Over-Training with Mixup May Hurt Generalization (*ICLR'23*)

Zixuan Liu[1*] **Ziqiao Wang**[1*] Hongyu Guo[2,1] Yongyi Mao[1]

[1]University of Ottawa [2]National Research Council Canada (NRC)

Contributions

- **Novel Observation**
  - Over-training with Mixup causes U-shaped test error curve.
- **Explanation**
  - Mixup induces label noise.
  - Overfitting to noise occcurs in over-training.

Background on Mixup

C-class classification setting

- ▷ Input space: $\mathcal{X} \subseteq \mathbb{R}^{d_0}$; Label space: $\mathcal{Y} = \{1, 2, \ldots, C\}$.
- ▷ Training set: $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where each $\mathbf{y}_i$ may be a one-hot vector.
- ▷ Predictor: $f_\theta : \mathcal{X} \to [0, 1]^C$; Loss: $\ell(\theta, \mathbf{x}, \mathbf{y})$;
  Empirical risk: $\hat{R}_S(\theta) := \frac{1}{n} \sum_{i=1}^n \ell(\theta, \mathbf{x}_i, \mathbf{y}_i)$.

Background on Mixup

C-class classification setting

- Input space: $\mathcal{X} \subseteq \mathbb{R}^{d_0}$; Label space: $\mathcal{Y} = \{1, 2, \ldots, C\}$.
- Training set: $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where each $\mathbf{y}_i$ may be a one-hot vector.
- Predictor: $f_\theta : \mathcal{X} \to [0, 1]^C$; Loss: $\ell(\theta, \mathbf{x}, \mathbf{y})$;
  Empirical risk: $\hat{R}_S(\theta) := \frac{1}{n}\sum_{i=1}^n \ell(\theta, \mathbf{x}_i, \mathbf{y}_i)$.
- Mixup synthetic dataset:

$$\widetilde{S}_\lambda := \{(\lambda\mathbf{x} + (1-\lambda)\mathbf{x}', \lambda\mathbf{y} + (1-\lambda)\mathbf{y}') : (\mathbf{x}, \mathbf{y}) \in S, (\mathbf{x}', \mathbf{y}') \in S\},$$

  where $\lambda \in [0, 1]$ is drawn from some prescribed distribution, independently across for all example pairs.
- "Mixup loss", is then

$$\mathbb{E}_\lambda \hat{R}_{\widetilde{S}_\lambda}(\theta) := \mathbb{E}_\lambda \frac{1}{|\widetilde{S}_\lambda|} \sum_{(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \in \widetilde{S}_\lambda} \ell(\theta, \tilde{\mathbf{x}}, \tilde{\mathbf{y}})$$

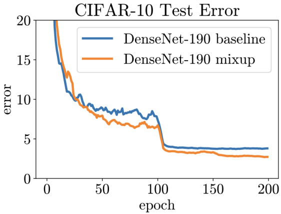
Visualization of Mixup

Figure 1: Example for Mixup Argumentation with $\lambda = 0.7$. Figure downloaded from *https://www.kaggle.com/code/kaushal2896/data-augmentation-tutorial-basic-cutout-mixup*

Mixup can effectively improve the performance

Dataset	Model	ERM	*mixup*
CIFAR-10	PreAct ResNet-18	5.6	**4.2**
	WideResNet-28-10	3.8	**2.7**
	DenseNet-BC-190	3.7	**2.7**
CIFAR-100	PreAct ResNet-18	25.6	**21.1**
	WideResNet-28-10	19.4	**17.5**
	DenseNet-BC-190	19.0	**16.8**

(a) Test errors for the CIFAR experiments.



(b) Test error evolution for the best ERM and *mixup* models.

Figure 2: Test errors for ERM and mixup on the CIFAR experiments from *Zhang, Hongyi, et al. "mixup: Beyond Empirical Risk Minimization." ICLR 2018.*

Lower Bound on Mixup Loss

Lemma 1

Let $\ell(\cdot)$ be the cross-entropy loss, and $\{\lambda\}$ is drawn i.i.d. from $\mathrm{Beta}(1,1)$ *(or the uniform distribution on* $[0,1]$*). Then for all $\theta \in \Theta$ and for any given training set S that is balanced,*

$$\mathbb{E}_\lambda \hat{R}_{\widetilde{S}_\lambda}(\theta) \geq \frac{C-1}{2C},$$

where the equality holds iff $f_\theta(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{y}}$ for each synthetic example $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \in \widetilde{S}_\lambda$.

For example, for 10-class classification tasks, the lower bound has value 0.45.

Observations: As the training loss continuously decays (left), the testing error first decreases then increases (right).

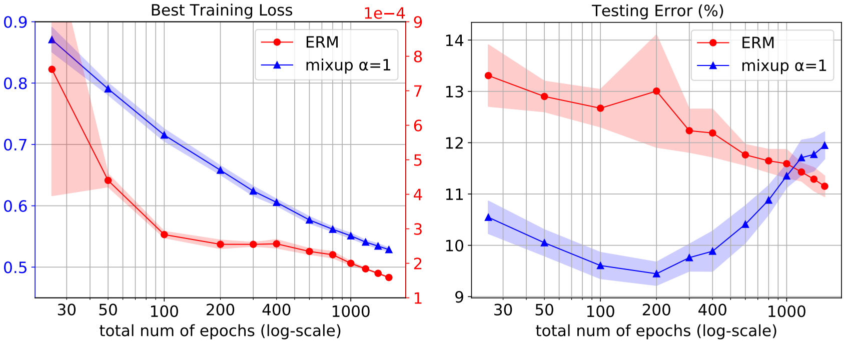
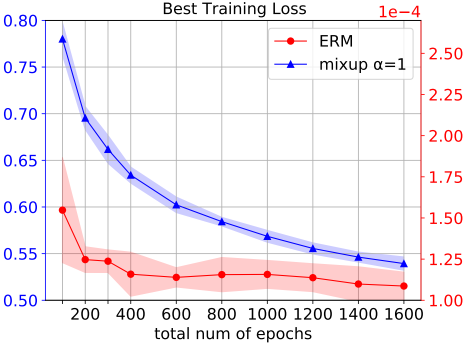


Figure 3: ResNet18 on CIFAR10

Observations



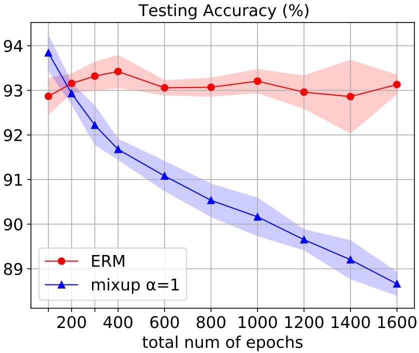


Figure 4: ResNet18 on SVHN (30%)

Observations

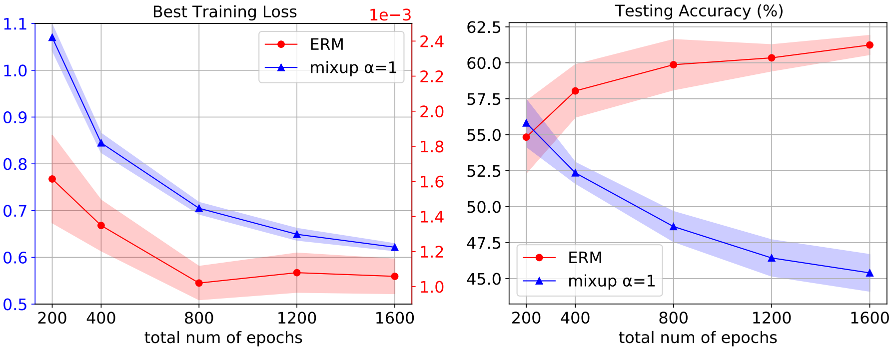


Figure 5: ResNet34 on CIFAR100

Also holds in

- ▷ different architecture, e.g., VGG16, ResNet34;
- ▷ different loss function, e.g., MSE;
- ▷ using other data augmentation (with reduced sample-size), e.g., “random crop” and “horizontal flip”;
- ▷ covariant shift, e.g., CIFAR10.1, CIFAR10.2.

Mixup Induces Label Noise

- Let $P(Y|X)$ be the ground-truth conditional distribution. Let $f : \mathcal{X} \to [0, 1]^C$, where $f_j(\mathbf{x}) \triangleq P(Y = j|X = \mathbf{x})$.
  e.g., $\mathbf{y} = \arg\max_{j\in\mathcal{Y}} f_j(\mathbf{x})$.

Mixup Induces Label Noise

- Let $P(Y|X)$ be the ground-truth conditional distribution. Let $f : \mathcal{X} \to [0,1]^C$, where $f_j(\mathbf{x}) \triangleq P(Y = j|X = \mathbf{x})$.
  e.g., $\mathbf{y} = \arg\max_{j\in\mathcal{Y}} f_j(\mathbf{x})$.
- Let $\widetilde{X} \triangleq \lambda X + (1-\lambda)X'$. There are two ways to assign a label to $\widetilde{X}$
  - Ground-truth: $\widetilde{Y}_{\mathrm{h}}^{*} \triangleq \arg\max_{j\in\mathcal{Y}} f_j(\widetilde{X})$

Mixup Induces Label Noise

- Let $P(Y|X)$ be the ground-truth conditional distribution. Let $f : \mathcal{X} \to [0,1]^C$, where $f_j(\mathbf{x}) \triangleq P(Y = j|X = \mathbf{x})$.
  e.g., $\mathbf{y} = \arg\max_{j\in\mathcal{Y}} f_j(\mathbf{x})$.
- Let $\widetilde{X} \triangleq \lambda X + (1-\lambda)X'$. There are two ways to assign a label to $\widetilde{X}$
  - Ground-truth: $\widetilde{Y}_{\mathrm{h}}^* \triangleq \arg\max_{j\in\mathcal{Y}} f_j(\widetilde{X})$
  - Mixup: $\widetilde{Y}_{\mathrm{h}} \triangleq \arg\max_{j\in\mathcal{Y}} P(\widetilde{Y} = j|\widetilde{X})$
    where $P(\widetilde{Y} = j|\widetilde{X}) = \lambda f_j(X) + (1-\lambda) f_j(X')$ for each j.

Mixup Induces Label Noise

- Let $P(Y|X)$ be the ground-truth conditional distribution. Let $f : \mathcal{X} \to [0,1]^C$, where $f_j(\mathbf{x}) \triangleq P(Y = j|X = \mathbf{x})$.
  e.g., $\mathbf{y} = \arg\max_{j\in\mathcal{Y}} f_j(\mathbf{x})$.
- Let $\widetilde{X} \triangleq \lambda X + (1-\lambda)X'$. There are two ways to assign a label to $\widetilde{X}$
  - Ground-truth: $\widetilde{Y}^*_{\mathrm{h}} \triangleq \arg\max_{j\in\mathcal{Y}} f_j(\widetilde{X})$
  - Mixup: $\widetilde{Y}_{\mathrm{h}} \triangleq \arg\max_{j\in\mathcal{Y}} P(\widetilde{Y} = j|\widetilde{X})$
    where $P(\widetilde{Y} = j|\widetilde{X}) = \lambda f_j(X) + (1-\lambda) f_j(X')$ for each j.
- When the two assignments disagree, $\widetilde{Y}_{\mathrm{h}} \neq \widetilde{Y}^*_{\mathrm{h}}$, then Mixup-assigned label $\widetilde{Y}_{\mathrm{h}}$ is noisy.

Mixup Induces Label Noise

Theorem 1

For any fixed X, X' and $\widetilde{X}$ related by $\widetilde{X} = \lambda X + (1-\lambda)X'$ for a fixed $\lambda \in [0,1]$, the probability of assigning a noisy label is lower bounded by

$$\begin{aligned} P(\widetilde{Y}_{\mathrm{h}} \neq \widetilde{Y}^*_{\mathrm{h}} | \widetilde{X}) \geq & \mathrm{TV}(P(\widetilde{Y}|\widetilde{X}), P(Y|X)) \\ \geq & \frac{1}{2} \sup_{j \in \mathcal{Y}} \left| f_j(\widetilde{X}) - [(1-\lambda) f_j(X) + \lambda f_j(X')] \right|, \end{aligned}$$

where $\mathrm{TV}(\cdot, \cdot)$ is the total variation.

Training with Noisy Labels

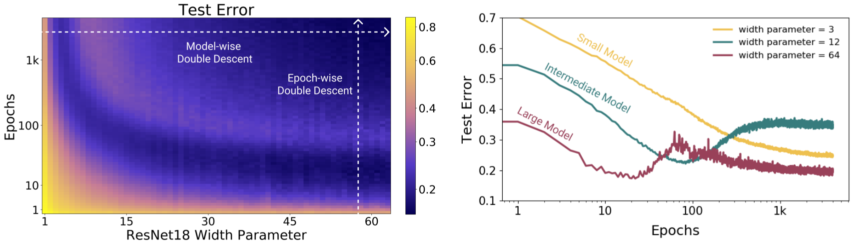


Figure 6: Double descent plots from *Nakkiran, Preetum, et al. "Deep Double Descent: Where Bigger Models and More Data Hurt." ICLR 2020.*

Reasoning about U-shaped Curve

- DNN is no longer over-parameterized ($d < m$)
- Mixup creates noisy labels

NNs learn clean data first

Neural networks are trained with a fraction of random labels, they will

- first learn the clean data

- then will overfit to the data with noisy labels.

NNs learn clean data first

Neural networks are trained with a fraction of random labels, they will

- first learn the clean data
- then will overfit to the data with noisy labels.

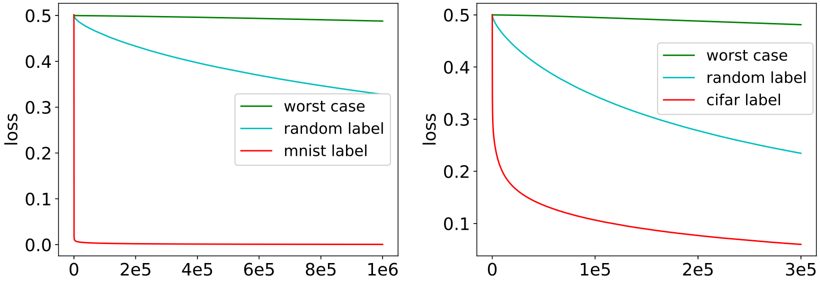


Figure 7: Convergence on clean data and noisy data from *Arora, Sanjeev, et al. "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks." ICML 2019.*

A Case Study: Regression Setting With Random Feature Models

- ▷ Let $\mathcal{Y} = \mathbb{R}$ so $f : \mathcal{X} \to \mathbb{R}$.
- ▷ Let $\widetilde{Y}^* = f(\widetilde{X})$ and $Z \triangleq \widetilde{Y} - \widetilde{Y}^*$.
  Then Z is the data-dependent noise introduced by Mixup.
  e.g., if f is strongly convex with some parameter $\rho > 0$, then $Z \geq \frac{\rho}{2}\lambda(1-\lambda)||X - X'||_2^2$.

A Case Study: Regression Setting With Random Feature Models

- Let $\mathcal{Y} = \mathbb{R}$ so $f : \mathcal{X} \to \mathbb{R}$.
- Let $\widetilde{Y}^* = f(\widetilde{X})$ and $Z \triangleq \widetilde{Y} - \widetilde{Y}^*$.
  Then Z is the data-dependent noise introduced by Mixup.
  e.g., if f is strongly convex with some parameter $\rho > 0$, then $Z \geq \frac{\rho}{2}\lambda(1-\lambda)||X - X'||_2^2$.
- Given $\widetilde{S} = \{(\widetilde{X}_i, \widetilde{Y}_i)\}_{i=1}^m$ and $\theta^T\phi(X)$, where $\phi : \mathcal{X} \to \mathbb{R}^d$ is fixed and $\theta \in \mathbb{R}^d$.
  Using the MSE loss

$$\hat{R}_{\widetilde{S}}(\theta) \triangleq \frac{1}{2m}\left|\left|\theta^T\widetilde{\Phi} - \widetilde{\mathbf{Y}}^T\right|\right|_2^2,$$

where $\widetilde{\Phi} = [\phi(\widetilde{X}_1), \phi(\widetilde{X}_2), \ldots, \phi(\widetilde{X}_m)] \in \mathbb{R}^{d\times m}$ and $\widetilde{\mathbf{Y}} = [\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_m] \in \mathbb{R}^m$.

A Case Study: Regression Setting With Random Feature Models

Given $\widetilde{S}$, the expected population risk is

$$R_t \triangleq \mathbb{E}_{\theta_t, X, Y} \left|\left|\theta_t^T \phi(X) - Y\right|\right|_2^2 .$$

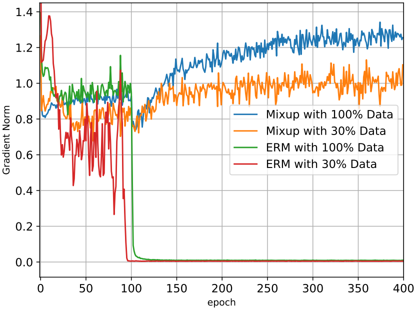
Theorem 2 (Dynamics of Population Risk)

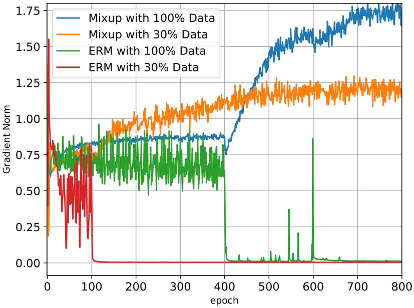
Given a synthesized dataset $\widetilde{S}$, assume $\theta_0 \sim \mathcal{N}(0, \xi^2 \mathrm{I}_d)$, $||\phi(X)||^2 \leq C_1/2$ for some constant $C_1 > 0$ and $|Z| \leq \sqrt{C_2}$ for some constant $C_2 > 0$, then we have

$$R_t - R^* \leq C_1 \sum_{k=1}^{d} \left[\underbrace{\left(\xi_k^2 + \theta_k^{*2}\right) e^{-2\eta\mu_k t}}_{\textit{Decresing Term}} + \underbrace{\frac{C_2}{\mu_k} \left(1 - e^{-\eta\mu_k t}\right)^2}_{\textit{Increasing Term}} \right] + 2\sqrt{C_1 R^* \zeta},$$

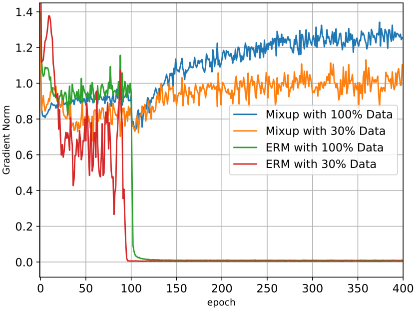
where $R^* = \mathbb{E}_{X,Y} \left|\left|Y - \theta^{*T} \phi(X)\right|\right|_2^2$, $\zeta = \sum_{k=1}^{d} \max\{\xi_k^2 + \theta_k^{*2}, \frac{C_2}{\mu_k}\}$ and μ_k is the k^{th} eigenvalue of the matrix $\frac{1}{m}\widetilde{\Phi}\widetilde{\Phi}^T$.

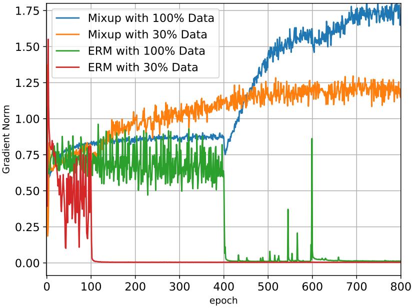
Gradient Norm in Mixup Training Does Not Vanish





Gradient Norm in Mixup Training Does Not Vanish





Take-home message:

A wrong objective also helps, only the trajectories/dynamics matter.

Thank you!

zwang286@uottawa.ca