Uncertainty estimation via Stochastic Batch Normalization

Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov,
Kirill Neklyudov, Dmitry Vetrov

Batch Normalization recap

Train phase

$$\mathrm{BN}^{\mathrm{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta$$

$$\mu(\mathcal{B}) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2(\mathcal{B}) = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu(\mathcal{B}))^2$$

“Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, Sergey Ioffe, Christian Szegedy 2015

Batch Normalization recap

Train phase

$$\mathrm{BN}^{\mathrm{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta$$

$$\mu(\mathcal{B}) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2(\mathcal{B}) = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu(\mathcal{B}))^2$$

Test phase

$$\mathrm{BN}^{\mathrm{test}}_{\gamma,\beta}(x_i) = \frac{x_i - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \epsilon}} \cdot \gamma + \beta$$

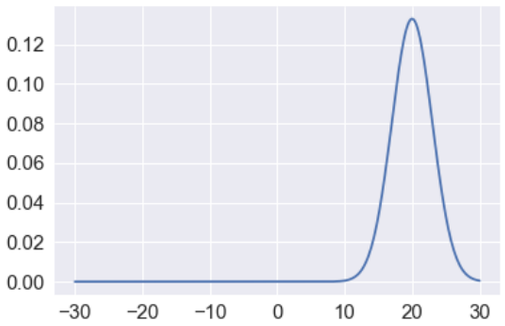
$\hat{\mu}, \hat{\sigma^2}$ exponential smoothed statistics:

$$\hat{\mu} \leftarrow \alpha\,\hat{\mu} + (1-\alpha)\,\mu(\mathcal{B})$$

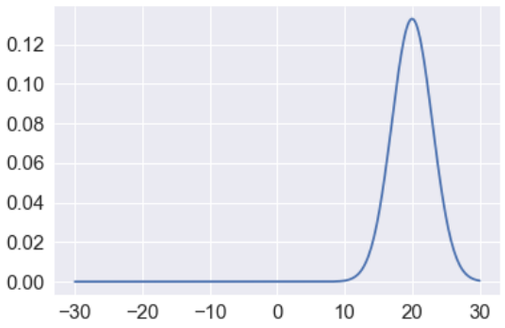
$$\hat{\sigma}^2 \leftarrow \alpha\,\hat{\sigma}^2 + (1-\alpha)\,\sigma^2(\mathcal{B})$$

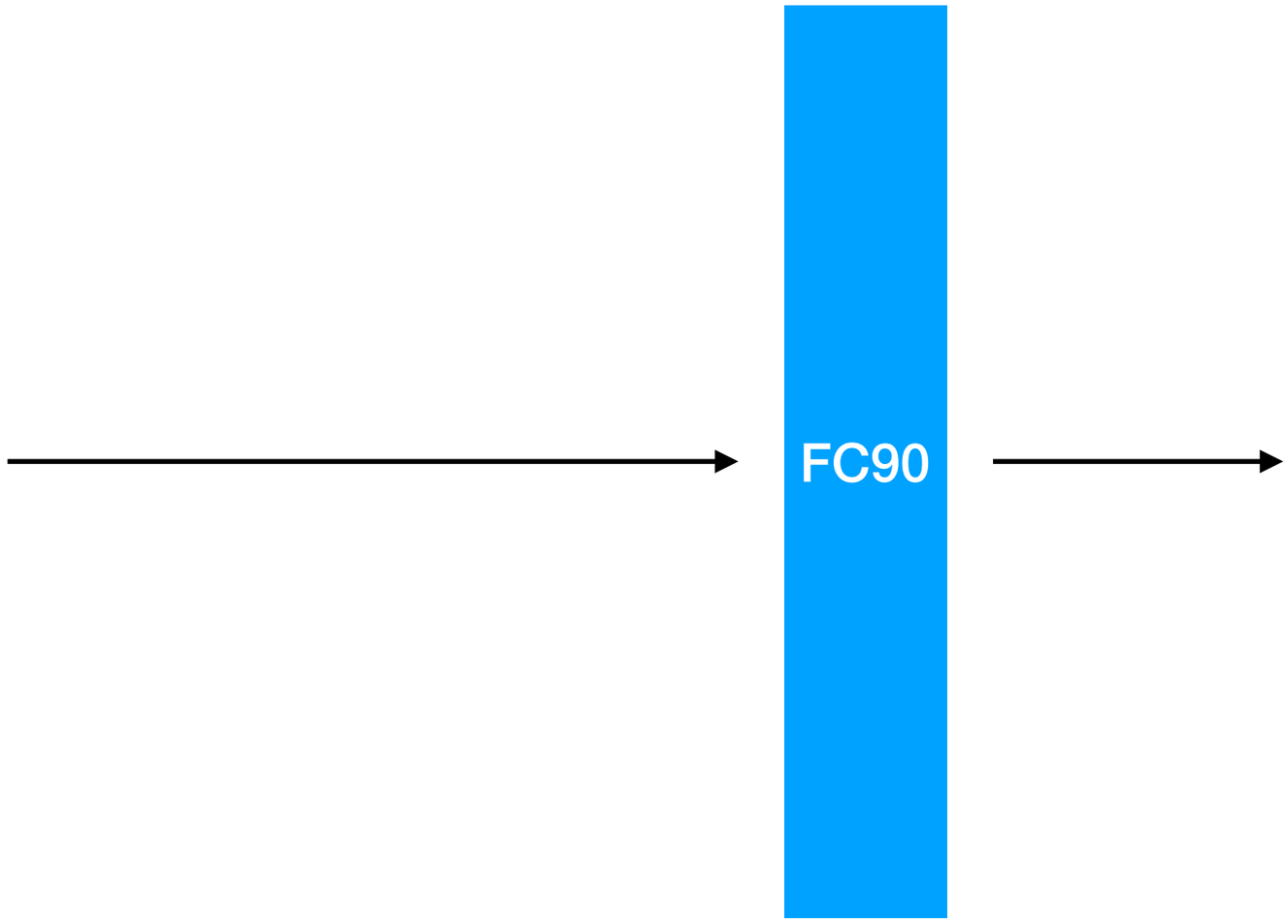
"Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Sergey Ioffe, Christian Szegedy 2015

Batch Normalization recap



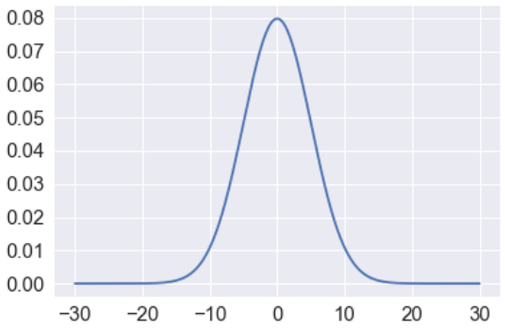
Batch Normalization recap

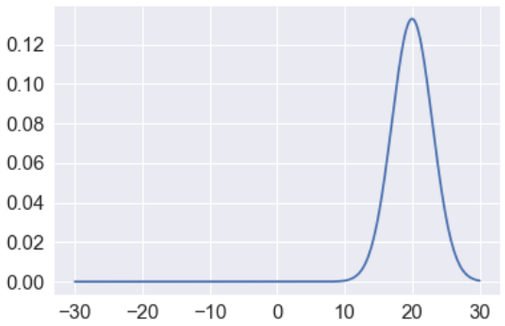


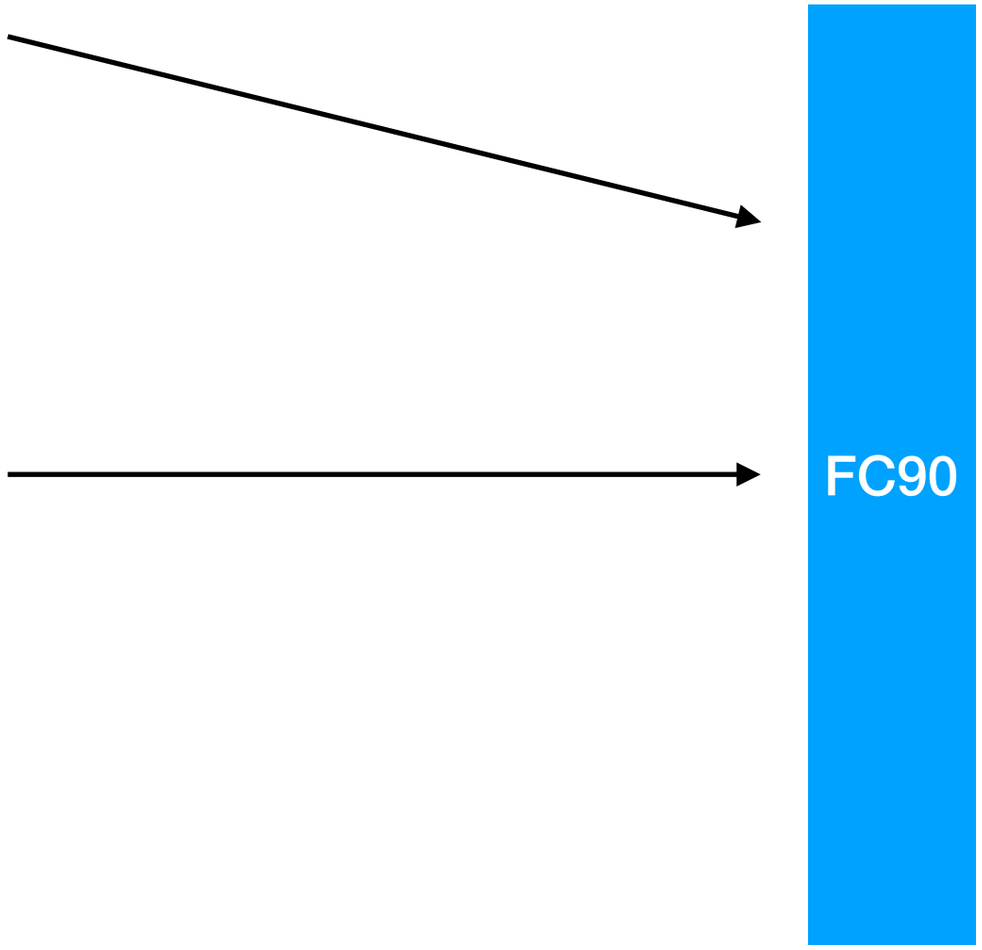


Fail

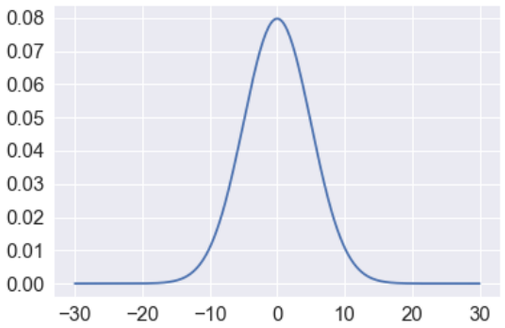
Batch Normalization recap

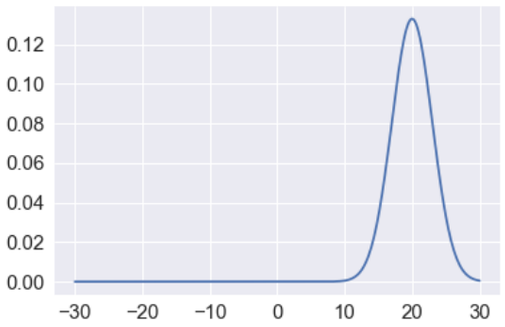


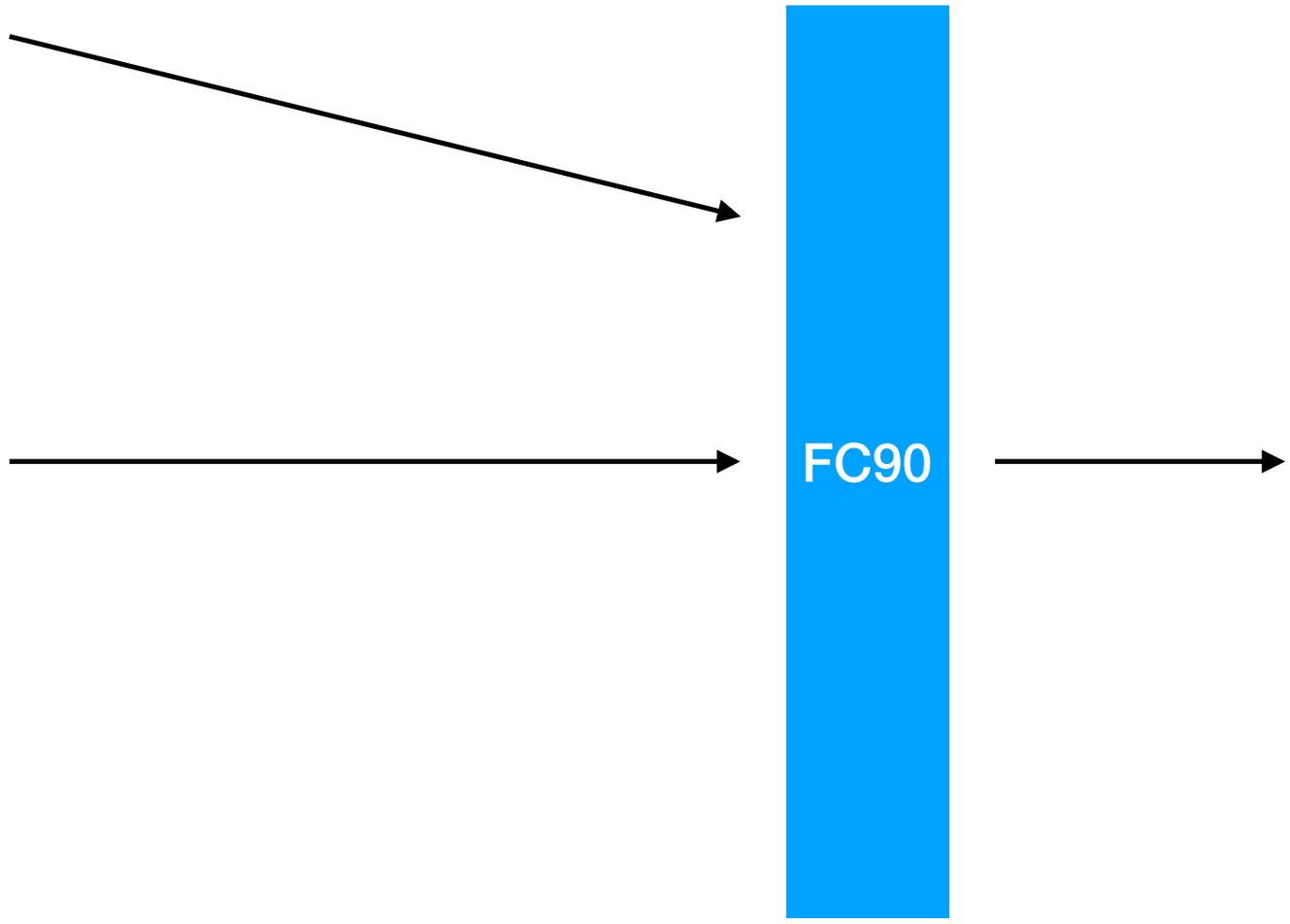




Batch Normalization recap

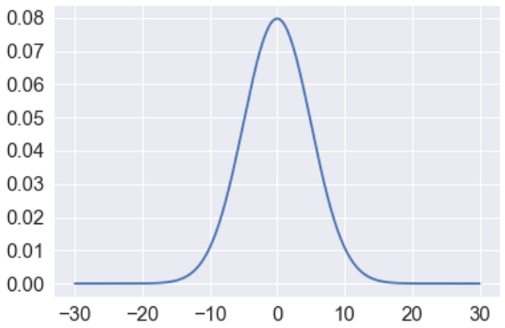


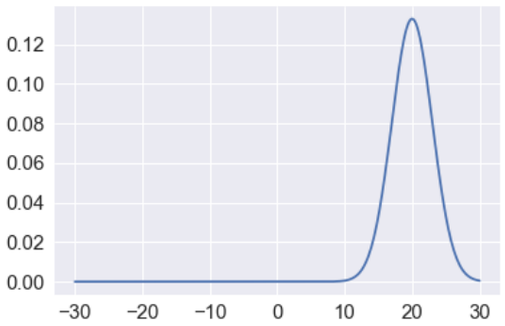


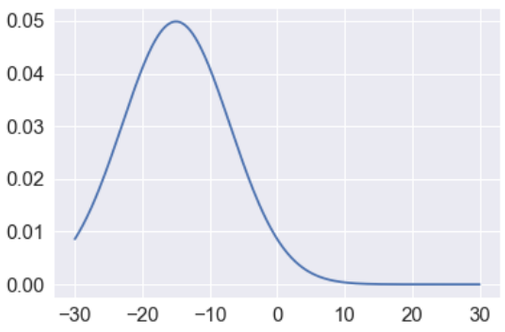


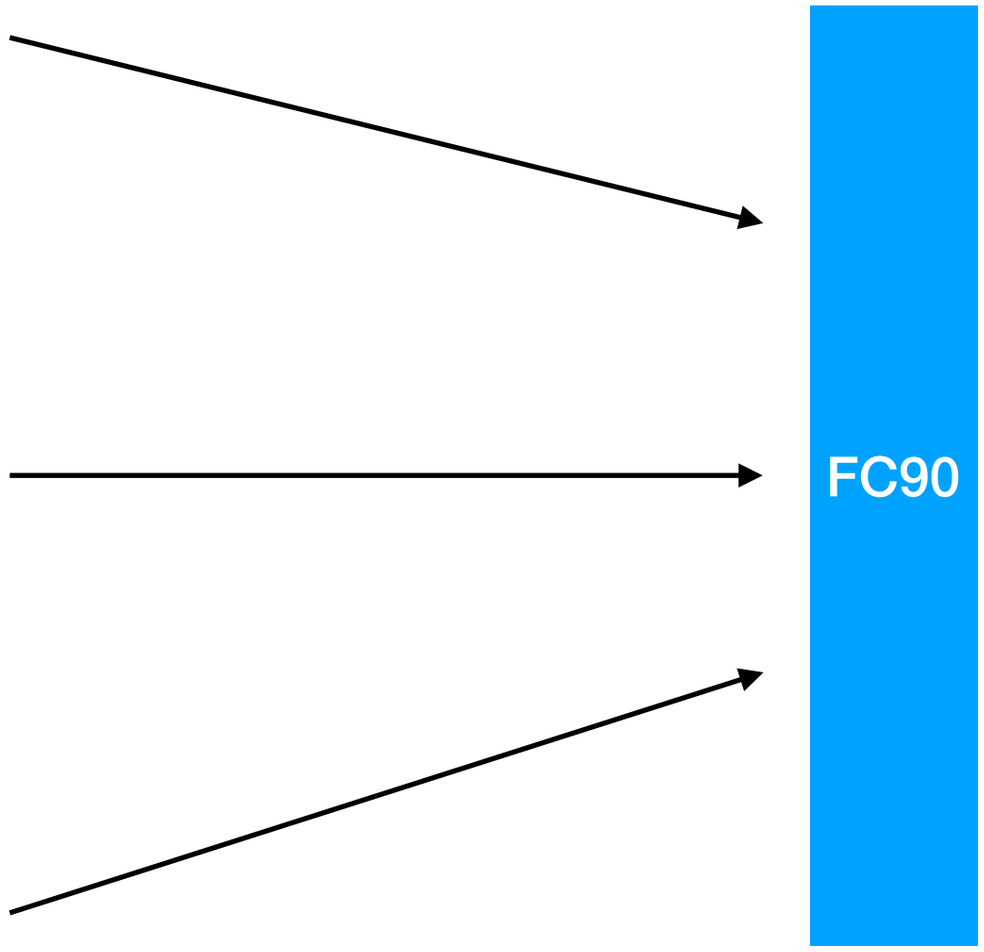
Fail

Batch Normalization recap



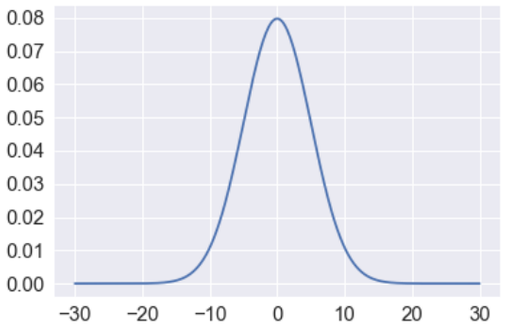


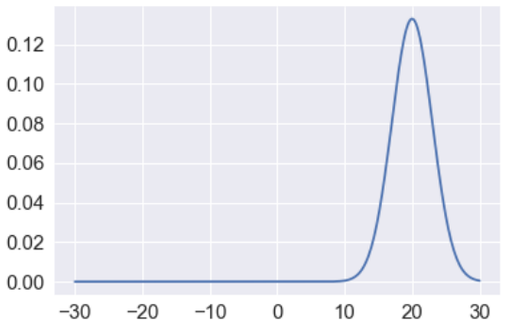


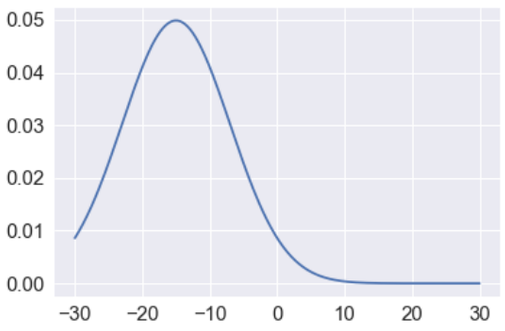


Fail

Batch Normalization recap

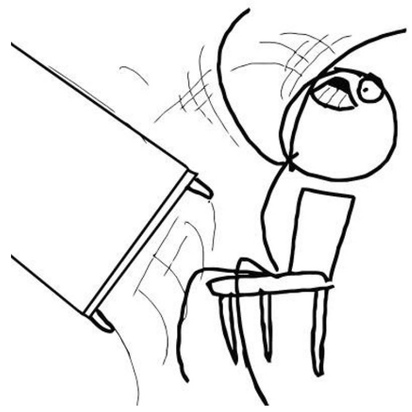




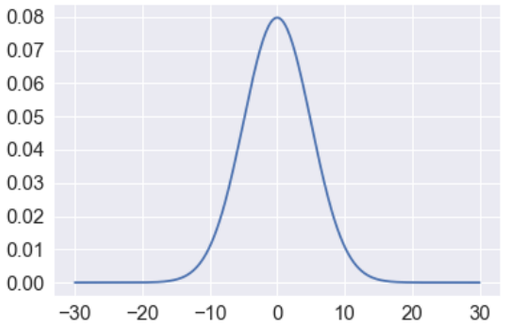


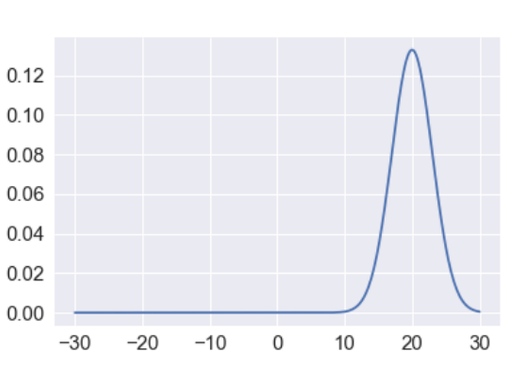
FC90

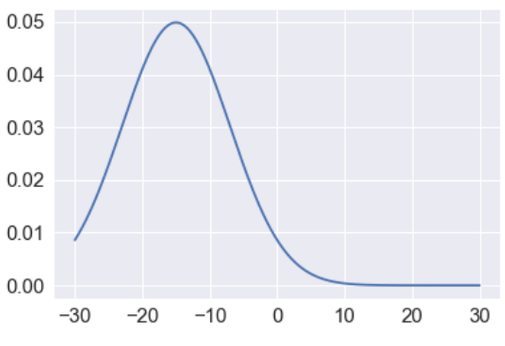
Fail



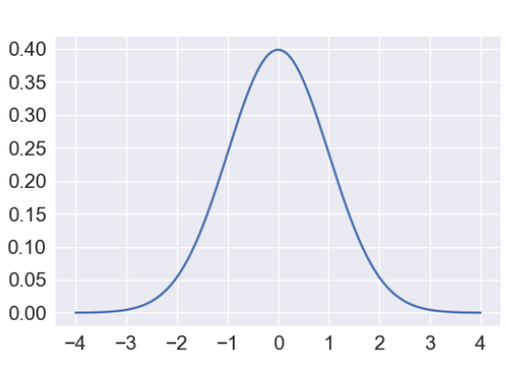
Batch Normalization recap







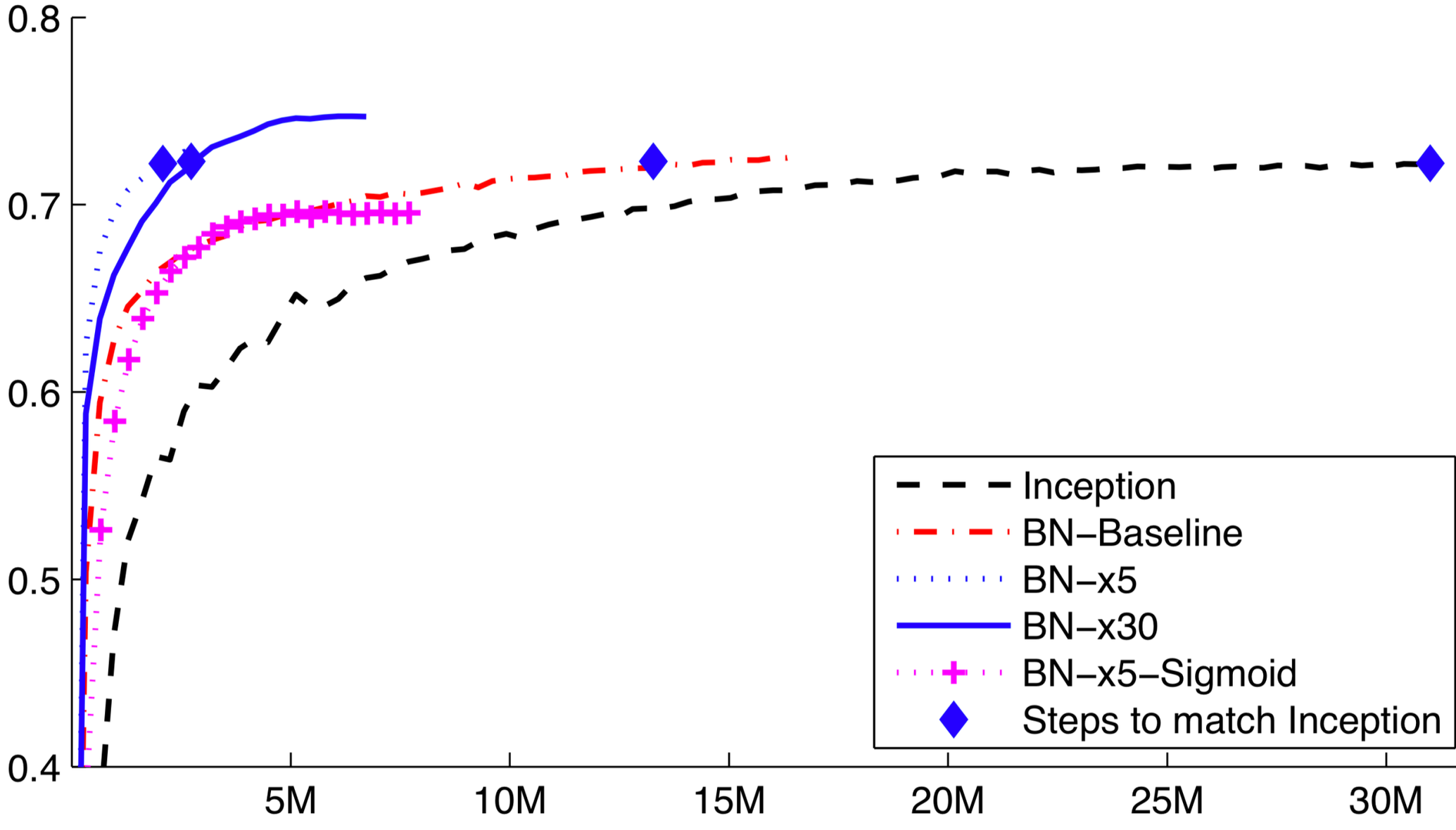
BN



FC90

Success

Batch Normalization recap



"Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Sergey Ioffe, Christian Szegedy 2015

Batchnorm: Probabilistic View

$$\mathrm{BN}^{\mathrm{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta$$

Batchnorm: Probabilistic View

$$\mathrm{BN}^{\mathrm{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta$$

Forward pass depends on an entire mini-batch by statistics:

$$p_\theta(y_i \mid x_i, \mathcal{B}_{\setminus i}) = p_\theta(y_i | x_i, \mu(\mathcal{B}), \sigma(\mathcal{B}))$$

Batchnorm: Probabilistic View

$$\mathrm{BN}^{\mathrm{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta$$

Forward pass depends on an entire mini-batch by statistics:

$$p_\theta(y_i \mid x_i, \mathcal{B}_{\setminus i}) = p_\theta(y_i | x_i, \mu(\mathcal{B}), \sigma(\mathcal{B}))$$

The random choice of a mini-batch induce distribution over statistics:

$$p_\theta(\mu, \sigma | x_i) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, p_\theta(\mu, \sigma \mid x_i, \mathcal{B}_{\setminus i}) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, \delta_{\mu(\mathcal{B})}(\mu)\delta_{\sigma(\mathcal{B})}(\sigma)$$

Batchnorm: Probabilistic View

$$\mathrm{BN}^{\mathrm{train}}_{\gamma,\beta}(x_i) = \frac{x_i - \mu(\mathcal{B})}{\sqrt{\sigma^2(\mathcal{B}) + \epsilon}} \cdot \gamma + \beta$$

Forward pass depends on an entire mini-batch by statistics:

$$p_\theta(y_i \mid x_i, \mathcal{B}_{\setminus i}) = p_\theta(y_i | x_i, \mu(\mathcal{B}), \sigma(\mathcal{B}))$$

The random choice of a mini-batch induce distribution over statistics:

$$p_\theta(\mu, \sigma | x_i) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, p_\theta(\mu, \sigma \mid x_i, \mathcal{B}_{\setminus i}) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, \delta_{\mu(\mathcal{B})}(\mu)\delta_{\sigma(\mathcal{B})}(\sigma)$$

Marginal likelihood:

$$p_\theta(y|x) = \mathbb{E}_{p_\theta(\mu,\sigma|x)} p_\theta(y|x,\mu,\sigma)$$

Batchnorm: Probabilistic View

MLE optimization problem:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(y_i|x_i) = \sum_{i=1}^{N} \log \mathbb{E}_{p_\theta(\mu,\sigma|x_i)}\, p_\theta(y_i|x_i,\mu,\sigma)$$

Batchnorm: Probabilistic View

MLE optimization problem:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(y_i|x_i) = \sum_{i=1}^{N} \log \mathbb{E}_{p_\theta(\mu,\sigma|x_i)}\, p_\theta(y_i|x_i,\mu,\sigma)$$

Lower bound using Jensen-Shannon inequality:

$$\mathcal{L}_{\text{BN}}(\theta) = \sum_{i=1}^{N} \mathbb{E}_{\mu,\sigma} \log p_\theta(y_i|x_i,\mu,\sigma) \leq \sum_{i=1}^{N} \log \mathbb{E}_{\mu,\sigma}\, p_\theta(y_i|x_i,\mu,\sigma) = \mathcal{L}(\theta)$$

$$\mathbb{E}_{\mu,\sigma} \log p_\theta(y_i|x_i,\mu,\sigma) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\ \log p_\theta(y_i|x_i,\mu(\mathcal{B}),\sigma(\mathcal{B}))$$

Batchnorm: Probabilistic View

MLE optimization problem:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(y_i|x_i) = \sum_{i=1}^{N} \log \mathbb{E}_{p_\theta(\mu,\sigma|x_i)}\, p_\theta(y_i|x_i,\mu,\sigma)$$

Lower bound using Jensen-Shannon inequality:

$$\mathcal{L}_{\text{BN}}(\theta) = \sum_{i=1} \mathbb{E}_{\mu,\sigma} \log p_\theta(y_i|x_i,\mu,\sigma) \leq \sum_{i=1} \log \mathbb{E}_{\mu,\sigma}\, p_\theta(y_i|x_i,\mu,\sigma) = \mathcal{L}(\theta)$$

$$\mathbb{E}_{\mu,\sigma} \log p_\theta(y_i|x_i,\mu,\sigma) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\ \log p_\theta(y_i|x_i,\mu(\mathcal{B}),\sigma(\mathcal{B}))$$

Batch Normalization performs unbiased estimation of gradients :

$$\nabla \hat{\mathcal{L}}_{\text{BN}}(\theta) = \frac{N}{M} \sum_{i=1}^{M} \nabla \log p_\theta(y_i|x_i,\mu(\mathcal{B}),\sigma(\mathcal{B}))$$

$$\mathbb{E}_{\mathcal{B}} \nabla \hat{\mathcal{L}}_{\text{BN}}(\theta) = \mathbb{E}_{x_i} \mathbb{E}_{\mathcal{B}_{\setminus i}} \nabla \hat{\mathcal{L}}_{\text{BN}}(\theta) = \nabla \mathcal{L}_{\text{BN}}(\theta)$$

Batchnorm: Probabilistic View

$$\mathrm{BN}^{\mathrm{test}}_{\gamma,\beta}(x_i) = \frac{x_i - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \epsilon}} \cdot \gamma + \beta$$

Proper predictive distribution:

$$p_\theta(y|x) = \mathbb{E}_{p_\theta(\mu,\sigma|x)} p(y|x,\mu,\sigma)$$

However Batchnorm uses running averaged statistics $\hat{\mu}, \hat{\sigma^2}$:

$$\mathbb{E}\mu \approx \hat{\mu}, \quad \mathbb{E}\sigma \approx \hat{\sigma}$$

$$\mathbb{E}_{p_\theta(\mu,\sigma|x_i)} p(y_i|x_i,\mu,\sigma) \approx p_\theta(y|x,\mathbb{E}\mu,\mathbb{E}\sigma)$$

Batchnorm: Probabilistic View

$$p_\theta(y|x) = \mathbb{E}_{p_\theta(\mu,\sigma|x)} p(y|x,\mu,\sigma)$$

Mini-batch parametrization:

$$\mathbb{E}_{\mu,\sigma} p_\theta(y_i|x_i,\mu,\sigma) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, p_\theta(y_i|x_i,\mu(\mathcal{B}),\sigma(\mathcal{B}))$$

Batchnorm: Probabilistic View

$$p_\theta(y|x) = \mathbb{E}_{p_\theta(\mu,\sigma|x)} p(y|x,\mu,\sigma)$$

Mini-batch parametrization:

$$\mathbb{E}_{\mu,\sigma} p_\theta(y_i|x_i,\mu,\sigma) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, p_\theta(y_i|x_i,\mu(\mathcal{B}),\sigma(\mathcal{B}))$$

Unbiased Monte Carlo estimation:

$$\mathbb{E}_{p_\theta(\mu,\sigma|x_i)}\, p_\theta(y_i|x_i,\mu,\sigma) \approx \frac{1}{S}\sum_{i=1}^{S} p_\theta(y_i|x_i,\mu(\mathcal{B}_i),\sigma(\mathcal{B}_i))$$

Batchnorm: Probabilistic View

$$p_\theta(y|x) = \mathbb{E}_{p_\theta(\mu,\sigma|x)} p(y|x,\mu,\sigma)$$

Mini-batch parametrization:

$$\mathbb{E}_{\mu,\sigma} p_\theta(y_i|x_i,\mu,\sigma) = \mathbb{E}_{\mathcal{B}_{\setminus i}}\, p_\theta(y_i|x_i,\mu(\mathcal{B}),\sigma(\mathcal{B}))$$

Unbiased Monte Carlo estimation:

$$\mathbb{E}_{p_\theta(\mu,\sigma|x_i)}\, p_\theta(y_i|x_i,\mu,\sigma) \approx \frac{1}{S}\sum_{i=1}^{S} p_\theta(y_i|x_i,\mu(\mathcal{B}_i),\sigma(\mathcal{B}_i))$$

- S forward passes through the network
- Access to the training data during inference

Stochastic Batch Normalization

Approximate true distribution by parametric one:

$$p_\theta(\mu, \sigma \mid x_i) \approx r(\mu) r(\sigma)$$

$$r(\mu) = \mathcal{N}(\mu | \mathrm{m}_\mu, \mathrm{s}_\mu^2) \quad r(\sigma) = \mathrm{Log}\mathcal{N}(\sigma | \mathrm{m}_\sigma, \mathrm{s}_\sigma^2)$$

Stochastic Batch Normalization

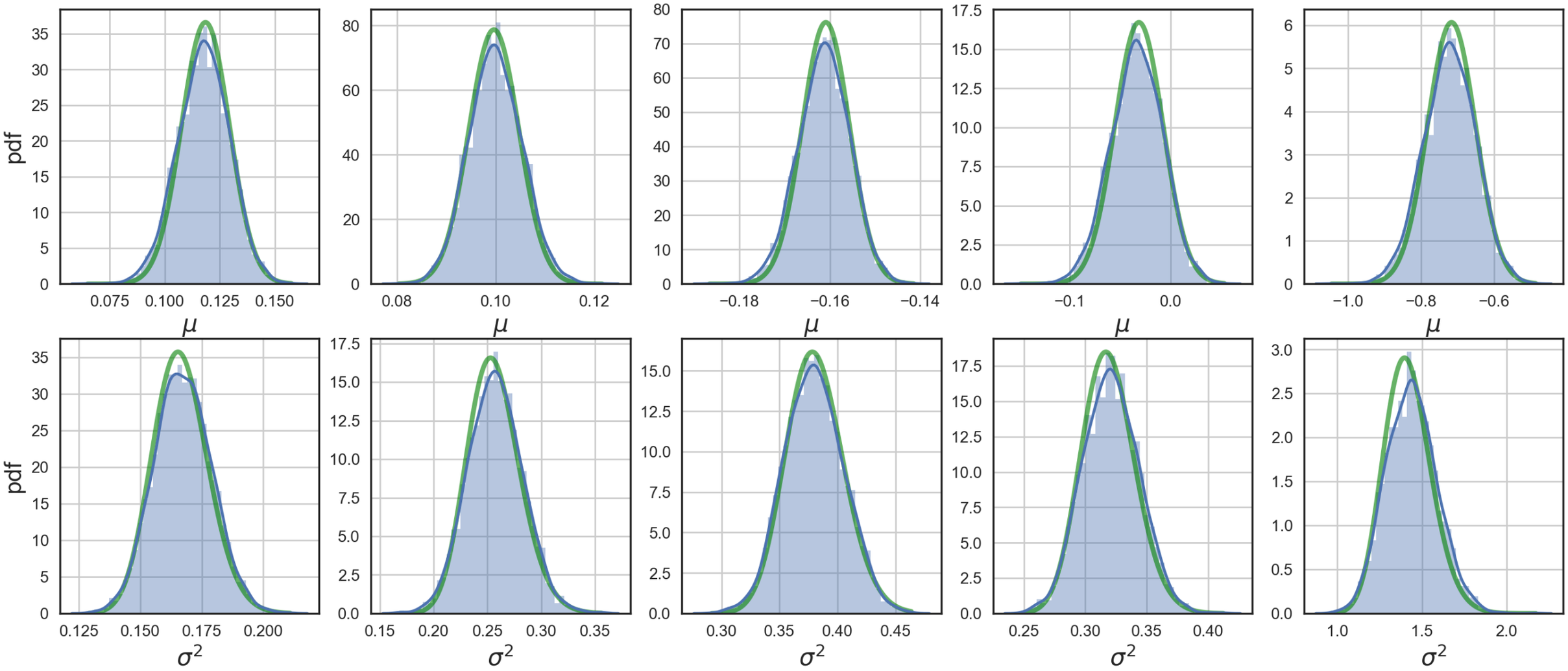
Approximate true distribution by parametric one:

$$p_\theta(\mu, \sigma \,|\, x_i) \approx r(\mu) r(\sigma)$$

$$r(\mu) = \mathcal{N}(\mu | \mathrm{m}_\mu, \mathrm{s}_\mu^2) \quad r(\sigma) = \mathrm{Log}\mathcal{N}(\sigma | \mathrm{m}_\sigma, \mathrm{s}_\sigma^2)$$

Minimize the KL-divergency between averaged statistics distribution and approximation:

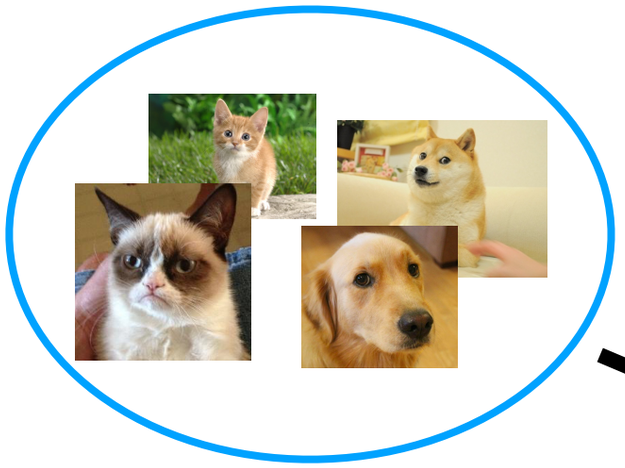
$$D_{\mathrm{KL}} \left(\frac{1}{N} \sum\nolimits_{i=1}^{N} p_\theta(\mu, \sigma \,|\, x_i) \,\middle|\middle|\, r(\mu) r(\sigma) \right) \longrightarrow \min_{m_\mu, s_\mu, m_\sigma, s_\sigma}$$



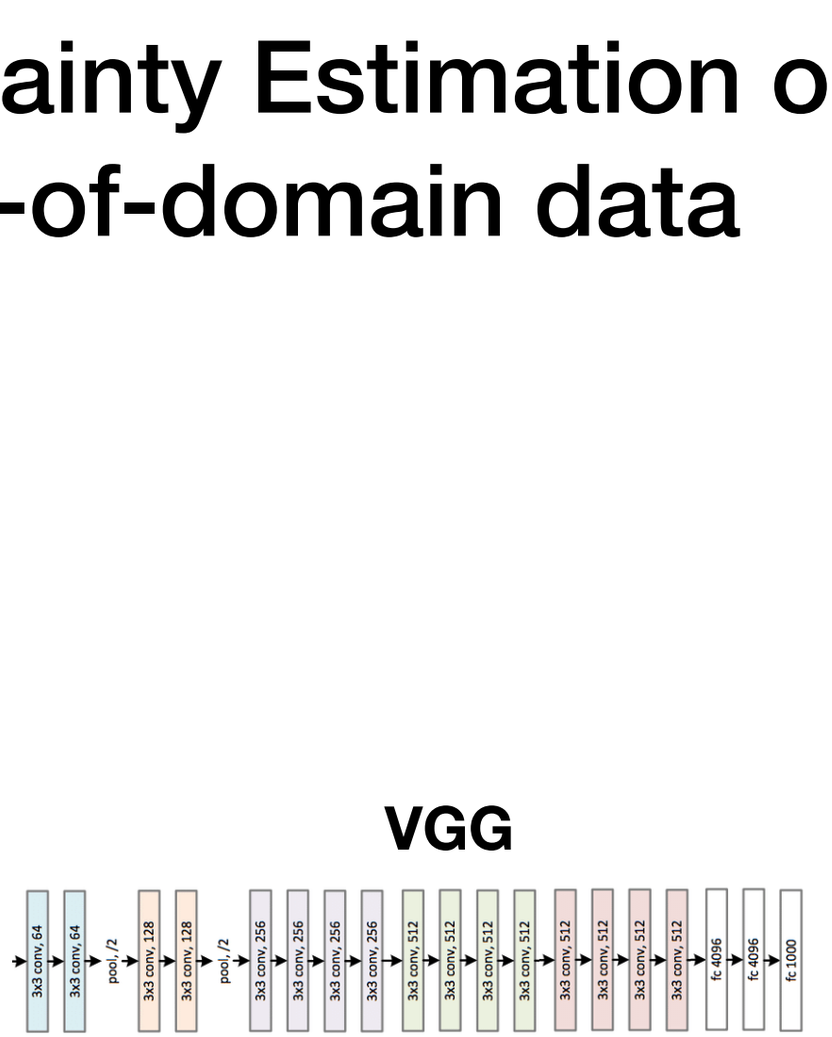
Marginals of true distribution (blue) and approximation (green)

Uncertainty Estimation on out-of-domain data

Training domain



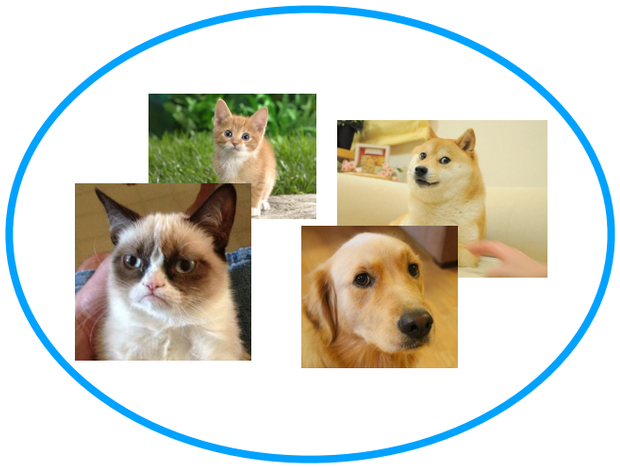
VGG



Out-of-domain data

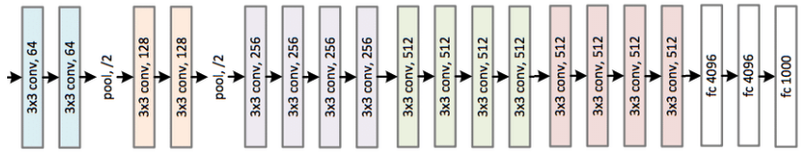
Uncertainty Estimation on out-of-domain data

Training domain



Out-of-domain data

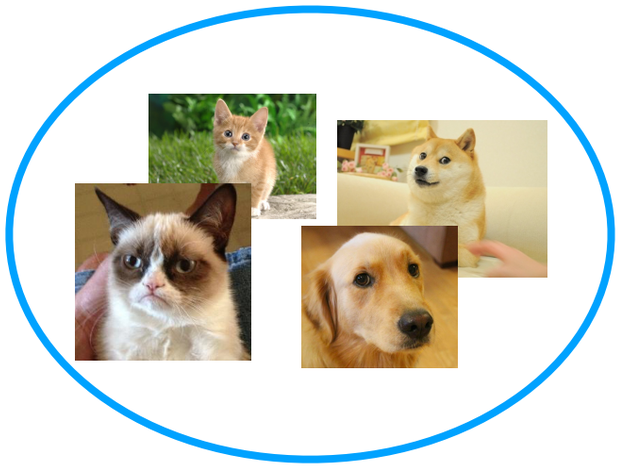
VGG



Cat **Dog**

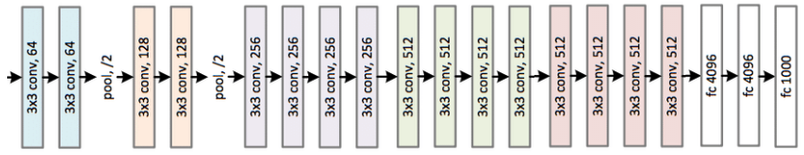
Uncertainty Estimation on out-of-domain data

Training domain



Out-of-domain data

VGG



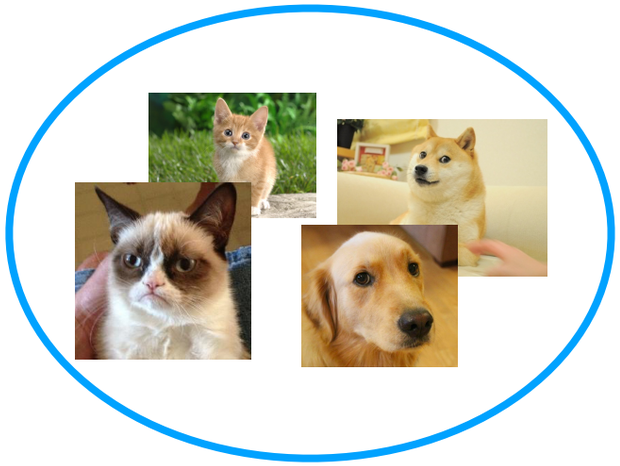
Cat **Dog**

\+

- Dropout
- MNF

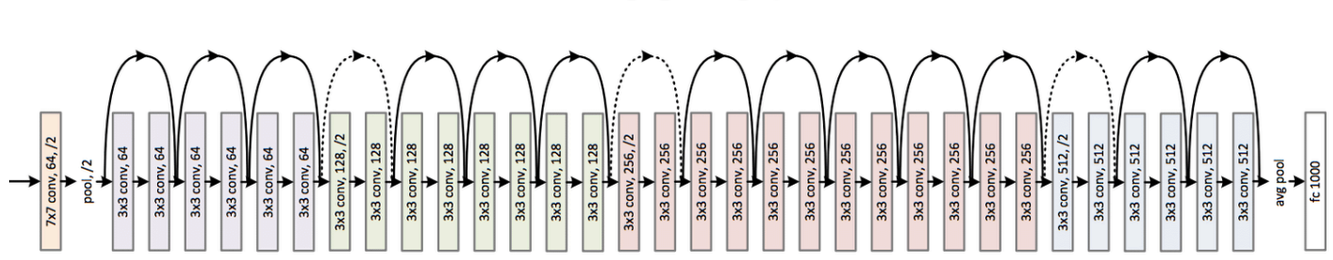
Uncertainty Estimation on out-of-domain data

Training domain



Out-of-domain data

ResNet

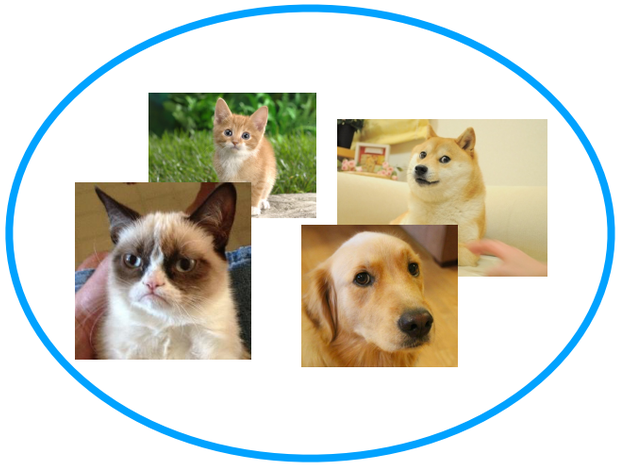


+

- ~~Dropout~~
- ~~MNF~~

Uncertainty Estimation on out-of-domain data

Training domain



Out-of-domain data

ResNet

Cat **Dog**

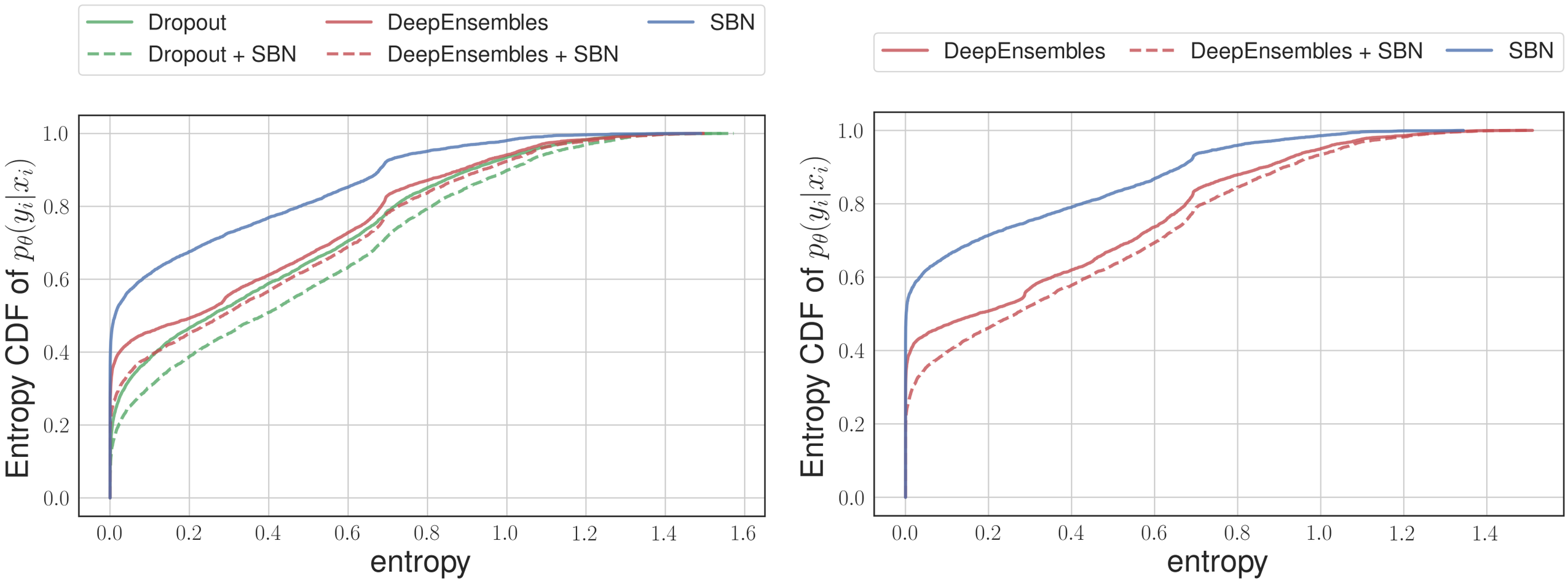
+

~~• Dropout~~
~~• MNF~~

+

Stochastic Batch Normalization

Stochastic Batch Normalization



Empirical predictive distribution entropy CDF for CIFAR5

Stochastic Batch Normalization

Network	Method	Error%		NLL	
		No SBN	SBN	No SBN	SBN
LeNet-5 MNIST	SBN	—	0.53 ± 0.05	—	0.025 ± 0.003
	Deep Ensembles	0.43 ± 0.00	0.43 ± 0.00	0.015 ± 0.001	0.014 ± 0.001
	Dropout	0.51 ± 0.00	0.49 ± 0.00	0.016 ± 0.000	0.015 ± 0.000
VGG-11 CIFAR5	SBN	—	5.76 ± 0.00	—	0.302 ± 0.002
	Deep Ensembles	$\mathbf{5.18 \pm 0.00}$	5.23 ± 0.00	0.177 ± 0.004	$\mathbf{0.154 \pm 0.002}$
	Dropout	$\mathbf{5.32 \pm 0.00}$	5.38 ± 0.00	0.155 ± 0.001	$\mathbf{0.149 \pm 0.001}$
ResNet-18 CIFAR5	SBN	—	4.35 ± 0.17	—	0.255 ± 0.018
	Deep Ensembles	$\mathbf{3.37 \pm 0.00}$	3.34 ± 0.00	0.138 ± 0.005	$\mathbf{0.110 \pm 0.004}$

Table 1: Test errors (%) and NLL scores for known classes. MNIST for LeNet-5 and CIFAR5 for VGG-11 and ResNet-18. SBN column correspond to methods with all Batch Normalization layers replaced by ours SBN.

Uncertainty estimation via Stochastic Batch Normalization

We proposed:

- Probabilistic model for Batch Normalization.
- Scaleable approximation technique for inference phase.
- Simple and efficient method for uncertainty estimation.

- Check out our paper on arXiv: https://goo.gl/FCMTGQ
- Code available at https://goo.gl/TfMxni
- Submission on ICLR Workshop track: https://goo.gl/tvHyoy