent with infinitesimal step size):

$$\frac{w(t+1)-w(t)}{\eta} = -\nabla_w L_S(w(t)) \xrightarrow{\eta \to 0} \frac{dw(t)}{dt} = -\nabla_w L_S(w(t))$$

Loss Path Kernel (LPK):

$$\mathrm{K}_T(z, z'; S) = \int_0^T \langle \nabla_w \ell(w(t), z), \nabla_w \ell(w(t), z') \rangle \, dt$$

where $z = (x, y)$



Chen et al., 2023. Analyzing Generalization of Neural Networks through Loss Path Kernels. NeurIPS 2023.

General NN and Loss Path Kernel (LPK)

Loss Path Kernel (LPK):

$$\mathrm{K}_T(z, z'; S) = \int_0^T \langle \nabla_w \ell(w(t), z), \nabla_w \ell(w(t), z') \rangle \, dt$$

We can derive equivalence:

$$\ell(w_T, z) = \sum_{i=1}^{n} -\frac{1}{n} \mathrm{K}_T(z, z_i; S) + \ell(w_0, z)$$

Loss function at time T

Kernel machine with **LPK**

Loss function at initialization

Very general equivalence!

Chen et al., 2023. Analyzing Generalization of Neural Networks through Loss Path Kernels. NeurIPS 2023.

General NN and Loss Path Kernel (LPK)

Stochastic gradient flow (SGD with infinitesimal step size):

$$\frac{w(t+1)-w(t)}{\eta} = -\nabla_w L_{S_t}(w(t)) \xrightarrow{\eta \to 0} \frac{dw(t)}{dt} = -\nabla_w L_{S_t}(w(t))$$

$S_t \subseteq \{1, \dots, n\}$ is the indices of batch data, $m = |S_t|$: batch size

Equivalence:

Sum of KMs with **LPK**

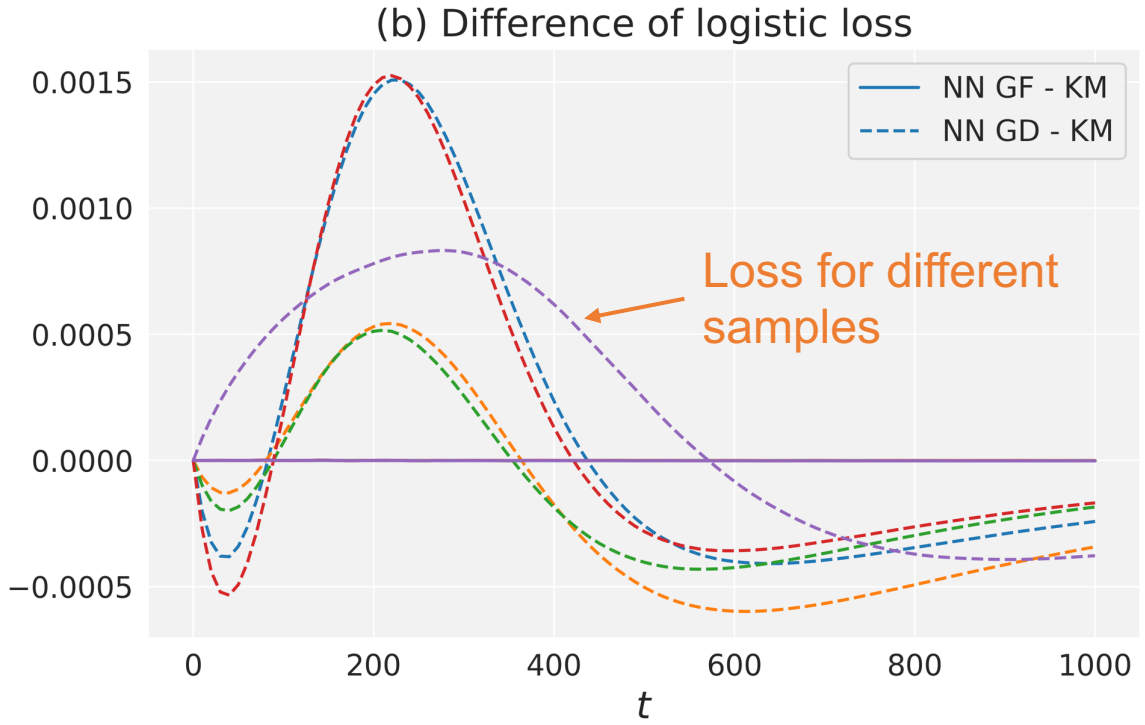
$$\ell(w_T, z) = \sum_{t=1}^{T-1} \sum_{i \in S_t} -\frac{1}{m} \mathrm{K}_T(z, z_i; S) + \ell(w_0, z)$$

Chen et al., 2023. Analyzing Generalization of Neural Networks through Loss Path Kernels. NeurIPS 2023.

General NN and Loss Path Kernel (LPK)

Verify the equivalence: two-layer NN

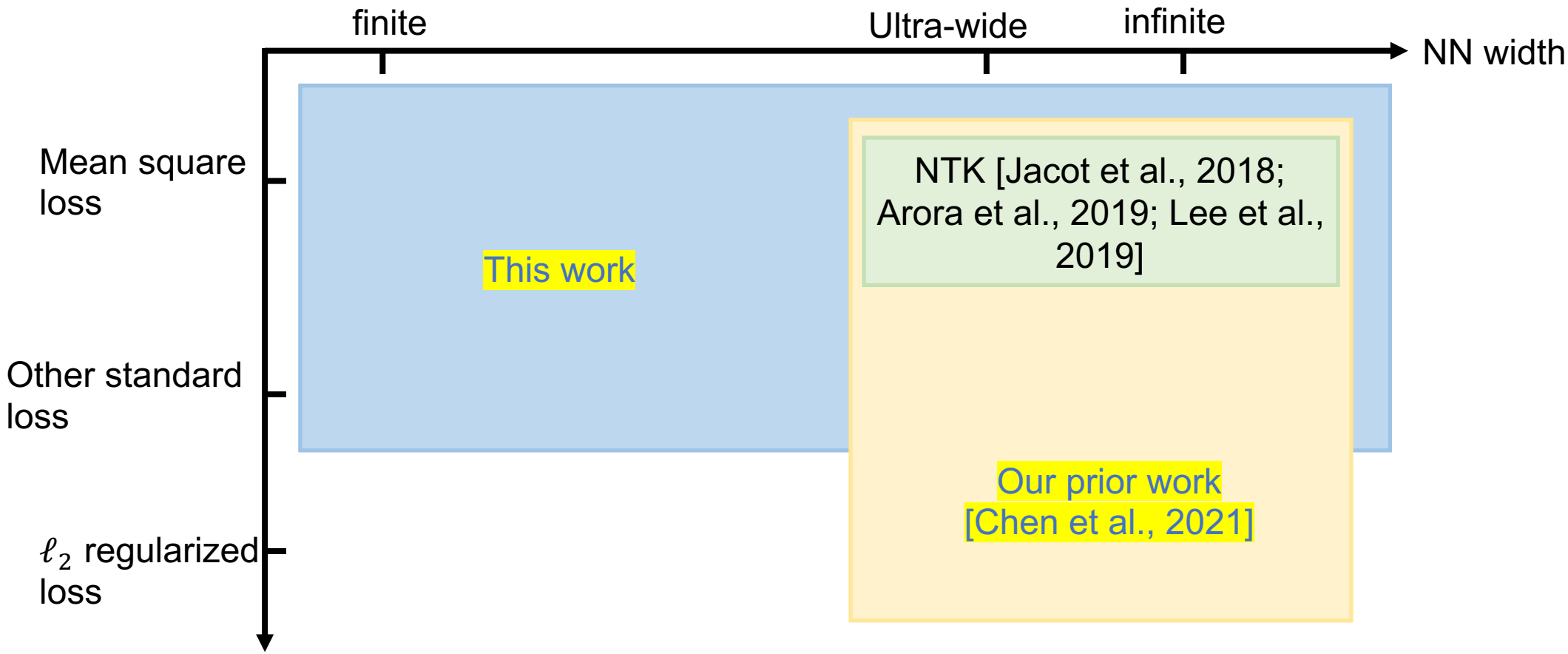
$$\ell(w_T, z) - \left(\sum_{i=1}^{n} -\frac{1}{n}\mathbf{K}_T(z, z_i; S) + \ell(w_0, z)\right)$$



- NN trained by gradient flow (GF) exactly equal to the KM
- NN trained by gradient descent (GD) is also close with the KM

General NN and Loss Path Kernel (LPK)

Compare with previous equivalence results:



Outline

1. Equivalence between neural networks and kernel machines
   - Wide NN and Neural Tangent Kernel (NTK)
   - General NN and Loss Path Kernel (LPK)

2. Analyzing generalization of neural networks through Loss Path Kernel
   - Generalization bound for NN through LPK
   - Case study and Application
     - Ultra-wide NN
     - Neural architecture search

3. Conclusion and future works

Generalization theory of neural networks

How do the neural networks (NN) generalize on test data?

generalization gap:

$$GAP = \underbrace{\mathbb{E}_{z\sim\mu}\left[\ell(w,z)\right]}_{L_\mu(w)} - \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell(w,z)}_{L_S(w)} \leq \boxed{?}$$

$L_\mu(w)$: population loss

$L_S(w)$: training loss

$$GAP \leq \sqrt{\frac{|\mathcal{G}|}{n}}$$

$\mathcal{G}$: NN function class

n: # of samples

Generalization theory: general NNs

1. VC dimension [Bartlett et al., 2019]

$$GAP \leq O\left(\sqrt{L\frac{\#\ of\ parameters}{n}}\log(n)\right)$$

L: # of layers
n: # of samples
W_l: weight of layer l

2. Norm-based bounds [Bartlett et al., 2017; …]

$$GAP \leq O\left(\frac{\prod_{l=1}^{L}\|W_l\|}{\sqrt{n}}\right)$$

☹
- Do not explain the generalization ability of overparameterized NNs. [Belkin et al., 2019]
- Vacuous: too large to be useful

- Other bounds:
  - PAC-Bayes bounds (mainly focus on stochastic NNs)
  - Information-theoretical approach (expected bound)

Bartlett, et al.. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. JMLR 2019.
Bartlett, et al.. Spectrally-normalized margin bounds for neural networks. NeurIPS 2017.

Generalization theory: ultra-wide NNs

- Arora et al., 2019: for ultra-wide two-layer FCNN,

$$GAP \leq \sqrt{\frac{2\,\mathbf{y}^\top (\mathbf{H}^\infty)^{-1}\mathbf{y}}{n}}$$

$\mathbf{H}^\infty$: NTK of the first layer

- Cao & Gu, 2019: for ultra-wide L-layer FCNN,

$$GAP \leq \tilde{O}\left(L \cdot \sqrt{\frac{2\,\mathbf{y}^\top (\Theta)^{-1}\mathbf{y}}{n}}\right)$$

☹ Tighter but only hold for ultra-wide NNs

Q. Can we establish tight generalization bounds for general NNs (vs ultra-wide NNs)?

Generalization bound for NN through LPK

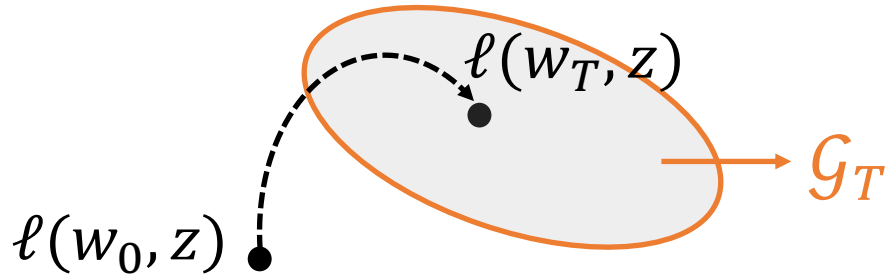
Different training set induces distinct LPK. Set of LPKs with constrained RKHS norm:

Set of LPKs

$$\mathcal{K}_T = \left\{ \mathrm{K}_T(\cdot,\cdot;S') : S' \in \mathrm{supp}(\mu^{\otimes n}), \frac{1}{n^2}\sum_{i,j} \mathrm{K}_T\left(z_i', z_j'; S'\right) \le B^2 \right\}$$

$$S = \{z_i\}_{i=1}^n, \quad S' = \{z_i'\}_{i=1}^n$$

Set of NNs trained to time T from all feasible S' :



$$\mathcal{G}_T = \left\{ g(z) = \sum_{i=1}^{n} -\frac{1}{n}\mathrm{K}(z, z_i'; S') + \ell(w_0, z) : \mathrm{K}(\cdot,\cdot;S') \in \mathcal{K}_T \right\}$$

$\ell(w_T, z)$ trained from S'

Generalization bound for NN through LPK

Compute the Rademacher complexity of $\mathcal{G}_T$,

$$GAP \leq \frac{2B}{n}\sqrt{\sup_{\mathrm{K}\in\mathcal{K}_T}\sum_{i=1}^{n}\mathrm{K}(z_i,z_i;S') + \sum_{i\neq j}\Delta(z_i,z_j)}$$

$$\Delta(z_i,z_j) = \frac{1}{2}[\sup_{\mathrm{K}\in\mathcal{K}_T}\mathrm{K}(z_i,z_j;S') - \inf_{\mathrm{K}\in\mathcal{K}_T}\mathrm{K}(z_i,z_j;S')]$$

maximum magnitude of the loss gradient in $\mathcal{K}_T$ evaluated with S throughout the training trajectory.

range of variation of LPK in $\mathcal{K}_T$

$$\mathrm{K}_T(z_i,z_i;S') = \int_0^T \|\nabla_w \ell(w,z_i)\|^2\, dt$$

- The generalization of NN depends on its training trajectory!
- For fixed dataset, NN converges faster and with smaller loss gradients will generalize better
- Intuitively, NN that changes less from initialization is less complex and has less chance to overfit

Generalization bound for NN through LPK

Analyze the covering number of $\mathcal{G}_T$,

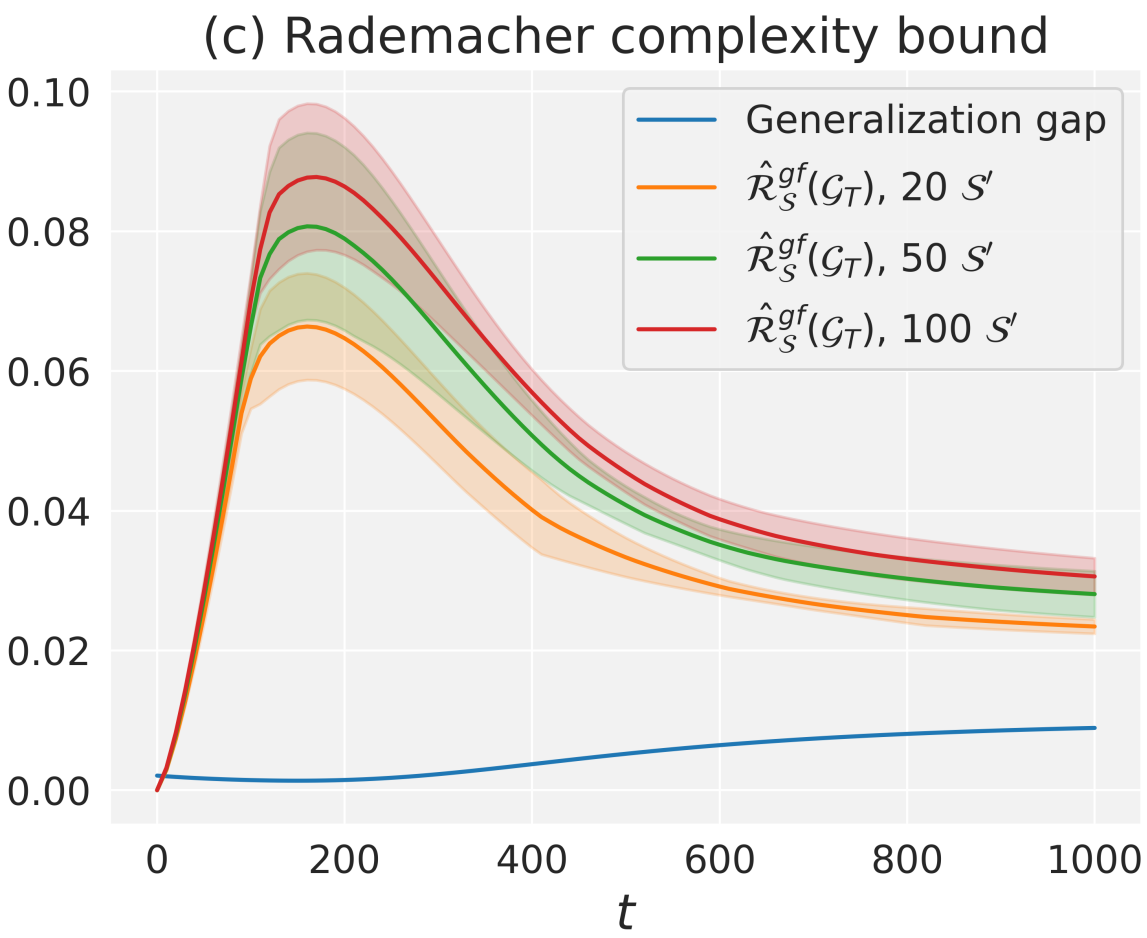
$$GAP \leq 2 \inf_{\epsilon>0} \left(\frac{\epsilon}{n} + \sqrt{\frac{2 \ln \mathcal{N}(\mathcal{G}_T^S, \epsilon, \| \|_1)}{n}} \right)$$

$\mathcal{G}_T^S = \{g(\mathbf{Z}) = (g(z_1), \dots, g(z_n)) : g \in \mathcal{G}_T\}$,
$\mathcal{N}(\mathcal{G}_T^S, \epsilon, \| \|_1)$ is the covering number of $\mathcal{G}_T^S$.

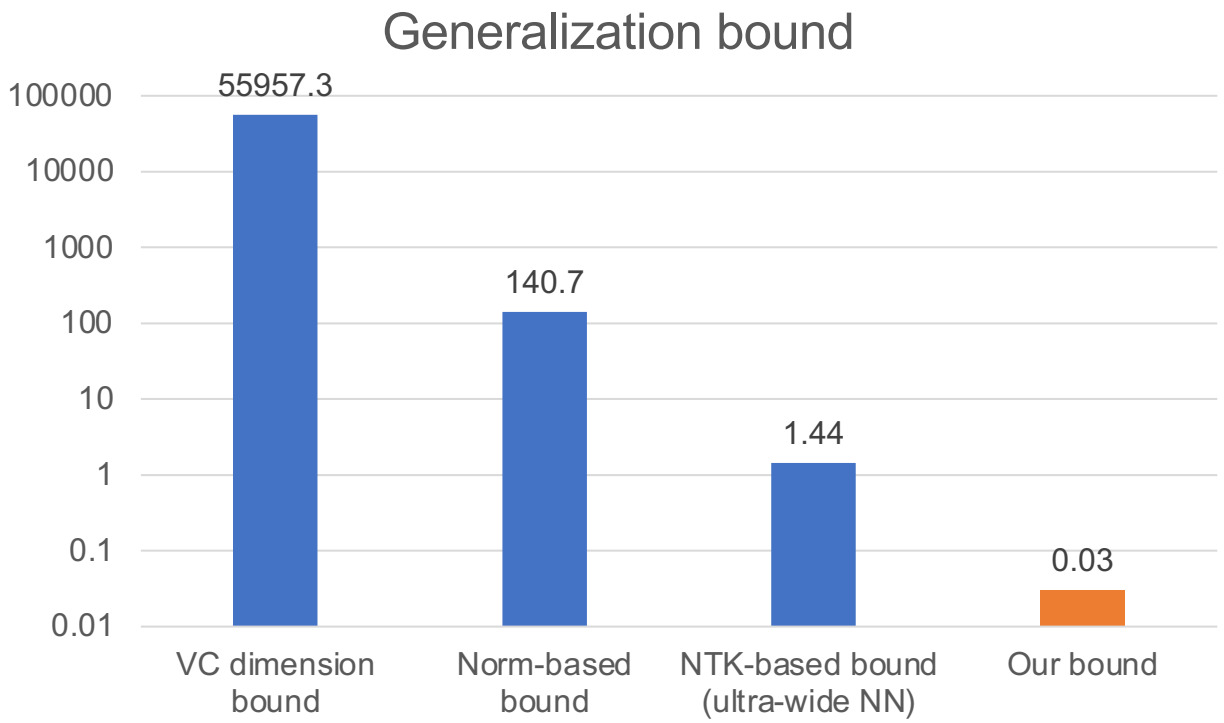
If the variation of the loss with different training data is small, GAP will be small.

Experiment: Generalization bound for NN through LPK

Experiment of two-layer ReLU NN

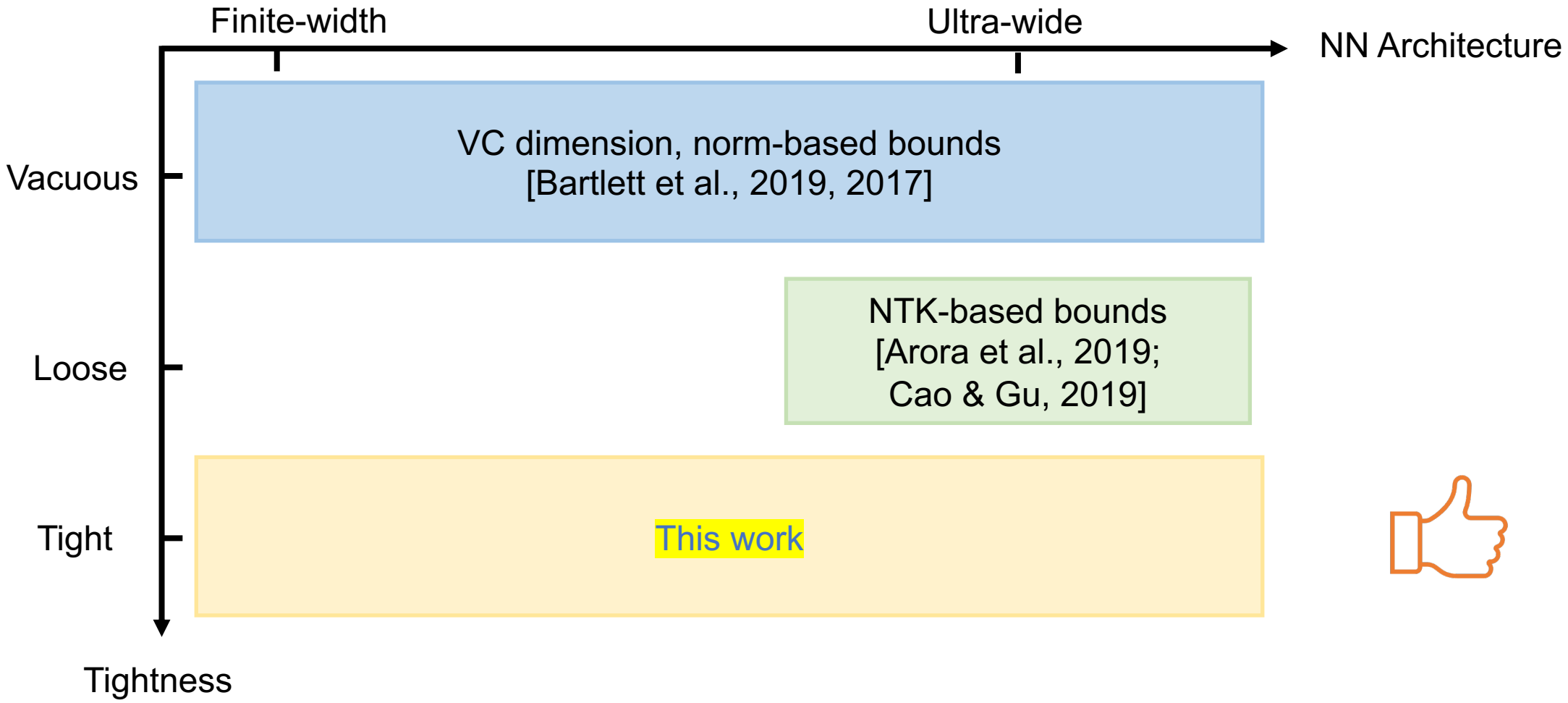


Tight bound!



Generalization bound for NN through LPK

Compare with previous generalization bounds:



Case study and application

1. Generalization bound for infinite-width NN. $\Theta(x, x')$: NTK of infinite-width NN

$$GAP \leq \frac{\rho B\sqrt{T}}{n}\sqrt{\sum_{i,j}\left|\Theta\left(x_i, x_j\right)\right|}$$

ρ: Lipschitz constant of $\ell(f, y)$

2. Neural architecture search (NAS). Use the bound to estimate the test loss and design minimum-training NAS algorithms:

$$\mathrm{Gene}(w, S) = L_S(w) + 2U_{sgd}$$

U_{sgd}: simplified bound of stochastic gradient flow

Algorithm	CIFAR-10 Accuracy	Best	CIFAR-100 Accuracy	Best
Baselines				
TENAS [13]	93.08±0.15	93.25	70.37±2.40	**73.16**
RS + LGA_3 [39]	93.64		69.77	
Ours				
RS + $\mathrm{Gene}(w, S)_1$	93.68±0.12	93.84	72.02±1.43	73.15
RS + $\mathrm{Gene}(w, S)_2$	**93.79**±0.18	**94.02**	**72.76**±0.33	73.15
Optimal	94.37		73.51	

Outline

1. Equivalence between neural networks and kernel machines
   - Wide NN and Neural Tangent Kernel (NTK)
   - General NN and Loss Path Kernel (LPK)
2. Analyzing generalization of neural networks through loss path kernels
   - Generalization bound for NN trained by gradient flow
   - Case study and Application
     - Ultra-wide NN
     - Neural architecture search
3. Conclusion and future works

Conclusion

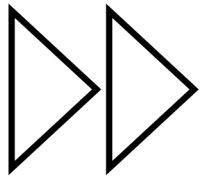
1 New equivalence between NN and KM

- NN-SVM equivalence
- NN-LPK: much more general equivalence

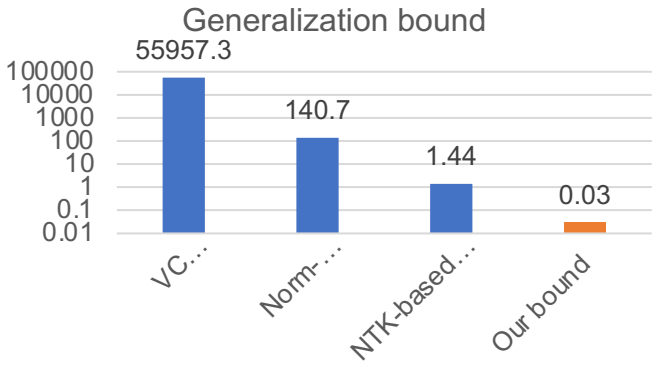
$$\ell(w_T, z) = \sum_{i=1}^{n} -\frac{1}{n} \mathrm{K}_T(z, z_i; S) + \ell(w_0, z)$$

Kernel machine with **LPK**

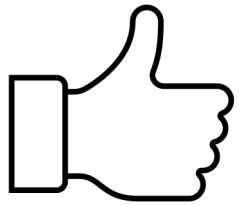
2 Generalization bound for NN



- Holds for general NNs
- Tighter bounds!



3 Useful in theory and practice



- Better bound for ultra-wide NNs
- Minimum-training NAS algorithms

Algorithm	CIFAR-10 Accuracy	Best	CIFAR-100 Accuracy	Best
Baselines				
TENAS [13]	93.08±0.15	93.25	70.37±2.40	**73.16**
RS + LGA_3 [39]	93.64		69.77	
Ours				
RS + Gene$(w, S)_1$	93.68±0.12	93.84	72.02±1.43	73.15
RS + Gene$(w, S)_2$	**93.79**±0.18	**94.02**	**72.76**±0.33	73.15
Optimal	94.37		73.51	

Future works

1 Applications based on the theory

- Analyze different NN architectures from this equivalence
- Design better kernel function based on LPK
- Quantify the influence of each training sample (core set selection, interpretability, robustness)
- …

2 Equivalence & generalization for other optimization algorithms

- SGD with momentum
- Adam

3 Improve the generalization bound

- Remove supremum
- Tighter and meaningful bound
- Fine-grained characterization of LPK

References

1. Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31, 2018.
2. Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. Advances in Neural Information Processing Systems, 32, 2019.
3. Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. Advances in neural information processing systems, 32, 2019.
4. Chen, Y., Huang, W., Nguyen, L., and Weng, T.-W. On the equivalence between neural network and support vector machine. Advances in Neural Information Processing Systems, 34, 2021.
5. Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences, 116(32):15849–15854, 2019.
6. Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. The Journal of Machine Learning Research, 2019.
7. Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In International Conference on Machine Learning, pp. 322–332. PMLR, 2019.
8. Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. Advances in neural information processing systems, 32, 2019.
9. Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 2002.
10. Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. Advances in neural information processing systems, 30, 2017.

Thank you for listening!