ハイパーネットによる
畳み込みニューラルネットワークの暗黙的事後分布推定

○鵜飼健矢・松原　崇・上原邦昭（神戸大）

Outline

- We introduce a novel **regularization method for large CNNs**
  - ✓ This estimates the **posterior** of the parameters **implicitly** by **hypernetworks**.
- By estimating the posteriors,
  - ✓ Probabilistic behavior of the parameters **regularizes** the training.
  - ✓ We can perform **model averaging** in the inference phase.
- In the experiment,
  our method **improved** image classification accuracy.

Background : Regularization

- Deep neural networks have a rich ability to learn complex representations.
- However, they are prone to **overfitting** due to the limited number of training samples.

Regularizing the learning process of neural networks is essential.

Background : Parameter Estimation

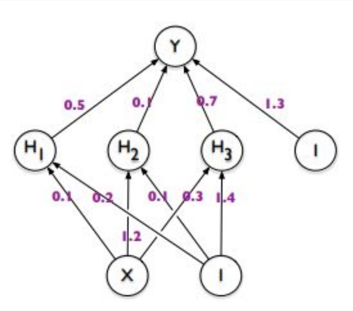
In bayesian statistics,

regularization and **parameter estimation** is deeply connected.

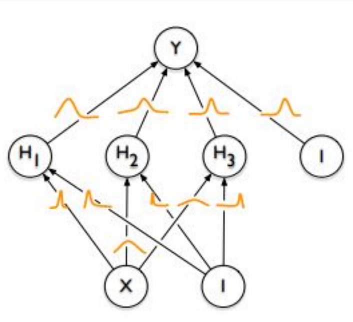
- Training of Typical Probabilistic Models :
  - estimate the **value** of the Parameters
  - MLE, MAP estimation

$$\mathbf{w}^{\text{MLE}} = \arg\max_{\mathbf{w}} \log P(\mathcal{D}|\mathbf{w}) = \arg\max_{\mathbf{w}} \sum_i \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}).$$

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}} \log P(\mathbf{w}|\mathcal{D}) = \arg\max_{\mathbf{w}} \log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w}).$$



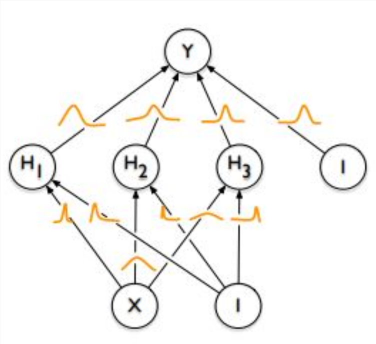
- Bayesian Probabilistic Models:
  - estimate the **posterior distribution** of the parameters
  - Able to treat uncertainty
  - Able to perform bayesian model averaging



Related works : Bayesian Neural Nets

- Typical training: minimize Kullback-Leibler divergence between
  - ***P(w|D)***: true posterior
  - ***q(w|θ)***: approximation posterior

$$
\begin{aligned}
\theta^\star &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w}|\mathcal{D})] \\
&= \arg\min_\theta \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})} \mathrm{d}\mathbf{w} \\
&= \arg\min_\theta \mathrm{KL}\left[q(\mathbf{w}|\theta) \;||\; P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right].
\end{aligned}
$$

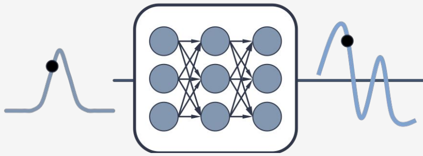


- Bayes by backprop [Blundell et al, 2015]
  - Factorized Gaussian prior ***P(w)*** & posterior ***q(w|θ)***
  - All the parameters are assumed to be **independent**

Related works : Bayesian Neural Nets

- Typical training: minimize Kullback-Leibler divergence between
  - ***P(w|D)***: true posterior
  - ***q(w|θ)***: approximation posterior

$$
\begin{aligned}
\theta^{\star} &= \arg\min_{\theta} \mathrm{KL}[q(\mathbf{w}|\theta) || P(\mathbf{w}|\mathcal{D})] \\
&= \arg\min_{\theta} \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})} \mathrm{d}\mathbf{w} \\
&= \arg\min_{\theta} \mathrm{KL}\left[q(\mathbf{w}|\theta) \;||\; P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right].
\end{aligned}
$$



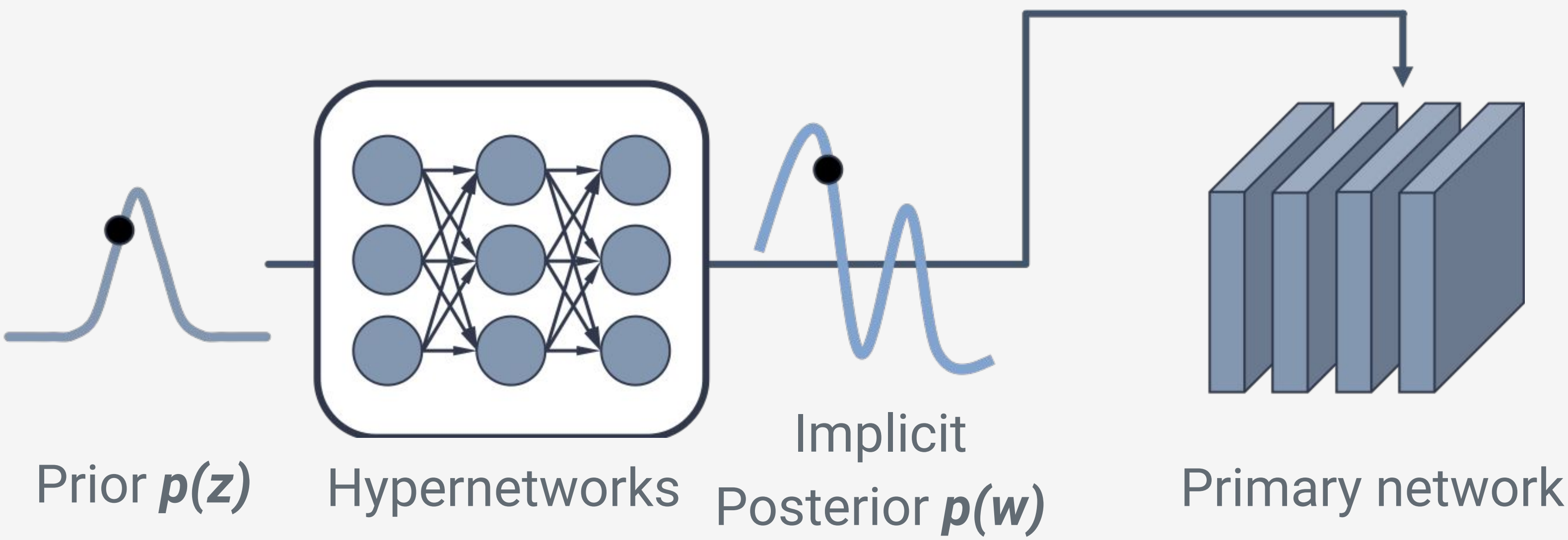
- Bayesian hypernetworks [Krueger et al, 2017]
  - Flexible **correlated** posterior *q(w/θ)* by hypernetworks
  - Problems for large scale CNNs because of the special-structured hypernetworks and Weight Normalization

Hypernetwork-based Posterior estimation of large scale CNNs is not accomplished.

Our Methods

- Approximate the **implicit posterior** by **hypernetworks**
- Simply **maximize the target likelihood** directly
  to relax the restriction of the hypernetworks

$$\cancel{\arg\min_{\theta} \mathrm{KL}\left[q(\mathbf{w}|\theta) \,||\, P(\mathbf{w})\right]} - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right].$$



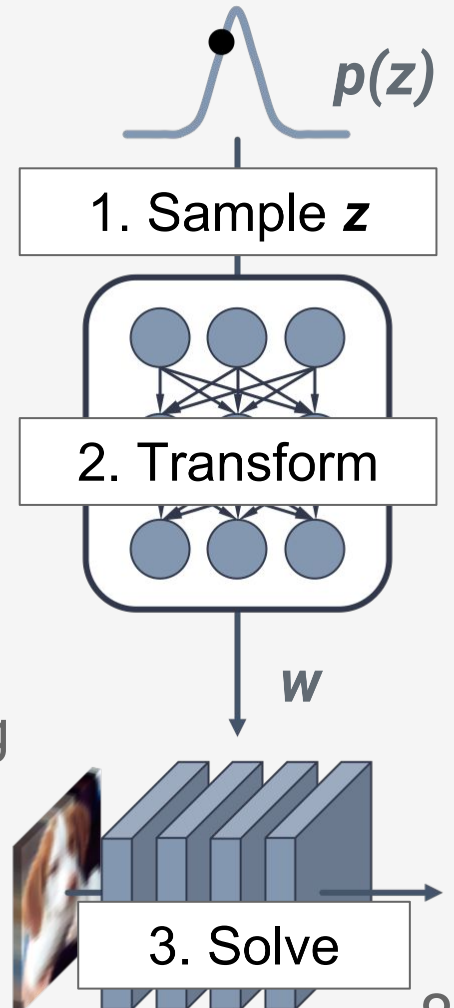
Our Methods

Procedure

1. Sample ***z*** from prior ***p(z)***
2. Transform ***z*** by hypernet ***w = g(z; θ)***
3. Solve the task with the parameter ***w***
4. Update ***θ*** by backpropagation

By using our methods,

- ✓ Probabilistic behavior **regularizes** the training
- ✓ Able to perform **model averaging**
- ✓ Able to **apply large scale CNNs**



Experimental Results - Settings

- We demonstrate the regularization effects by **classification accuracy**
- We examined
  - Effect of the prior *p(z)*
  - Learned posterior of the parameters
- We evaluated the methods using
  - Networks: WideResNet, ResNeXT, Pyramidal ResNet
  - Datasets: CIFAR10, SVHN, Imagenet

Classification Errors - CIFAR10

Methods	WideResNet 28-4		WideResNet 28-10	
	x1	x16	x1	x16
MLE (No weight decay)	6.05%	—	5.49%	—
MAP estimation (Weight decay)	4.23%	—	3.90%	—
Ours Posterior estimation	**4.21%**	**4.19%**	**3.76%**	**3.73%**

Our method **improved** image classification accuracy.

Classification Errors - CIFAR10

Methods	ResNeXT 29-8-64d		Pyramidal ResNet 110-48	
	x1	x16	x1	x16
MAP estimation (Weight decay)	4.03%	—	4.59%	—
Ours Posterior estimation	**3.92%**	**3.91%**	4.64%	4.61%

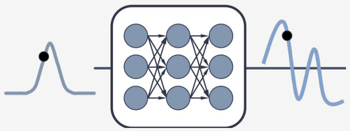
Our method is applicable to large scale & redundant structure

such as WideResNet, ResNeXT

Effect of the prior p(z) - CIFAR10

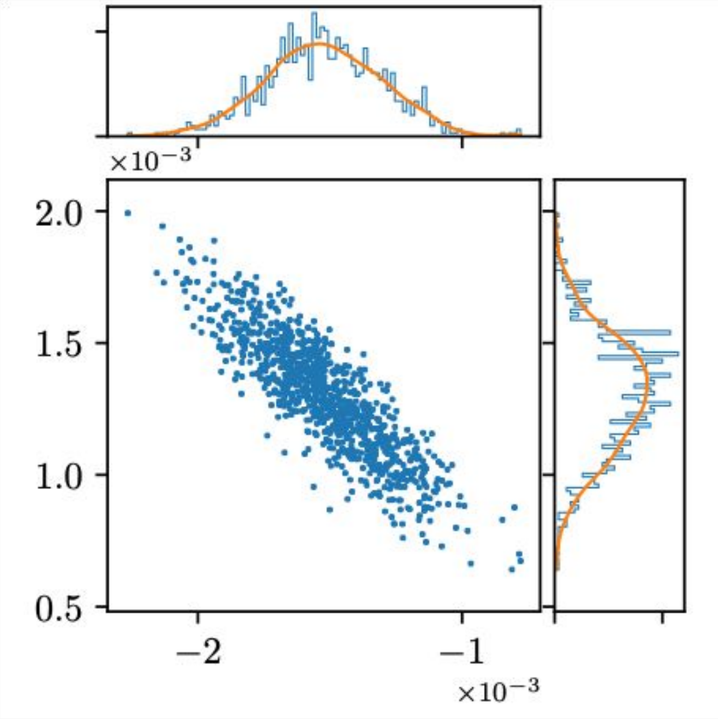
We compared classification errors with various *p(z)*

- U(0, 1) : better regularization at x1
- N(0, 1) : model averaging is effective



Methods	*p(z)*	WideResNet 28-4	
		x1	x16
MAP estimation (Weight decay)	—	4.23%	—
Ours Posterior estimation	*U(0, 1)*	**4.21%**	4.19%
	N(0, 1)	4.65%	**4.06%**

Learned Posteriors



Samples of randomly chosen two parameters

- Two parameters are **correlated**.
- Complicated posterior

We also examined the effectiveness of correlation in detail in the paper.

Conclusion

- We introduced a novel **regularization method for large CNNs**
  - ✓ This estimates the **posterior** of the parameters **implicitly** by **hypernetworks**.
- By estimating the posteriors,
  - ✓ Probabilistic behavior of the parameters **regularizes** the training.
  - ✓ We can perform **model averaging** in the inference phase.
- In the experiment,
  our method **improved** image classification accuracy.

Appendix

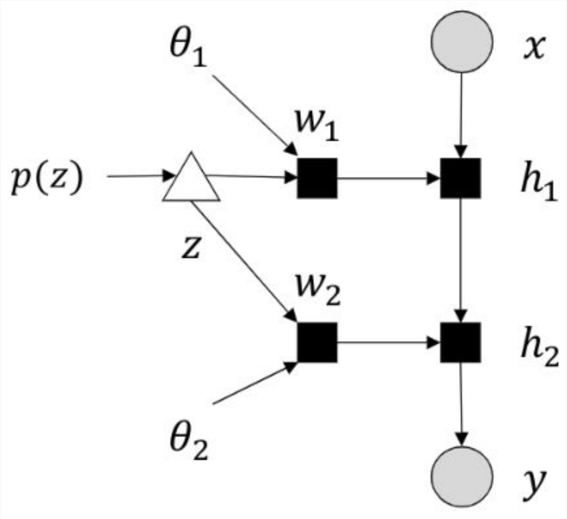
Classification Errors - Strategies and Priors

Methods	Strategy	prior	WideResNet 28-4		WideResNet 28-10	
			x1	x16	x1	x16
MLE	—	—	6.05%	—	5.49%	—
MAP	—	—	4.23%	—	3.90%	—
hypernet	all-in-one	N(0, 1)	4.65%	**4.06%**	4.13%	**3.79%**
	all-in-one	U(0, 1)	**4.21%**	**4.19%**	**3.76%**	**3.73%**
	block-wise	N(0, 1)	4.70%	**4.03%**	4.42%	**3.85%**
	block-wise	U(0, 1)	4.34%	4.34%	4.02%	3.97%

Correlation of the posteriors

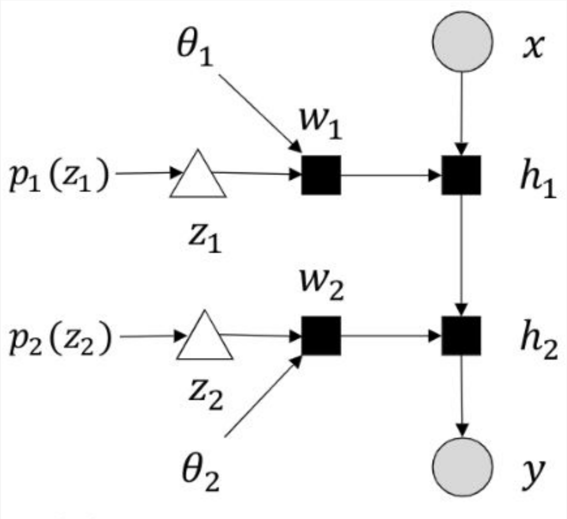
We compared two models to examine the effect of **correlation**

All-in-one strategy



Share prior among all layers

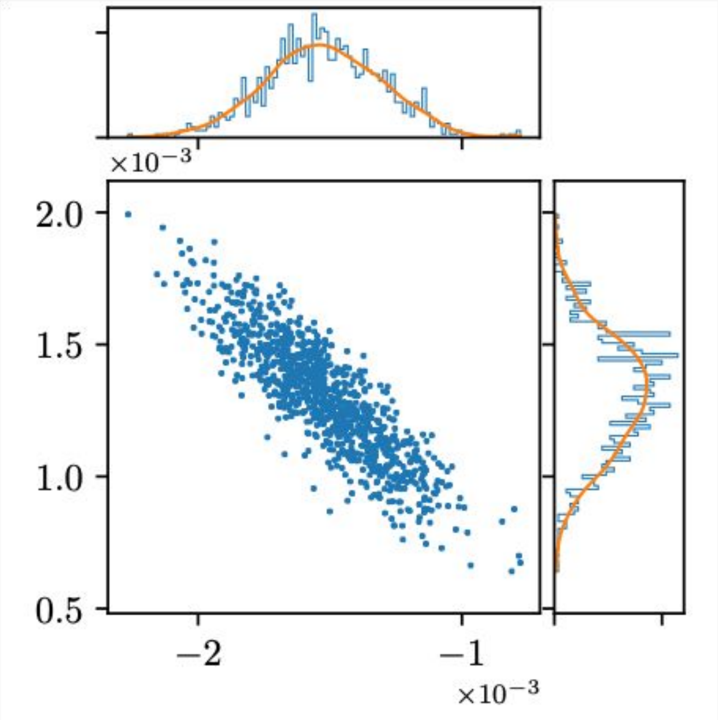
Block-wise strategy



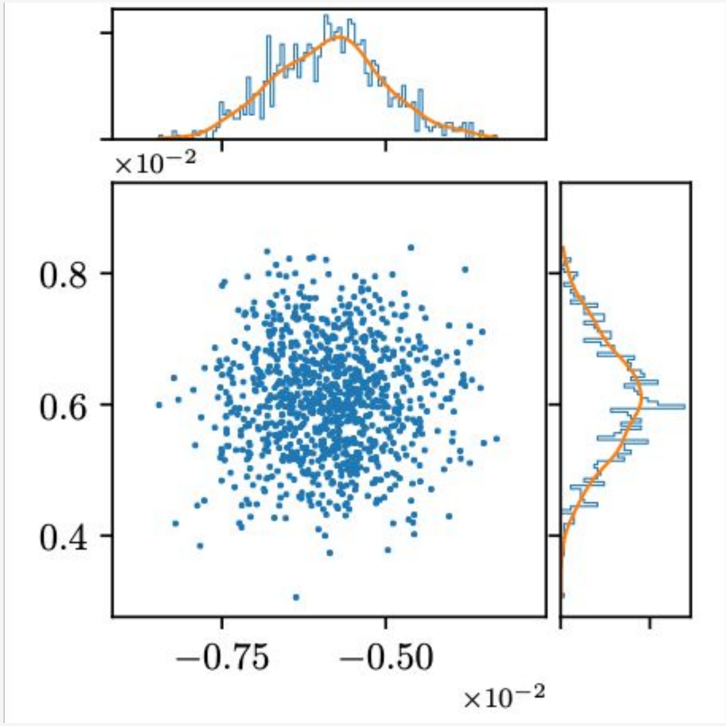
Share prior within each ResBlock

Difference of Correlation

All-in-one



Block-wise



Samples of two parameters randomly chosen from different ResBlocks.

Classification Accuracy - Prior p(z)

prior $p(z)$	$\times 1$	$\times 16$
$\mathcal{U}(0,1)$	4.21%	4.19%
$\mathcal{U}(-1,1)$	4.74%	4.22%
$\mathcal{N}(0,1)$	4.65%	4.06%
$\lvert\mathcal{N}(0,1)\rvert$	4.64%	4.58%

Classification Accuracy - SVHN & ImageNet

Table 5: Test error rates on SVHN for WideResNet16-4.

Methods	prior $p(z)$	$\times 1$	$\times 16$
MAP	—	1.90%	—
hypernetwork	$\mathcal{U}(0,1)$	1.93%	1.93%
	$\mathcal{N}(0,1)$	1.90%	**1.80%**

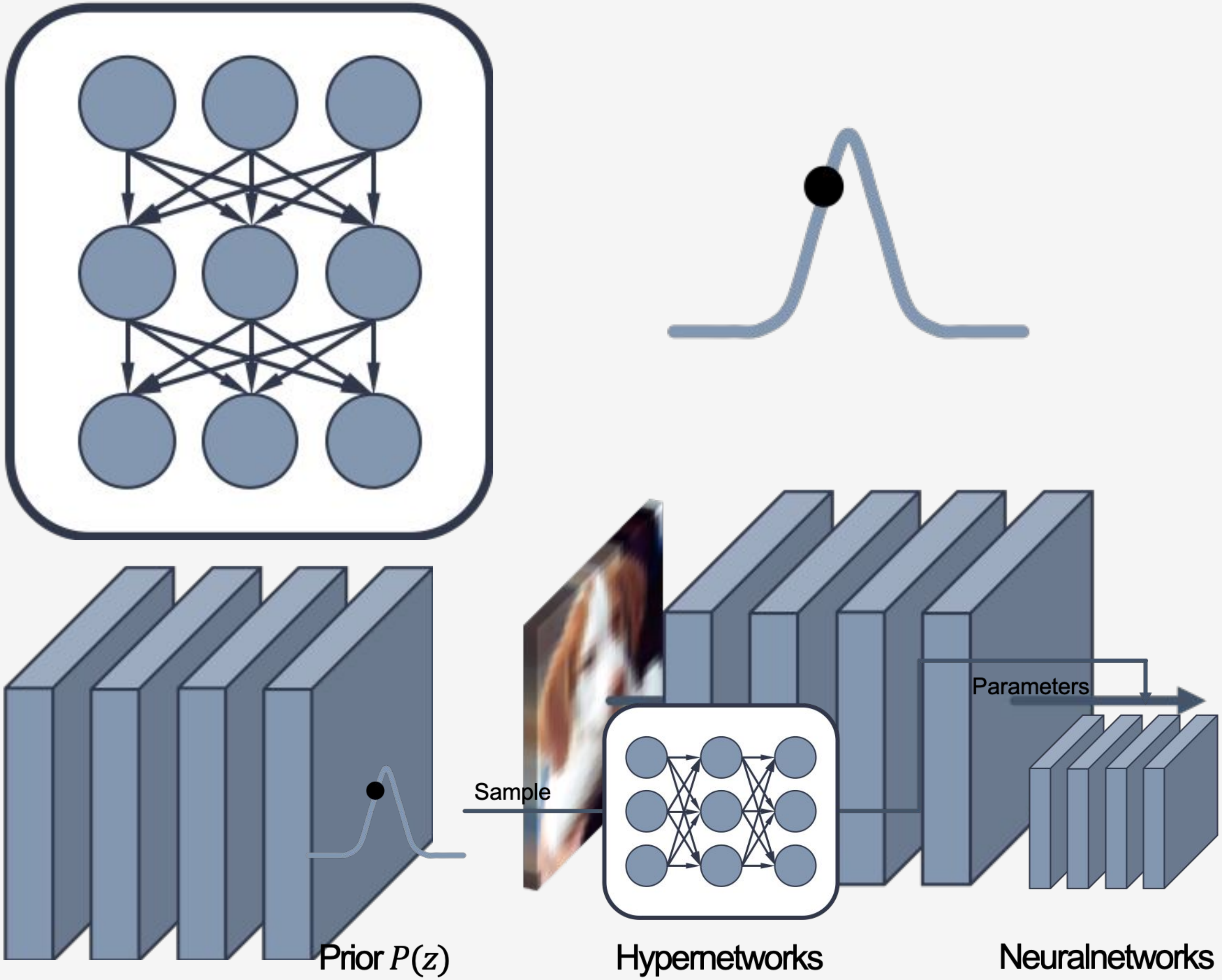
Table 6: Test error rates on ImageNet for ResNet50 with the prior of $U(0,1)$.

	Top 1		Top 5	
Methods	$\times 1$	$\times 16$	$\times 1$	$\times 16$
MAP	23.84%	—	7.06%	—
hypernetwork	25.97%	25.87%	8.17%	8.16%

Classification Accuracy - Various CNNs

	Pyramidal ResNet-110-48		ResNeXt-29-8-64d	
Methods	$\times 1$	$\times 16$	$\times 1$	$\times 16$
MAP	4.59%	—	4.03%	—
hypernetwork	4.64%	4.61%	**3.92%**	**3.91%**

Works better with ResNeXT
not better with Pyramidal ResNet


Parameters
Sample
Prior $P(z)$
Hypernetworks
Neuralnetworks