Removing the Feature Correlation Effect of Multiplicative Noise

Zijun Zhang, Yining Zhang, Zongpeng Li

University of Calgary

Multiplicative Noise

- Multiplicative noise is widely used as a **regularization** technique for deep neural networks (DNNs). General form:
$$\tilde{x}_i = u_i x_i, \forall i \in \mathcal{H}^l. \tag{1}$$
The noise u_i satisfies $\mathbb{E}[u_i] = 1$, such that $\mathbb{E}[\tilde{x}_i] = x_i$.
- E.g., **dropout**. Let m_i be the dropout mask sampled from a Bernoulli distribution, $\mathrm{Bern}(p)$, then the equivalent multiplicative noise is given by
$$u_i = m_i / p. \tag{2}$$
- Multiplicative noise can **adapt the scale** of noise to the scale of features, which may contribute to its empirical success.

Training with Noise

- In a DNN, if noise is applied to the activations of layer l, the **pre-activations** (without biases) of the next layer is

$$z_j = \sum_{i \in \mathcal{H}^l} w_{ij} \tilde{x}_i, \forall j \in \mathcal{H}^{l+1}. \tag{3}$$

- It can be decomposed into **signal and noise components** as

$$z_j^s = \sum_i w_{ij} x_i, \text{ and } z_j^n = z_j - z_j^s = \sum_i w_{ij} \left(u_i - 1 \right) x_i. \tag{4}$$

- To reduce the interference of noise, a simple strategy that can be learned is to **increase the signal-to-noise ratio** (SNR) of pre-activations.

The Feature Correlation Effect

- We can model the tendency of **increasing SNR** as an implicit objective function:

$$\text{maximize} \quad \mathrm{SNR}\left(z_j\right) = \frac{\mathbb{E}\left[\left(z_j^s - \mathbb{E}\left[z_j^s\right]\right)^2\right]}{\mathbb{E}\left[\left(z_j^n\right)^2\right]}. \tag{5}$$

- Maximizing $\mathrm{SNR}\left(z_j\right)$ is **equivalent** to

$$\boxed{\text{maximize} \quad \frac{2\,\mathbb{E}\left[\sum_{i'\neq i}\sum_i \left(w_{ij}x_i\right)\left(w_{i'j}x_{i'}\right)\right]}{\mathbb{E}\left[\sum_i \left(w_{ij}x_i\right)^2\right]}} - \frac{\mathbb{E}\left[z_j^s\right]^2}{\mathbb{E}\left[\sum_i \left(w_{ij}x_i\right)^2\right]}. \tag{6}$$

- Training with **multiplicative noise** $\Longrightarrow$ Increasing **feature correlation**

Removing the Correlation Effect

- An immediate solution is to **truncate the gradient** through the noise component:

$$\text{maximize} \quad \mathrm{SNR}(z_j) = \frac{\mathbb{E}\left[\left(z_j^s - \mathbb{E}\left[z_j^s\right]\right)^2\right]}{\boxed{\mathbb{E}\left[\left(z_j^n\right)^2\right]}}. \tag{7}$$

- However, maximizing $\mathrm{SNR}(z_j)$ is now equivalent to **increasing the magnitude** of the signal component.
- A better solution:

noise gradient truncation + batch normalization

Non-Correlating Multiplicative Noise (NCMN)

- **NCMN**-1: decomposes **batch-normalized pre-activations** (before scaling and shifting), and truncates the gradient through the noise component.

$$\hat{z}_j' = \mathrm{BN}\left(z_j^s\right) + \mathrm{AsConst}\left(\mathrm{BN}\left(z_j\right) - \mathrm{BN}\left(z_j^s\right)\right). \tag{8}$$

- **NCMN**-0: **approximates** NCMN-1 by directly applying noise to batch-normalized pre-activations.

$$\hat{z}_j' = \hat{z}_j^s + \mathrm{AsConst}\left(v_j \hat{z}_j^s\right). \tag{9}$$

- NCMN-0 is computationally efficient, and is as **simple as dropout**.

Non-Correlating Multiplicative Noise

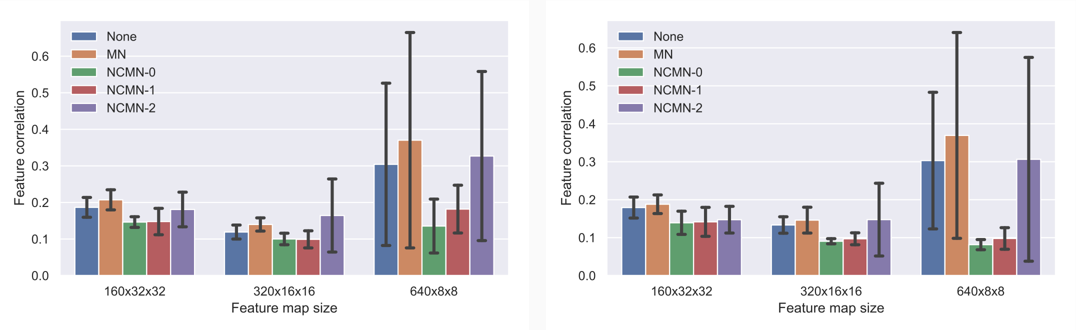
- **NCMN**-2: the decomposition is done once every two layers, works better on **residual networks**.

$$\hat{z}_k^s = \Psi_k^{l+2}\left(\Phi^{l+1}\left(\mathbf{x}^l\right)\right), \text{ and } \hat{z}_k^n = \Psi_k^{l+2}\left(\mathbf{u}^{l+1} \odot \Phi^{l+1}\left(\mathbf{u}^l \odot \mathbf{x}^l\right)\right) - \hat{z}_k^s, \quad (10)$$

$$\hat{z}_k^l = \hat{z}_k^s + \mathrm{AsConst}\left(\hat{z}_k^n\right), \quad (11)$$

- NCMN-2 can be seen as a simplified version of **shake-shake regularization** that does not require extra residual branches.

Results - Feature Correlations

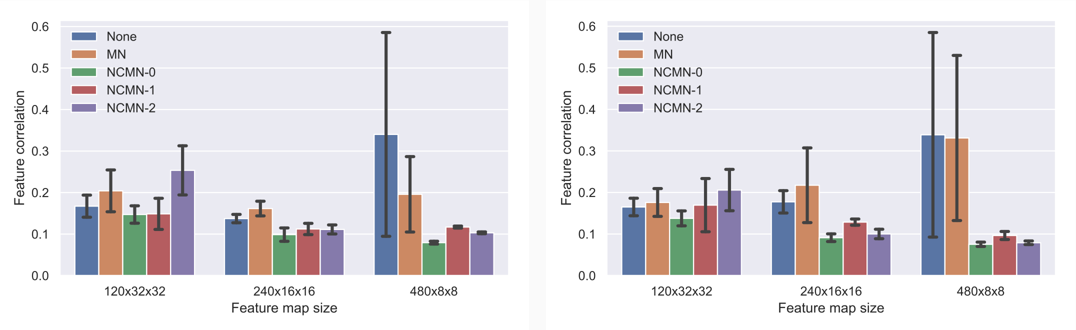


(a) Results on CIFAR-10.

(b) Results on CIFAR-100.

Figure 1: Feature correlations of CNN-16-10 networks trained with different types of noise.

Results - Feature Correlations



(a) Results on CIFAR-10.

(b) Results on CIFAR-100.

Figure 2: Feature correlations of WRN-22-7.5 networks trained with different types of noise.

Results - Classification Accuracies

Table 1: CIFAR-10/100 error rates (%) of CNN-16-10 networks trained with different types of noise.

Noise type	CIFAR-10	CIFAR-100
None	4.05 ± 0.05	19.22 ± 0.05
MN	3.76 ± 0.00	18.08 ± 0.03
NCMN-0	3.51 ± 0.07	$\mathbf{17.37} \pm 0.05$
NCMN-1	$\mathbf{3.41} \pm 0.07$	17.55 ± 0.06
NCMN-2	3.44 ± 0.03	18.16 ± 0.04

Table 2: CIFAR-10/100 error rates (%) of WRN-22-7.5 networks trained with different types of noise.

Noise type	CIFAR-10	CIFAR-100
None	3.68 ± 0.02	19.29 ± 0.07
MN	3.59 ± 0.06	18.60 ± 0.03
NCMN-0	3.34 ± 0.02	17.05 ± 0.08
NCMN-1	3.02 ± 0.06	17.09 ± 0.10
NCMN-2	$\mathbf{3.00} \pm 0.05$	$\mathbf{16.70} \pm 0.13$

Results - Classification Accuracies

Table 3: More results on CIFAR-10/100 for comparison.

Model	Params	Epochs	Noise type	CIFAR-10	CIFAR-100
DenseNet-BC (250, 24) [2]	15.3M	300	None	3.62	17.60
ResNeXt-26 (2×96d) [1]	26.2M	1800	Shake/None	**2.86**/3.58	—
ResNeXt-29 (8×64d) [1]	34.4M	1800	Shake/None	—	**15.85**/16.34
WRN-28-10 [3]	36.5M	200	Dropout/None	3.89/4.00	18.85/19.25
DenseNet-BC (40, 48)	3.9M	300	NCMN-0/None	3.51/4.07	17.68/19.92
CNN-16-3	1.6M	200	NCMN-0/None	4.47/5.10	21.92/24.97
CNN-16-10	17.1M	200	NCMN-1/None	3.41/4.05	17.55/19.22
WRN-22-2	1.1M	200	NCMN-0/None	4.56/5.19	23.54/25.90
WRN-22-7.5	15.1M	200	NCMN-2/None	3.00/3.68	16.70/19.29
WRN-22-5.4×2	15.5M	200	Shake/None	3.51/4.04	17.77/19.71
WRN-28-10	36.5M	200	NCMN-2/None	**2.78**/3.70	**15.86**/18.42

Conclusion

- We identified the feature correlation effect of multiplicative noise, and developed non-correlating multiplicative noise as a better alternative to dropout for batch-normalized neural networks.

Poster

Thu Dec 6th 10:45 AM – 12:45 PM @ Room 210 & 230 AB **#107**

X. Gastaldi.
Shake-shake regularization of 3-branch residual networks.
In *Workshop of International Conference on Learning Representations*, 2017.

G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten.
Densely connected convolutional networks.
In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

S. Zagoruyko and N. Komodakis.
Wide residual networks.
arXiv preprint arXiv:1605.07146, 2016.