Noisy Feature Mixup

Soon Hoe Lim (Nordita), N. Benjamin Erichson (U of Pittsburgh),
Francisco Utrera (U of Pittsburgh and ICSI), Winnie Xu (U of Toronto),
& Michael Mahoney (ICSI and UC Berkeley)

The Tenth International Conference on Learning Representations (ICLR 2022)

Introduction

Motivation: Increasing model robustness to data perturbations and improving test performance are critical for many safety and sensitive applications.

Our Solution: We propose and study a simple yet effective data augmentation method, which we call *Noisy Feature Mixup* (NFM). This method combines mixup and noise injection, thereby inheriting the benefits of both methods, and it can be seen as a generalization of input mixup and manifold mixup.

The main novelty of NFM against manifold mixup lies in the injection of noise when taking convex combinations of pairs of input and hidden layer features.

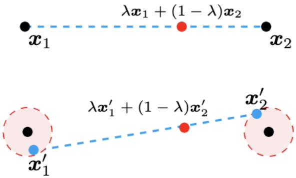

Figure: Rather than training with convex combinations of pairs of examples and their labels, we use noise-perturbed convex combinations of datapoints in both input and feature space.

Main Contributions

- We study NFM in the context of implicit regularization, showing that NFM amplifies the regularizing effects of manifold mixup and noise injection.
- We provide mathematical analysis to show that NFM can further improve model robustness when compared to manifold mixup and noise injection.
- We provide empirical results to support our theoretical findings, showing that NFM improves robustness with respect to various forms of data perturbation across a wide range of state-of-the-art architectures on computer vision benchmark tasks.

Setting

We consider multi-class classification with K classes.

- Input space: $\mathcal{X} \subset \mathbb{R}^d$; output space: $\mathcal{Y} = \mathbb{R}^K$
- Classifier: g, constructed from a learnable map $f : \mathcal{X} \to \mathbb{R}^K$, mapping an input x to its label, $g(x) = \arg\max_k f^k(x) \in [K]$.
- Training set: $\mathcal{Z}_n := \{(x_i, y_i)\}_{i=1}^n$, consisting of n pairs of input and one-hot label, with each training pair $z_i := (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ drawn i.i.d. from a ground-truth distribution $\mathcal{D}$.

We consider training a deep neural network $f := f_k \circ g_k$, where $g_k : \mathcal{X} \to g_k(\mathcal{X})$ maps an input to a hidden representation at layer k, and $f_k : g_k(\mathcal{X}) \to g_L(\mathcal{X}) := \mathcal{Y}$ maps the hidden representation to a one-hot label at layer L.

Here, $g_k(\mathcal{X}) \subset \mathbb{R}^{d_k}$ for $k \in [L]$, $d_L := K$, $g_0(x) = x$ and $f_0(x) = f(x)$.

Noisy Feature Mixup (NFM)

Training f using NFM consists of the following steps:

1. Select a random layer k from a set, $\mathcal{S} \subset \{0\} \cup [L]$, of eligible layers in the neural network.
2. Process two random data minibatches (x, y) and (x', y') as usual, until reaching layer k. This gives us two immediate minibatches $(g_k(x), y)$ and $(g_k(x'), y')$.
3. Perform mixup on these intermediate minibatches, producing the mixed minibatch:
$$(\tilde{g}_k, \tilde{y}) := (M_\lambda(g_k(x), g_k(x')), M_\lambda(y, y')),$$
where the mixing level $\lambda \sim Beta(\alpha, \beta)$, with the hyper-parameters $\alpha, \beta > 0$.
4. Produce noisy mixed minibatch by injecting additive and multiplicative noise:
$$(\tilde{\tilde{g}}_k, \tilde{y}) := ((\nVdash + \sigma_{mult}\xi_k^{mult}) \odot M_\lambda(g_k(x), g_k(x')) + \sigma_{add}\xi_k^{add}, M_\lambda(y, y')),$$
where the ξ_k^{add} and ξ_k^{mult} are $\mathbb{R}^{d_k}$-valued independent random variables modeling the additive and multiplicative noise respectively, and $\sigma_{add}, \sigma_{mult} \geq 0$ are pre-specified noise levels.
5. Continue the forward pass from layer k until output using the noisy mixed minibatch $(\tilde{\tilde{g}}_k, \tilde{y})$.
6. Compute the loss and gradients that update all the parameters of the network.

We backpropagate gradients through the entire computational graph, including those layers before the mixup layer k.

Visualizing the Effects of NFM

NFM is most effective at smoothing the decision boundary of the trained classifiers; compared to noise injection and mixup alone, it imposes the strongest smoothness on this dataset.

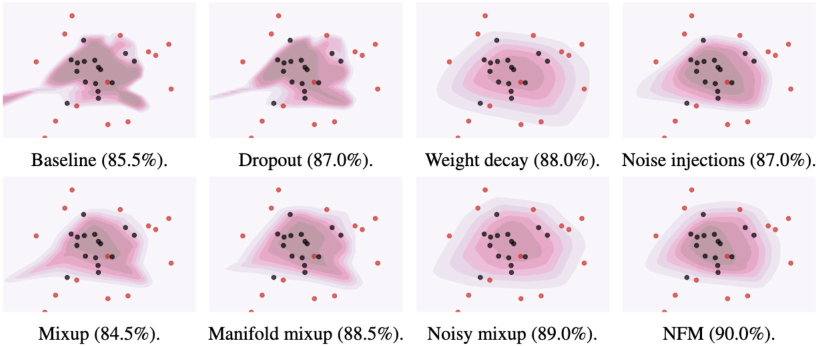

Figure: The decision boundaries and test accuracy (in parenthesis) for different training schemes on a toy dataset in binary classification.

Main Results (Theory)

- NFM can be formulated within the framework of vicinal risk minimization and interpreted as a stochastic learning strategy.
- We prove that minimizing the NFM loss function is approximately equivalent to minimizing a sum of the original loss and feature-dependent regularizers (see Theorem 1 in the paper), amplifying the regularizing effects of manifold mixup and noise injection, and implicitly reducing the feature-output Jacobians and Hessians according to the mixing and noise levels.
- Further, under reasonable assumptions, we show that NFM loss is approximately the upper bound on a regularized version of an adversarial loss (see Theorem 2 in the paper), and thus training with NFM not only improves robustness but can also mitigate robust over-fitting.

Main Results (Experiments)

We demonstrate that various model architectures trained with NFM have favorable trade-offs between predictive accuracy on clean data and robustness with respect to various types of data perturbation on CIFAR-10, CIFAR-10c, CIFAR-100, and ImageNet.

We consider input perturbations that are common in the literature: (a) white noise; (b) salt and pepper; and (c) adversarial perturbations

Table 1: Robustness of ResNet-18 w.r.t. white noise (σ) and salt and pepper (γ) perturbations evaluated on CIFAR-10. The results are averaged over 5 models trained with different seed values.

Scheme	Clean (%)	σ (%)			γ (%)		
		0.1	0.2	0.3	0.02	0.04	0.1
Baseline	94.6	90.4	76.7	56.3	86.3	76.1	55.2
Baseline + Noise	94.4	94.0	87.5	71.2	89.3	82.5	64.9
Baseline + Label Smoothing	95.0	91.3	77.5	56.9	87.7	79.2	60.0
Mixup ($\alpha = 1.0$) [81]	95.6	93.2	85.4	71.8	87.1	76.1	55.2
CutMix [78]	**96.3**	86.7	60.8	32.4	90.9	81.7	54.7
PuzzleMix [36]	**96.3**	91.7	78.1	59.9	91.4	81.8	54.4
Manifold Mixup ($\alpha = 1.0$) [70]	95.7	92.7	82.7	67.6	88.9	80.2	57.6
Noisy Mixup ($\alpha = 1.0$) [76]	78.9	78.6	66.6	46.7	66.6	53.4	25.9
Noisy Feature Mixup ($\alpha = 1.0$)	95.4	**95.0**	**91.6**	**83.0**	**91.9**	**87.4**	**73.3**

Table 2: Robustness of Wide-ResNet-18 w.r.t. white noise (σ) and salt and pepper (γ) perturbations evaluated on CIFAR-100. The results are averaged over 5 models trained with different seed values.

Scheme	Clean (%)	σ (%)			γ (%)		
		0.1	0.2	0.3	0.02	0.04	0.1
Baseline	76.9	64.6	42.0	23.5	58.1	39.8	15.1
Baseline + Noise	76.1	75.2	60.5	37.6	64.9	51.3	23.0
Mixup ($\alpha = 1.0$) [81]	80.3	72.5	54.0	33.4	62.5	43.8	16.2
CutMix [78]	77.8	58.3	28.1	13.8	70.3	58.	24.8
PuzzleMix (200 epochs) [36]	78.6	66.2	41.1	22.6	69.4	56.3	23.3
PuzzleMix (1200 epochs) [36]	80.3	53.0	19.1	6.2	69.3	51.9	15.7
Manifold Mixup ($\alpha = 1.0$) [70]	79.7	70.5	45.0	23.8	62.1	42.8	14.8
Noisy Mixup ($\alpha = 1.0$) [76]	78.9	78.6	66.6	46.7	66.6	53.4	25.9
Noisy Feature Mixup ($\alpha = 1.0$)	**80.9**	**80.1**	**72.1**	**55.3**	**72.8**	**62.1**	**34.4**

Table 3: Robustness of ResNet-50 w.r.t. white noise (σ) and salt and pepper (γ) perturbations evaluated on ImageNet. Here, the NFM training scheme improves both the predictive accuracy on clean data and robustness with respect to data perturbations.

Scheme	Clean (%)	σ (%)			γ (%)		
		0.1	0.25	0.5	0.06	0.1	0.15
Baseline	76.0	73.5	67.0	50.1	53.2	50.4	45.0
Manifold Mixup ($\alpha = 0.2$) [70]	76.7	74.9	70.3	57.5	58.1	54.6	49.5
Noisy Feature Mixup ($\alpha = 0.2$)	**77.0**	**76.5**	**72.0**	**60.1**	58.3	56.0	52.3
Noisy Feature Mixup ($\alpha = 1.0$)	76.8	76.2	71.7	60.0	**60.9**	**58.8**	**54.4**

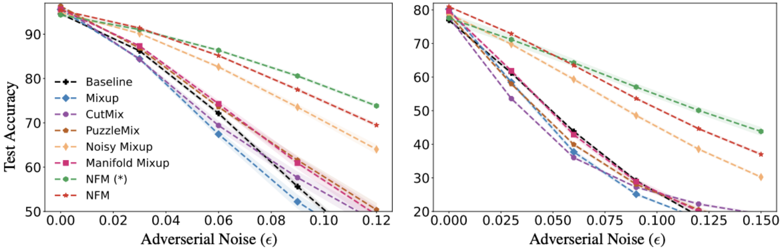

Figure 5: Pre-actived ResNet-18 evaluated on CIFAR-10 (left) and Wide ResNet-18 evaluated on CIFAR-100 (right) with respect to adversarially perturbed inputs.

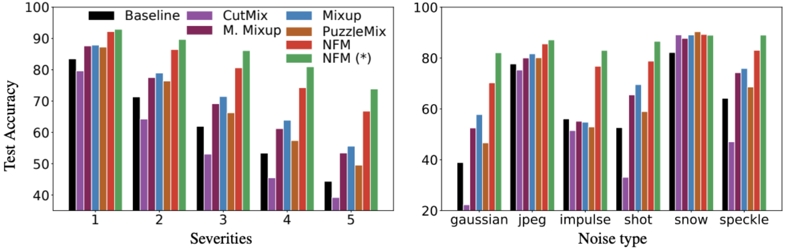

Figure 6: Pre-actived ResNet-18 evaluated on CIFAR-10c.

Conclusion

- We introduce NFM, an effective data augmentation method that combines mixup and noise injection.
- We identify the implicit regularization effects of NFM, showing that the effects are amplifications of those of manifold mixup and noise injection.
- Moreover, we demonstrate the benefits of NFM in terms of superior model robustness, both theoretically and experimentally.
- Our work inspires a range of interesting future directions, including theoretical investigations of the trade-offs between accuracy and robustness for NFM and applications of NFM beyond computer vision tasks.

References

Paper: `https://arxiv.org/abs/2110.02180`
Code: `https://github.com/erichson/NFM`