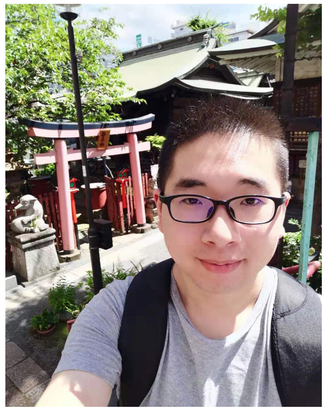
Understanding Adaptive Gradient Methods in Training Neural Networks: Risk Bounds and Practical Implications

Yuan Cao

Department of Statistics and Actuarial Science
University of Hong Kong

The 9th International Forum on Statistics

Collaborators



Zixiang Chen

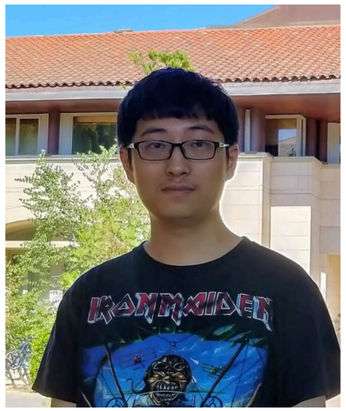
Mikhail Belkin

Quanquan Gu

Difan Zou

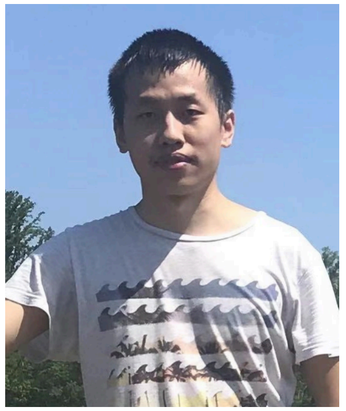
Yuanzhi Li

Jinghui Chen



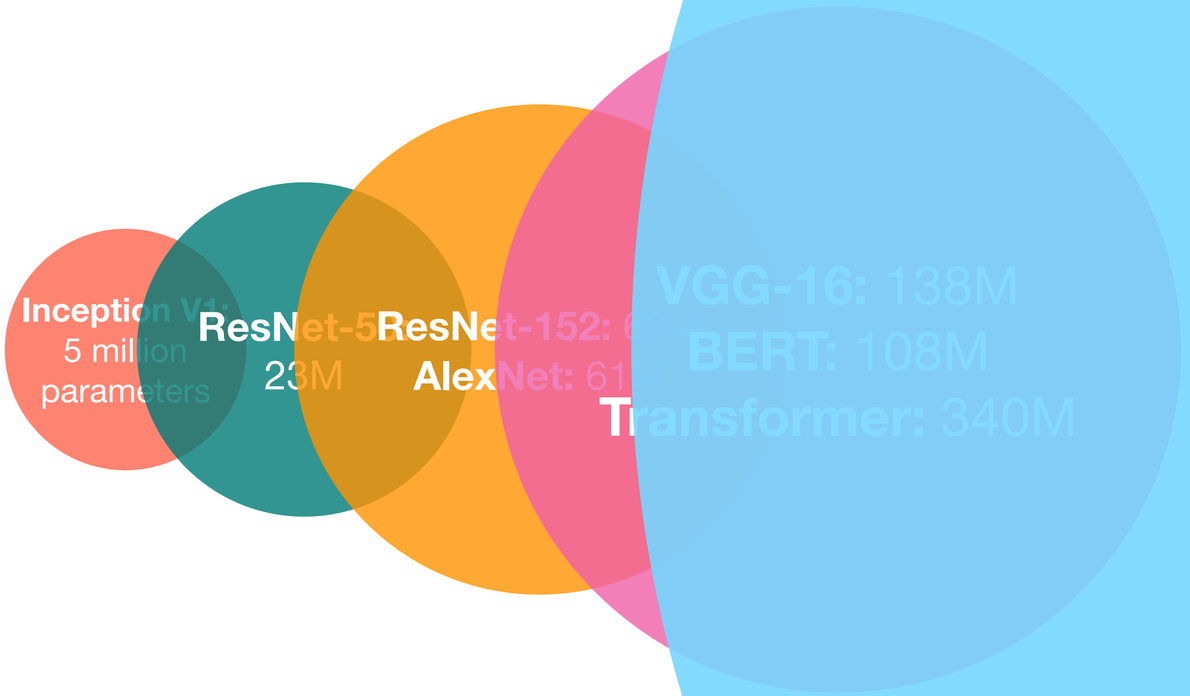
Dongruo Zhou

Ziyan Yang



Yiqi Tang

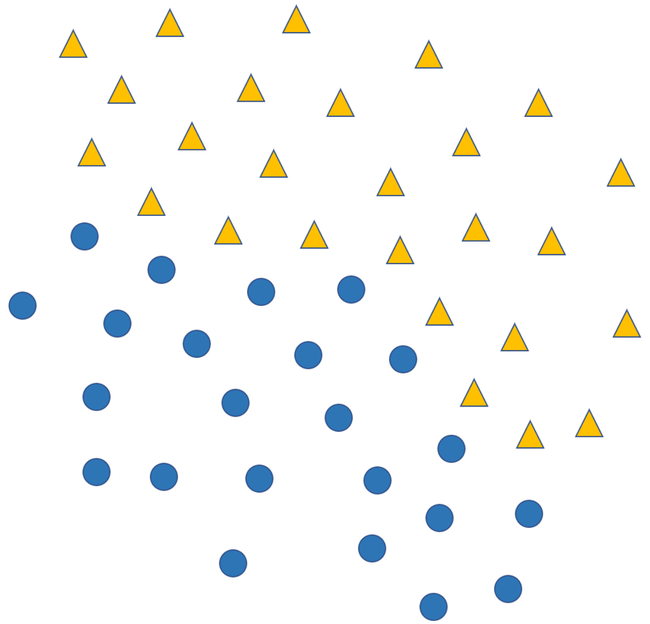
Modern Neural Networks are Over-parameterized



GPT-3:
175 billion parameters

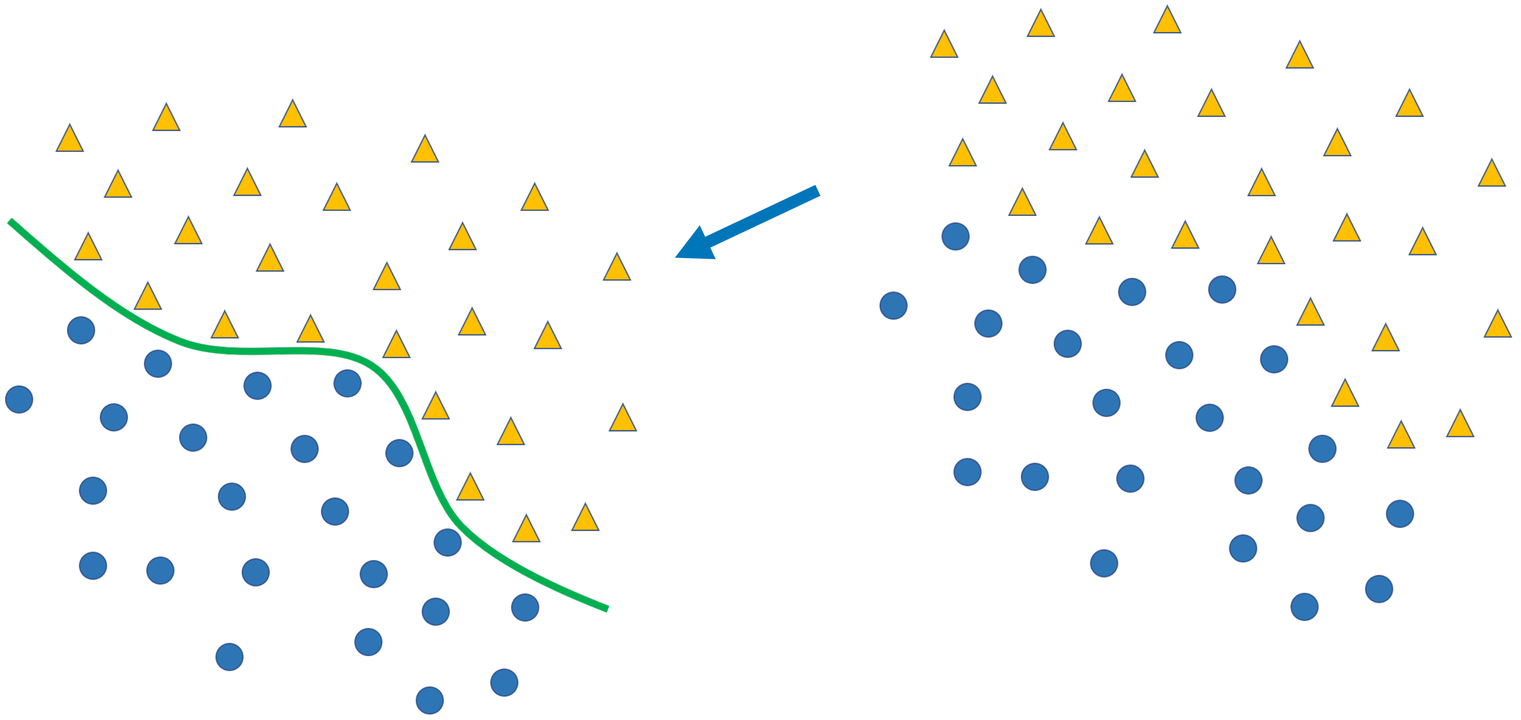
Optimization Matters for Achieving Good Generalization

When learning over-parameterized models, there can be infinitely many predictors that can perfectly fit the limited number of training data.



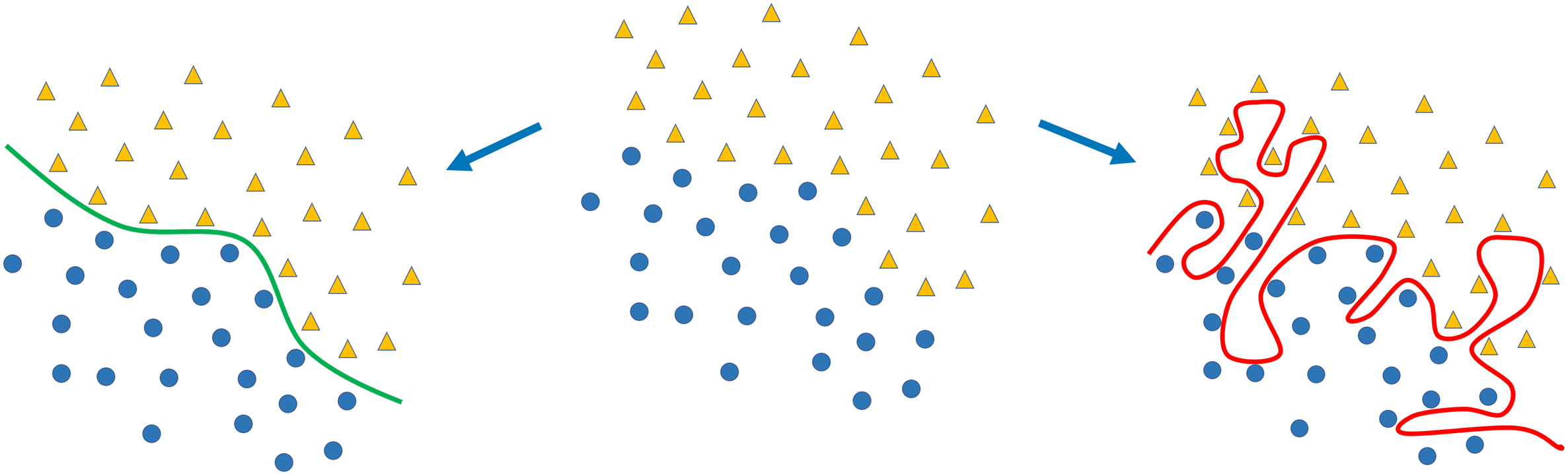
Optimization Matters for Achieving Good Generalization

When learning over-parameterized models, there can be infinitely many predictors that can perfectly fit the limited number of training data.



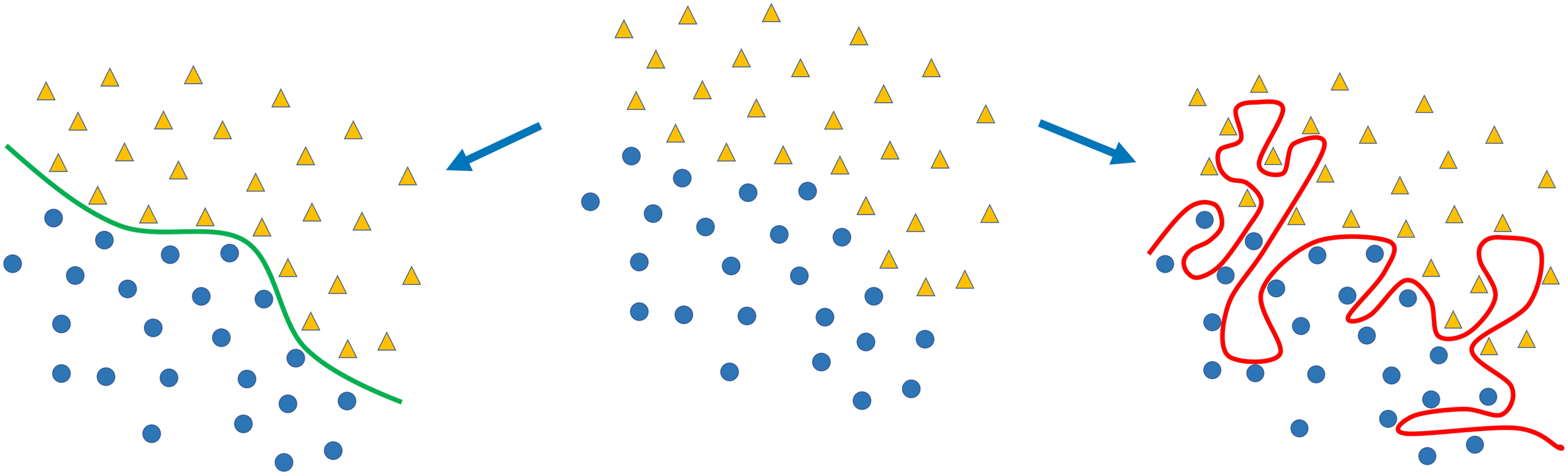
Optimization Matters for Achieving Good Generalization

When learning over-parameterized models, there can be infinitely many predictors that can perfectly fit the limited number of training data.



Optimization Matters for Achieving Good Generalization

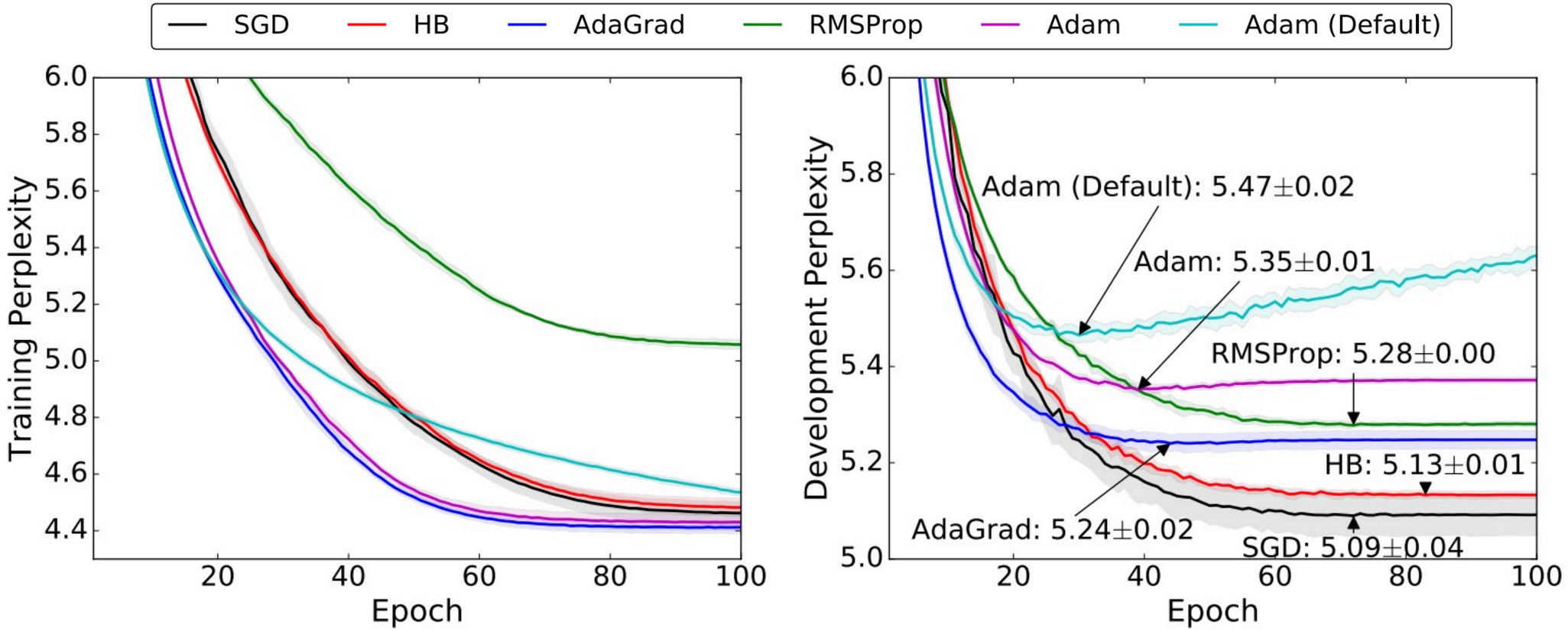
When learning over-parameterized models, there can be infinitely many predictors that can perfectly fit the limited number of training data.



We need to study the impact of the learning algorithm: Which predictor will the training algorithm actually converge to, among the infinitely many minimizers of the training loss?

Adam vs. Gradient Descent

Adaptive gradient methods can converge faster than SGD, but sometimes generalize worse for largely over-parameterized problems [Wilson et al.,2017]



Wilson, Ashia C., Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. "The marginal value of adaptive gradient methods in machine learning." In *Advances in neural information processing systems*, pp. 4148-4158. 2017.

Outline

- A simple statistical learning task and theoretical guarantees for learning two-layer CNNs with GD

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. "Benign Overfitting in Two-layer Convolutional Neural Networks." In Advances in Neural Information Processing Systems, 2022.

Outline

- A simple statistical learning task and theoretical guarantees for learning two-layer CNNs with GD

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. "Benign Overfitting in Two-layer Convolutional Neural Networks." In Advances in Neural Information Processing Systems, 2022.

- The generalization gap between Adam and GD

Difan Zou, Yuan Cao, Yuanzhi Li and Quanquan Gu, "Understanding the Generalization of Adam in Learning Neural Networks with Proper Regularization.” in International Conference on Learning Representations, 2023

Outline

- A simple statistical learning task and theoretical guarantees for learning two-layer CNNs with GD

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. "Benign Overfitting in Two-layer Convolutional Neural Networks." In Advances in Neural Information Processing Systems, 2022.

- The generalization gap between Adam and GD

Difan Zou, Yuan Cao, Yuanzhi Li and Quanquan Gu, "Understanding the Generalization of Adam in Learning Neural Networks with Proper Regularization." in International Conference on Learning Representations, 2023

- Practical implications: Padam and beyond

Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao and Quanquan Gu, "Closing the Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks." in International Joint Conference on Artificial Intelligence 2020

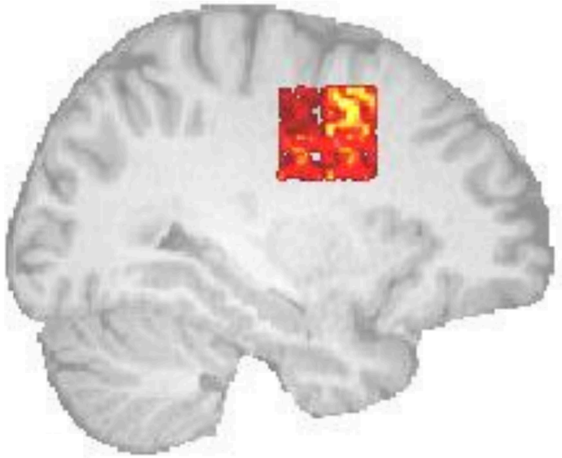
A simple statistical learning task and theoretical guarantees for learning two-layer CNNs with GD

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]

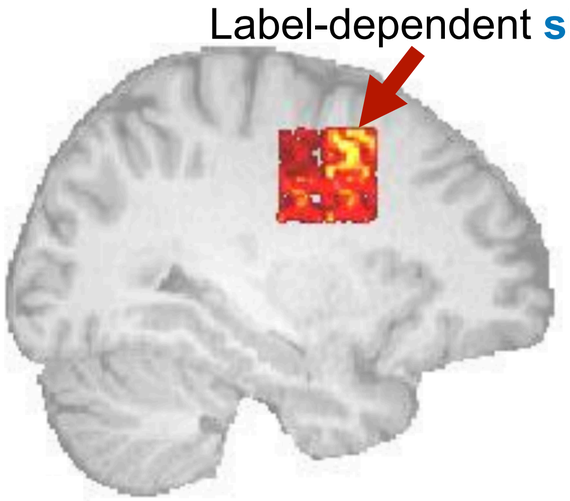
A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]



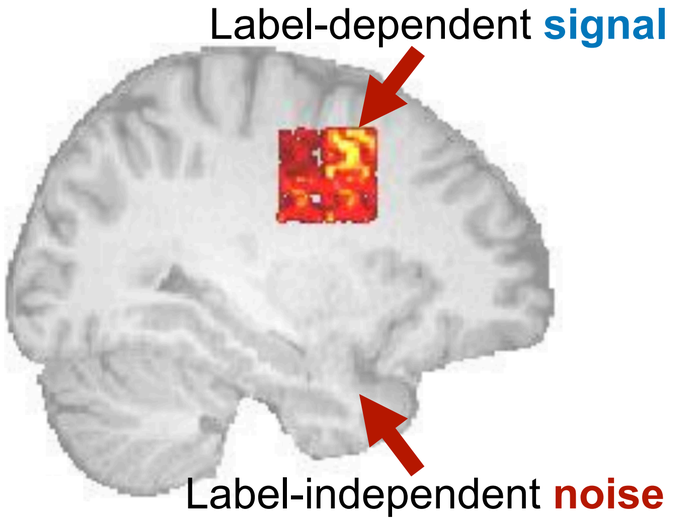
A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]



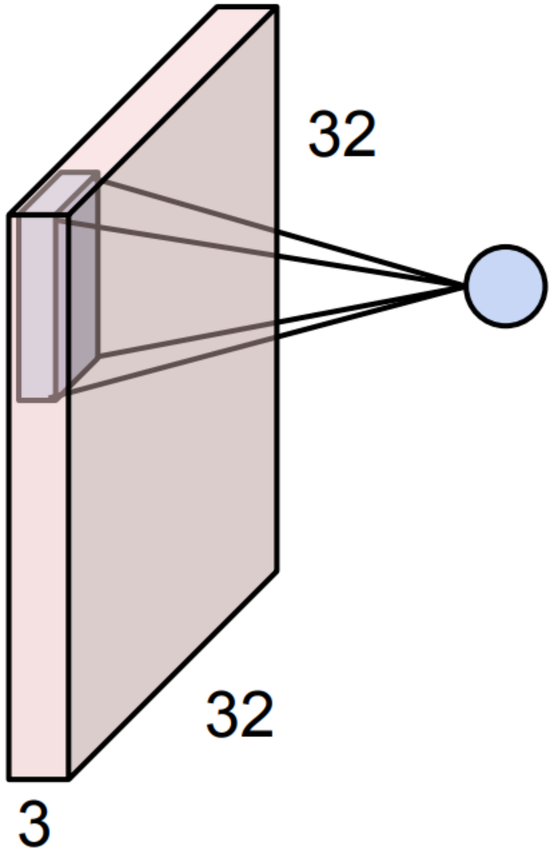
A Simple Statistical Learning Task

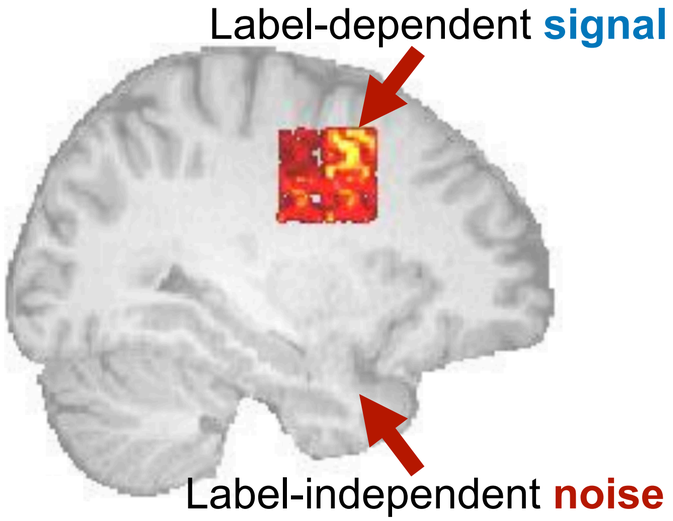
In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]



A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]

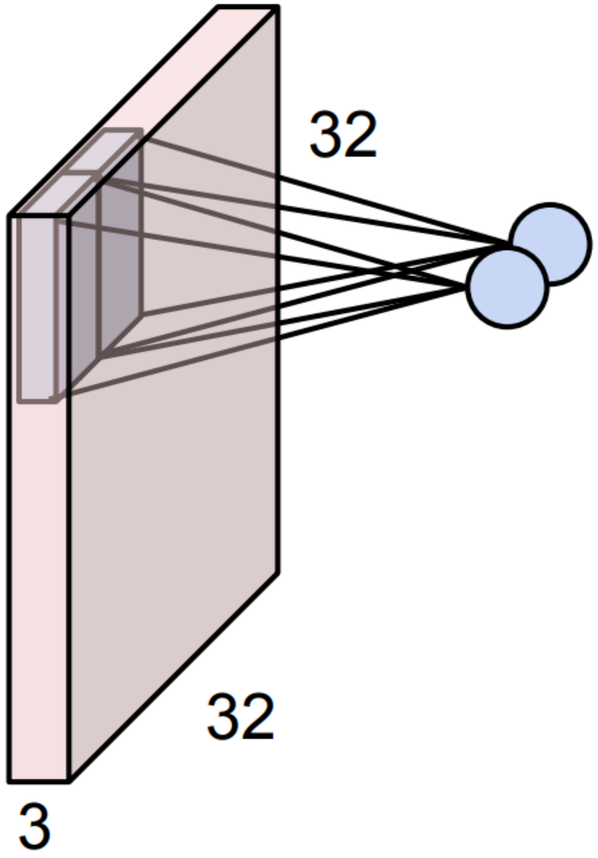


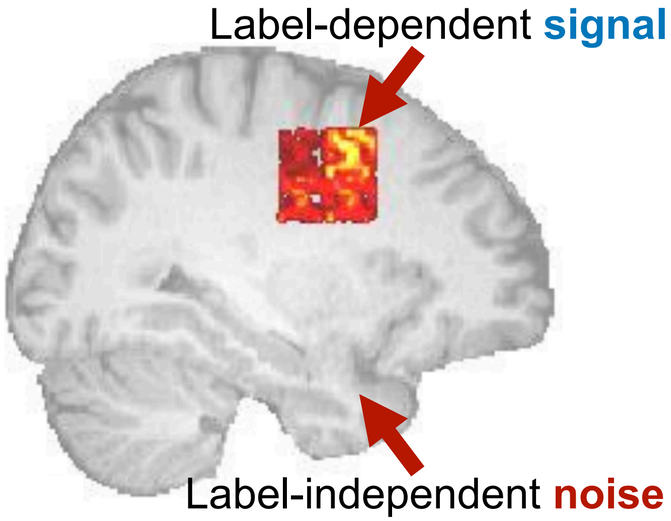


In CNNs, we use convolution filters to extract information from different patches of an image

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]

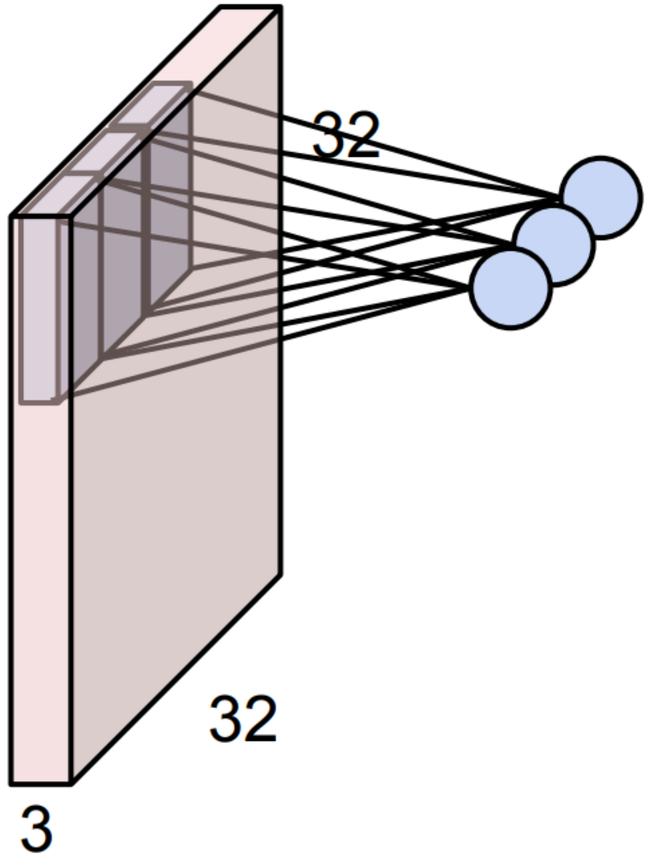


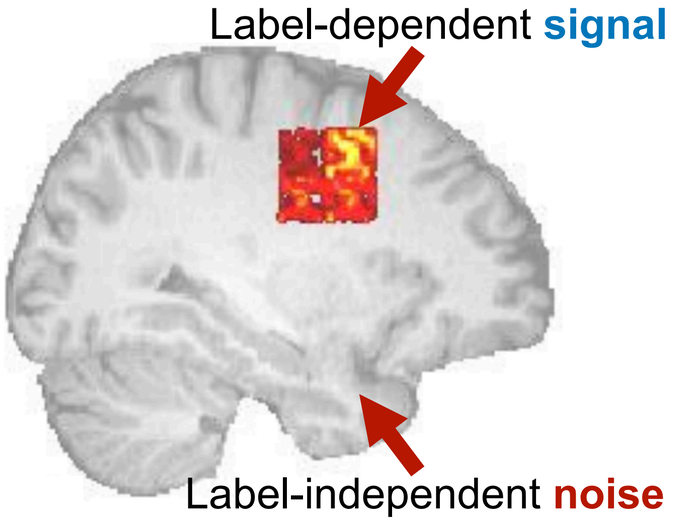


In CNNs, we use convolution filters to extract information from different patches of an image

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]

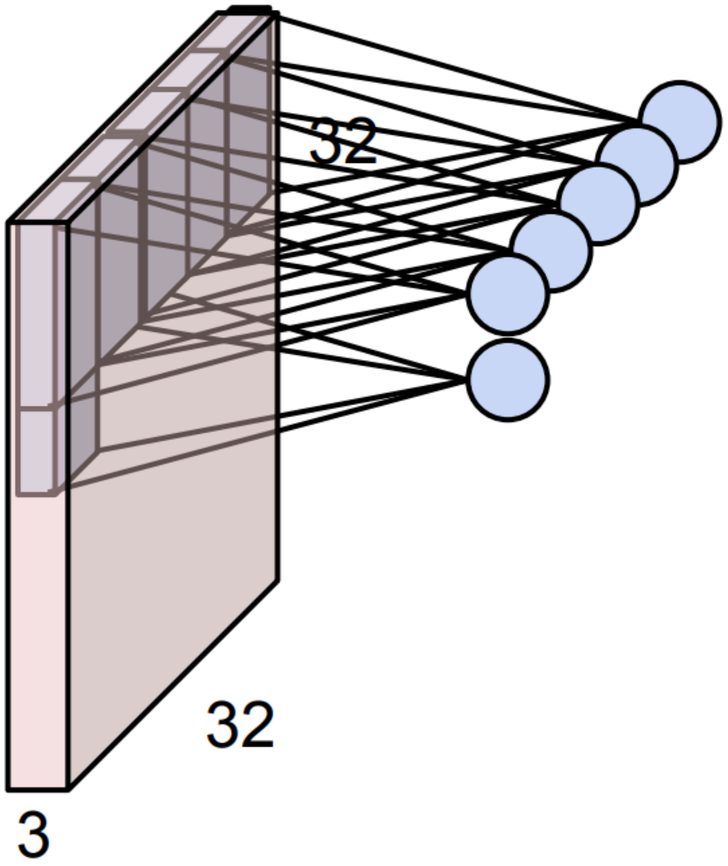


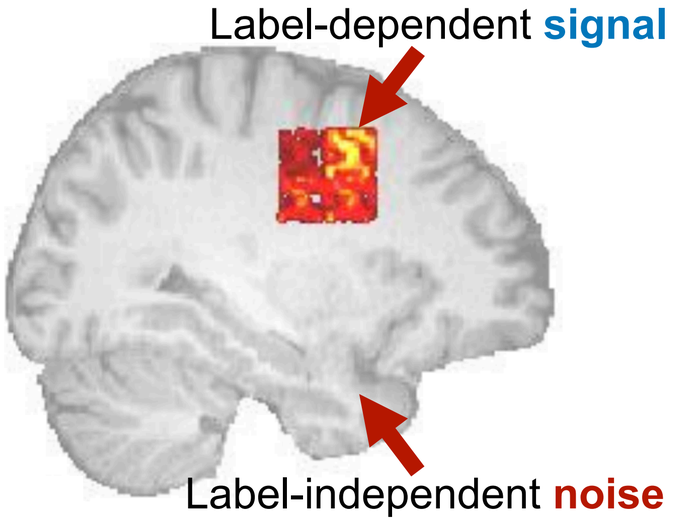


In CNNs, we use convolution filters to extract information from different patches of an image

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]

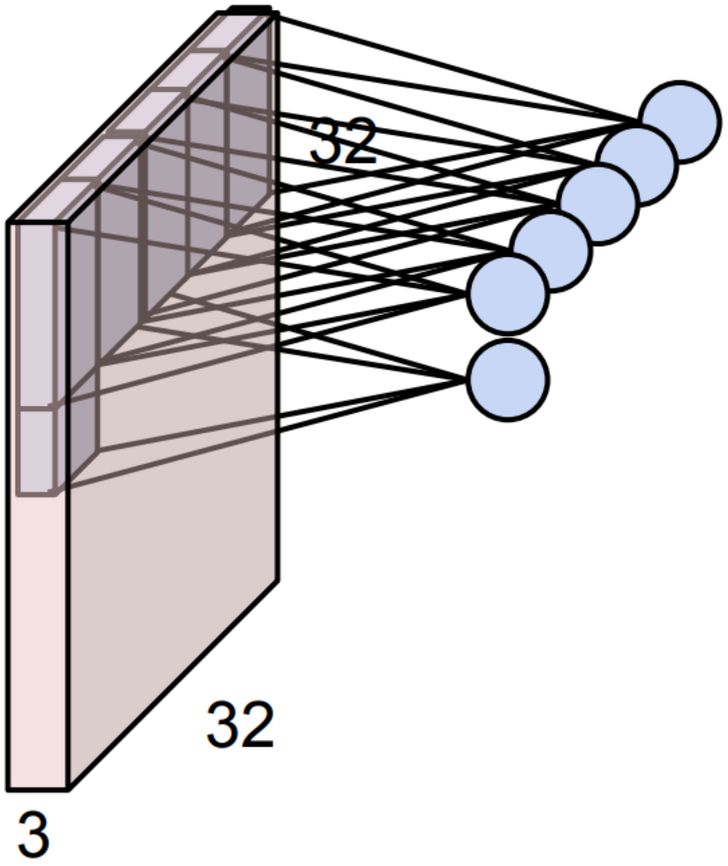


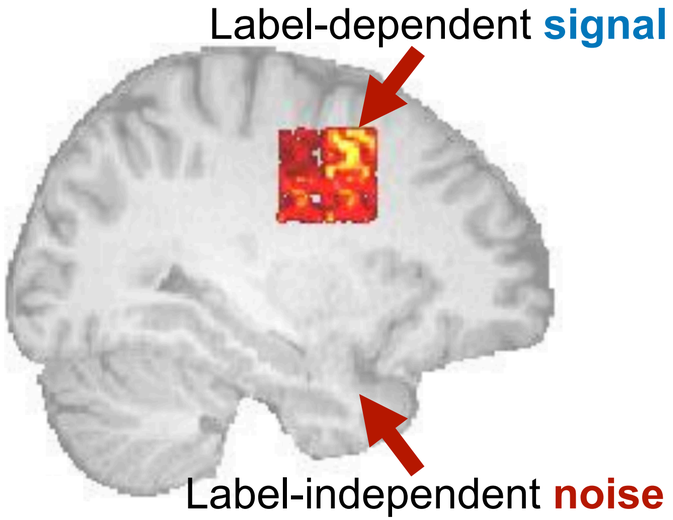


In CNNs, we use convolution filters to extract information from different patches of an image

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]





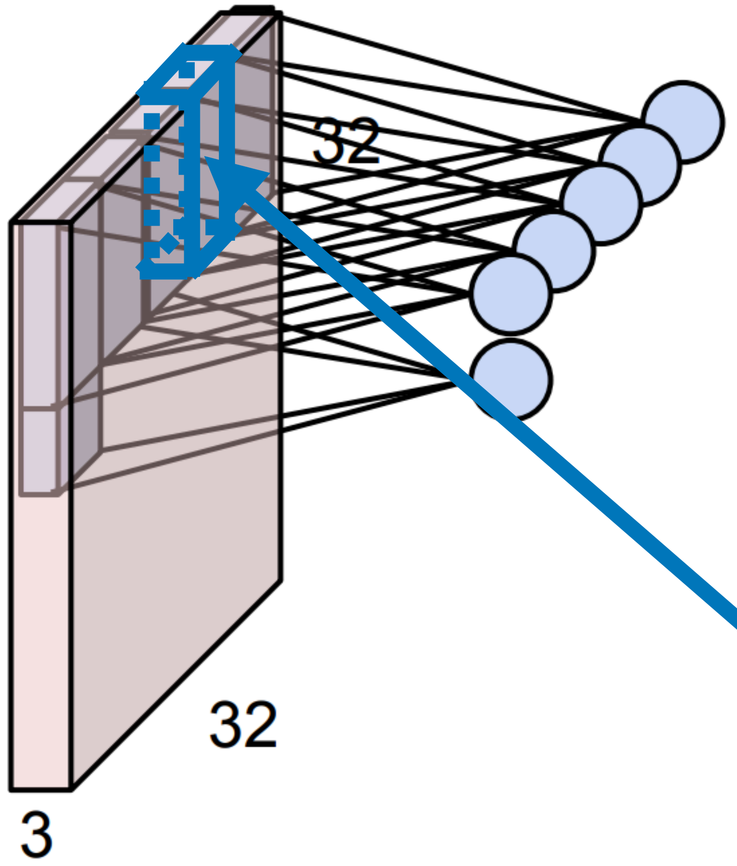
In CNNs, we use convolution filters to extract information from different patches of an image

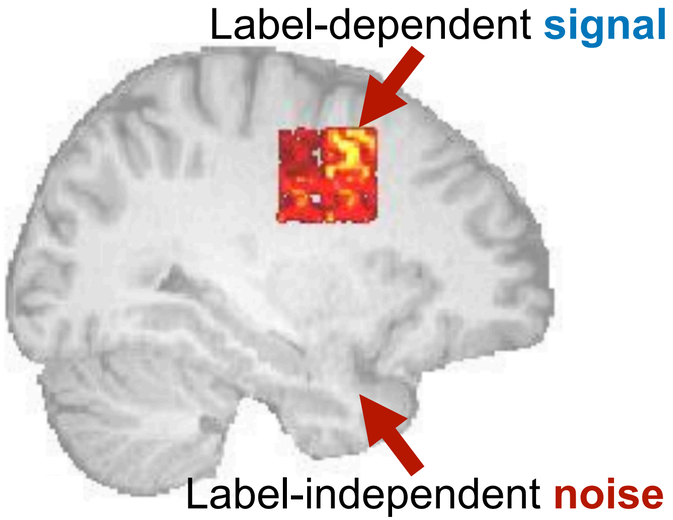
Our data model $\mathscr{D}$:

1. $y \sim \text{Rademacher}$
2.

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]





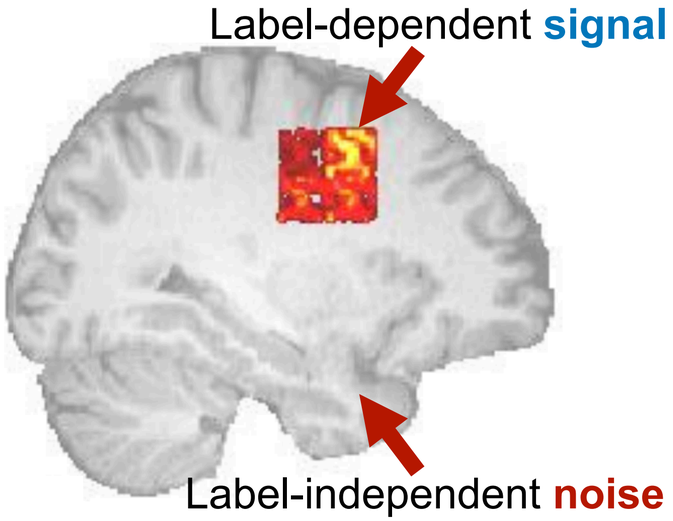
In CNNs, we use convolution filters to extract information from different patches of an image

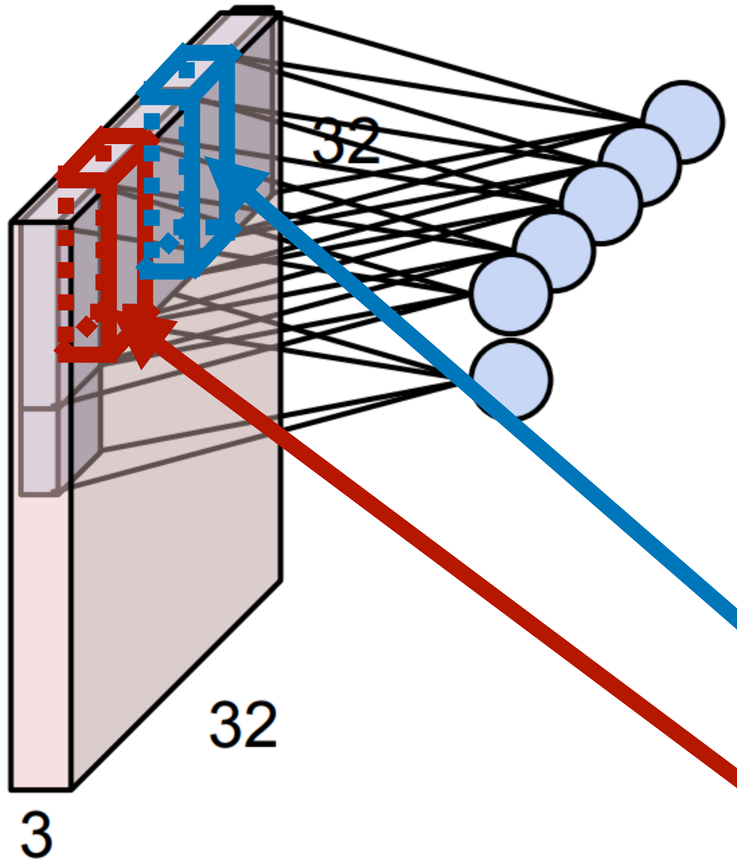
Our data model $\mathscr{D}$:

1. $y \sim \text{Rademacher}$
2. Signal patch : $y \cdot \boldsymbol{\mu}$

A Simple Statistical Learning Task

In image-based prediction tasks, it is common to assume that only a small patch of the image is relavant to the labels. [Wu & Feng, 2022]





In CNNs, we use convolution filters to extract information from different patches of an image

Our data model $\mathscr{D}$:

1. $y \sim \text{Rademacher}$
2. Signal patch : $y \cdot \boldsymbol{\mu}$

Noise patch : $\boldsymbol{\xi} \sim N(0, \sigma_p^2 \mathbf{I})$

CNN models

The corresponding two-layer CNN:

$$f_{\mathbf{W}}(\mathbf{x}) = F_{+1}(\mathbf{W}_{+1}, \mathbf{x}) - F_{-1}(\mathbf{W}_{-1}, \mathbf{x})$$

$$F_j(\mathbf{W}_j, \mathbf{x}) = \frac{1}{m} \sum_{r=1}^{m} \left[\sigma(\langle \mathbf{w}_{j,r}, \mathbf{x}^{(1)} \rangle) + \sigma(\langle \mathbf{w}_{j,r}, \mathbf{x}^{(2)} \rangle)\right], \; j \in \{\pm 1\}$$

CNN models

The corresponding two-layer CNN:

$$f_{\mathbf{W}}(\mathbf{x}) = F_{+1}(\mathbf{W}_{+1}, \mathbf{x}) - F_{-1}(\mathbf{W}_{-1}, \mathbf{x})$$

$$F_j(\mathbf{W}_j, \mathbf{x}) = \frac{1}{m} \sum_{r=1}^{m} \left[\sigma(\langle \mathbf{w}_{j,r}, \mathbf{x}^{(1)} \rangle) + \sigma(\langle \mathbf{w}_{j,r}, \mathbf{x}^{(2)} \rangle)\right], \; j \in \{\pm 1\}$$

- We consider the case $\sigma(x) = \mathrm{ReLU}^q(x)$ for $q > 2$. The power q here ensures smoothness. The results can also be applied to the "Huberized ReLU":

$$\sigma(z) = \begin{cases} 0, & z < 0 \\ z^q/(qh^{q-1}), & z \in [0, h] \\ z - (q-1)h/q, & \text{otherwise.} \end{cases}$$

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n} \sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})], \;\; L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}} \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n} \sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}} \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n} \sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}} \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

The dimension d is sufficiently large

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n} \sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}} \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

The dimension d is sufficiently large

The learning rate η and the weight random initialization scale σ_0 are appropriately chosen.

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n} \sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}} \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

The dimension d is sufficiently large

The learning rate η and the weight random initialization scale σ_0 are appropriately chosen.

Then:

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p\sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n}\sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}}\ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

The dimension d is sufficiently large

The learning rate η and the weight random initialization scale σ_0 are appropriately chosen.

Then:

- If $n \cdot SNR^q = \tilde{\Omega}\big(1\big)$, then gradient descent will output a CNN $\hat{\mathbf{W}}$ such that $L_S(\hat{\mathbf{W}}) \approx 0$, $L_{\mathscr{D}}(\hat{\mathbf{W}}) \approx 0$.

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p\sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n}\sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}}\ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

The dimension d is sufficiently large

The learning rate η and the weight random initialization scale σ_0 are appropriately chosen.

Then:

- If $n \cdot SNR^q = \tilde{\Omega}\big(1\big)$, then gradient descent will output a CNN $\hat{\mathbf{W}}$ such that $L_S(\hat{\mathbf{W}}) \approx 0$, $L_{\mathscr{D}}(\hat{\mathbf{W}}) \approx 0$.
- If $n^{-1} \cdot SNR^{-q} = \tilde{\Omega}\big(1\big)$, then gradient descent will output a CNN $\hat{\mathbf{W}}$ such that $L_S(\hat{\mathbf{W}}) \approx 0$, $L_{\mathscr{D}}(\hat{\mathbf{W}}) = \Theta(1)$.

Risk Bounds of Two-Layer CNNs Trained by GD

Definition. Signal-to-noise ratio $SNR = \dfrac{\|\boldsymbol{\mu}\|_2}{\sigma_p \sqrt{d}} \approx \dfrac{\|\boldsymbol{\mu}\|_2}{\|\boldsymbol{\xi}\|_2}$.

Training/test losses $L_S(\mathbf{W}) := \dfrac{1}{n} \sum_{i=1}^{n} \ell[y_i \cdot f_{\mathbf{W}}(\mathbf{x_i})]$, $L_{\mathscr{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathscr{D}} \ell[y \cdot f_{\mathbf{W}}(\mathbf{x})]$.

Logistic loss: $\ell(z) := \log[1 + \exp(-z)]$

Theorem. Suppose that

The dimension d is sufficiently large

The learning rate η and the weight random initialization scale σ_0 are appropriately chosen.

Then:

- If $n \cdot SNR^q = \tilde{\Omega}\big(1\big)$, then gradient descent will output a CNN $\hat{\mathbf{W}}$ such that $L_S(\hat{\mathbf{W}}) \approx 0$, $L_{\mathscr{D}}(\hat{\mathbf{W}}) \approx 0$.
- If $n^{-1} \cdot SNR^{-q} = \tilde{\Omega}\big(1\big)$, then gradient descent will output a CNN $\hat{\mathbf{W}}$ such that $L_S(\hat{\mathbf{W}}) \approx 0$, $L_{\mathscr{D}}(\hat{\mathbf{W}}) = \Theta(1)$.

Sharp phase transition!

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i \,.$$

Signal-noise decomposition

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i \,.$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}, \underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i \,.$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$;

Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same independent

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same independent

If some $\gamma_{j,r}^{(t)}$ are large while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are small, then training and test losses will both be small

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same independent

If some $\gamma_{j,r}^{(t)}$ are large while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are small, then training and test losses will both be small

If $\gamma_{j,r}^{(t)}$ are small while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are large, then training loss will be small, but test loss will be large

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same independent

If some $\gamma_{j,r}^{(t)}$ are large while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are small, then training and test losses will both be small

If $\gamma_{j,r}^{(t)}$ are small while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are large, then training loss will be small, but test loss will be large

Gradient descent

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same independent

If some $\gamma_{j,r}^{(t)}$ are large while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are small, then training and test losses will both be small

If $\gamma_{j,r}^{(t)}$ are small while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are large, then training loss will be small, but test loss will be large

Discrete dynamical system of coefficients

Gradient descent →

(signal learning)
$$\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{nm} \cdot \sum_{i=1}^{n} \ell_i'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(0)}, y_i \cdot \boldsymbol{\mu} \rangle + jy\gamma_{j,r}^{(t)}) \cdot \|\boldsymbol{\mu}\|_2^2$$

(noise memorization)
$$\overline{\rho}_{j,r,i}^{(t+1)} = \overline{\rho}_{j,r,i}^{(t)} - \frac{\eta}{nm} \cdot \ell_i'^{(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i \rangle + \sum_{i'=1}^{n} \overline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} + \sum_{i'=1}^{n} \underline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} \right) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbf{1}\{y_i = j\}$$

(noise memorization)
$$\underline{\rho}_{j,r,i}^{(t+1)} = \underline{\rho}_{j,r,i}^{(t)} + \frac{\eta}{nm} \cdot \ell_i'^{(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i \rangle + \sum_{i'=1}^{n} \overline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} + \sum_{i'=1}^{n} \underline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} \right) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbf{1}\{y_i = -j\}$$

Proof Sketch - Signal-Noise Decomposition

By the gradient descent update rule, we observe that $\mathbf{w}_{j,r}^{(t)}$ lies in the span of $\mathbf{w}_{j,r}^{(0)}$, $\boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$, $i \in [n]$.

Then there exist unique coefficients $\gamma_{j,r}^{(t)} \geq 0$, $\overline{\rho}_{j,r,i}^{(t)} \geq 0$, $\underline{\rho}_{j,r,i}^{(t)} \leq 0$ such that

$$\mathbf{w}_{j,r}^{(t)} = \mathbf{w}_{j,r}^{(0)} + j \cdot \gamma_{j,r}^{(t)} \cdot \|\boldsymbol{\mu}\|_2^{-2} \cdot \boldsymbol{\mu} + \sum_{i=1}^{n} \overline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i + \sum_{i=1}^{n} \underline{\rho}_{j,r,i}^{(t)} \cdot \|\boldsymbol{\xi}_i\|_2^{-2} \cdot \boldsymbol{\xi}_i .$$

Signal-noise decomposition

$\gamma_{j,r}^{(t)}$ — signal learning

$\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ — noise memorization

Training data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_i$; Test data: signal $\boldsymbol{\mu}$, noise $\boldsymbol{\xi}_{\text{test}}$

same independent

If some $\gamma_{j,r}^{(t)}$ are large while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are small, then training and test losses will both be small

If $\gamma_{j,r}^{(t)}$ are small while $\overline{\rho}_{j,r,i}^{(t)}$, $\underline{\rho}_{j,r,i}^{(t)}$ are large, then training loss will be small, but test loss will be large

Discrete dynamical system of coefficients

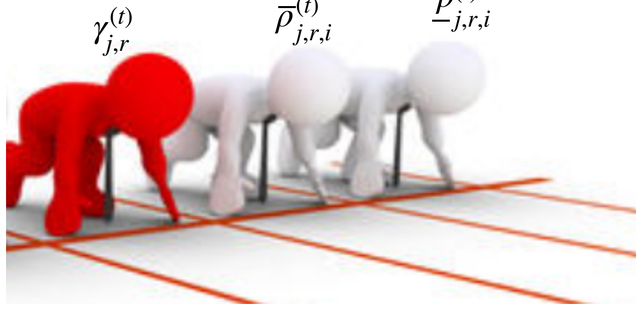
Gradient descent →

(signal learning)
$$\gamma_{j,r}^{(t+1)} = \gamma_{j,r}^{(t)} - \frac{\eta}{nm} \cdot \sum_{i=1}^{n} \ell_i'^{(t)} \cdot \sigma'(\langle \mathbf{w}_{j,r}^{(0)}, y_i \cdot \boldsymbol{\mu} \rangle + j y \gamma_{j,r}^{(t)}) \cdot \|\boldsymbol{\mu}\|_2^2$$

(noise memorization)
$$\overline{\rho}_{j,r,i}^{(t+1)} = \overline{\rho}_{j,r,i}^{(t)} - \frac{\eta}{nm} \cdot \ell_i'^{(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i \rangle + \sum_{i'=1}^{n} \overline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} + \sum_{i'=1}^{n} \underline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} \right) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbf{1}\{y_i = j\}$$

(noise memorization)
$$\underline{\rho}_{j,r,i}^{(t+1)} = \underline{\rho}_{j,r,i}^{(t)} + \frac{\eta}{nm} \cdot \ell_i'^{(t)} \sigma' \left(\langle \mathbf{w}_{j,r}^{(0)}, \boldsymbol{\xi}_i \rangle + \sum_{i'=1}^{n} \overline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} + \sum_{i'=1}^{n} \underline{\rho}_{j,r,i'}^{(t)} \frac{\langle \boldsymbol{\xi}_{i'}, \boldsymbol{\xi}_i \rangle}{\|\boldsymbol{\xi}_{i'}\|_2^2} \right) \cdot \|\boldsymbol{\xi}_i\|_2^2 \cdot \mathbf{1}\{y_i = -j\}$$

→



The Generalization Gap Between Adam and GD

The Adam Algorithm

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1 - \beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1 - \beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

The Adam Algorithm

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1 - \beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1 - \beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

entry-wise operation

The Adam Algorithm

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

entry-wise operation

Such entry-wise operation can **increase the speed of noise memorization**!

Gradient Descent and Adam

Consider an extreme case where the learning rate η is very small.

Gradient Descent and Adam

Consider an extreme case where the learning rate η is very small.

Then adjacent iterations should be very similar to each other:

$$\mathbf{W}^{(t)} \approx \mathbf{W}^{(t-1)} \approx \cdots \approx \mathbf{W}^{(t-k)}$$

$$\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)}) \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-1)}) \approx \cdots \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-k)})$$

Gradient Descent and Adam

Consider an extreme case where the learning rate η is very small.

Then adjacent iterations should be very similar to each other:

$$\mathbf{W}^{(t)} \approx \mathbf{W}^{(t-1)} \approx \cdots \approx \mathbf{W}^{(t-k)}$$

$$\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)}) \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-1)}) \approx \cdots \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-k)})$$

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1 - \beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1 - \beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

Gradient Descent and Adam

Consider an extreme case where the learning rate η is very small.

Then adjacent iterations should be very similar to each other:

$$\mathbf{W}^{(t)} \approx \mathbf{W}^{(t-1)} \approx \cdots \approx \mathbf{W}^{(t-k)}$$

$$\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)}) \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-1)}) \approx \cdots \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-k)})$$

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

The impact of $\mathbf{W}^{(t-k-1)}$ has a β_1^k, β_2^k factor — very small

Gradient Descent and Adam

Consider an extreme case where the learning rate η is very small.

Then adjacent iterations should be very similar to each other:

$$\mathbf{W}^{(t)} \approx \mathbf{W}^{(t-1)} \approx \cdots \approx \mathbf{W}^{(t-k)}$$

$$\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)}) \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-1)}) \approx \cdots \approx \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t-k)})$$

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

The impact of $\mathbf{W}^{(t-k-1)}$ has a β_1^k, β_2^k factor — very small

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}} \approx \mathbf{W}^{(t)} - \eta \cdot \text{sign}[\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]$$

Gradient Descent and Adam

Now suppose that we just have ONE sample from the noisy MNIST data set:

$$\mathbf{x}_1 = \quad , \quad y_1 = 1$$

Gradient Descent and Adam

Now suppose that we just have ONE sample from the noisy MNIST data set:

$$\mathbf{x}_1 = \quad , \quad y_1 = 1$$

Then for the gradient descent update of linear logistic regression is

$$\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} = -\eta \cdot \ell'(y_1 \cdot \langle \boldsymbol{\theta}, \mathbf{x_1} \rangle) \cdot y_1 \cdot \mathbf{x_1} \,//\, \mathbf{x}_1$$

Gradient Descent and Adam

Now suppose that we just have ONE sample from the noisy MNIST data set:

$$\mathbf{x}_1 = \text{  }, \quad y_1 = 1$$

Then for the gradient descent update of linear logistic regression is

$$\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} = -\eta \cdot \ell'(y_1 \cdot \langle \boldsymbol{\theta}, \mathbf{x}_1 \rangle) \cdot y_1 \cdot \mathbf{x}_1 \,/\!/\, \mathbf{x}_1$$

Therefore if you visualize the actual update of parameters per iteration $\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}$, you will essentially get 

Gradient Descent and Adam

Now suppose that we just have ONE sample from the noisy MNIST data set:

$$\mathbf{x}_1 = \quad , \quad y_1 = 1$$

But for Adam, recall that $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)}/\sqrt{\mathbf{V}^{(t)}} \approx \mathbf{W}^{(t)} - \eta \cdot \text{sign}[\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]$

Therefore for the linear model case, we will have $\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} = \eta \cdot \text{sign}(\mathbf{x_1})$

Gradient Descent and Adam

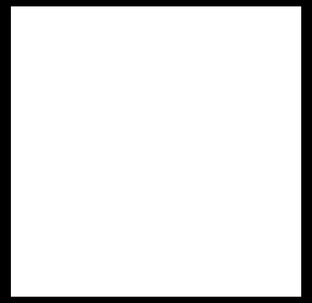
Now suppose that we just have ONE sample from the noisy MNIST data set:

$$\mathbf{x}_1 = \quad , \quad y_1 = 1$$

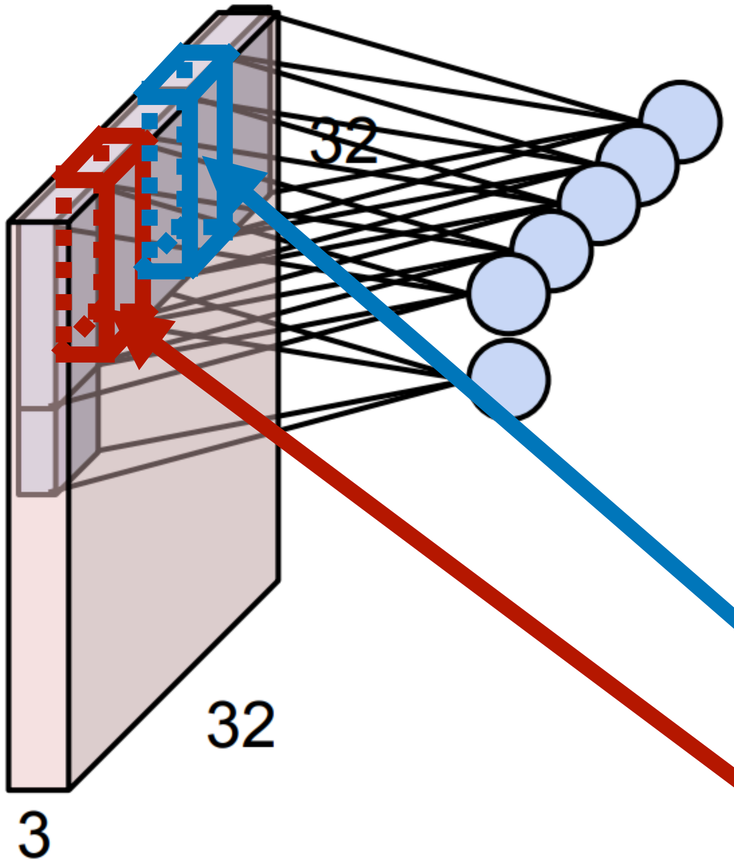
But for Adam, recall that $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)}/\sqrt{\mathbf{V}^{(t)}} \approx \mathbf{W}^{(t)} - \eta \cdot \text{sign}[\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]$

Therefore for the linear model case, we will have $\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} = \eta \cdot \text{sign}(\mathbf{x_1})$

Therefore if you visualize the actual update of parameters per iteration $\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}$, you will essentially get



Data Model for the CNN Case

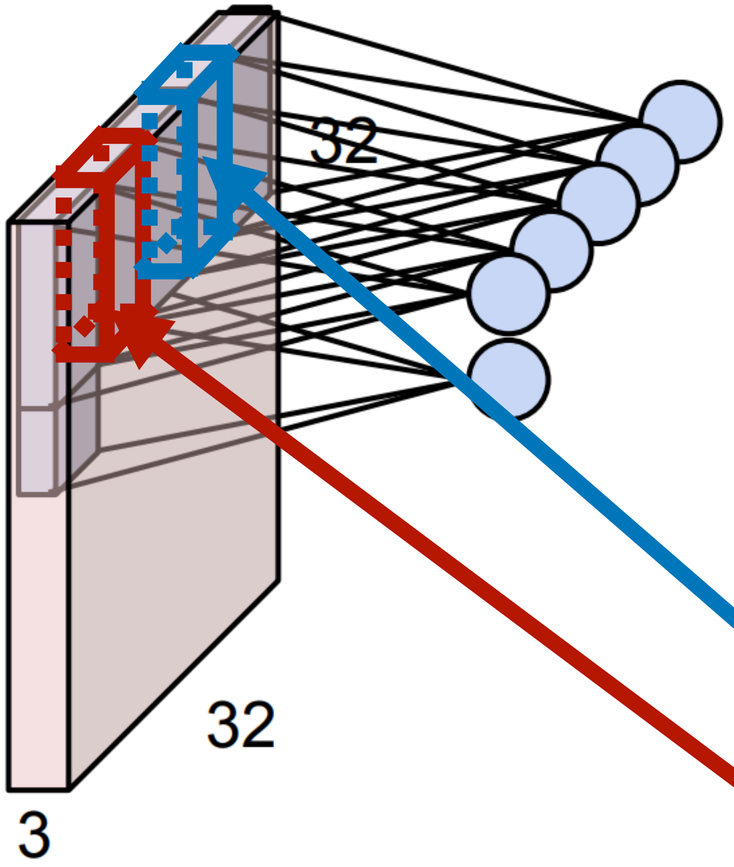


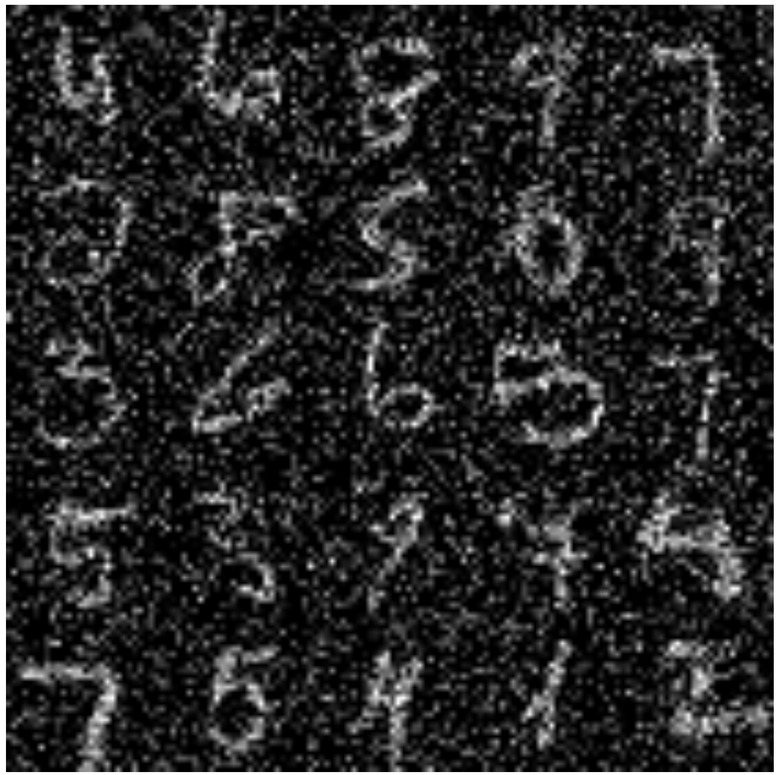
Our data model $\mathscr{D}$:

1. $y \sim \text{Rademacher}$
2. Signal patch: sparse, but non-zero entries are relatively large

Noise patch: dense, but entry-wisely small

Data Model for the CNN Case





Our data model $\mathscr{D}$:

1. $y \sim \text{Rademacher}$
2. Signal patch: sparse, but non-zero entries are relatively large

Noise patch: dense, but entry-wisely small

Main Results

Theorem

Main Results

Theorem

Under certain conditions, there exists $\lambda_0 > 0$ such that for all weight decay regularization parameter $\lambda < \lambda_0$, the following hold with high probability:

Main Results

Theorem

Under certain conditions, there exists $\lambda_0 > 0$ such that for all weight decay regularization parameter $\lambda < \lambda_0$, the following hold with high probability:

- Adam gives a CNN model $\hat{\mathbf{W}}_{\text{Adam}}$ with approximately 0 training loss, and approximately 0.5 test error.

Main Results

Theorem

Under certain conditions, there exists $\lambda_0 > 0$ such that for all weight decay regularization parameter $\lambda < \lambda_0$, the following hold with high probability:

- Adam gives a CNN model $\hat{\mathbf{W}}_{\text{Adam}}$ with approximately 0 training loss, and approximately 0.5 test error.
- GD gives a CNN model $\hat{\mathbf{W}}_{\text{GD}}$ with approximately 0 training loss, and approximately 0 test error.

Main Results

The result is more than what we have expected following the intuition from the convex setting!

Main Results

The result is more than what we have expected following the intuition from the convex setting!

- When the training loss function is convex, then weight decay will ensure that Adam and GD converge to the same unique global optimum!

Main Results

The result is more than what we have expected following the intuition from the convex setting!

- When the training loss function is convex, then weight decay will ensure that Adam and GD converge to the same unique global optimum!
- Our result shows that when training neural networks, weight decay regularization cannot resolve the issue of Adam!

Practical implications: Padam and beyond

The Padam Algorithm - “Interpolation” between Adam & SGD

Adam:

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

The Padam Algorithm - “Interpolation” between Adam & SGD

Adam:

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

Padam:

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / (\mathbf{V}^{(t)})^{p}$$

The Padam Algorithm - “Interpolation” between Adam & SGD

Adam:

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / \sqrt{\mathbf{V}^{(t)}}$$

Padam:

$$\mathbf{M}^{(t+1)} = \beta_1 \cdot \mathbf{M}^{(t)} + (1-\beta_1) \cdot \nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})$$

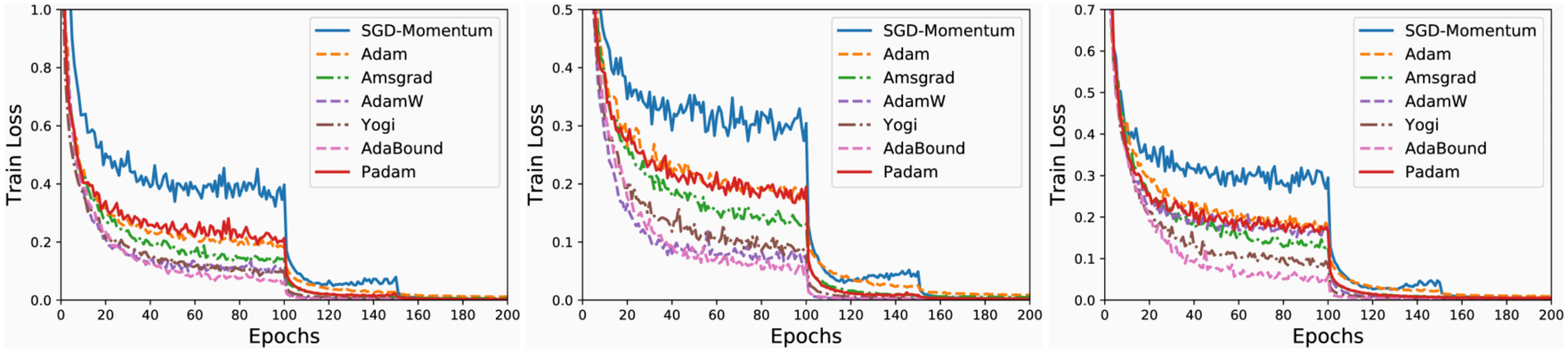
$$\mathbf{V}^{(t+1)} = \beta_2 \cdot \mathbf{V}^{(t)} + (1-\beta_2) \cdot [\nabla_{\mathbf{W}} L_S(\mathbf{W}^{(t)})]^2$$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \cdot \mathbf{M}^{(t)} / (\mathbf{V}^{(t)})^{p}$$

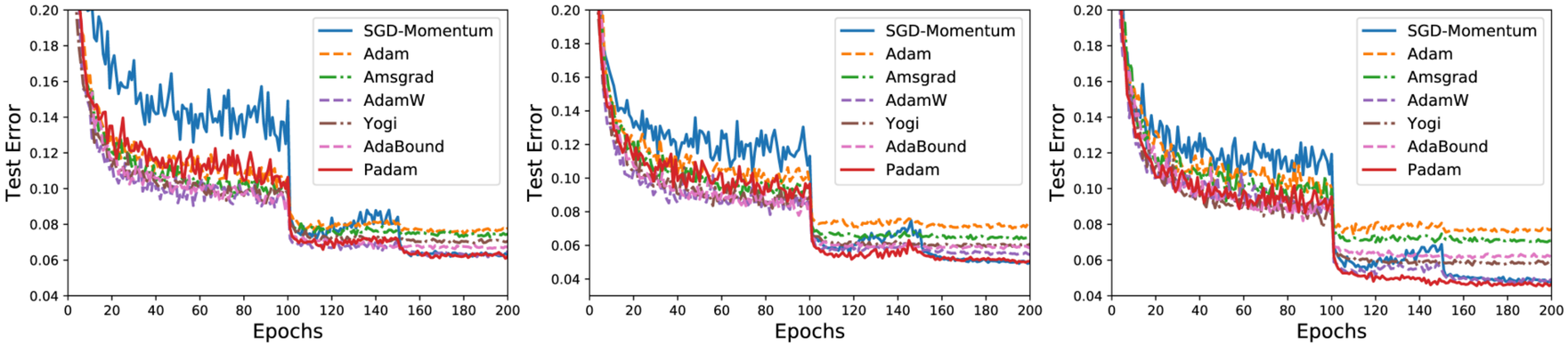
- p controls the level of **adaptivity** in the optimization algorithm
- If we set $p = 1/2$, it recovers the Adam; and if $p = 0$, it recovers SGD with momentum

Experimental Results

Training loss and test error (top-1 error) on CIFAR-10.



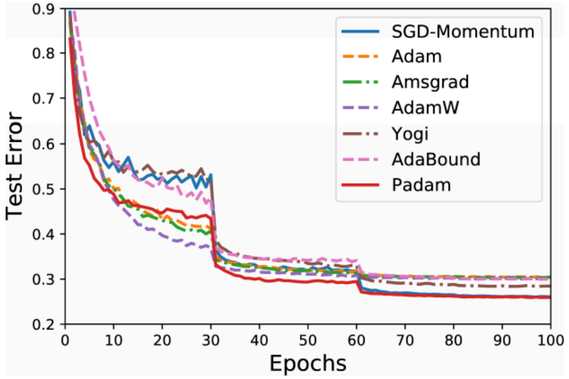
(a) Train Loss for VGGNet (b) Train Loss for ResNet (c) Train Loss for WideResNet



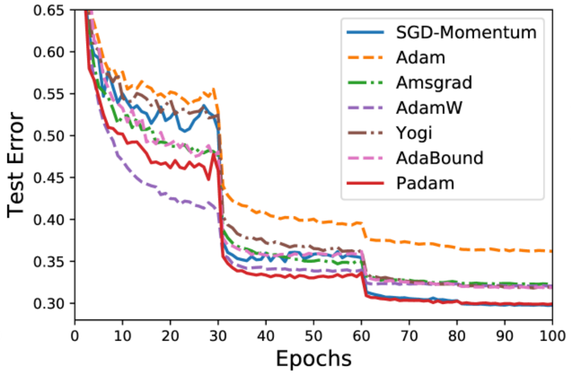
(d) Test Error for VGGNet (e) Test Error for ResNet (f) Test Error for WideResNet

Experimental Results

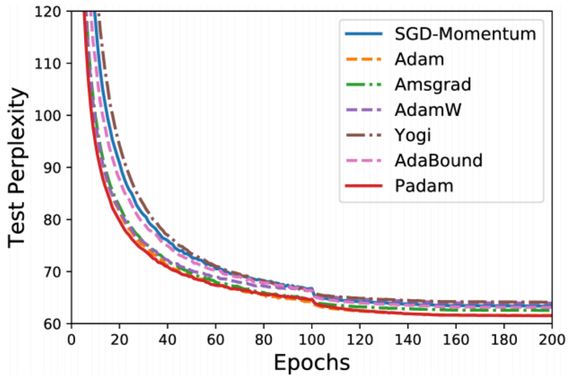
Test error on the ImageNet dataset (left and middle columns), and test perplexity on the Penn Treebank dataset (right column)



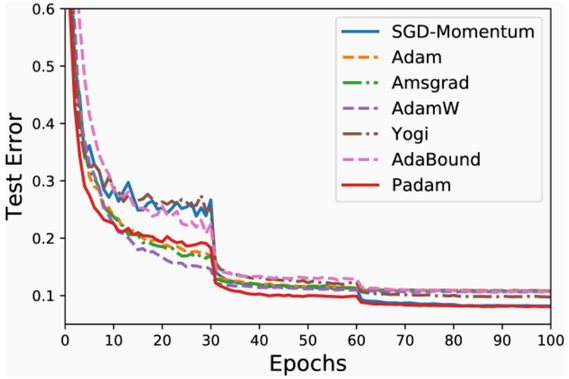
(a) Top-1 Error, VGGNet



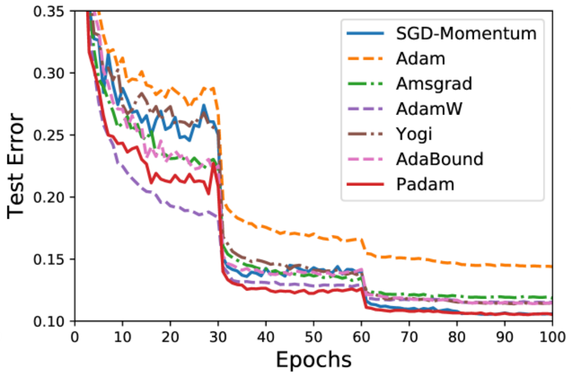
(b) Top-1 Error, ResNet



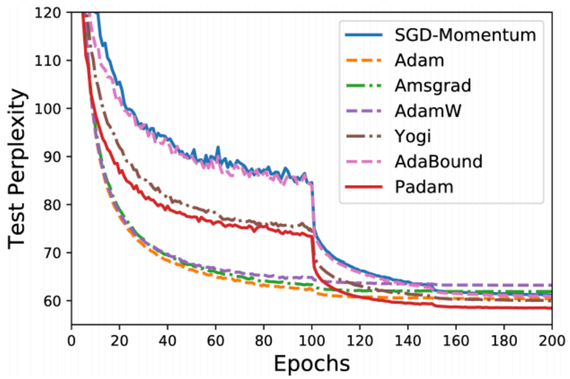
(c) 2-layer LSTM



(d) Top-5 Error, VGGNet



(e) Top-5 Error, ResNet



(f) 3-layer LSTM

Experimental Results

Test Accuracy on the CIFAR10 dataset after 200 epochs

Models	SGD-Momentum	Adam	Amsgrad	AdamW	Yogi	AdaBound	Padam
VGGNet	93.71	92.21	92.54	93.54	92.94	93.28	**93.78**
ResNet	**95.00**	92.89	93.53	94.56	93.92	94.16	94.94
WideResNet	95.26	92.27	92.91	95.08	94.23	93.85	**95.34**

Test Accuracy on the ImageNet dataset after 200 epochs

Models	Test Accuracy	SGD-Momentum	Adam	Amsgrad	AdamW	Yogi	AdaBound	Padam
Resnet	Top-1	**70.23**	63.79	67.69	67.93	68.23	68.13	70.07
	Top-5	89.40	85.61	88.15	88.47	88.59	88.55	**89.47**
VGGNet	Top-1	73.93	69.52	69.61	69.89	71.56	70.00	**74.04**
	Top-5	91.82	89.12	89.19	89.35	90.25	89.27	**91.93**

Test Perplexity on the Penn Treebank dataset after 200 epochs

Models	SGD-Momentum	Adam	Amsgrad	AdamW	Yogi	AdaBound	Padam
2-layer LSTM	63.37	61.58	62.56	63.93	64.13	63.14	**61.53**
3-layer LSTM	61.22	60.44	61.92	63.24	60.01	60.89	**58.48**

Experimental Results

Test Accuracy on the CIFAR10 dataset after 200 epochs

Models	SGD-Momentum	Adam	Amsgrad	AdamW	Yogi	AdaBound	Padam
VGGNet	93.71	92.21	92.54	93.54	92.94	93.28	**93.78**
ResNet	**95.00**	92.89	93.53	94.56	93.92	94.16	94.94
WideResNet	95.26	92.27	92.91	95.08	94.23	93.85	**95.34**

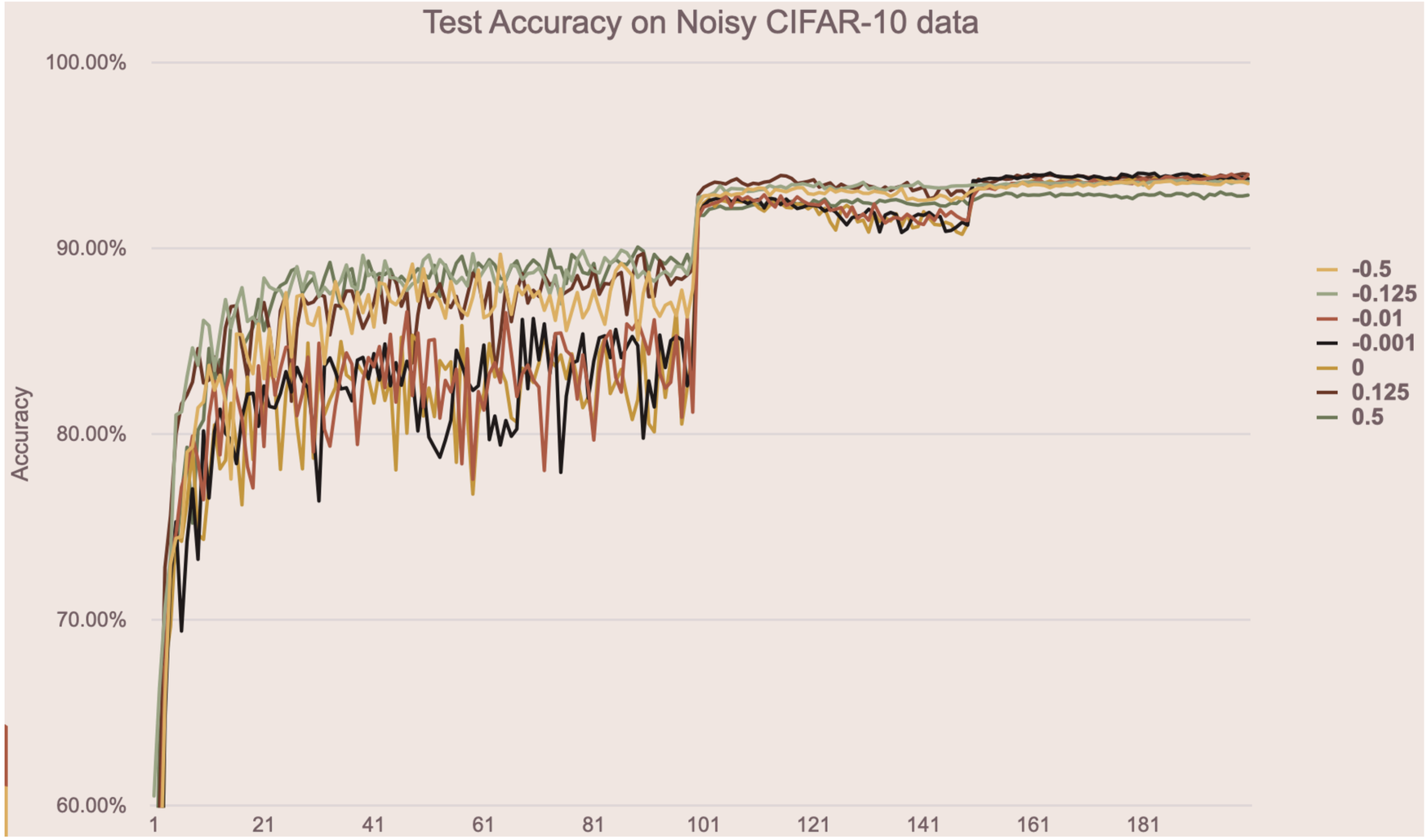
Test Accuracy on the ImageNet dataset after 200 epochs

Models	Test Accuracy	SGD-Momentum	Adam	Amsgrad	AdamW	Yogi	AdaBound	Padam
Resnet	Top-1	**70.23**	63.79	67.69	67.93	68.23	68.13	70.07
	Top-5	89.40	85.61	88.15	88.47	88.59	88.55	**89.47**
VGGNet	Top-1	73.93	69.52	69.61	69.89	71.56	70.00	**74.04**
	Top-5	91.82	89.12	89.19	89.35	90.25	89.27	**91.93**

Test Perplexity on the Penn Treebank dataset after 200 epochs

Models	SGD-Momentum	Adam	Amsgrad	AdamW	Yogi	AdaBound	Padam
2-layer LSTM	63.37	61.58	62.56	63.93	64.13	63.14	**61.53**
3-layer LSTM	61.22	60.44	61.92	63.24	60.01	60.89	**58.48**

Sometimes, “Extrapolation” Also Works!



Summary

We perform a case study on a simple statistical learning problem motivated by brain image data:

Summary

We perform a case study on a simple statistical learning problem motivated by brain image data:

We show that when training a 2-layer CNN, GD can achieve good test accuracy
while Adam will perform no better than random guess.

Summary

We perform a case study on a simple statistical learning problem motivated by brain image data:

- We show that when training a 2-layer CNN, GD can achieve good test accuracy while Adam will perform no better than random guess.
- Motivated by our theoretical explanation, we propose new algorithms that can potentially enjoy better prediction accuracy when learning certain type of data.

Summary

We perform a case study on a simple statistical learning problem motivated by brain image data:

- We show that when training a 2-layer CNN, GD can achieve good test accuracy while Adam will perform no better than random guess.
- Motivated by our theoretical explanation, we propose new algorithms that can potentially enjoy better prediction accuracy when learning certain type of data.

Thank you!