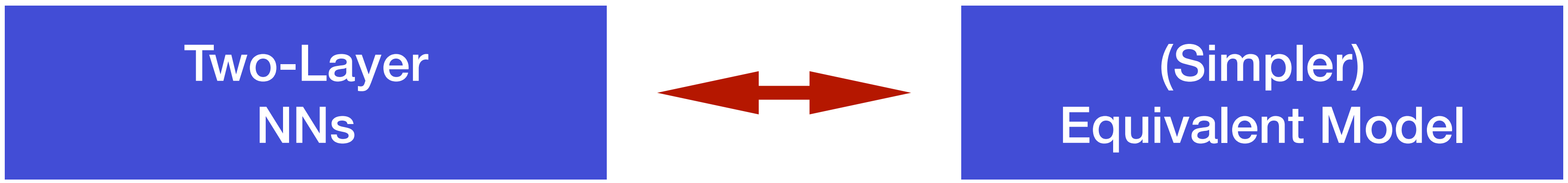
Asymptotic Analysis of Two-Layer Neural Networks after One Gradient Step under Gaussian Mixtures Data with Structure

Samet Demir & Zafer Dogan
MLIP Research Group, KUIS AI Center, Koç University

Theory of Neural Networks via Equivalent Models

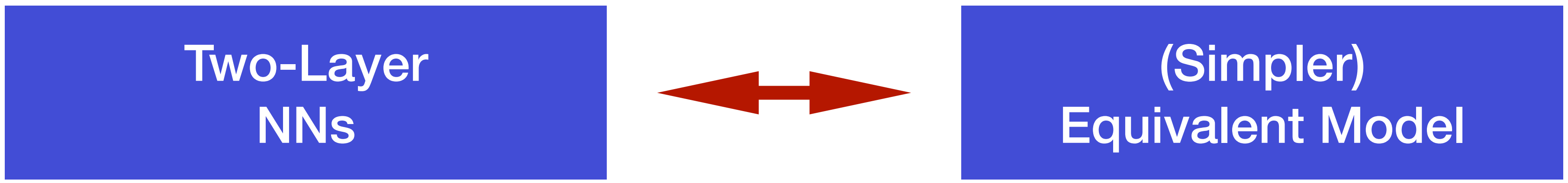
- We focus on the analysis of generalization performance of two-layer neural networks (NNs) for supervised learning via <u>equivalent models</u>.



<u>Generalization (and training) errors of the two models are equivalent.</u>

Theory of Neural Networks via Equivalent Models

- We focus on the analysis of generalization performance of two-layer neural networks (NNs) for supervised learning via equivalent models.



Generalization (and training) errors of the two models are equivalent.

- We consider the asymptotically proportional limit:
  - input dimension, number of neurons, and sample size diverge with finite ratios

Theory of Neural Networks via Equivalent Models

- We focus on the analysis of generalization performance of two-layer neural networks (NNs) for supervised learning via <u>equivalent models</u>.

Two-Layer NNs ⟷ (Simpler) Equivalent Model

<u>Generalization (and training) errors of the two models are equivalent.</u>

- We consider the asymptotically proportional limit:
  - input dimension, number of neurons, and sample size diverge with finite ratios
- Limitations of the literature
  - Lack of feature learning (e.g., random feature model) [1]
  - Limited data assumption (e.g., standard Gaussian inputs) [1,2,3,4]

[1] Hu and Lu. Universality laws for high-dimensional learning with random features. IEEE Trans. Inf. Theory, 69(3):1932–1964, Mar. 2023.
[2] Dandi et al. How two-layer neural networks learn, one (giant) step at a time, arXiv:2305.18270, 2023.
[3] Cui et al. Asymptotics of feature learning in two-layer networks after one gradient-step. ICML 2024.
[4] Moniri et al. A theory of non-linear feature learning with one gradient step in two-layer neural networks. ICML 2024.

Theory of Neural Networks via Equivalent Models

- We focus on the analysis of generalization performance of two-layer neural networks (NNs) for supervised learning via equivalent models.

Two-Layer NNs ⟷ (Simpler) Equivalent Model

Generalization (and training) errors of the two models are equivalent.

- We consider the asymptotically proportional limit:
  - input dimension, number of neurons, and sample size diverge with finite ratios
- Limitations of the literature
  - Lack of feature learning (e.g., random feature model) [1]
  - Limited data assumption (e.g., standard Gaussian inputs) [1,2,3,4]

[1] Hu and Lu. Universality laws for high-dimensional learning with random features. IEEE Trans. Inf. Theory, 69(3):1932–1964, Mar. 2023.
[2] Dandi et al. How two-layer neural networks learn, one (giant) step at a time, arXiv:2305.18270, 2023.
[3] Cui et al. Asymptotics of feature learning in two-layer networks after one gradient-step. ICML 2024.
[4] Moniri et al. A theory of non-linear feature learning with one gradient step in two-layer neural networks. ICML 2024.

Data: Gaussian Mixtures with Structured Covariance

- Our goal: a realistic data assumption

Data: Gaussian Mixtures with Structured Covariance

- Our goal: a realistic data assumption
- Gaussian Mixtures Data
  - Motivation: Real-world data can be better modeled with mixtures instead of single Gaussian.
  - e.g., classification problems

$$\textbf{\textit{Input:}}\ \boldsymbol{x} \sim \sum_{j=1}^{\mathcal{C}} \rho_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\,,\quad \textbf{\textit{Label:}}\ y := \sigma_*\left(\boldsymbol{\xi}^T \boldsymbol{x}, c\right)$$

Data: Gaussian Mixtures with Structured Covariance

- Our goal: a realistic data assumption
- Gaussian Mixtures Data
  - Motivation: Real-world data can be better modeled with mixtures instead of single Gaussian.
  - e.g., classification problems

$$\textbf{\textit{Input:}}\ \boldsymbol{x} \sim \sum_{j=1}^{\mathcal{C}} \rho_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j),\quad \textbf{\textit{Label:}}\ y := \sigma_*\left(\boldsymbol{\xi}^T \boldsymbol{x}, c\right)$$

- Structured Covariance
  - Motivation: Real-world data includes low-dimensional structure.
  - e.g., MNIST and CIFAR10 have intrinsic dimensions of approx. 15 and 35, resp. [5]

$$\boldsymbol{\Sigma}_c = \boldsymbol{I}_n + \underbrace{\sum_{i=1}^{d_c} \theta_{c,i} \boldsymbol{\gamma}_{c,i} \boldsymbol{\gamma}_{c,i}^T}_{\textit{Structure (spikes)}}$$

[5] Spigler et al. Asymptotic learning curves of kernel methods: empirical data versus teacher-student paradigm. Journal of Statistical Mechanics: Theory and Experiment, 2020(12).

Feature Learning by One Gradient Step

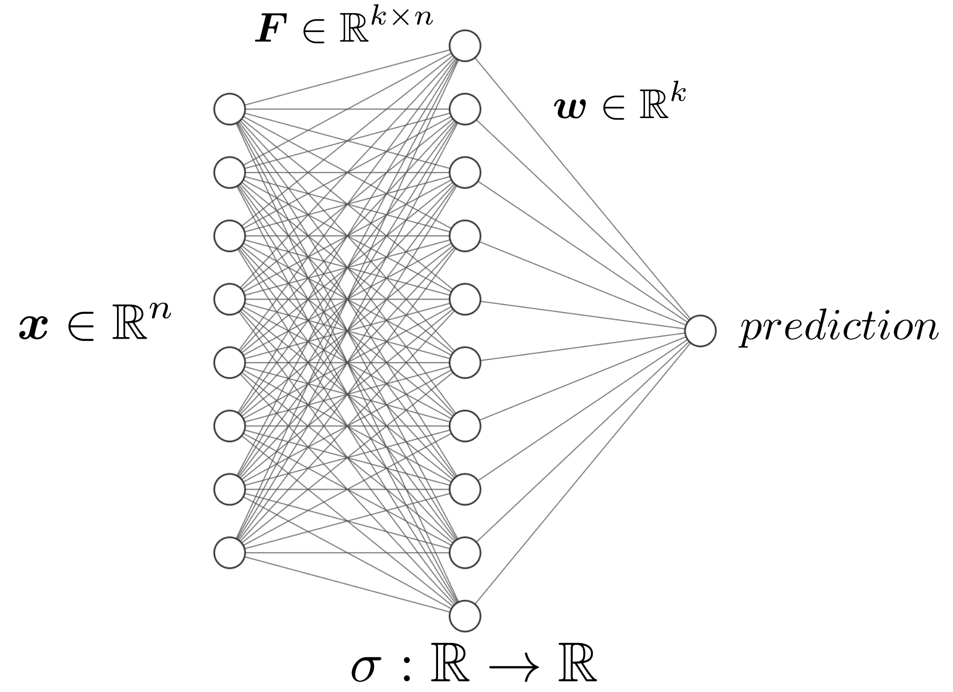
- Consider two-layer NNs for the described learning problem

- To simplify the analysis, train the model with two stages [6]
  - i) Training of the first layer:
    - One step gradient descent

$$\hat{\boldsymbol{F}} := \boldsymbol{F} + \eta \boldsymbol{G}$$

learning rate: $\eta > 0$

gradient matrix: $\boldsymbol{G}$



[6] Ba et al. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. NeurIPS 2022.

Feature Learning by One Gradient Step

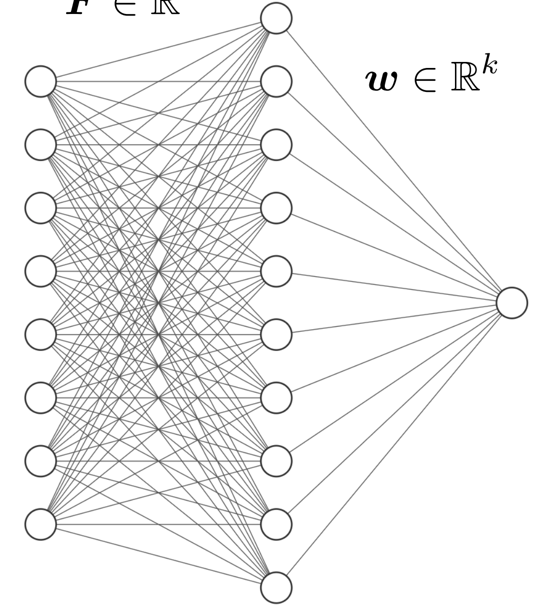
- Consider two-layer NNs for the described learning problem

- To simplify the analysis, train the model with two stages [6]
  - i) Training of the first layer:
    - One step gradient descent

$$\hat{\boldsymbol{F}} := \boldsymbol{F} + \eta \boldsymbol{G}$$

learning rate: $\eta > 0$

gradient matrix: $\boldsymbol{G}$



  - ii) Training of the second layer:
    - Ridge regression $\hat{\boldsymbol{w}} := \underset{\boldsymbol{w} \in \mathbb{R}^k}{\arg\min} \frac{1}{2m} \sum_{i=1}^{m} \left(y_i - \frac{1}{\sqrt{k}} \boldsymbol{w}^T \sigma(\hat{\boldsymbol{F}} \boldsymbol{x}_i) \right)^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|^2$

[6] Ba et al. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. NeurIPS 2022.

Feature Learning by One Gradient Step

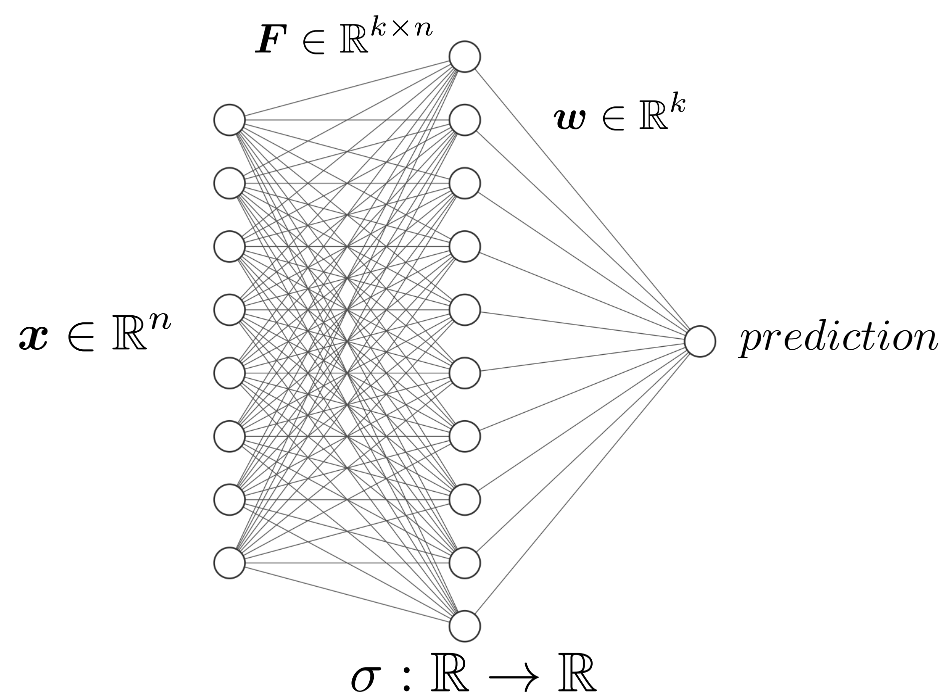
- Consider two-layer NNs for the described learning problem

- To simplify the analysis, train the model with two stages [6]
  - i) Training of the first layer:
    - One step gradient descent

$$\hat{\boldsymbol{F}} := \boldsymbol{F} + \eta \boldsymbol{G}$$

learning rate: $\eta > 0$

gradient matrix: $\boldsymbol{G}$



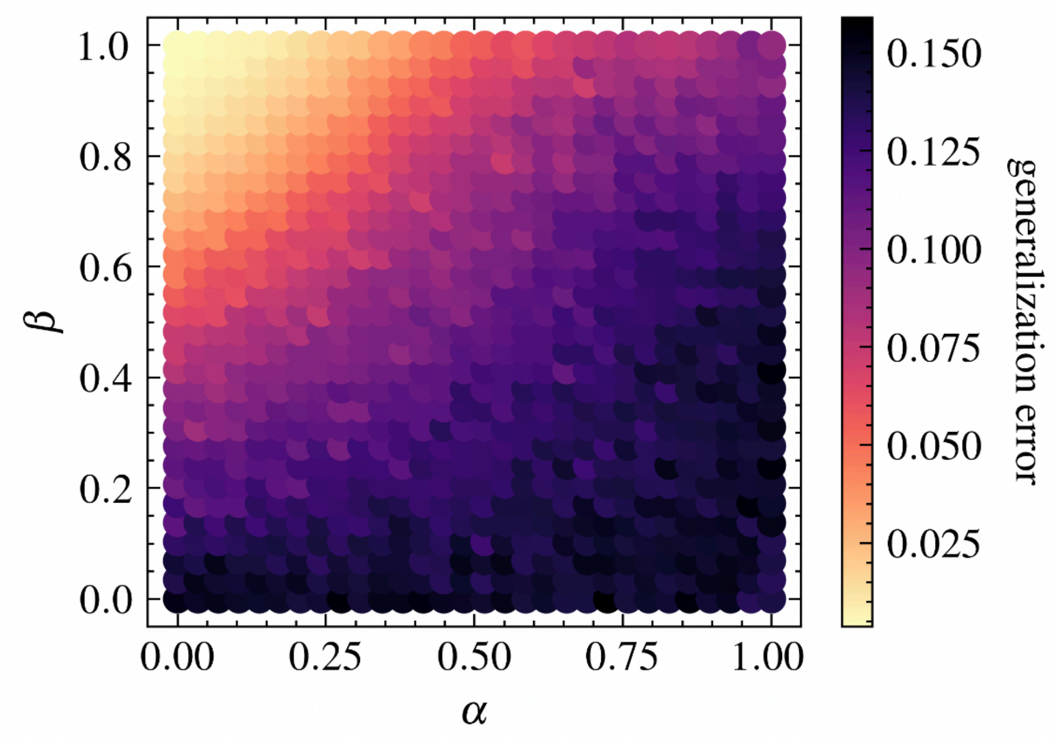
  - ii) Training of the second layer:
    - Ridge regression $\hat{\boldsymbol{w}} := \arg\min_{\boldsymbol{w} \in \mathbb{R}^k} \frac{1}{2m} \sum_{i=1}^{m} \left(y_i - \frac{1}{\sqrt{k}} \boldsymbol{w}^T \sigma(\hat{\boldsymbol{F}} \boldsymbol{x}_i) \right)^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|^2$

- Generalization error: $\mathbb{E}_{(\boldsymbol{x}, y)} \left[\left(y - \frac{1}{\sqrt{k}} \hat{\boldsymbol{w}}^T \sigma(\hat{\boldsymbol{F}} \boldsymbol{x}) \right)^2 \right]$

[6] Ba et al. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. NeurIPS 2022.

Scalings: data spread and learning rate

- Scalings of learning rate η and data spread $\|\mathbf{\Sigma}\|$ shape the generalization
- To control these, we define
  - A "strength parameter" $\beta \in [0, 1]$ governing joint scaling: $\eta\|\mathbf{\Sigma}\| \asymp n^{\beta}$
  - A "weighting parameter" $\alpha \in [0, 1]$ controlling individual scalings:
    - data spread $\|\mathbf{\Sigma}\| \asymp n^{\beta(1-\alpha)}$ and learning rate $\eta \asymp n^{\beta\alpha}$



Theorem: Conditional Gaussian Equivalence

- *(Informal Statement) The following two feature maps are equivalent in terms of generalization (and training) errors:*

  - *Original feature map* $\phi(\boldsymbol{x}) := \sigma(\hat{\boldsymbol{F}}\boldsymbol{x})$

  - *A conditional feature map defined as*

$$\hat{\phi}(\boldsymbol{x}; c, \boldsymbol{\kappa}_c) := \underbrace{\boldsymbol{\nu}(c, \boldsymbol{\kappa}_c)}_{\text{mean}} + \underbrace{\boldsymbol{\Psi}(c, \boldsymbol{\kappa}_c)\boldsymbol{z}^\perp}_{\text{cross-covariance}} + \underbrace{\boldsymbol{\Phi}(c, \boldsymbol{\kappa}_c)^{1/2}\boldsymbol{g}}_{\text{remaining covariance}} \quad with \quad \boldsymbol{g} \sim \mathcal{N}(0, \boldsymbol{I}_n) \quad and \quad \boldsymbol{z}^\perp \; derived\ from\ \boldsymbol{x}$$

This is an equivalent conditional Gaussian feature map,
<u>conditioned on</u>
(i) c **: mixture component index,**

(ii) κ_c **: alignment of the input sample with the spikes in data covariance and gradient.**

Theorem: Equivalent Hermite (Polynomial) Model

- *(Informal Statement) If strength parameter satisfies* $\frac{l-2}{l-1} < \beta < \frac{l-1}{l}$ *, then, we can replace original activation function with the following one without any change on the generalization (and training) performance.*

$$\hat{\sigma}_l(x) := \left(\sum_{j=0}^{l-1} \frac{1}{j!} h_j H_j(x/b) \right) + h_l^* z \quad with \quad z \sim \mathcal{N}(0,1)$$

(finite-order) Hermite expansion ***remainder term to match the variance***

Note: $H_j : \mathbb{R} \to \mathbb{R}$ ***denotes j-th (probabilist's) Hermite polynomial.***

Theorem: Equivalent Hermite (Polynomial) Model

- *(Informal Statement) If strength parameter satisfies* $\frac{l-2}{l-1} < \beta < \frac{l-1}{l}$ *, then, we can replace original activation function with the following one without any change on the generalization (and training) performance.*

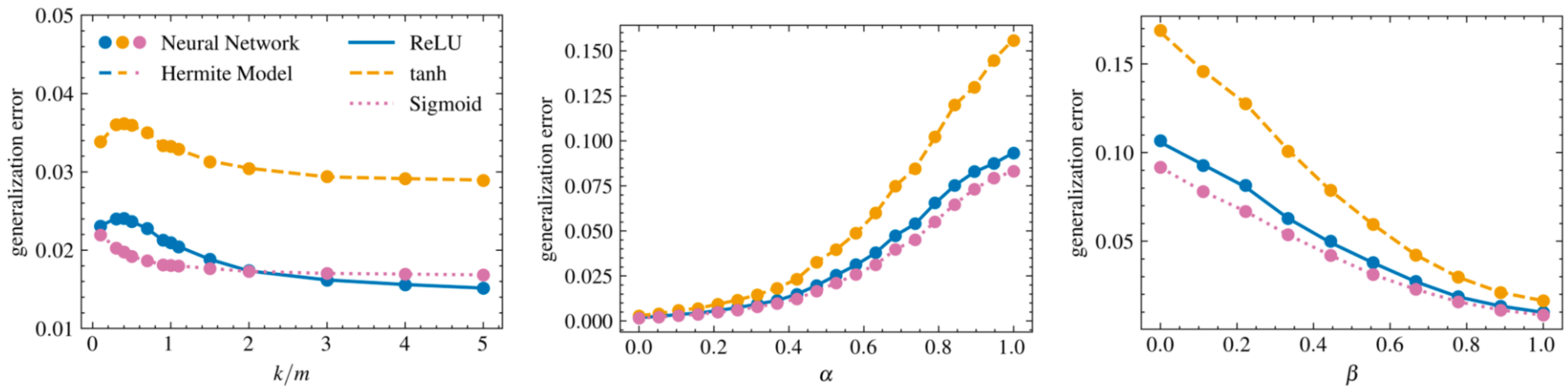
$$\hat{\sigma}_l(x) := \left(\sum_{j=0}^{l-1} \frac{1}{j!} h_j H_j(x/b)\right) + h_l^* z \quad \textit{with} \quad z \sim \mathcal{N}(0,1)$$

(finite-order) Hermite expansion ***remainder term to match the variance***

Note: $H_j : \mathbb{R} \to \mathbb{R}$ ***denotes j-th (probabilist's) Hermite polynomial.***

- Equivalent "Hermite Model": $\frac{1}{\sqrt{k}} \boldsymbol{w}^T \hat{\sigma}_l(\hat{\boldsymbol{F}} \boldsymbol{x})$

Simulation: Impacts of Data Spread and Learning Rate



(a) Impact of k/m ($\beta = 3/4^-$ and $\alpha = 1/2$)

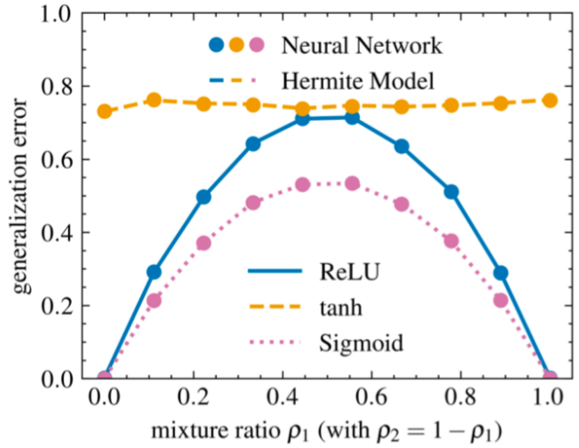
(b) Impact of α ($\beta = 3/4^-$ and $k/m = 1$)

(c) Impact of β ($\alpha = 1/2$ and $k/m = 1$)

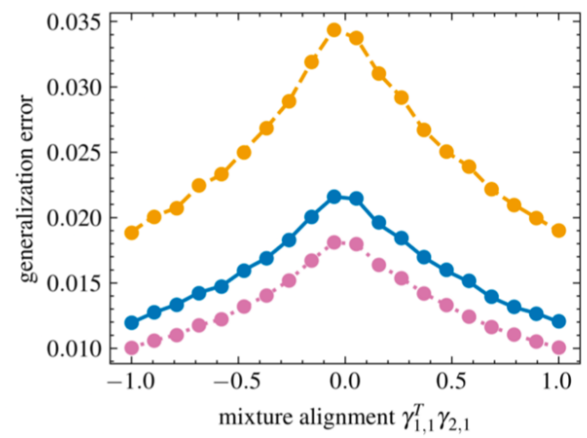
- Reminder: data spread $\|\boldsymbol{\Sigma}\| \asymp n^{\beta(1-\alpha)}$ and learning rate $\eta \asymp n^{\beta\alpha}$

- (a, b, c) The generalization errors of the NNs and the Hermite model closely align.
- (b) High data spread leads to better performance compared to high learning rate.
- (c) Larger strength parameter results in improved generalization in general.

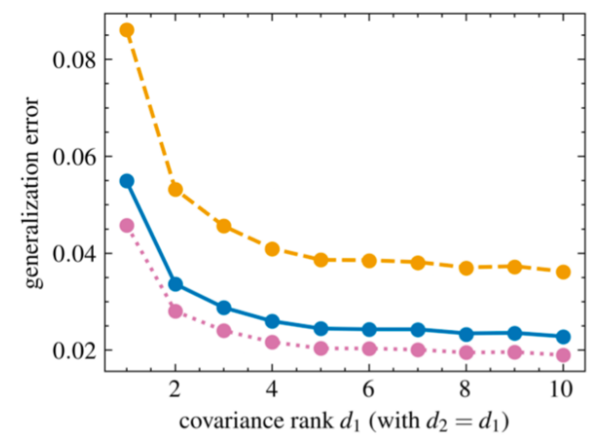
Simulation: Impacts of Mixture Properties



(a) Impact of mixture ratio for classification



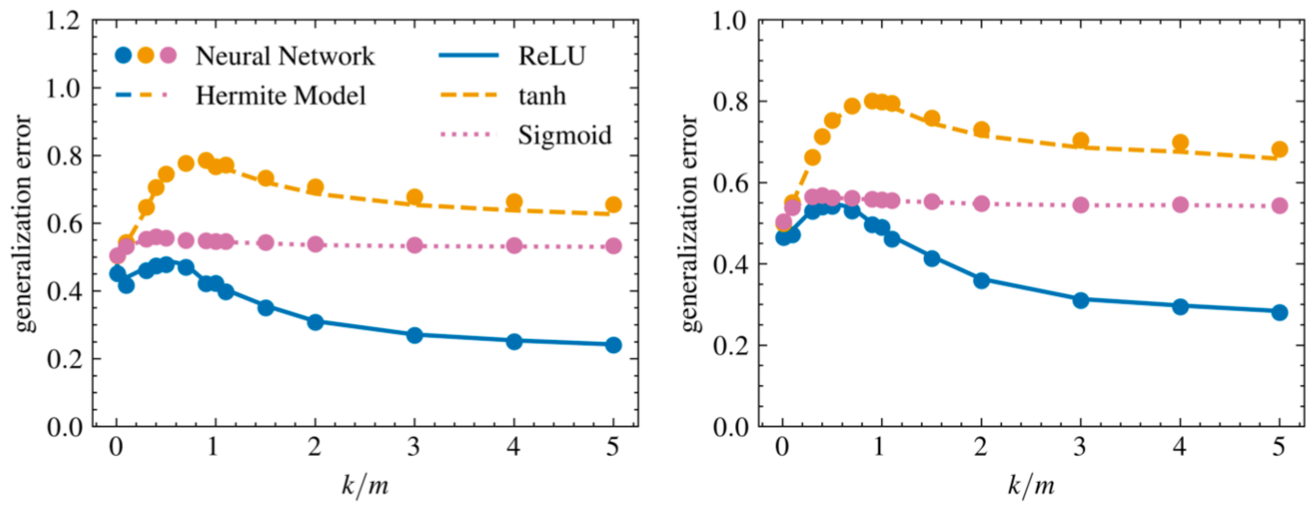
(b) Impact of mixture alignment for regression



(c) Impact of covariance rank for regression

- The generalization errors of the NNs and the Hermite model closely align.
- Mixture properties significantly affect the generalization errors.

Real Data: Fashion-MNIST Classification



(a) Class 0 vs. Class 1

(T-shirt/top vs. Trouser)

(b) Class 5 vs. Class 9

(Sandal vs. Ankle boot)

- The generalization errors of the NNs and the Hermite model closely align.

Summary

- **Takeaway**: Data distribution (data spread) impacts the generalization performance of neural networks together with the feature learning.

- Under Gaussian Mixtures data assumption and with feature learning via one gradient step, we found simpler models equivalent to two-layer NNs:
  - A conditional Gaussian model,
  - A polynomial model formed by Hermite polynomials.