Learning for Single-Shot Confidence Calibration in Deep Neural Networks through Stochastic Inferences



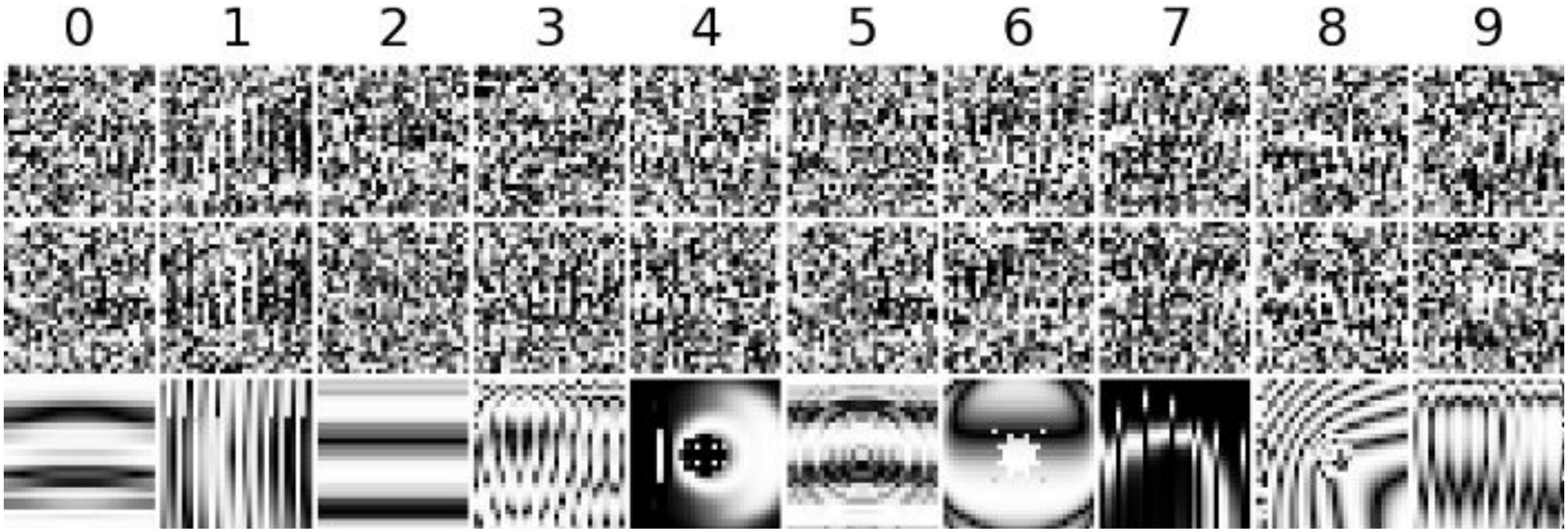
Seonguk Seo*[1] Paul Hongsuck Seo*[1,2] Bohyung Han[1]

서 울 대 학 교
SEOUL NATIONAL UNIVERSITY

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Overconfidence Issues

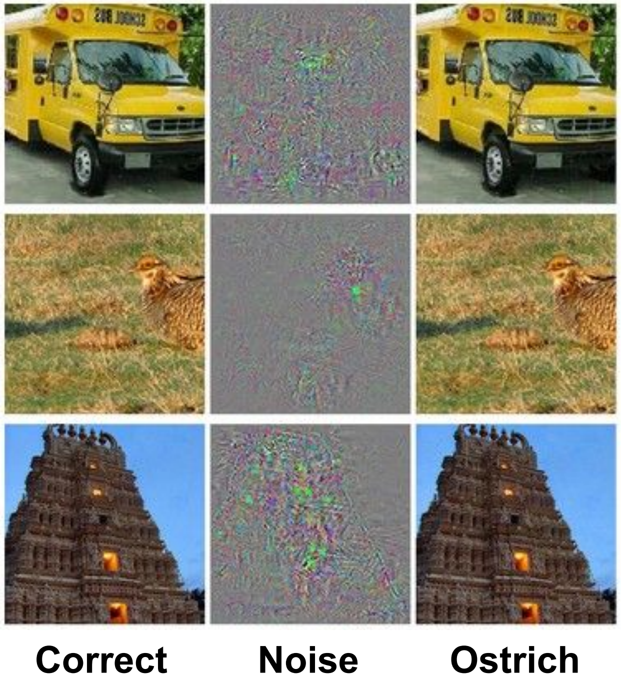
- Overconfidence to unseen examples
  - 99.9+% sure for the following predictions

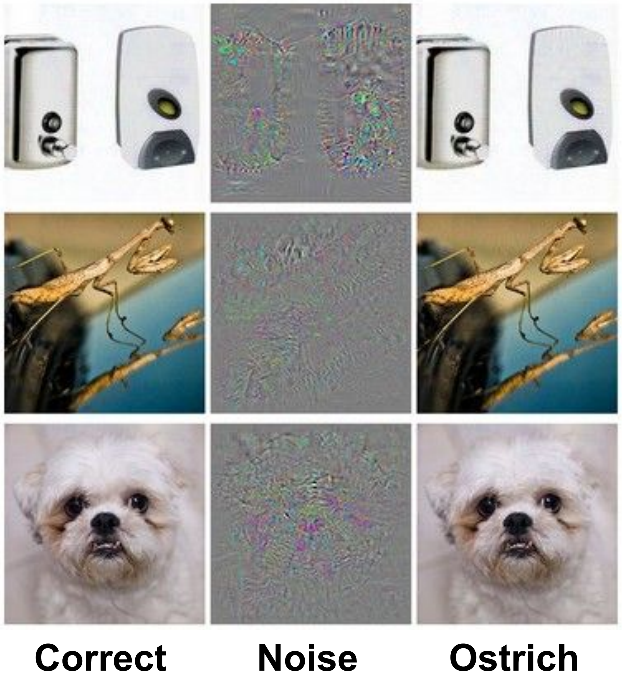


[Nguyen15] A. Nguyen, J. Yosinski, J. Clune: **Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images**. CVPR 2015

Vulnerability

- Vulnerability to noise

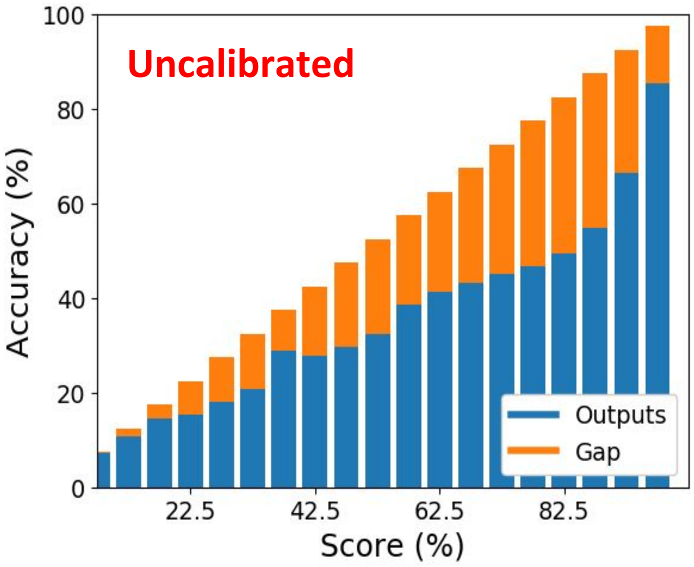


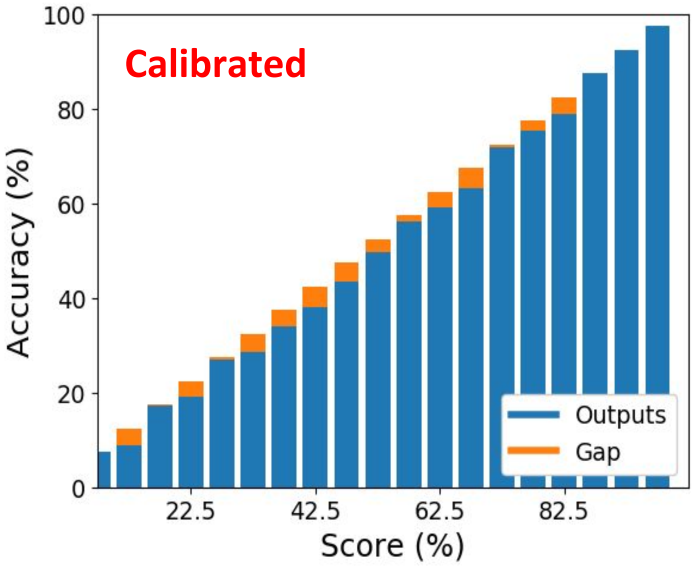


[Szegedy14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus: **Intriguing properties of neural networks**. ICLR 2014

Goals

- Confidence calibration
  - Reducing the discrepancy between confidence (score) and expected accuracy
  - Adopting idea of stochastic regularization





Stochastic Regularization

- Regularization by noise: reducing overfitting problem by adding noise (randomness) to data or models
  - Noise injection to training data
  - Dropout[Srivastava14]
  - DropConnect[Wan13]
  - Learning with stochastic depth[Huang16]

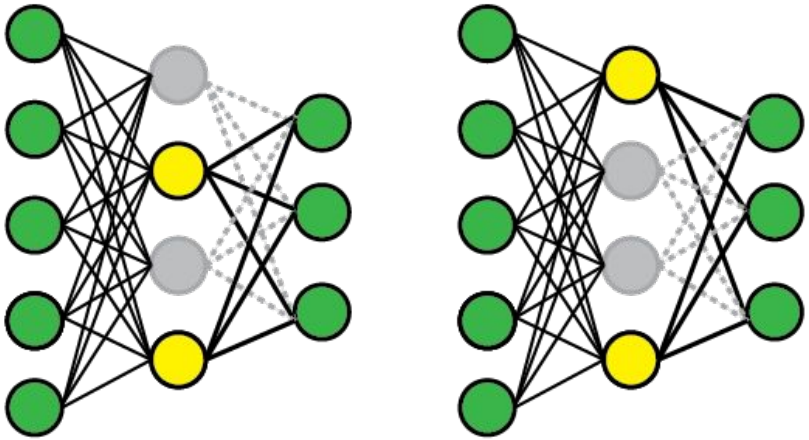
[Srivastava14] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov: **Dropout: a simple way to prevent neural networks from overfitting.** JMLR 2014
[Wan13] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, R. Fergus. **Regularization of neural networks using dropconnect**. ICML 2013
[Huang16] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Q. Weinberger: **Deep networks with stochastic depth**. ECCV 2016

Stochastic Regularization

- Objective (in classification)
  - Perturbing parameters by element-wise multiplication during training

$$\hat{\mathcal{L}}_{\mathrm{SR}}(\theta) = -\frac{1}{M}\sum_{i=1}^{M}\log p\left(y_i|x_i,\hat{\omega}_i\right) + \lambda||\theta||_2^2 \quad \text{where} \quad \hat{\omega}_i = \theta \odot \epsilon_i$$

- Dropout



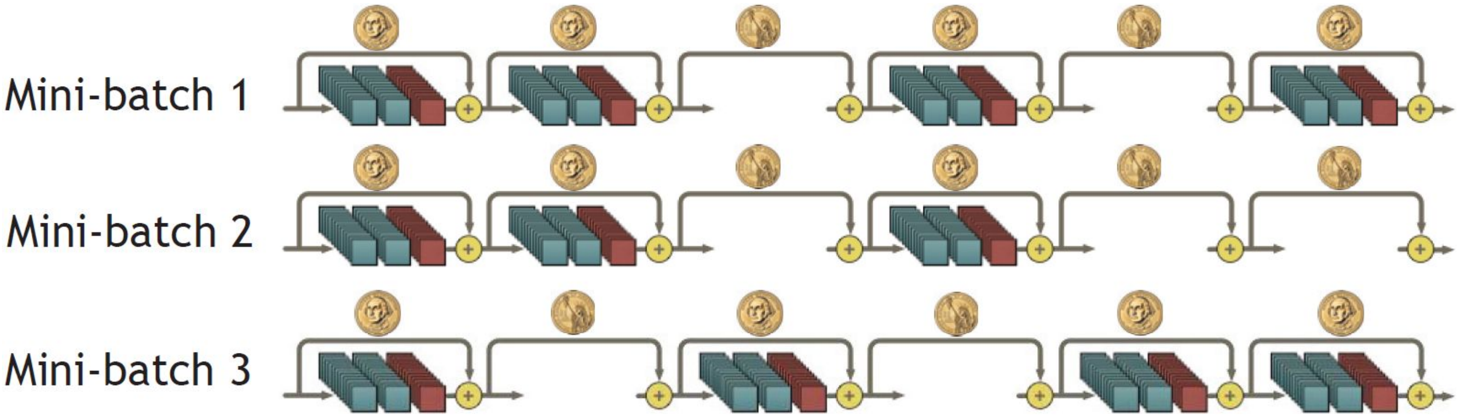
[Srivastava14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov: **Dropout: A Simple Way to Prevent Neural Networks from Overfitting**. JMLR 2014

Stochastic Regularization

- Objective (in classification)
  - Perturbing parameters by element-wise multiplication during training

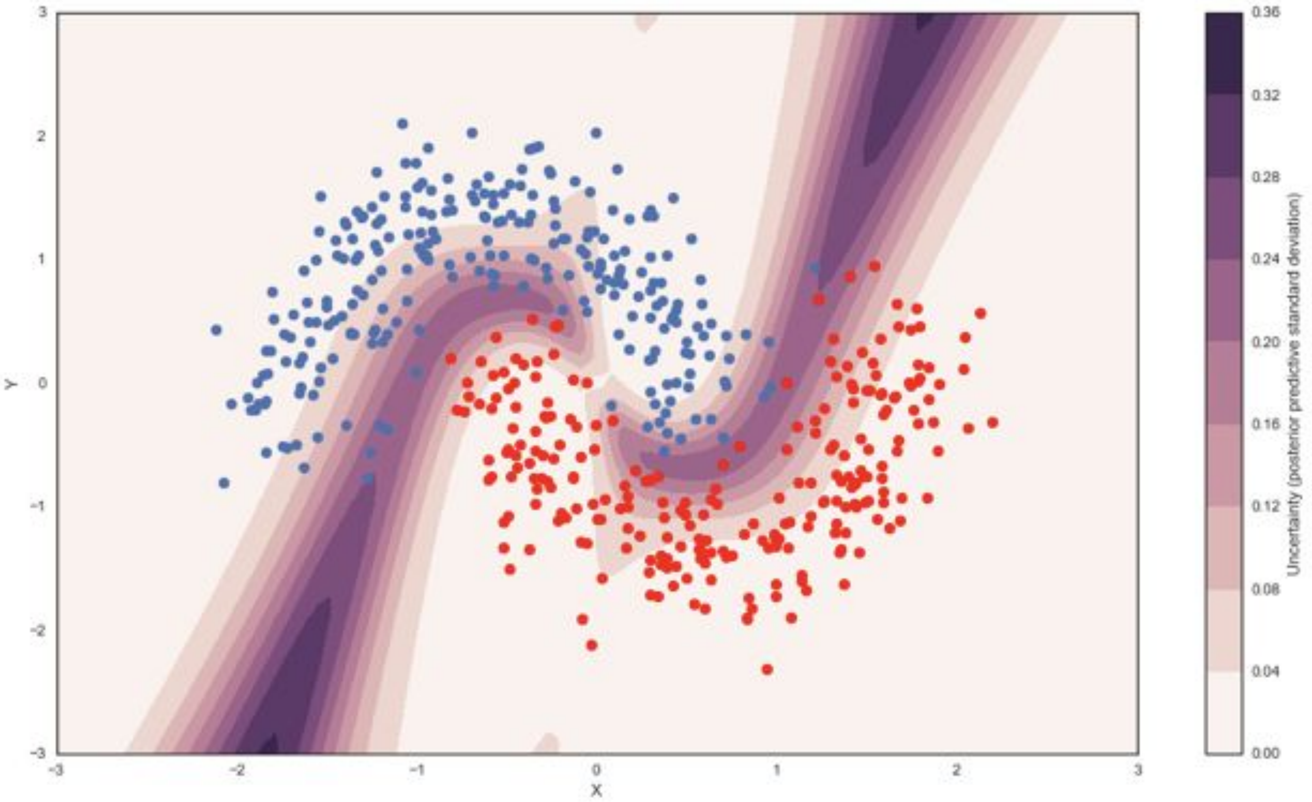
$$\hat{\mathcal{L}}_{\mathrm{SR}}(\theta) = -\frac{1}{M}\sum_{i=1}^{M} \log p\left(y_i | x_i, \hat{\omega}_i\right) + \lambda ||\theta||_2^2 \quad \text{where} \quad \hat{\omega}_i = \theta \odot \epsilon_i$$

- Stochastic depth



[Huang16] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Weinberger: **Deep Networks with Stochastic Depth**. ECCV 2016

Uncertainty in Deep Neural Networks



Bayesian Uncertainty Estimation

- Integrating stochastic regularization techniques for inferences
    - Dropout, stochastic depth, etc.
    - Individual inferences produce different outputs.

- Uncertainty can be measured by multiple stochastic inferences.

[Gal16] Y. Gal and Z. Ghahramani. **Dropout as a Bayesian approximation: Representing model uncertainty in deep learning.** ICML 2016

Bayesian Uncertainty Estimation

- Bayesian interpretation of stochastic regularization
  - Learning objective: maximizing marginal likelihood by estimating posterior $p(\omega|\mathcal{D})$

$$p(y|x,\mathcal{D}) = \int_{\omega} p(y|x,\omega)p(\omega|D)d\omega.$$

  - Variational approximation (but intractable integration)

$$\mathcal{L}_{\mathrm{VA}}(\theta) = -\sum_{i=1}^{N}\int_{\omega} q_{\theta}(\omega)\log p(y_i|x_i,\omega)d\omega + D_{\mathrm{KL}}(q_{\theta}(\omega)||p(\omega))$$

  - Variational approximation with Monte Carlo: by sampling $\hat{\omega}_{i,j} \sim q_{\theta}(\omega)$

$$\hat{\mathcal{L}}_{\mathrm{VA}}(\theta) = -\frac{N}{MS}\sum_{i=1}^{M}\sum_{j=1}^{S}\log p\left(y_i|x_i,\hat{\omega}_{i,j}\right) + D_{\mathrm{KL}}\left(q_{\theta}(\omega)||p(\omega)\right)$$

[Gal16] Y. Gal and Z. Ghahramani. **Dropout as a Bayesian approximation: Representing model uncertainty in deep learning.** ICML 2016

Bayesian Uncertainty Estimation

- Bayesian interpretation of stochastic regularization
  - Variational approximation with Monte Carlo: by sampling $\hat{\omega}_{i,j} \sim q_\theta(\omega)$

$$\hat{\mathcal{L}}_{\mathrm{VA}}(\theta) = -\frac{N}{MS}\sum_{i=1}^{M}\sum_{j=1}^{S}\log p\left(y_i|x_i,\hat{\omega}_{i,j}\right) + D_{\mathrm{KL}}\left(q_\theta(\omega)||p(\omega)\right)$$

  - Learning with stochastic regularization with weight decay: same objective with Gaussian assumption of true and approximated posteriors

$$\hat{\mathcal{L}}_{\mathrm{SR}}(\theta) = -\frac{1}{M}\sum_{i=1}^{M}\log p\left(y_i|x_i,\hat{\omega}_i\right) + \lambda||\theta||_2^2,$$

- **The average prediction and its uncertainty can be computed directly from multiple stochastic inferences**.

$$\mathbb{E}_{\hat{p}}[y=c] \approx \frac{1}{T}\sum_{i=1}^{T}\hat{p}(y=c|x,\hat{\omega}_i) \quad \text{and} \quad \mathrm{Cov}_{\hat{p}}[\mathbf{y}] \approx \mathbb{E}_{\hat{p}}[\mathbf{y}\mathbf{y}^\intercal] - \mathbb{E}_{\hat{p}}[\mathbf{y}]\mathbb{E}_{\hat{p}}[\mathbf{y}]^\intercal$$

[Gal16] Y. Gal and Z. Ghahramani. **Dropout as a Bayesian approximation: Representing model uncertainty in deep learning.** ICML 2016

Bayesian Uncertainty Estimation

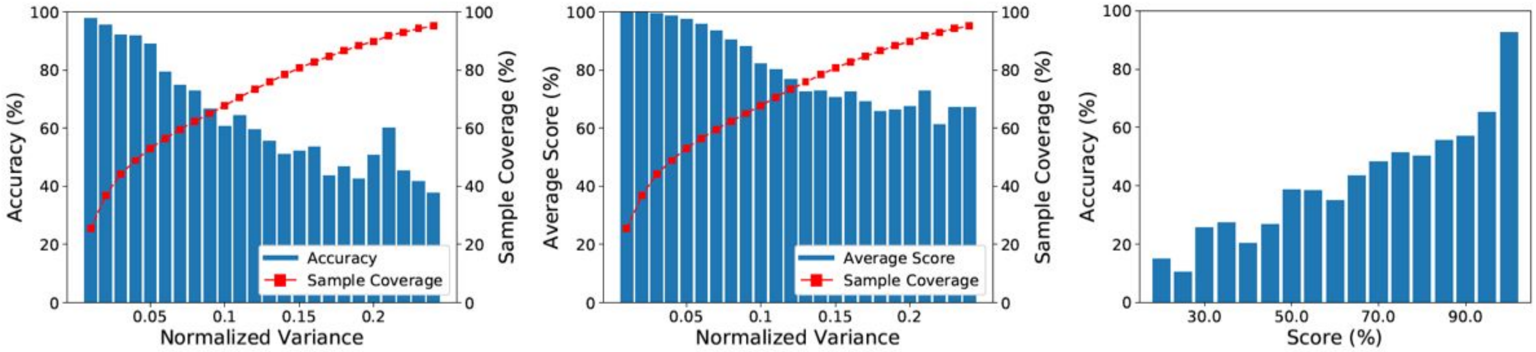
- Integrating stochastic regularization techniques for inferences
  - Dropout, stochastic depth, etc.
  - Individual inferences produce different outputs.

- Uncertainty can be measured by multiple stochastic inferences.

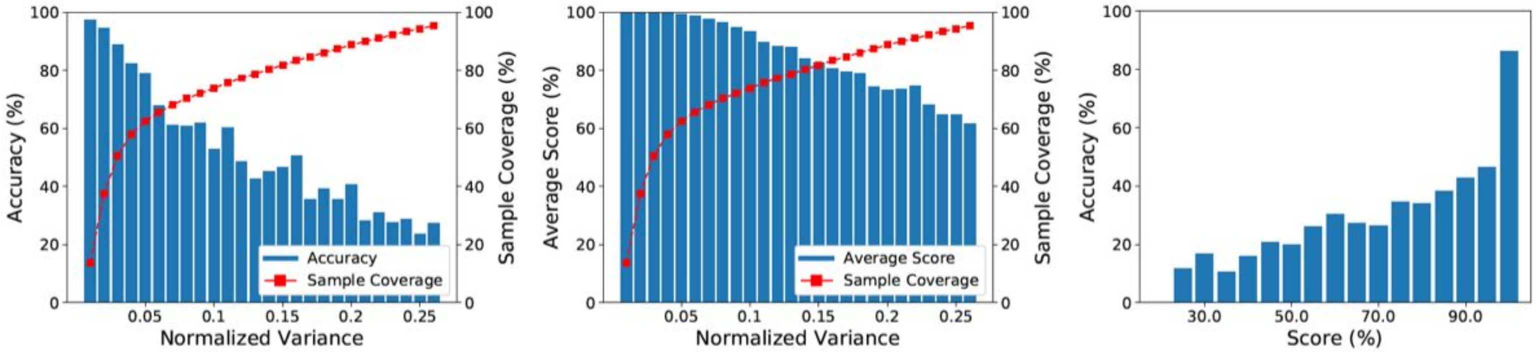
The uncertainty of a prediction can be estimated using the variation of multiple stochastic inferences.

[Gal16] Y. Gal and Z. Ghahramani. **Dropout as a Bayesian approximation: Representing model uncertainty in deep learning.** ICML 2016

Empirical Observations



(a) Prediction uncertainty characteristics with stochastic depth in ResNet-34



(b) Prediction uncertainty characteristics with dropout in VGGNet with 16 layers

Uncertainty through Stochastic Inferences

- Limitation of the simple uncertainty estimation method by multiple stochastic inferences
  - Requires multiple inferences for each example
- Solution
  - Designing a loss function to learn uncertainty
  - Exploiting multiple stochastic inferences results for training
  - Learning a model for the single-shot confidence calibration
- Desired score distribution
  - Confident examples have prediction scores close to one-hot vectors.
  - Uncertain examples produce relatively flat score distributions.

We propose a loss function to make the confidence (the prediction score) proportional to the expected accuracy.

Confidence-Integrated Loss

- A naive loss function for accuracy-score calibration
  - A linear combination of two loss terms with respect to ground-truth and uniform distribution
  - Blindly augmenting a loss term with a uniform distribution

$$\begin{aligned}\mathcal{L}(\theta) &= \mathcal{L}_{\mathrm{GT}}(\theta) + \beta \mathcal{L}_{\mathrm{U}}(\theta) \\ &= \sum_{i=1}^{N} H(p_{\mathrm{GT}}(y_i|x_i), p(y|x_i,\theta)) + \beta H(\mathcal{U}(y), p(y|x_i,\theta)) \\ &= \sum_{i=1}^{N} \underbrace{\boxed{-\log p(y_i|x_i,\theta)}}_{\text{Accuracy term}} + \beta \underbrace{\boxed{D_{\mathrm{KL}}(\mathcal{U}(y)||p(y|x_i,\theta))}}_{\text{Confidence term}} + \xi.\end{aligned}$$

Confidence-Integrated Loss

- The same loss functions are discussed for different purposes
  - [Pereyra17]: for accuracy improved via regularization
  - [Lee18]: for identifying out-of-distribution examples
  - No attempt to estimate the confidence of predictions

$$
\begin{aligned}
\mathcal{L}(\theta) &= \mathcal{L}_{\mathrm{GT}}(\theta) + \beta \mathcal{L}_{\mathrm{U}}(\theta) \\
&= \sum_{i=1}^{N} H(p_{\mathrm{GT}}(y_i|x_i), p(y|x_i,\theta)) + \beta H(\mathcal{U}(y), p(y|x_i,\theta)) \\
&= \sum_{i=1}^{N} -\log p(y_i|x_i,\theta) + \beta D_{\mathrm{KL}}(\mathcal{U}(y)||p(y|x_i,\theta)) + \xi.
\end{aligned}
$$

[Pereyra17] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, G. Hinton. **Regularizing neural networks by penalizing confident output distributions**. arXiv 2017
[Lee18] K. Lee, H. Lee, K. Lee, J. Shin. **Training confidence- calibrated classifiers for detecting out-of-distribution samples**. ICLR 2018

Confidence-Integrated Loss

- A simple loss function for accuracy-score calibration
  - All samples have the same weight of the confidence loss term regardless of example-specific characteristics.
  - Interpretation of this loss function is very hard.
  - Needs for a global hyper-parameter β

$$\begin{aligned}\mathcal{L}(\theta) &= \mathcal{L}_{\mathrm{GT}}(\theta) + \beta \mathcal{L}_{\mathrm{U}}(\theta) \\ &= \sum_{i=1}^{N} H(p_{\mathrm{GT}}(y_i|x_i), p(y|x_i, \theta)) + \beta H(\mathcal{U}(y), p(y|x_i, \theta)) \\ &= \sum_{i=1}^{N} -\log p(y_i|x_i, \theta) + \beta D_{\mathrm{KL}}(\mathcal{U}(y) || p(y|x_i, \theta)) + \xi.\end{aligned}$$

Variance-Weighted Confidence-Integrated Loss

- A more sophisticated loss function for accuracy-score calibration
  - An interpolation of two cross-entropy terms
  - The two terms are weighted by the variance of stochastic inferences
  - Generalization of the confidence-integrated loss function

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \boxed{(1-\alpha_i)} \mathcal{L}_{\mathrm{GT}}^{(i)}(\theta) + \boxed{\alpha_i} \mathcal{L}_{\mathrm{U}}^{(i)}(\theta)$$

$$= \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{T} -(1-\alpha_i) \log p(y_i | x_i, \hat{\omega}_{i,j}) + \alpha_i D_{\mathrm{KL}}(\mathcal{U}(y) || p(y|x_i, \hat{\omega}_{i,j})) + \xi_i$$

α_i: normalized variance

Variance-Weighted Confidence-Integrated Loss

- A more sophisticated loss function for accuracy-score calibration
  - Motivated by Bayesian interpretation of stochastic regularization and our empirical observation
  - No hyper-parameter to balance two terms

$$
\begin{aligned}
\mathcal{L}(\theta) &= \sum_{i=1}^{N} \boxed{(1-\alpha_i)} \mathcal{L}_{\mathrm{GT}}^{(i)}(\theta) + \boxed{\alpha_i} \mathcal{L}_{\mathrm{U}}^{(i)}(\theta) \\
&= \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{T} -(1-\alpha_i) \log p(y_i | x_i, \hat{\omega}_{i,j}) + \alpha_i D_{\mathrm{KL}}(\mathcal{U}(y) || p(y | x_i, \hat{\omega}_{i,j})) + \xi_i
\end{aligned}
$$

α_i: normalized variance

Experiments

- Datasets
  - CIFAR-100
  - Tiny ImageNet
- Architectures
  - ResNet
  - VGG
  - WideResNet
  - DenseNet

Experiments

- Evaluation metrics
  - Classification accuracy
  - Calibration scores
    - Expected Calibration Error (ECE): $\mathrm{ECE} = \sum_{m=1}^{M} \frac{B_m}{N'} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|$
    - Maximum Calibration Error (MCE): $\mathrm{MCE} = \max_{m \in \{1,\ldots,M\}} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|$
    - Negative Log Likelihood (NLL): $\mathrm{NLL} = -\sum_{i=1}^{N'} \log p(y_i | x_i, \theta)$
    - Brier Score: $\mathrm{Brier} = -\sum_{i=1}^{N'} \sum_{j=1}^{C} \left(p(y_i = j | x_i, \theta) - \delta(y_i - j) \right)^2$

Results

- On Tiny ImageNet

Dataset	Architecture	Method	Accuracy[%]	ECE	MCE	NLL	Brier Score
Tiny ImageNet	ResNet-34	Baseline	50.82	0.067	0.147	2.050	0.628
		CI	50.09 ± 1.08	0.134 ± 0.079	0.257 ± 0.098	2.270 ± 0.212	0.665 ± 0.037
		VWCI	**52.80**	**0.027**	**0.076**	**1.949**	**0.605**
		CI[Oracle]	51.45	0.035	0.171	2.030	0.620
	VGG-16	Baseline	46.58	0.346	0.595	4.220	0.844
		CI	46.82 ± 0.81	0.226 ± 0.095	0.435 ± 0.107	3.224 ± 0.468	0.761 ± 0.054
		VWCI	**48.03**	**0.053**	**0.142**	**2.373**	**0.659**
		CI[Oracle]	47.39	0.122	0.320	2.812	0.701
	WideResNet-16-8	Baseline	55.92	0.132	0.237	1.974	0.593
		CI	55.80 ± 0.44	0.115 ± 0.040	0.288 ± 0.100	1.980 ± 0.114	0.594 ± 0.017
		VWCI	**56.66**	**0.046**	**0.136**	**1.866**	**0.569**
		CI[Oracle]	56.38	0.050	0.208	1.851	0.572
	DenseNet-40-12	Baseline	42.50	**0.020**	0.154	2.423	0.716
		CI	40.18 ± 1.68	0.059 ± 0.061	0.152 ± 0.082	2.606 ± 0.208	0.748 ± 0.035
		VWCI	**43.25**	0.025	**0.089**	**2.410**	**0.712**
		CI[Oracle]	41.21	0.025	0.094	2.489	0.726

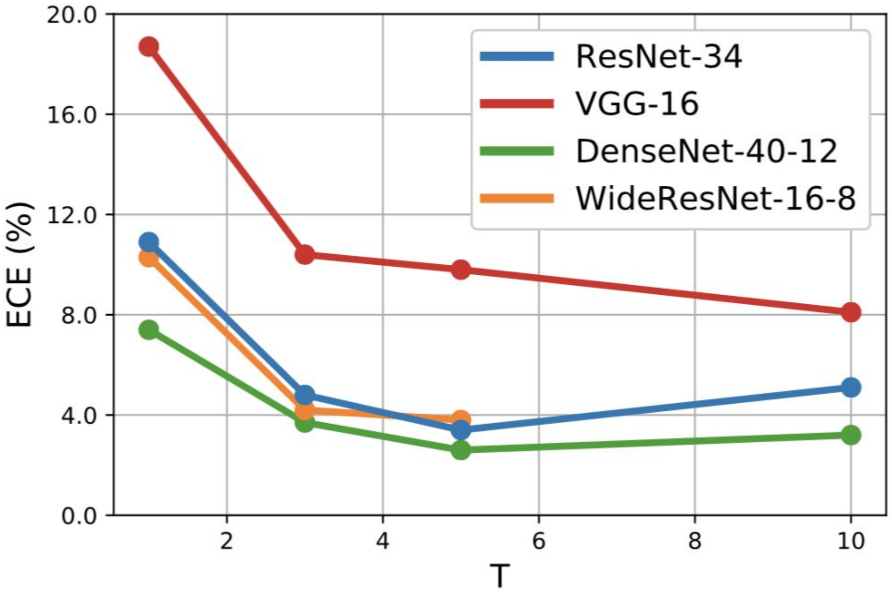
Results

- On Tiny ImageNet

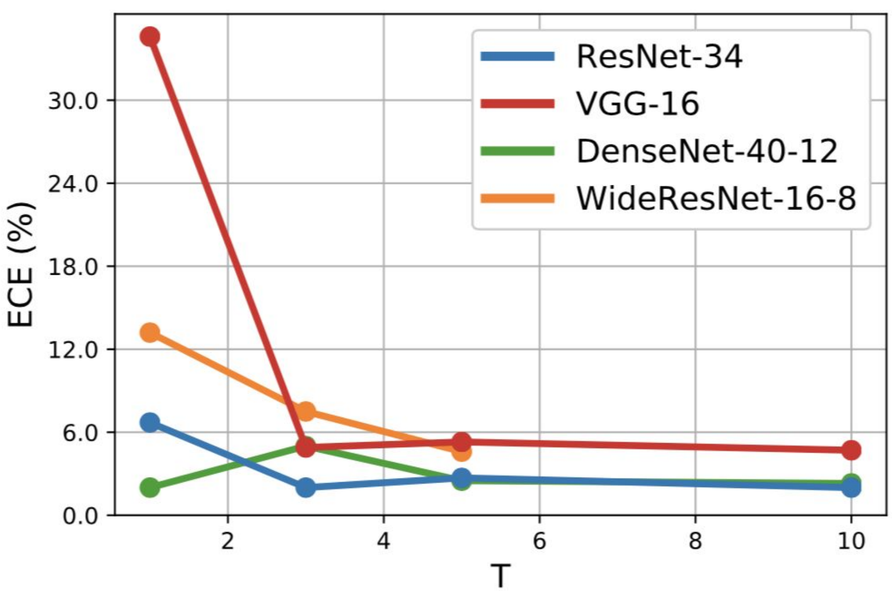
Dataset	Architecture	Method	Accuracy[%]	ECE	MCE	NLL	Brier Score
CIFAR-100	ResNet-34	Baseline	77.19	0.109	0.304	1.020	0.345
		CI	77.56 ± 0.60	0.134 ± 0.131	0.251 ± 0.128	1.064 ± 0.217	0.360 ± 0.057
		VWCI	**78.64**	**0.034**	**0.089**	**0.908**	**0.310**
		CI[Oracle]	78.54	0.029	0.087	0.921	0.321
	VGG-16	Baseline	73.78	0.187	0.486	1.667	0.437
		CI	73.75 ± 0.35	0.183 ± 0.079	0.489 ± 0.214	1.526 ± 0.175	0.436 ± 0.034
		VWCI	**73.87**	**0.098**	**0.309**	**1.277**	**0.391**
		CI[Oracle]	73.78	0.083	0.285	1.289	0.396
	WideResNet-16-8	Baseline	77.52	0.103	0.278	0.984	0.336
		CI	77.35 ± 0.21	0.133 ± 0.091	0.297 ± 0.108	1.062 ± 0.180	0.356 ± 0.044
		VWCI	**77.74**	**0.038**	**0.101**	**0.891**	**0.314**
		CI[Oracle]	77.53	0.074	0.211	0.931	0.327
	DenseNet-40-12	Baseline	65.91	0.074	0.134	1.238	0.463
		CI	64.72 ± 1.46	0.070 ± 0.040	0.138 ± 0.055	1.312 ± 0.125	0.482 ± 0.028
		VWCI	**67.45**	**0.026**	**0.094**	**1.161**	**0.439**
		CI[Oracle]	66.20	0.019	0.053	1.206	0.456

Ablation Study

- Calibration performance w.r.t. the number of stochastic inferences during training



CIFAR-100



Tiny ImageNet

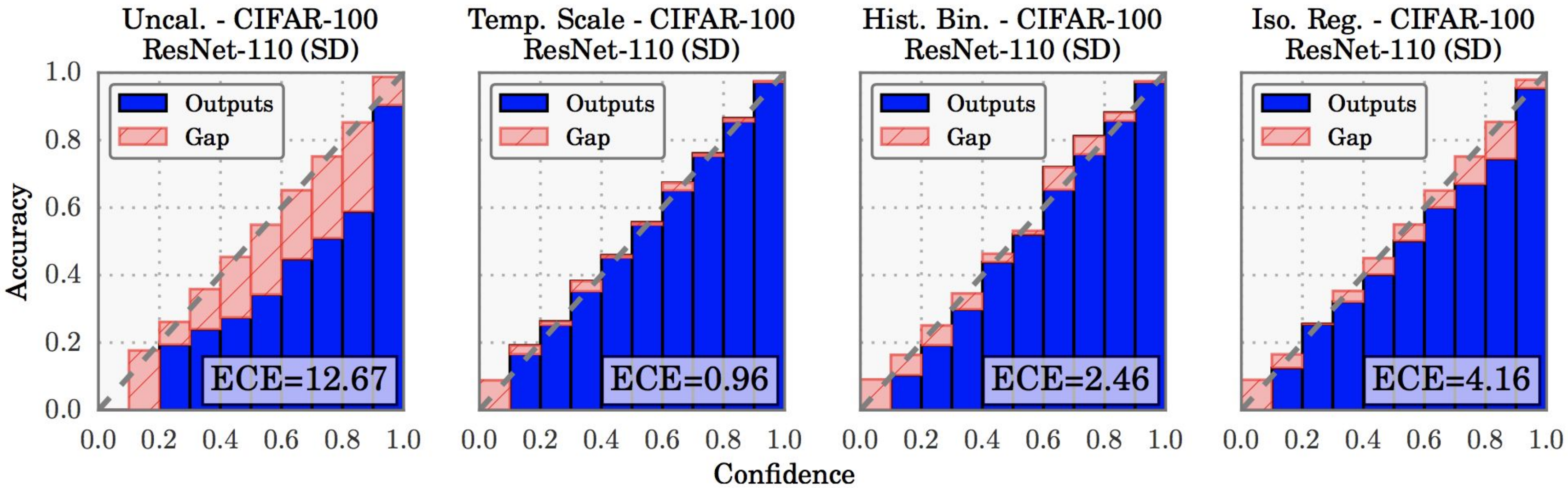
Ablation Study

- Performance of the models fine-tuned with the VWCI losses
  - From the uncalibrated pretrained networks
  - On CIFAR-100
  - About 25% of the additional iterations are sufficient for good calibration.

Architecture	Method	Acc. [%]	ECE	MCE	NLL	Brier
VGG-16	Baseline	73.78	0.187	0.486	1.667	0.437
	VWCI	73.87	0.098	0.309	1.277	0.391
	Baseline $\rightarrow$ VWCI	**74.17**	**0.074**	**0.243**	**1.227**	**0.385**
ResNet-34	Baseline	77.19	0.109	0.304	1.020	0.345
	VWCI	**78.64**	0.034	0.089	**0.908**	**0.310**
	Baseline $\rightarrow$ VWCI	77.87	**0.026**	**0.069**	1.013	0.346

Temperature Scaling

- A simple confidence calibration technique
  - Optimizes temperature of softmax function
  - Simple to implement and train
  - Does not change prediction results



[Guo17] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger: **On Calibration of Modern Neural Networks**. ICML 2017

Results

- Comparison with temperature scaling[Guo17]
  - Case 1: using the entire training set for both training and calibration
  - Case 2: using 90% of training set for training and the rest for calibration
  - It may suffers from binning artifacts

Dataset	Architecture	Method	Accuracy[%]	ECE	MCE	NLL	Brier Score
Tiny ImageNet	ResNet-34	TS (case 1)	50.82	0.162	0.272	2.241	0.660
		TS (case 2)	47.20	**0.021**	0.080	2.159	0.661
		VWCI	**52.80**	0.027	**0.076**	**1.949**	**0.605**
	VGG-16	TS (case 1)	46.58	0.358	0.604	4.425	0.855
		TS (case 2)	46.53	**0.028**	**0.067**	**2.361**	0.671
		VWCI	**48.03**	0.053	0.142	2.373	**0.659**
	WideResNet-16-8	TS (case 1)	55.92	0.200	0.335	2.259	0.627
		TS (case 2)	53.95	**0.027**	0.224	1.925	0.595
		VWCI	**56.66**	0.046	**0.136**	**1.866**	**0.569**
	DenseNet-40-12	TS (case 1)	42.50	0.037	0.456	2.436	0.717
		TS (case 2)	41.63	**0.024**	0.109	2.483	0.728
		VWCI	**43.25**	0.025	**0.089**	**2.410**	**0.712**

[Guo17] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger. **On calibration of modern neural networks**. ICML 2017

Summary on Confidence Calibration

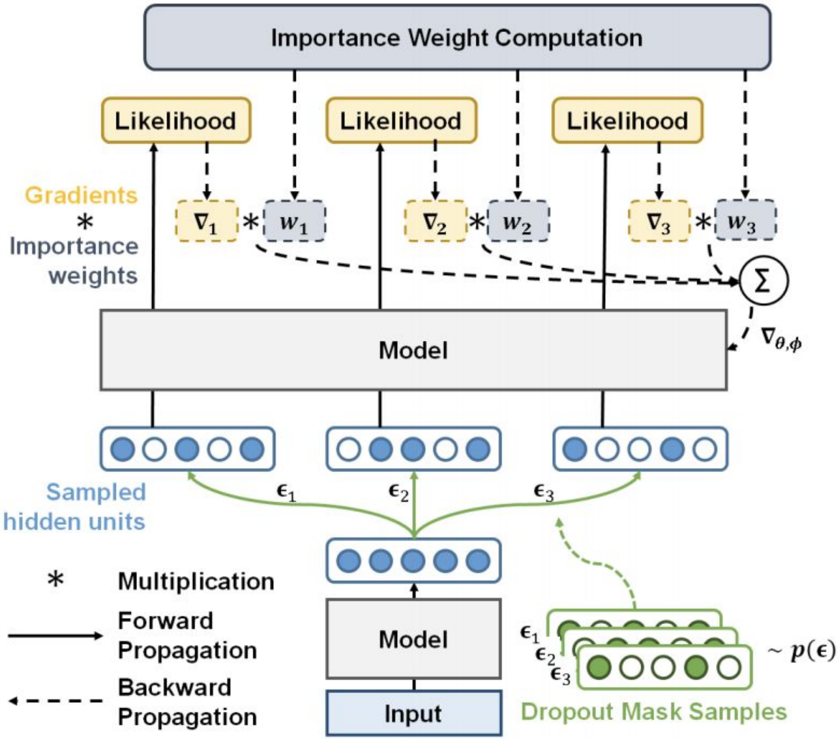
- A Bayesian interpretation of generic stochastic regularization techniques with multiplicative noise

- A generic framework to calibrate accuracy and confidence (score) of a prediction
  - Through stochastic inferences in deep neural networks
  - Introducing Variance-Weighted Confidence-Integrated (VWCI) loss
  - Capable of estimating prediction uncertainty using a single prediction
  - Supported by empirical observations

- Promising and consistent performance on multiple datasets and stochastic inference techniques

Other Works Related to Stochastic Learning

- Regularization by noise
  - Sampling multiple dropout masks
  - Learning with importance weighted stochastic gradient
- Interpretation and benefit
  - Improving the lower-bound of marginal likelihood by increasing the number of samples
  - Better accuracy in several domains

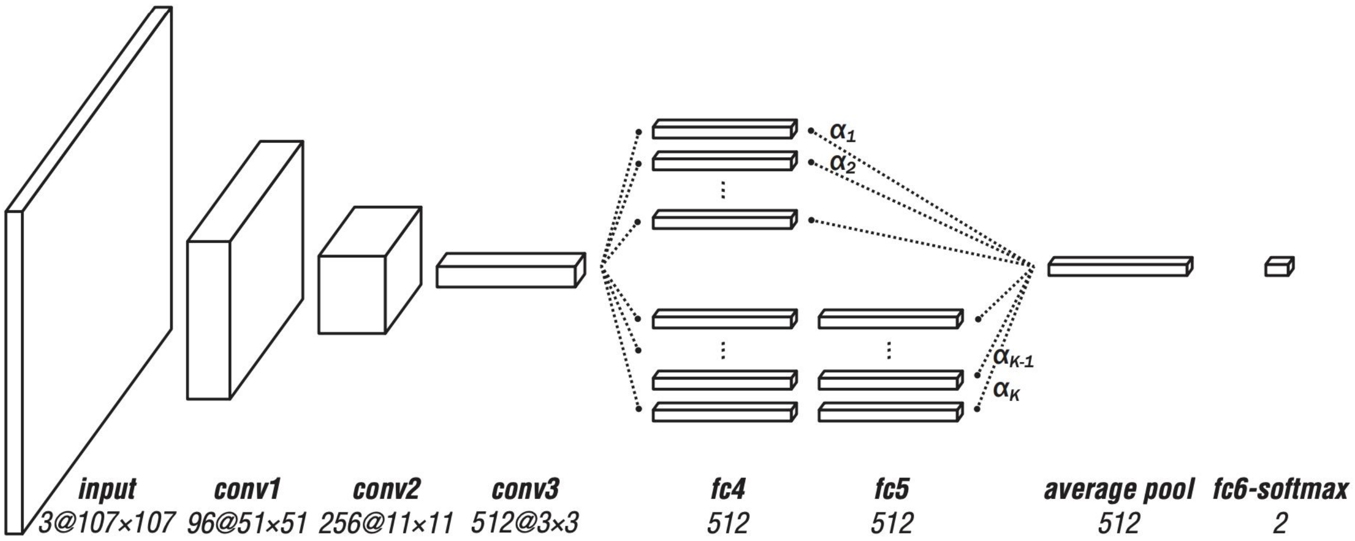


[Noh17] H. Noh, T. You, J. Mun, B. Han, **Regularizing Deep Neural Networks by Noise: Its Interpretation and Optimization**. NIPS 2017

Other Works Related to Stochastic Learning

Computer Vision Lab
Seoul National University

- Stochastic online few-shot ensemble learning
  - Preventing correlation of representations obtained from multiple branches
  - Randomly selecting branches for updates



[Han17]B. Han, J. Sim, H. Adam: **BranchOut: Regularization for Online Ensemble Tracking with Convolutional Neural Networks**. CVPR 2017

Other Research (in ML Perspective)

- Weakly supervised learning[NIPS2015, CVPR2016, AAAI2017, CVPR2017a, CVPR2018]
- Multi-modal learning[CVPR2016, AAAI2017, ICCV2017, NIPS2017]
- Metric learning[CVPR2017b]
- Multiple choice learning[NeurIPS2018]
- Zero-shot transfer learning[arXiv2018]
- Combinatorial learning
- Meta-learning
- Continual learning
- AutoML