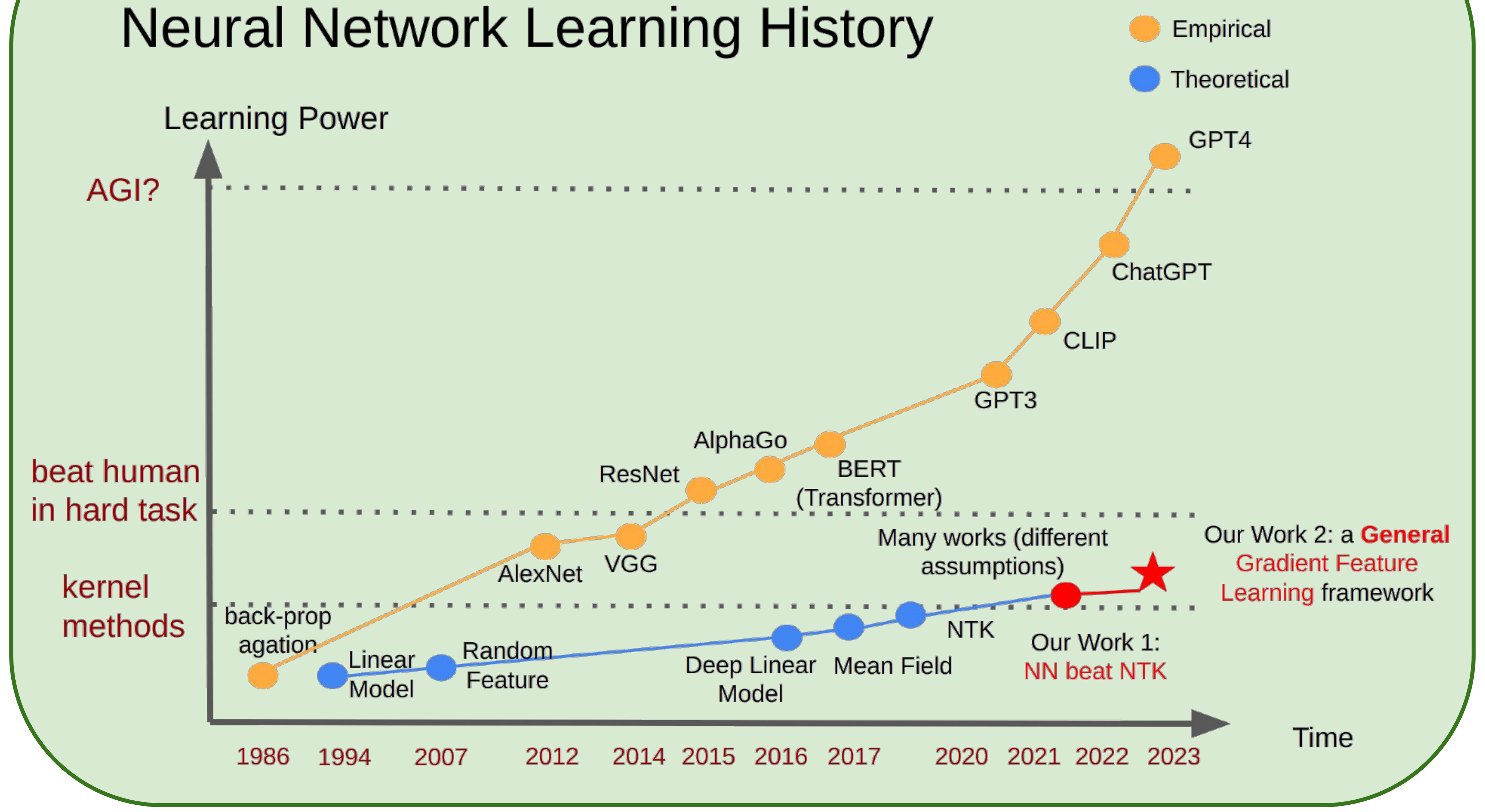
Provable Guarantees for Neural Networks via Gradient Feature Learning

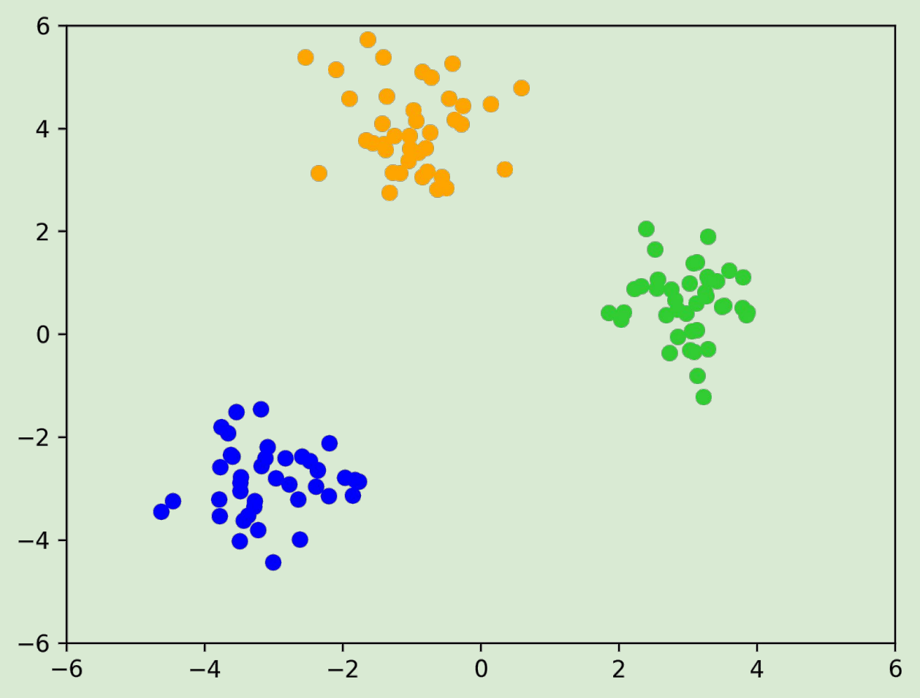
Zhenmei Shi*, Junyi Wei*, Yingyu Liang

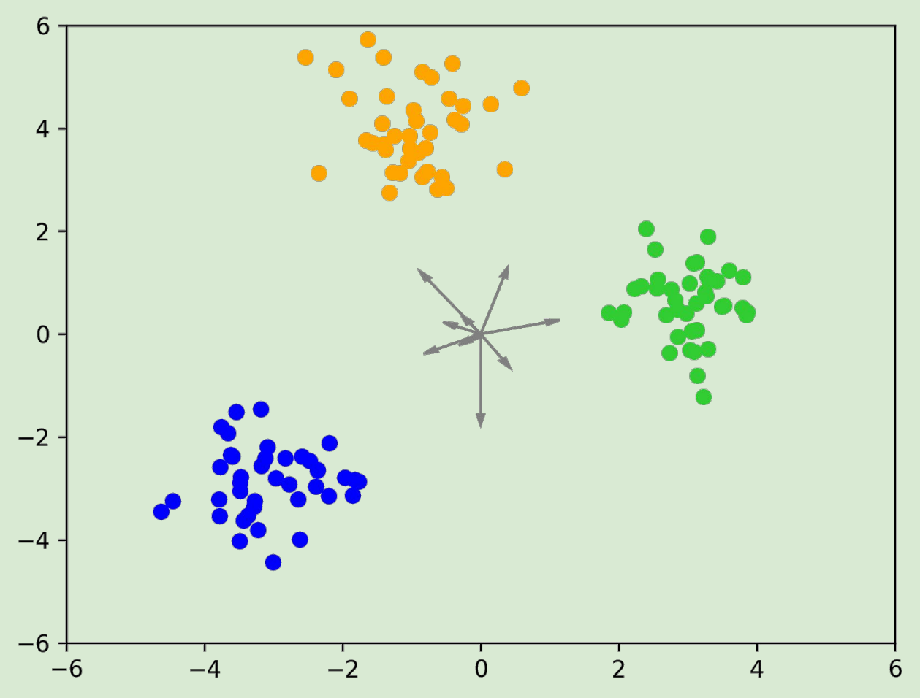
Background

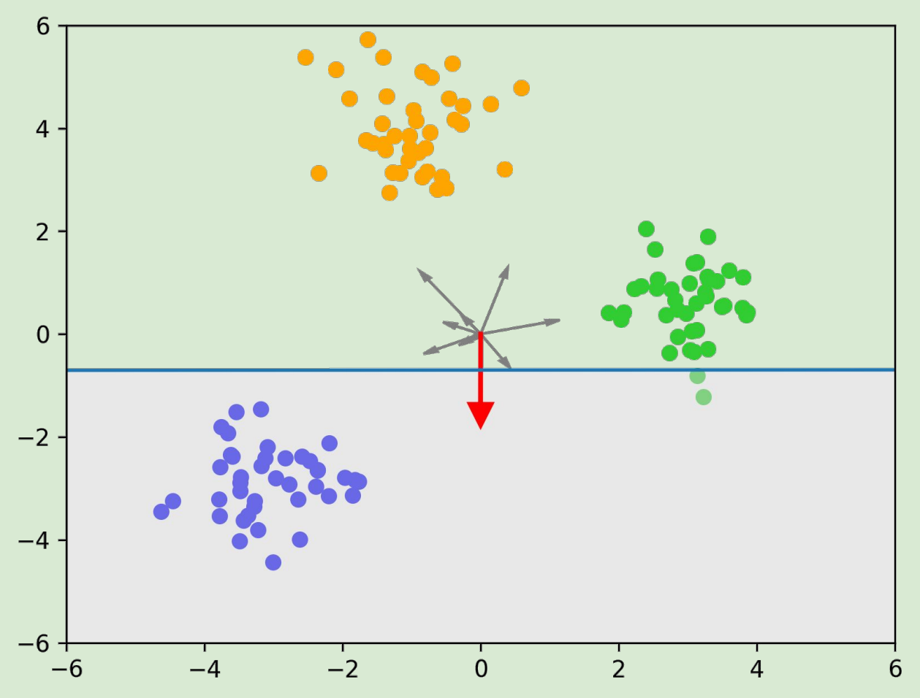
Neural Network Learning History

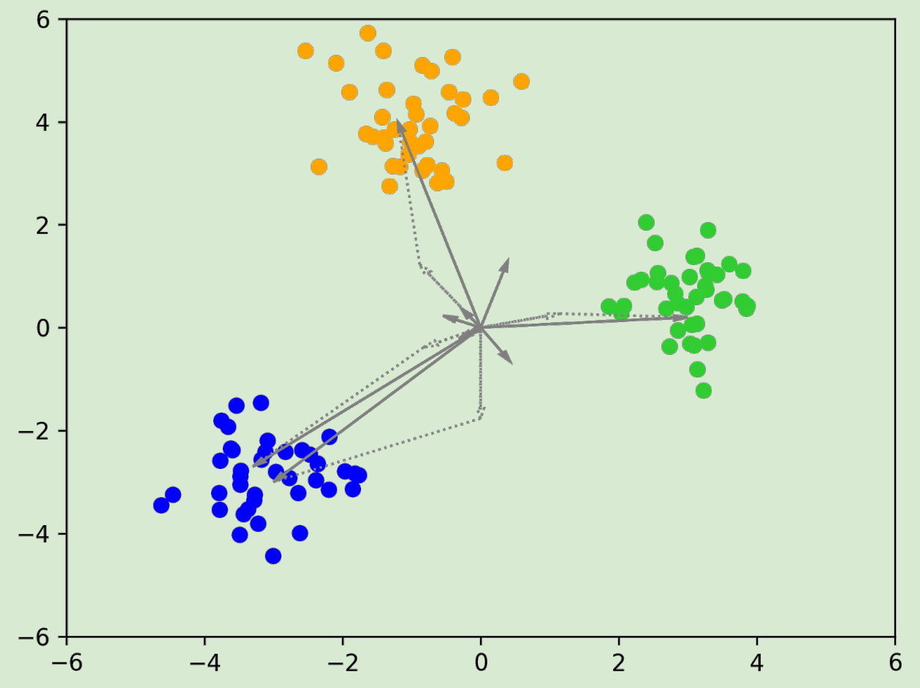


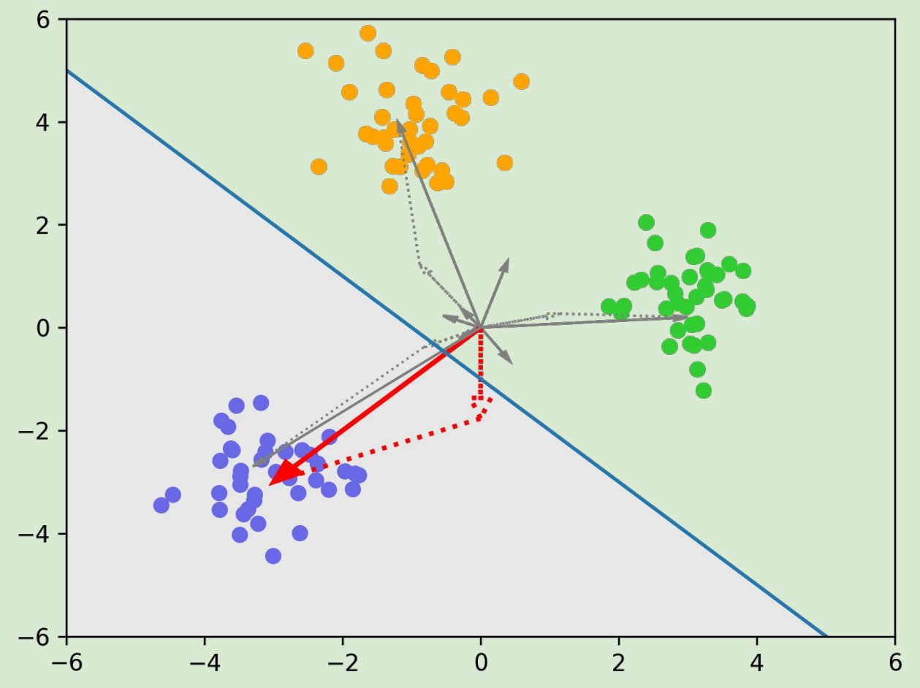
Motivation


Three Gaussian data clusters.


With Random init weight vectors.

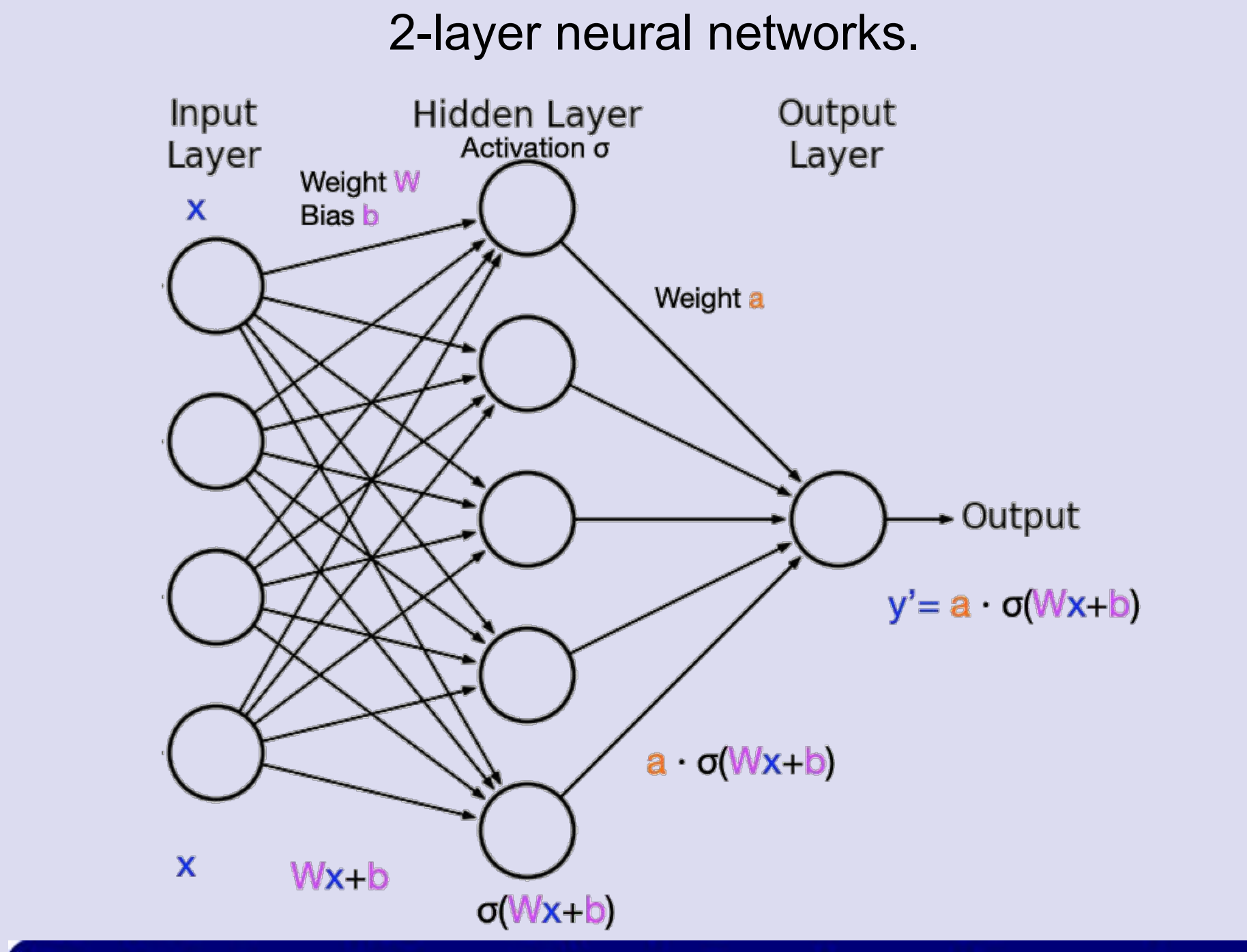

Activation under one neuron.

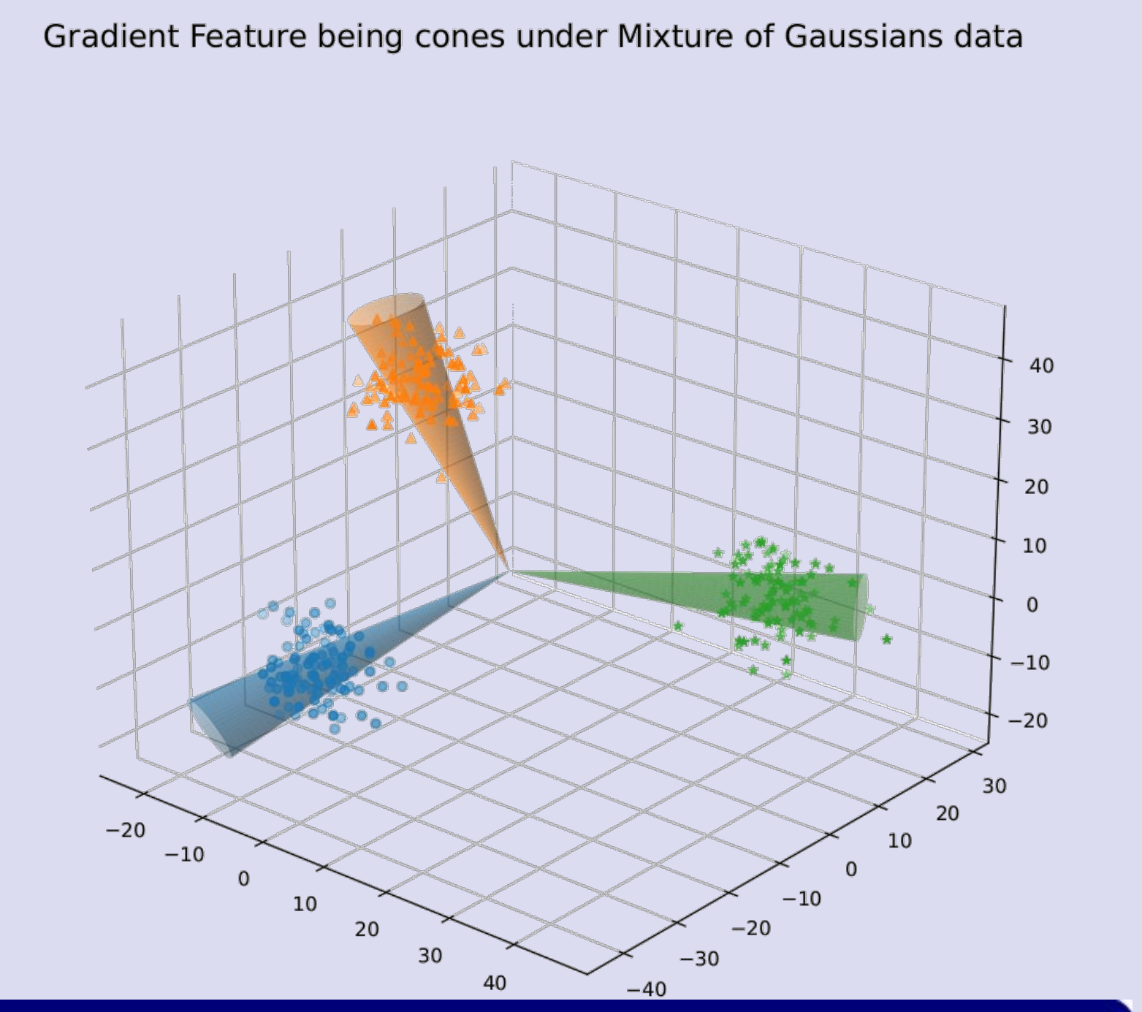

Weight after one gradient step.


Updated activation pattern.

One step gradient update is good for feature learning under mixture of Gaussians: **Gradient Feature Learning Framework**

Core Concept





Definition (Simplified Gradient Vector)

For any $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, a Simplified Gradient Vector is defined as

$$G(\mathbf{w}, b) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[y\mathbf{x}\mathbb{I}[\mathbf{w}^\top \mathbf{x} > b]]. \tag{2}$$

Definition (Gradient Feature)

For a unit vector $D \in \mathbb{R}^d$ with $\|D\|_2 = 1$, and a $\gamma \in (0,1)$. Let $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ be random variables drawn from some distribution $\mathcal{W}, \mathcal{B}$. A Gradient Feature set with parameters p, γ, B_G is defined as:

$$S_{p,\gamma,B_G}(\mathcal{W},\mathcal{B}) := \left\{ (D, s) \;\middle|\; \Pr_{\mathbf{w},b}\left[\frac{|\langle G(\mathbf{w},b), D\rangle|}{\|G(\mathbf{w},b)\|_2} > (1-\gamma), \right.\right. \tag{3}$$

$$\left.\left. \|G(\mathbf{w},b)\|_2 \geq B_G,\ s = \frac{b}{|b|} \right] \geq p \right\}. \tag{4}$$

Definition (Gradient Feature Induced Networks)

The Gradient Feature Induced Networks are defined as:

$$\mathcal{F}_{d,m,B_F,S} := \left\{ f_{(\mathbf{a},\mathbf{W},\mathbf{b})} \in \mathcal{F}_{d,m} \;\middle|\; \forall i \in [m], ..., \left(\mathbf{w}_i, \frac{\mathbf{b}_i}{|\mathbf{b}_i|}\right) \in S,\ |\mathbf{b}_i| \leq B_b \right\}, \tag{5}$$

Definition (Optimal Approximation via Gradient Features)

The Optimal Approximation network and loss using gradient feature induced networks $\mathcal{F}_{d,r,B_F,S}$ are defined as:

$$f^* := \text{argmin}_{f \in \mathcal{F}_{d,r,B_F,S}} L_{\mathcal{D}}(f), \qquad \text{OPT}_{d,r,B_F,S} := \min_{f \in \mathcal{F}_{d,r,B_F,S}} L_{\mathcal{D}}(f). \tag{6}$$

Main Results

Theorem

Assume $\mathbb{E}[\|\mathbf{x}\|_2] \leq B_{x1}$, $\mathbb{E}[\|\mathbf{x}\|_2^2] \leq B_{x2}$. *For any* $\epsilon, \delta \in (0,1)$, *if* $m \leq e^d$ *and* $m = \Omega\left(\textit{poly}\left(\frac{1}{p}, \frac{1}{\epsilon}, \frac{1}{\delta}, B_{a1}, B_{x1}, \log(r)\right)\right)$, $T = \Omega\left(\textit{poly}\left(m, \frac{1}{p}, \frac{1}{\epsilon}, \frac{1}{\delta}, B_{a2}, B_{x1}, \sqrt{r}\right)\right)$, *then with proper hyper-parameter values, we have with probability* $\geq 1-\delta$, *there exists* $t \in [T]$ *with*

$$\Pr[\text{sign}(f_{\Xi^{(t)}}(\mathbf{x})) \neq y] \leq L_{\mathcal{D}}(f_{\Xi^{(t)}}) \leq OPT_{d,r,B_F,S_{p,\gamma,B_G}} + rB_{a1}B_{x1}\sqrt{2\gamma} + \epsilon.$$

- It shows how neural network converge to almost best solution given learned gradient features.
- After the first step, both layers continue to learn with the same learning rate, but second layer weights grow while the first layer weights stay in a neighborhood.

Applications and Implications

Apply to four case studies by our Gradient Feature Learning Framework directly:

(1) mixtures of Gaussians, (2) parity functions, (3) linear data, (4) multiple-index models.

Beyond the Kernel Regime:

- There exists a data distribution in the parity learning that (1) any fixed feature methods (including NTK) needs exponentially large size to learn successful; (2) Gradient Feature only needs polynomially large model, runtime, and sample complexity to learn successful.
- There exists a data distribution in the mixtures of Gaussians that (1) any fixed feature methods (including NTK) needs Ω(d^2) features and Ω(d^2) samples to learn successful; (2) Gradient Feature only needs Ω(log d) neurons and Ω((log d)^2) samples to learn successful.

Lottery Ticket Hypothesis (LTH):

- Show the existence of the winning lottery subnetwork.
- Show subnetwork can learn to similar loss in similar runtime as the whole network (novel).

Implicit Regularization / Simplicity Bias:

- Networks first learn simpler functions and then more sophisticated ones.

Learning over Different Data Distributions:

- Data-dependent non-vacuous guarantees to measure the "complexity" of the problem. For easier problems, this quantity is smaller, give a better error bound to derive guarantees.

New Perspectives about Roadmaps Forward:

- Our framework: the strong representation power of NN is the key to successful learning.
- Traditional ones: strong representation power leads to vacuous generalization bounds.
- Traditional analysis typically first reasons about the optimal based on the whole function class then analyzes how NN learns proper features and reaches the optimal. In contrast, our framework defines feature family first, and then reasons about the optimal based on it.