Regularizing Deep Neural Network by Noise: Its Interpretation and Optimization

Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, Bohyung Han
Dept. of Computer Science and Engineering, POSTECH, Korea

Regularization by Noise

- ❏ Injecting noises to neural activations during training
- ❏ Examples: Dropout, Adding Gaussian noise

Our contribution

- Novel interpretation of the regularization by noise
- Better optimization for the regularization by noise

Interpretation

- Noise injection→ Sampling from stochastic neuron
- Lower bound analysis on the standard training objective

Optimization

- multiple noise sample + importance weighting

Interpretation and Optimization

1. Noise Injection

= Sampling from stochastic neuron

$$\mathbf{z} = g(\mathbf{h}_\phi(\mathbf{x}), \epsilon) \sim p_\phi(\mathbf{z}|\mathbf{x})$$

2. Objective for Stochastic NN

- Stochastic neural networks require optimizing marginal likelihood

$$\mathcal{L}_{\text{marginal}} = \log \mathbb{E}_{p(\epsilon)} \left[p_\theta(\mathbf{y}|g(\mathbf{h}_\phi(\mathbf{x}), \epsilon), \mathbf{x}) \right]$$

3. Lower-bound Analysis [Burda et al., 2016]

- Standard training optimizes lower bound (S=1)

$$\mathcal{L}_{\text{marginal}} \geq \underbrace{\mathcal{L}_{\text{SGD}}(S > 1)}_{\text{Proposed training objective}} \geq \underbrace{\mathcal{L}_{\text{SGD}}(S = 1) = \mathcal{L}_{\text{dropout}}}_{\text{Standard dropout training}}$$

$$\mathcal{L}_{\text{SGD}}(S > 1) = \mathbb{E}_{p(\mathcal{E})} \left[\log \frac{1}{S} \sum_{\epsilon \in \mathcal{E}} p_\theta(\mathbf{y}|g(\mathbf{h}_\phi(\mathbf{x}), \epsilon), \mathbf{x}) \right]$$

$$\mathcal{L}_{\text{SGD}}(1) = \mathcal{L}_{\text{dropout}} = \mathbb{E}_{p(\epsilon)} \left[\log p_\theta(\mathbf{y}|g(\mathbf{h}_\phi(\mathbf{x}), \epsilon), \mathbf{x}) \right]$$

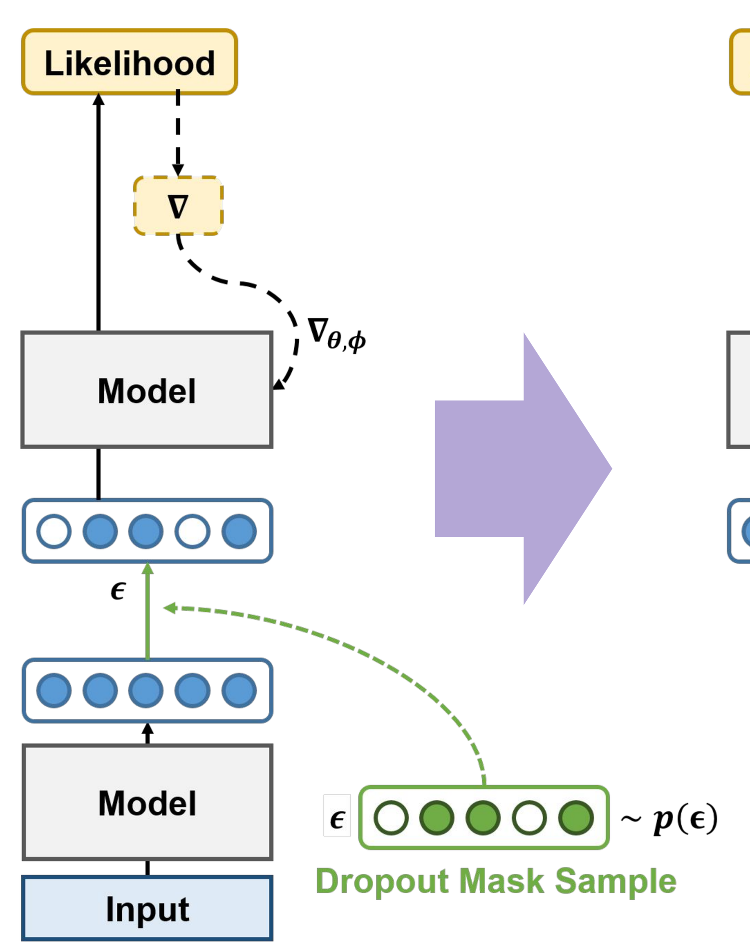
Importance weighted SGD

- Optimize tighter lower bound (S > 1)

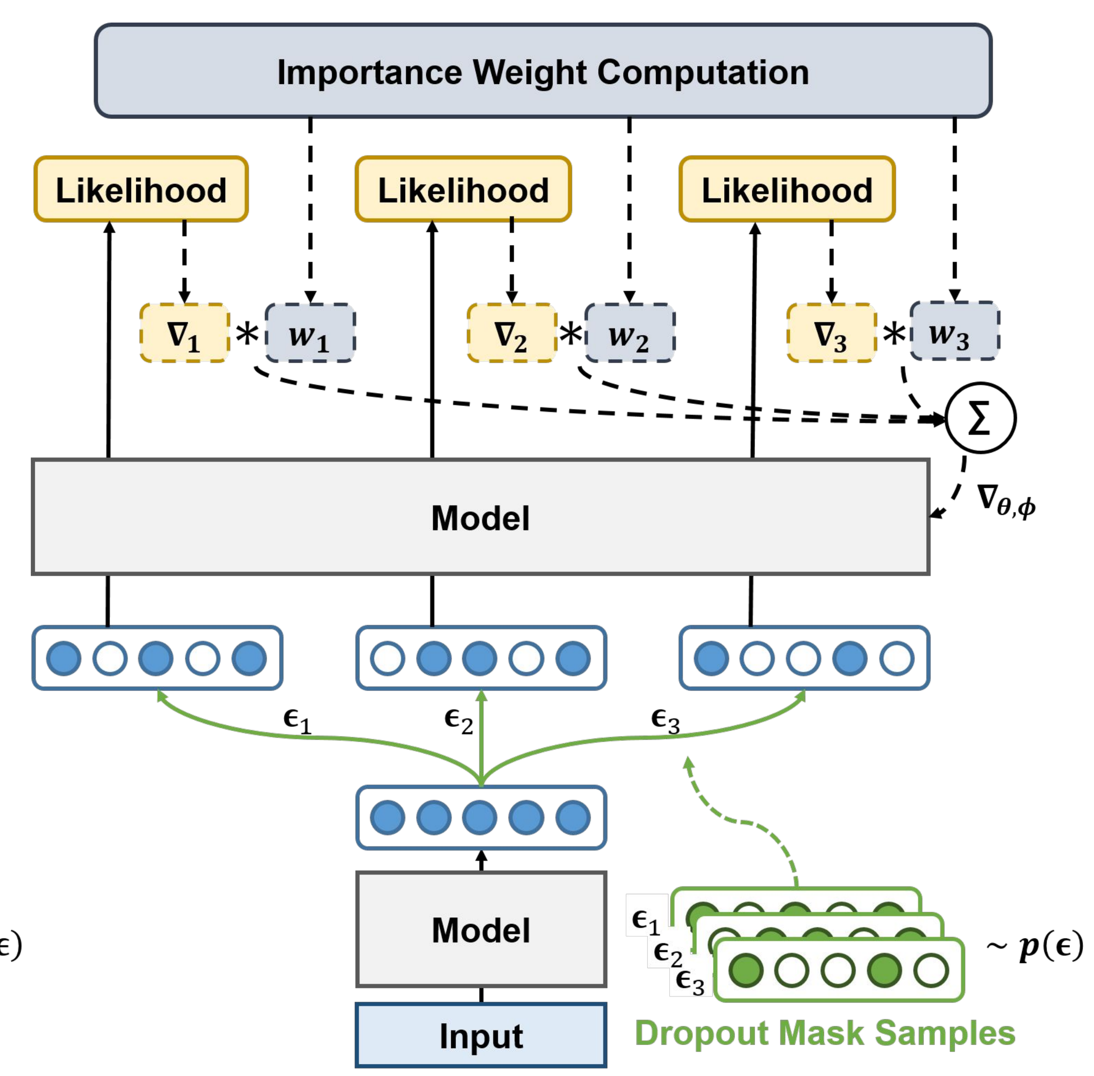
$$\nabla_{\theta,\phi} \mathcal{L}_{SGD}(S) = \mathbb{E}_{p(\mathcal{E})} \left[\sum_{\epsilon \in \mathcal{E}} w_\epsilon \nabla_{\theta,\phi} \log p_\theta \left(\mathbf{y}|g(\mathbf{h}_\phi(\mathbf{x}), \epsilon), \mathbf{x} \right) \right] \qquad w_\epsilon = \frac{p_\theta \left(\mathbf{y}|g(\mathbf{h}_\phi(\mathbf{x}), \epsilon), \mathbf{x} \right)}{\sum_{\epsilon' \in \mathcal{E}} p_\theta \left(\mathbf{y}|g(\mathbf{h}_\phi(\mathbf{x}), \epsilon'), \mathbf{x} \right)}$$

Application to Dropout

- Sample (S > 1) dropout masks per training example
- Update parameter with importance weighted gradients



Standard dropout training

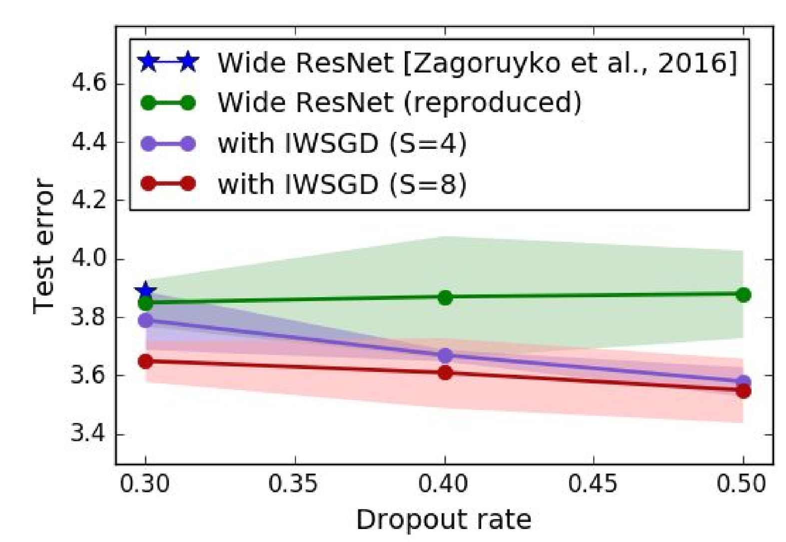


Importance weighted SGD (S=3)

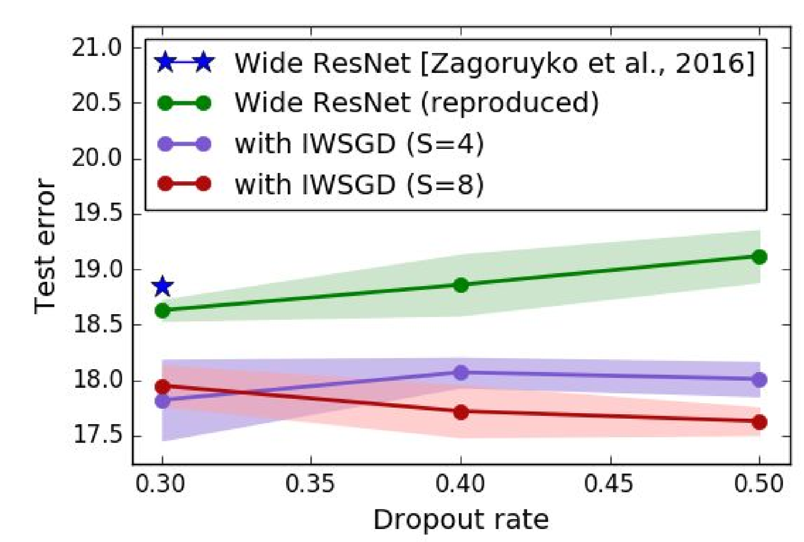
Experiments

Main Results

- Application to Wide ResNet [Zagoruyko et al., 2016]



Test error in CIFAR 10 (depth=28)



Test error in CIFAR 100 (depth=28)

- ❏ Consistent improvement over standard dropout
- ❏ Stable training with large dropout rate
- ❏ Better performance with more samples

	CIFAR-10	CIFAR-100
DenseNet [Huang et al., 2016]	3.46	17.18
Wide ResNet (depth=40) [Zagoruyko et al., 2016]	3.80	18.30
Wide ResNet (depth=28, dropout=0.3) [Zagoruyko et al., 2016]	3.89	18.85
Wide ResNet (depth=28, dropout=0.5) (reproduced)	3.88 (0.15)	19.12 (0.24)
Wide ResNet (depth=28, dropout=0.5) with IWSGD (S=4)	3.58 (0.05)	18.01 (0.13)
Wide ResNet (depth=28, dropout=0.5) with IWSGD (S=8)	**3.55 (0.11)**	**17.63 (0.13)**
Wide ResNet (depth=28, dropout=0.5) (X4 iterations)	4.48 (0.15)	20.70 (0.19)

Drop-in Replacement for Dropout

- Application to open source implementation
- Consistent improvement over various models / applications

❏ Visual Question Answering

VQA [Antol et al., 2015]	Open-Ended				Multiple-Choice			
	All	Y/N	Num	Others	All	Y/N	Num	Others
SAN [Yang et al., 2016]	58.68	79.28	36.56	46.09	-	-	-	-
SAN (reproduced)	60.19	79.69	36.74	48.84	64.77	79.72	39.03	57.82
with IWSGD (S=5)	60.31	80.74	34.70	48.66	65.01	80.73	36.36	58.05
with IWSGD (S=8)	60.41	80.86	35.56	48.56	65.21	80.77	37.56	58.18

❏ Image Captioning

MSCOCO [Lin et al., 2014]	BLEU	METEOR	CIDEr
Google-NIC [Vinyals et al., 2015]	27.7	23.7	85.5
Google-NIC (reproduced)	26.8	22.6	82.2
with IWSGD (S=5)	27.5	22.9	83.6

❏ Action Recognition

UCF-101 [Soomro et al., 2012]	Acc
TwoStreamFusion [Feichtenhofer et al., 2016]	92.50
TwoStreamFusion (reproduced)	92.49
with IWSGD (S=5)	92.73
with IWSGD (S=10)	92.69
with IWSGD (S=15)	92.72