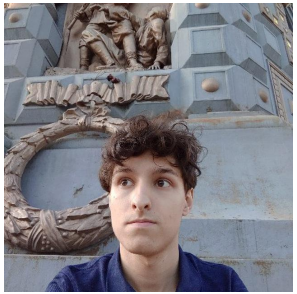
The State of Ensembling (and Uncertainty)

Arsenii Ashukha

Joint work with



Alexander Lyzhov*

Dmitry Molchanov*

Dmitry Vetrov

History of Ensembles in Machine Learning

- 1979: Bootstrap(Efron 79', Tibshirani 93'), Bagging(Breiman 96')
- 1990: Boosting (Schapire 90'), Adaboost(Freund and Schapire 96')
- 1992: Stacking (Wolpert 92')
- 1997: NN ensembles (Naftaly 97')
- 2012: DNN ensembles (Krizhevsky 12')

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

Bagging predictors

L Breiman - Machine learning, 1996 - Springer

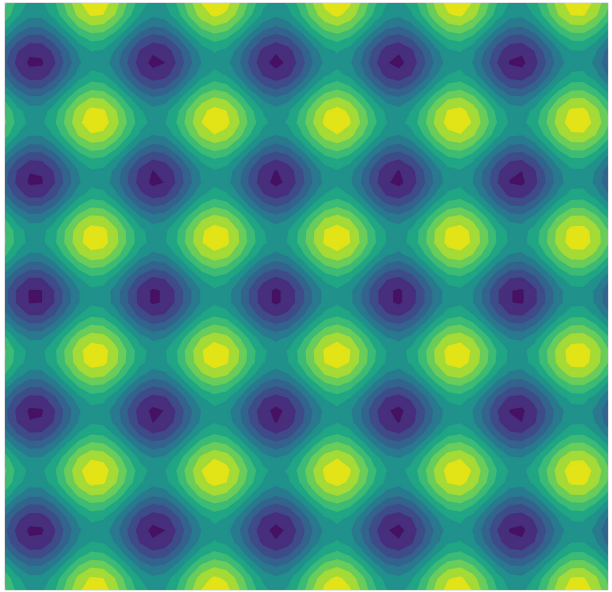
Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The …

☆ Cited by 22212 Related articles All 65 versions

What is Ensemble of DNNs? Do we have a lot of them?

$$p(y_i \mid x_i) = \frac{1}{K} \sum_{k=1}^{K} p(y_i \mid x_i, \omega_k), \quad \omega_k \sim q_m(\omega)$$

- Deep Ensembles
- Snapshot Ensembles
- Cyclical SGLD
- Fast Geometric Assembling
- K-FAC Laplace
- Dropout Ensemble
- Variational Inference Ensemble
- SWA-Gaussian (SWAG)
- Data augmentation
- ....



Loss surface of a DNN

How community estimates the quality of ensambles?

- Accuracy
- Log-likelihood
- Brier-score
- Calibration
- Misclassification detection

Measures of Predictive Performance

Measures of Predictive Performance and Uncertainty

Measures of Uncertainty

Downstream task

Why do we need uncertainty?

Many real-world risk-sensitive scenarios require reliable machine learning models.

A model should give us a proper estimate of a predictive distribution

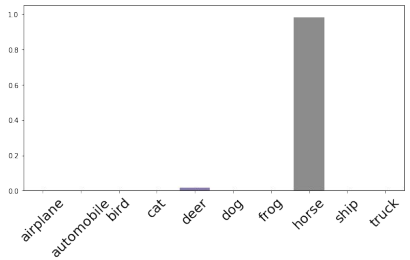
Benefits:

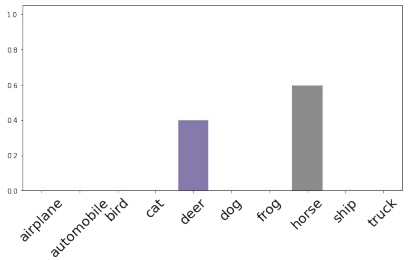
- report a level of confidence in a prediction
- robustness under dataset shift
- misclassification detection
- classification with rejection

$$LL = \frac{1}{n}\sum_{i=1}^{n} \log \hat{p}(y = y_i^* \mid x_i)$$

$$BS = \frac{1}{n}\frac{1}{C}\sum_{i=1}^{n}\sum_{c=1}^{C}(\mathbb{I}[y_i^* = c] - \hat{p}(y = c \mid x_i))^2$$

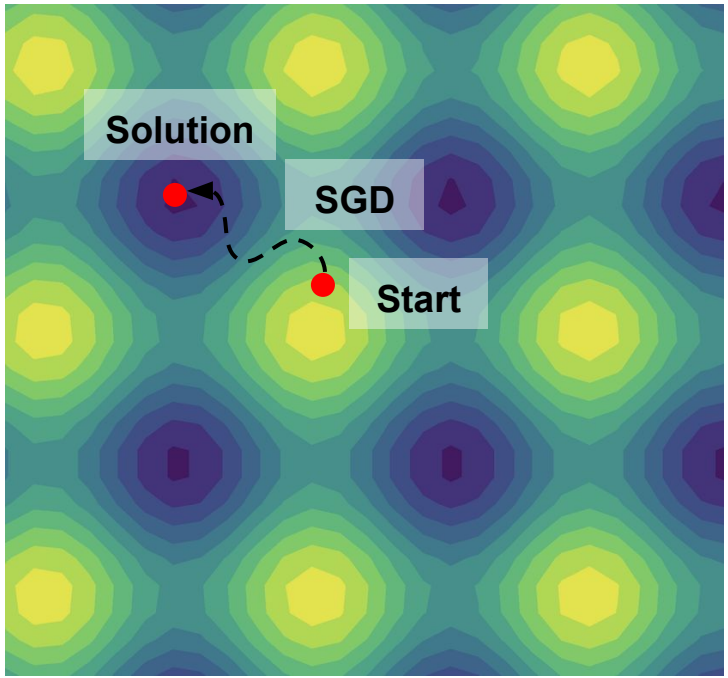
deer

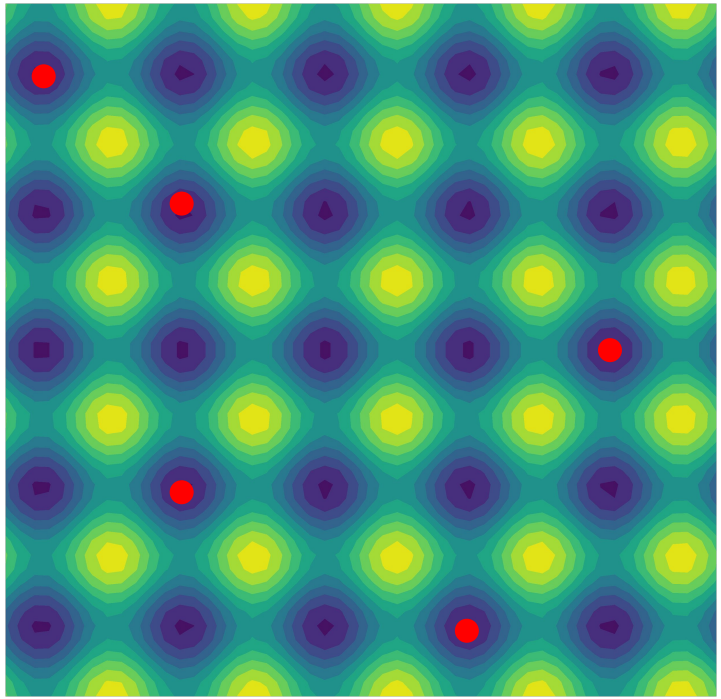




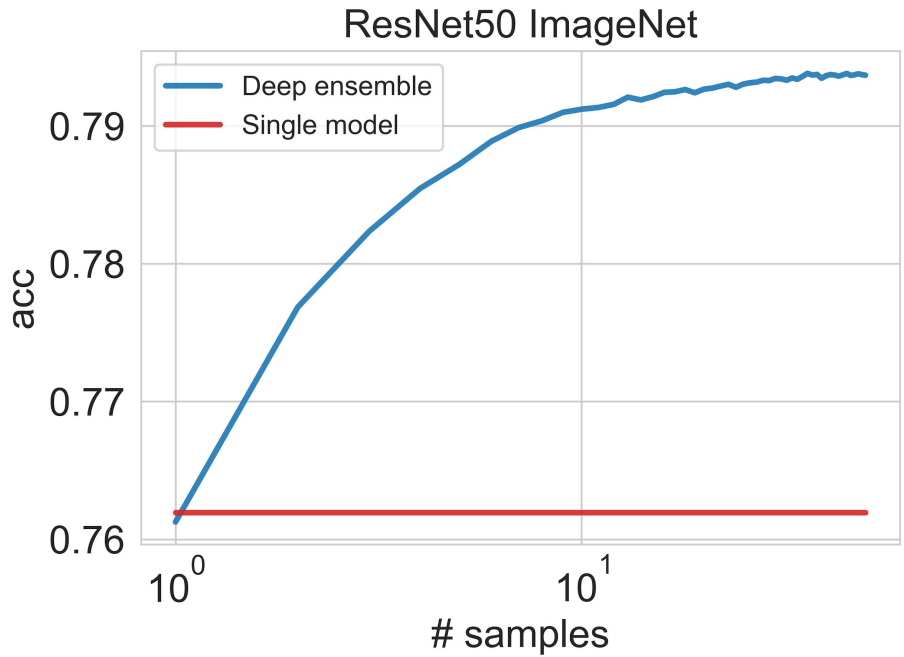
What is the simplest ensemble?

$$p(y_i \mid x_i) = \frac{1}{K} \sum_{k=1}^{K} p(y_i \mid x_i, \omega_k), \quad \omega_k \sim q_m(\omega)$$

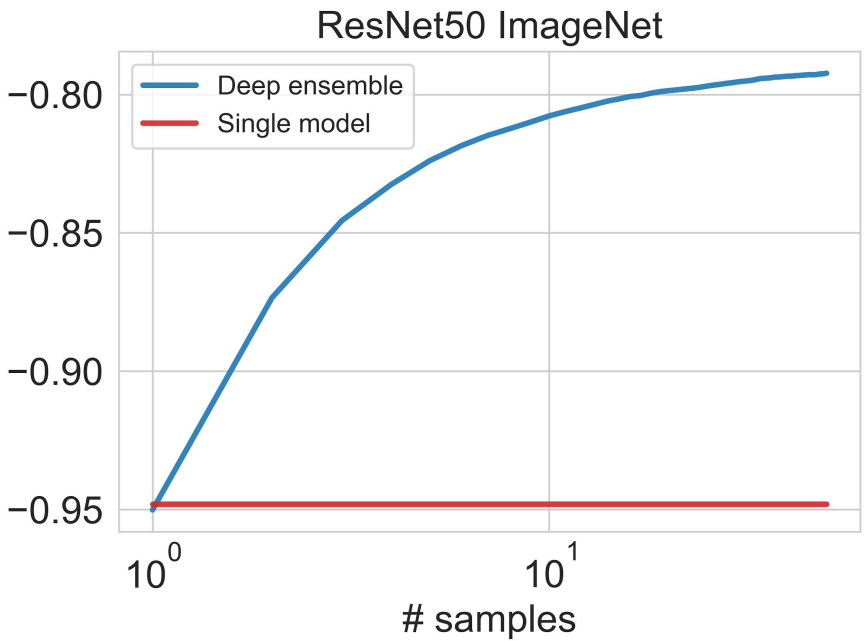




Metrics for a simple ensemble



=

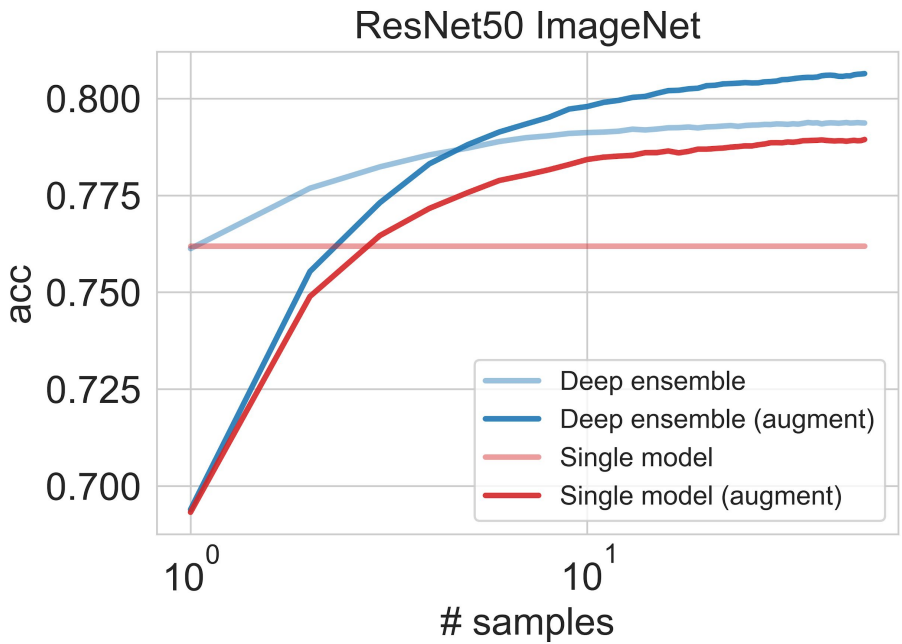


Augmentation

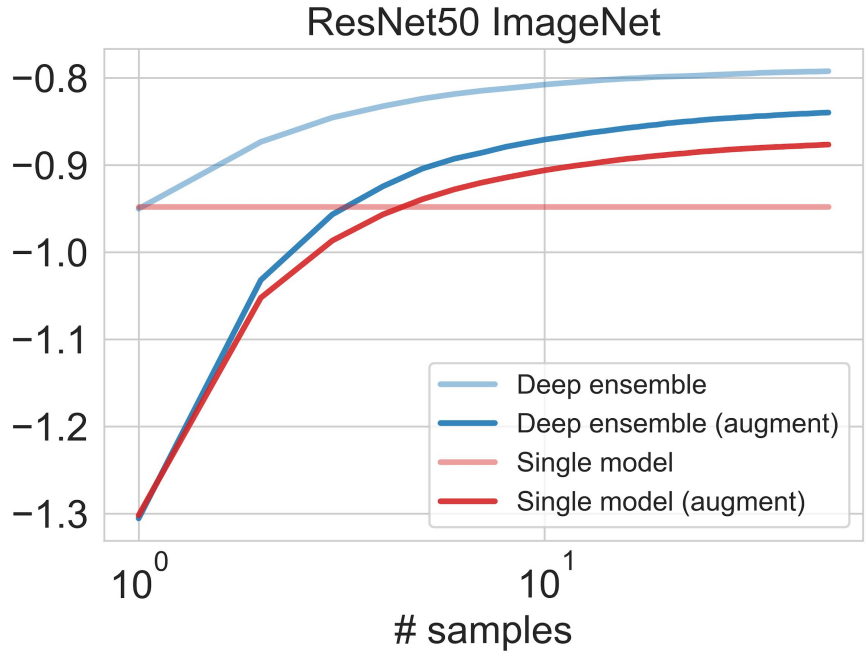
Original Image

What a network sees during training

Metrics for simple ensemble + test-time augmentation



=



Temperature scaling

$$softmax(z)_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

$$T^* = \arg\max_T LL(\text{Validation data}, \text{model}(T))$$

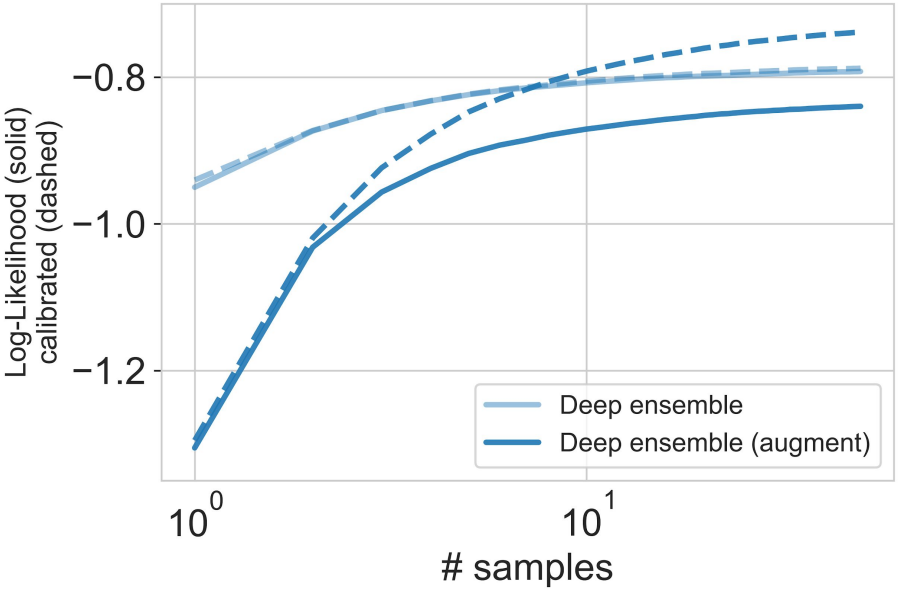
On Calibration of Modern Neural Networks

Chuan Guo [*1] Geoff Pleiss [*1] Yu Sun [*1] Kilian Q. Weinberger [1]

Dataset	Model	Uncalibrated	Hist. Binning	Isotonic	BBQ	Temp. Scaling	Vector Scaling	Matrix Scaling
Birds	ResNet 50	0.9786	1.6226	1.4128	1.2539	**0.8792**	0.9021	2.334
Cars	ResNet 50	0.5488	0.7977	0.8793	0.6986	0.5311	**0.5299**	1.0206
CIFAR-10	ResNet 110	0.3285	0.2532	0.2237	0.263	0.2102	0.2088	**0.2048**
CIFAR-10	ResNet 110 (SD)	0.2959	0.2027	0.1867	0.2159	0.1718	**0.1709**	0.1766
CIFAR-10	Wide ResNet 32	0.3293	0.2778	0.2428	0.2774	0.2283	0.2275	**0.2229**
CIFAR-10	DenseNet 40	0.2228	0.212	0.1969	0.2087	**0.1750**	0.1757	0.176
CIFAR-10	LeNet 5	0.4688	0.529	0.4757	0.4984	0.459	**0.4568**	0.4607
CIFAR-100	ResNet 110	1.4978	1.4379	1.207	1.5466	**1.0442**	1.0485	2.5637
CIFAR-100	ResNet 110 (SD)	1.1157	1.1985	1.0317	1.1982	**0.8613**	0.8655	1.8182
CIFAR-100	Wide ResNet 32	1.3434	1.4499	1.2086	1.459	**1.0565**	1.0648	2.5507
CIFAR-100	DenseNet 40	1.0134	1.2156	1.0615	1.1572	0.9026	**0.9011**	1.9639
CIFAR-100	LeNet 5	1.6639	2.2574	1.8173	1.9893	**1.6560**	1.6648	2.1405
ImageNet	DenseNet 161	0.9338	1.4716	1.1912	1.4272	0.8885	**0.8879**	-
ImageNet	ResNet 152	0.8961	1.4507	1.1859	1.3987	**0.8657**	0.8742	-
SVHN	ResNet 152 (SD)	0.0842	0.1137	0.095	0.1062	**0.0821**	0.0844	0.0924
20 News	DAN 3	0.7949	1.0499	0.8968	0.9519	0.7387	**0.7296**	0.9089
Reuters	DAN 3	0.102	0.2403	0.1475	0.1167	0.0994	**0.0990**	0.1491
SST Binary	TreeLSTM	0.3367	0.2842	0.2908	0.2778	**0.2739**	**0.2739**	**0.2739**
SST Fine Grained	TreeLSTM	1.1475	1.1717	1.1661	1.149	1.1168	**1.1085**	1.1112

Table S3. NLL (%) on standard vision and NLP datasets before calibration and with various calibration methods. The number following a model's name denotes the network depth. To summarize, NLL roughly follows the trends of ECE.

Metrics for a simple ensemble

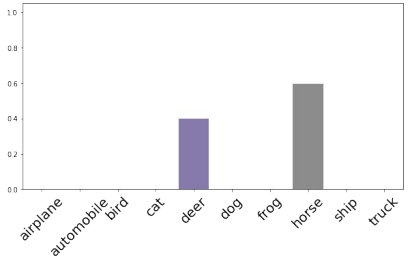


$$softmax(z)_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

Conclusion 1: Comparison of the log-likelihood should only be performed at the optimal temperature!

Misclassification detection

deer



label

Confidence : max p(y|x)

Entropy : H p(y|x)

High confidence (1.0)

+

+

+

-

Low confidence (0.1)

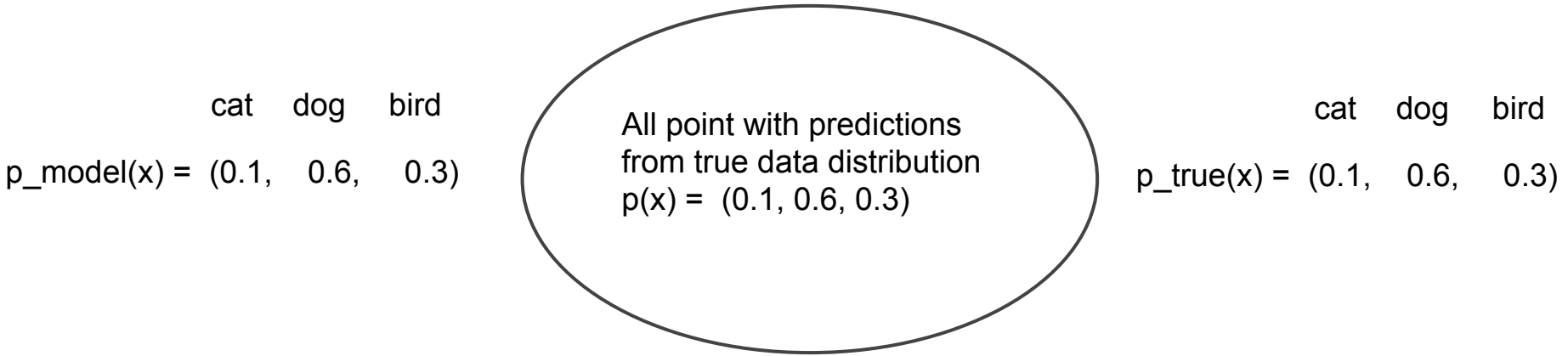
-

Conclusion 2: AUCs for misclassification detection can not be compared between different models.

Calibration

A classifier is calibrated if any predicted class probability is equal to the true class probability according to the underlying data distribution

$$P(y \in \cdot \,|\, g(x)) = g(x)$$



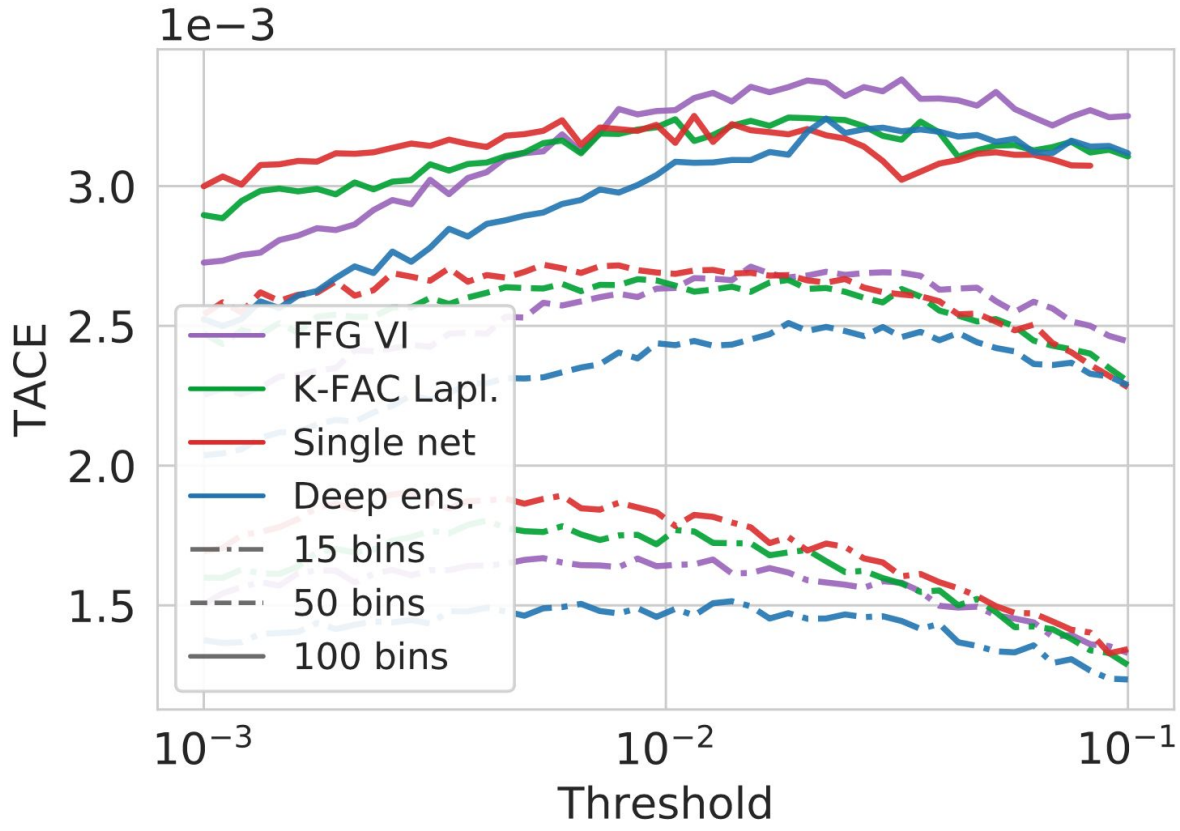
Calibration

A classifier is calibrated if any predicted class probability is equal to the true class probability according to the underlying data distribution

$$P(y \in \cdot \,|\, g(x)) = g(x)$$

$$\mathrm{TACE} = \frac{1}{CM} \sum_{c=1}^{C} \sum_{m=1}^{M} \frac{|B_m^{\mathrm{TA}}|}{n} \left| \mathrm{objs}(B_m^{\mathrm{TA}}, c) - \mathrm{conf}(B_m^{\mathrm{TA}}, c) \right|$$

Calibration



Calibration

Evaluating model calibration in classification

Juozas Vaicenavicius
Uppsala University; Veoneer Inc.

David Widmann
Uppsala University

Carl Andersson
Uppsala University

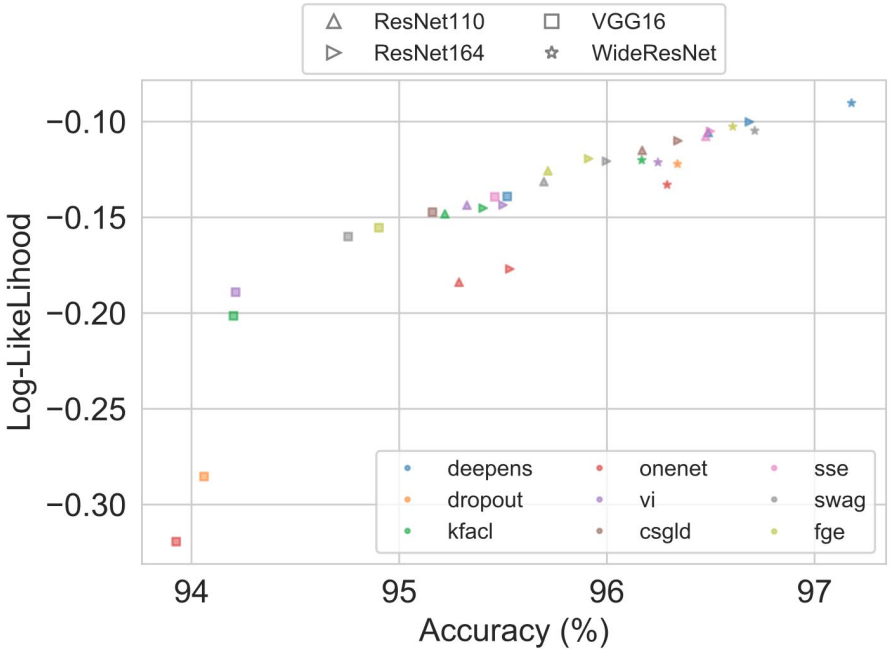
Fredrik Lindsten
Linköping University

Jacob Roll
Veoneer Inc.

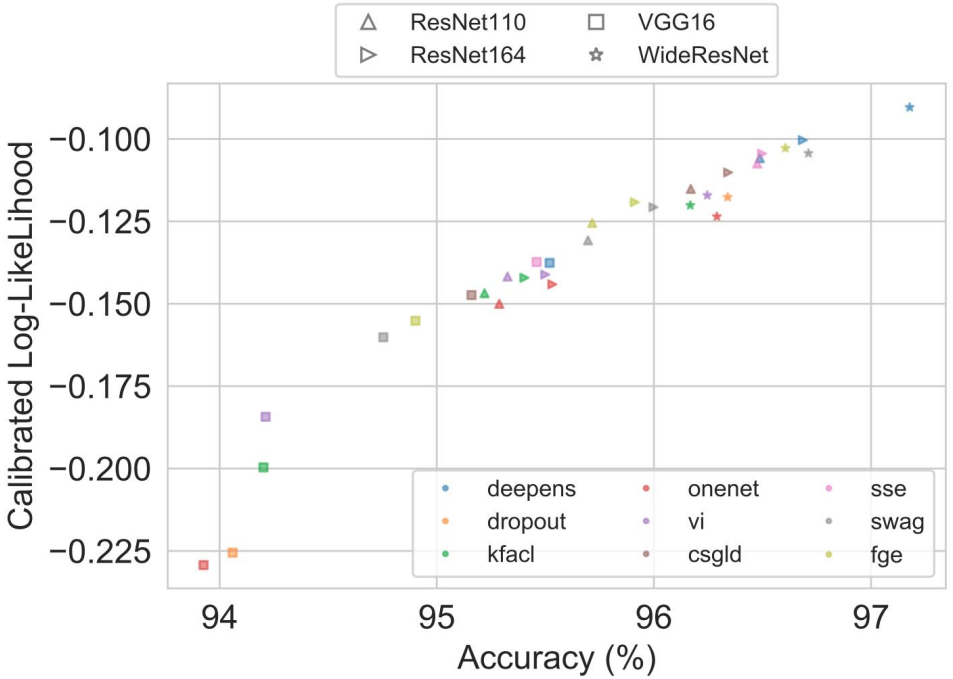
Thomas B. Schön
Uppsala University

"Alas, we do not know anything about how much the biases [of different models] differ (it is easy to come up with examples showing that the two biases can differ significantly, see Appendix A.1).

Correlation of Accuracy and LL



(a) CIFAR-10, $\rho = 0.869$



(b) CIFAR-10, $\rho = 0.962$

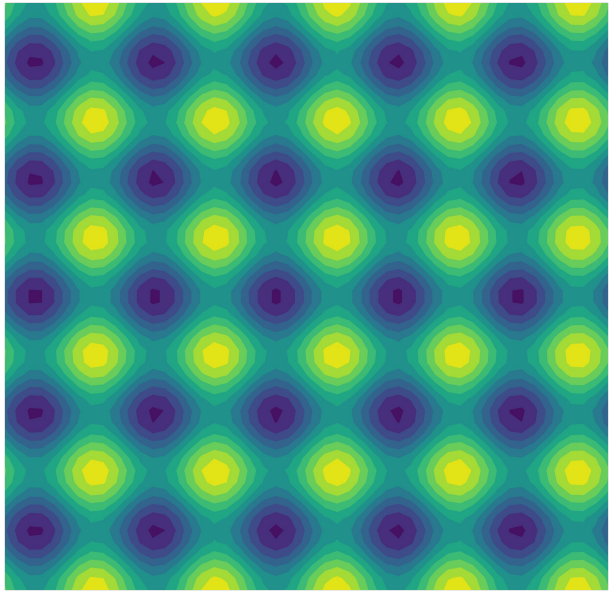
Test-time cross validation

1. A test set is randomly shuffled and divided into K folds of the same size
2. A temperature T* is adjusted by K-1 folds, T* = argmax_T LL(Model(T), Data(K-1 folds))
3. The model at the optimal temperature T* is used to evaluate metrics on the Kth fold.
4. The steps 1-3 are repeated several times, the metric values are averaged.

What is Ensemble of DNNs? Do we have a lot of them?

$$p(y_i \mid x_i) = \frac{1}{K} \sum_{k=1}^{K} p(y_i \mid x_i, \omega_k), \quad \omega_k \sim q_m(\omega)$$

- Deep Ensembles
- Snapshot Ensembles
- Cyclical SGLD
- Fast Geometric Assembling
- K-FAC Laplace
- Dropout Ensemble
- Variational Inference Ensemble
- SWA-Gaussian (SWAG)
- Data augmentation
- ....



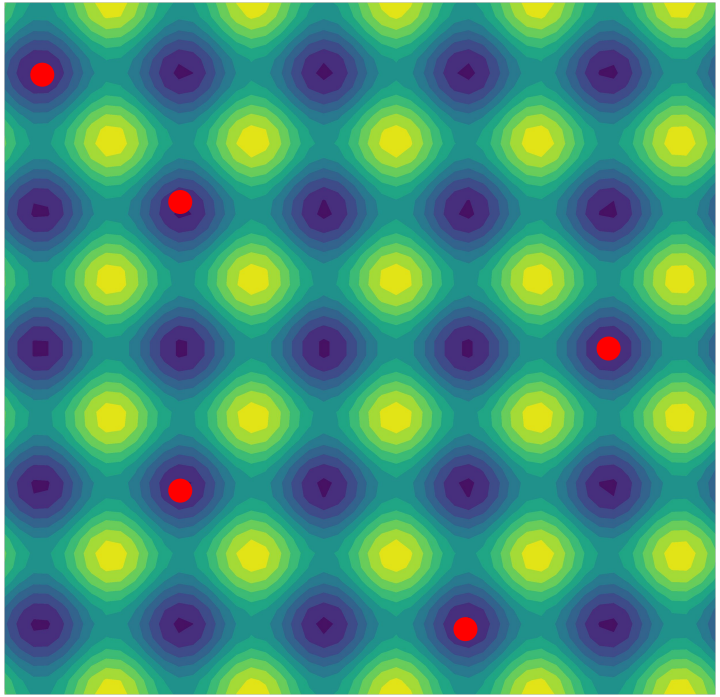
Loss surface of a DNN

Deep Ensembles

- **Deep Ensembles**: Independently trained networks (Lakshminarayanan 2017).

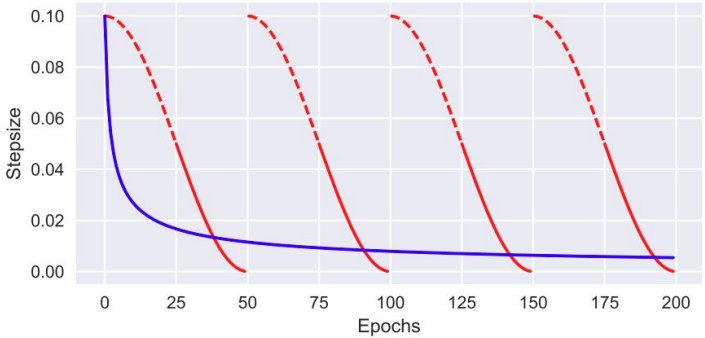
Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan **Alexander Pritzel** **Charles Blundell**
DeepMind
{balajiln,apritzel,cblundell}@google.com



Snapshot Ensembles

- **Snapshot Ensembles**: Training models with cyclical learning rate (Huang 2017).

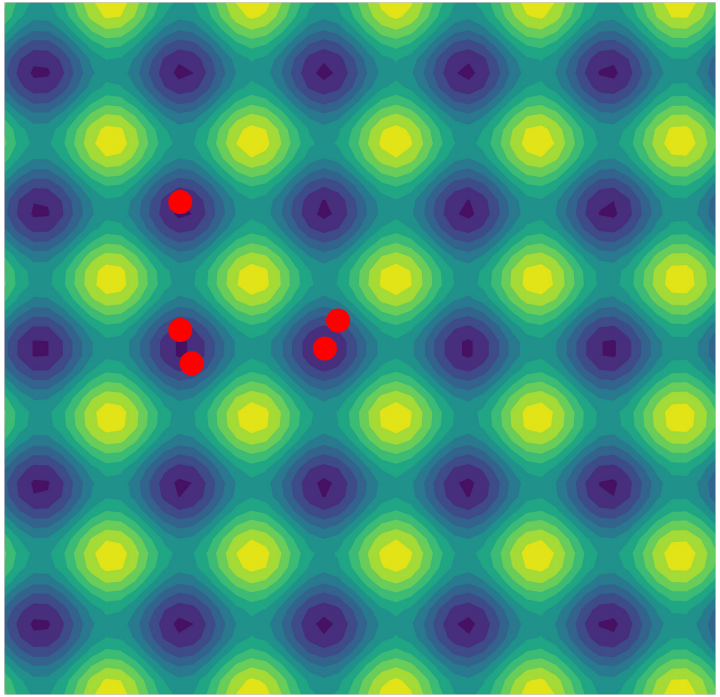


SNAPSHOT ENSEMBLES: TRAIN 1, GET M FOR FREE

Gao Huang*, **Yixuan Li***, **Geoff Pleiss**
Cornell University
{gh349, yl2363}@cornell.edu, geoff@cs.cornell.edu

Zhuang Liu
Tsinghua University
liuzhuangthu@gmail.com

John E. Hopcroft, Kilian Q. Weinberger
Cornell University
jeh@cs.cornell.edu, kqw4@cornell.edu



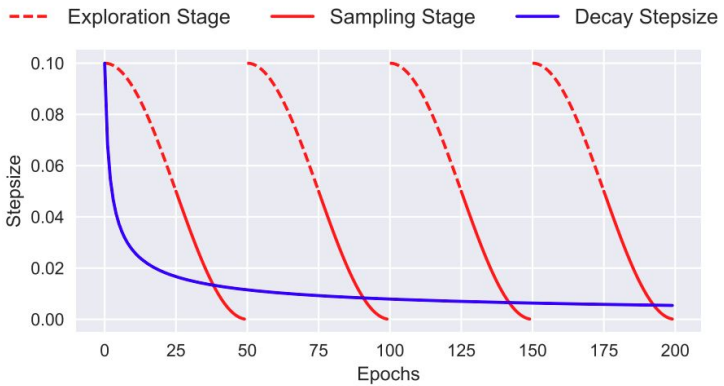
Cyclical SGLD

- **Cyclical SGLD**: Cyclical learning rate and gradient noise (Zhang 2019)

$$p(\theta|\mathcal{D}) \propto \exp(-U(\theta))$$

$$U(\theta) = -\log p(\mathcal{D}|\theta) - \log p(\theta)$$

$$\theta_k = \theta_{k-1} - \alpha_k \nabla \tilde{U}(\theta_k) + \sqrt{2\alpha_k}\epsilon_k$$

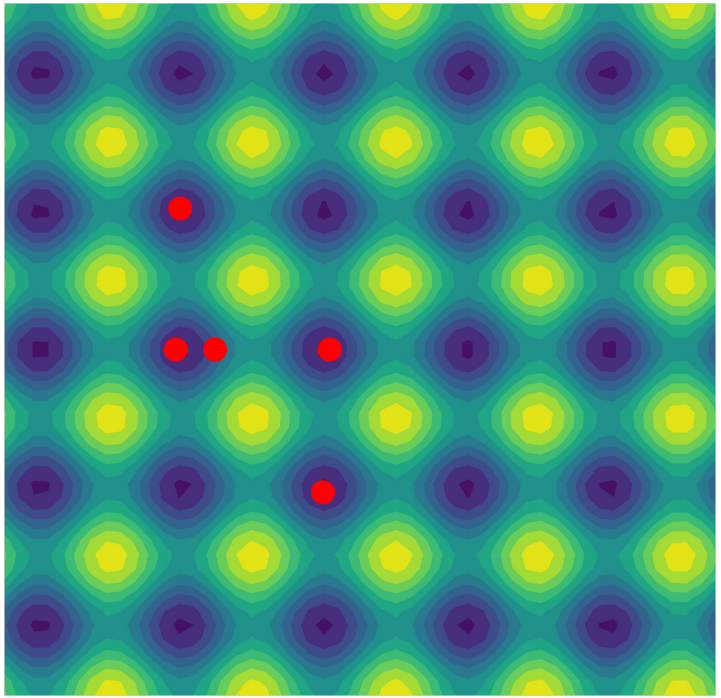


Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning

Ruqi Zhang [1] Chunyuan Li [2] Jianyi Zhang [3] Changyou Chen [4] Andrew Gordon Wilson [1]

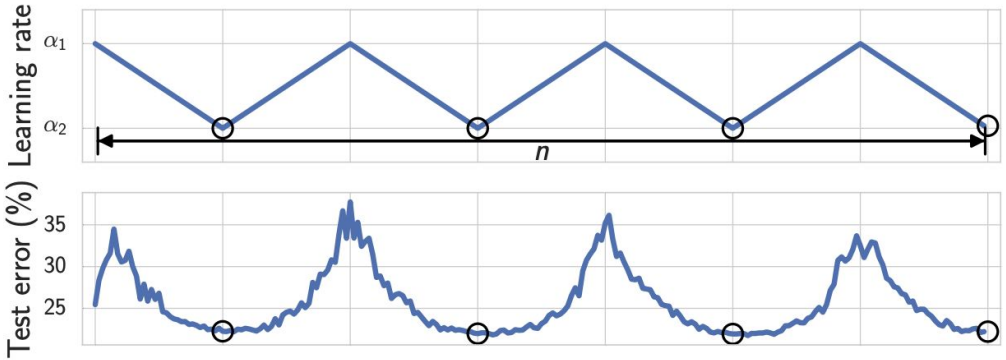
Bayesian Learning via Stochastic Gradient Langevin Dynamics

Max Welling WELLING@ICS.UCI.EDU
D. Bren School of Information and Computer Science, University of California, Irvine, CA 92697-3425, USA
Yee Whye Teh YWTEH@GATSBY.UCL.AC.UK
Gatsby Computational Neuroscience Unit, UCL, 17 Queen Square, London WC1N 3AR, UK



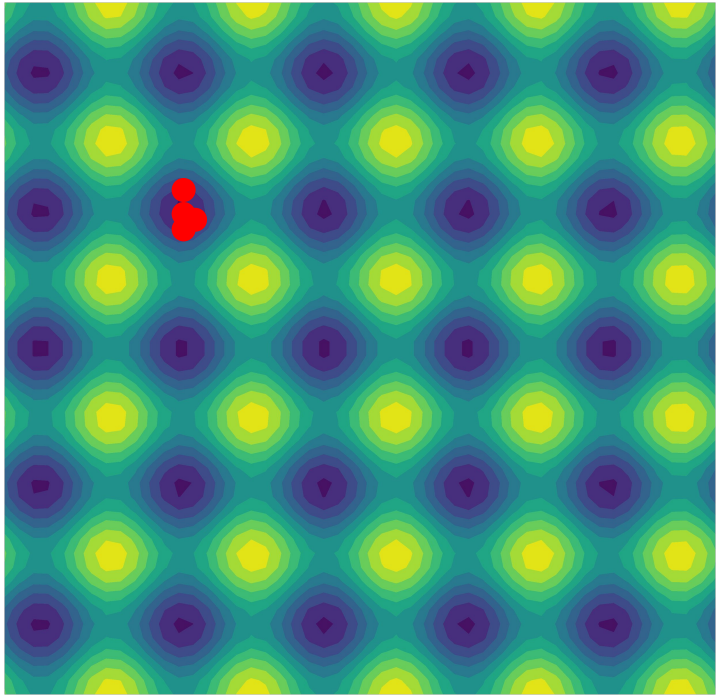
Fast Geometric Ensembling

- **Fast Geometric Ensembling**: Short epoch with cyclical learning rate (Izmailov 2018).



Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs

Timur Garipov[*1,2] Pavel Izmailov[*3] Dmitrii Podoprikhin[*4]
Dmitry Vetrov[5] Andrew Gordon Wilson[3]



Correlated Gaussian

- SWA Gaussian: Correlated Gaussian Noise (Maddox 2019)

$$\widetilde{\theta} = \theta_{SWA} + \Sigma_{\text{diag}}^{\frac{1}{2}} z_1, \quad z_1 \sim \mathcal{N}(0, I_d)$$

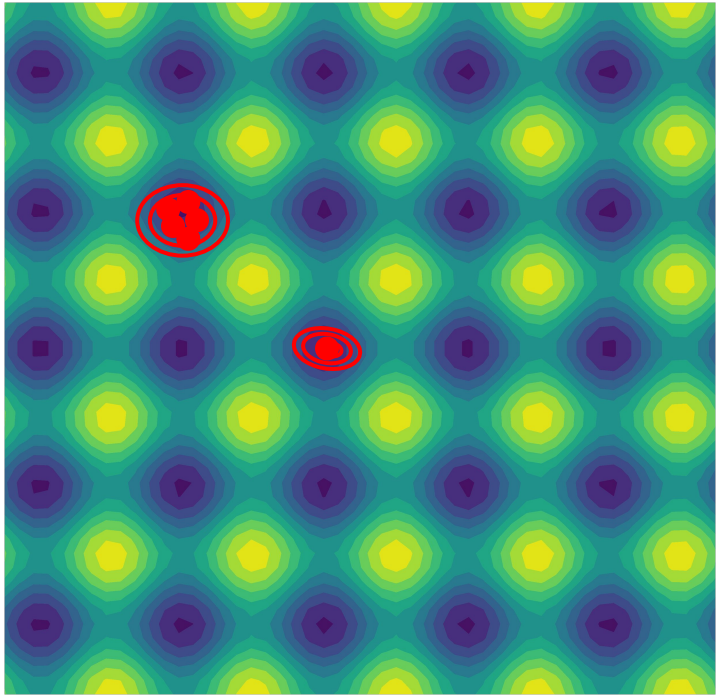
- K-FAC Laplace: Correlated Gaussian Noise (Ritter 2018)

A Simple Baseline for Bayesian Uncertainty in Deep Learning

Wesley Maddox [*1] Timur Garipov [*2] Pavel Izmailov [*1] Dmitry Vetrov [2 3] Andrew Gordon Wilson [1]

A SCALABLE LAPLACE APPROXIMATION FOR NEURAL NETWORKS

Hippolyt Ritter [1*], Aleksandar Botev [1], David Barber [1 2]
[1]University College London [2]Alan Turing Institute

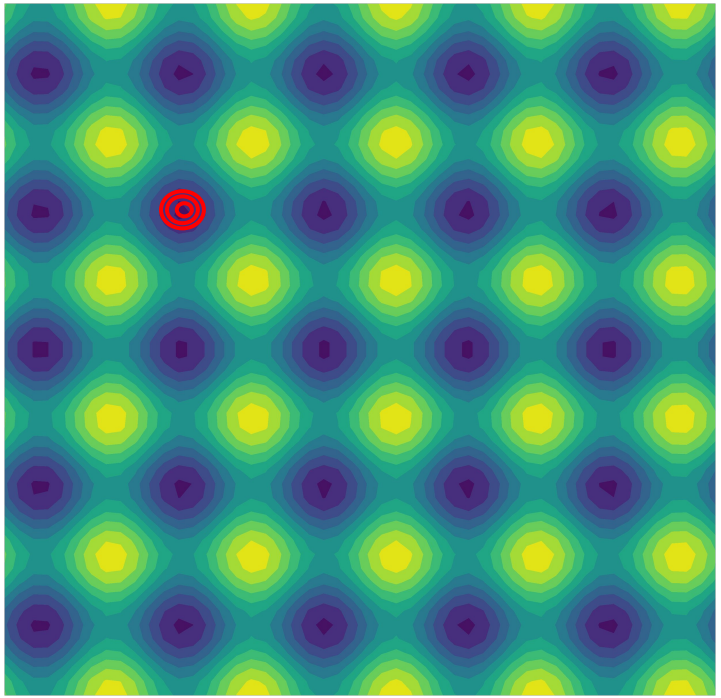


Independent Gaussian Noise

- **Independent Noise**:
  - Dropout
  - Gaussian Dropout
  - FFG Variational Inference

$$E_{q(W)} L(X, Y, W) \to \min_{\mu}$$

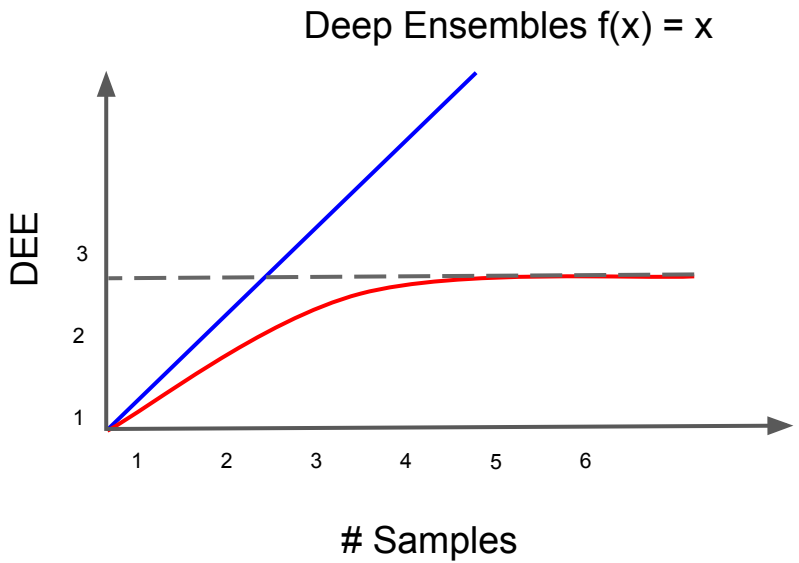
$$q(w_{ij}) = N(w_{ij} \,|\, \mu_{ij}, \sigma_{ij}^2)$$



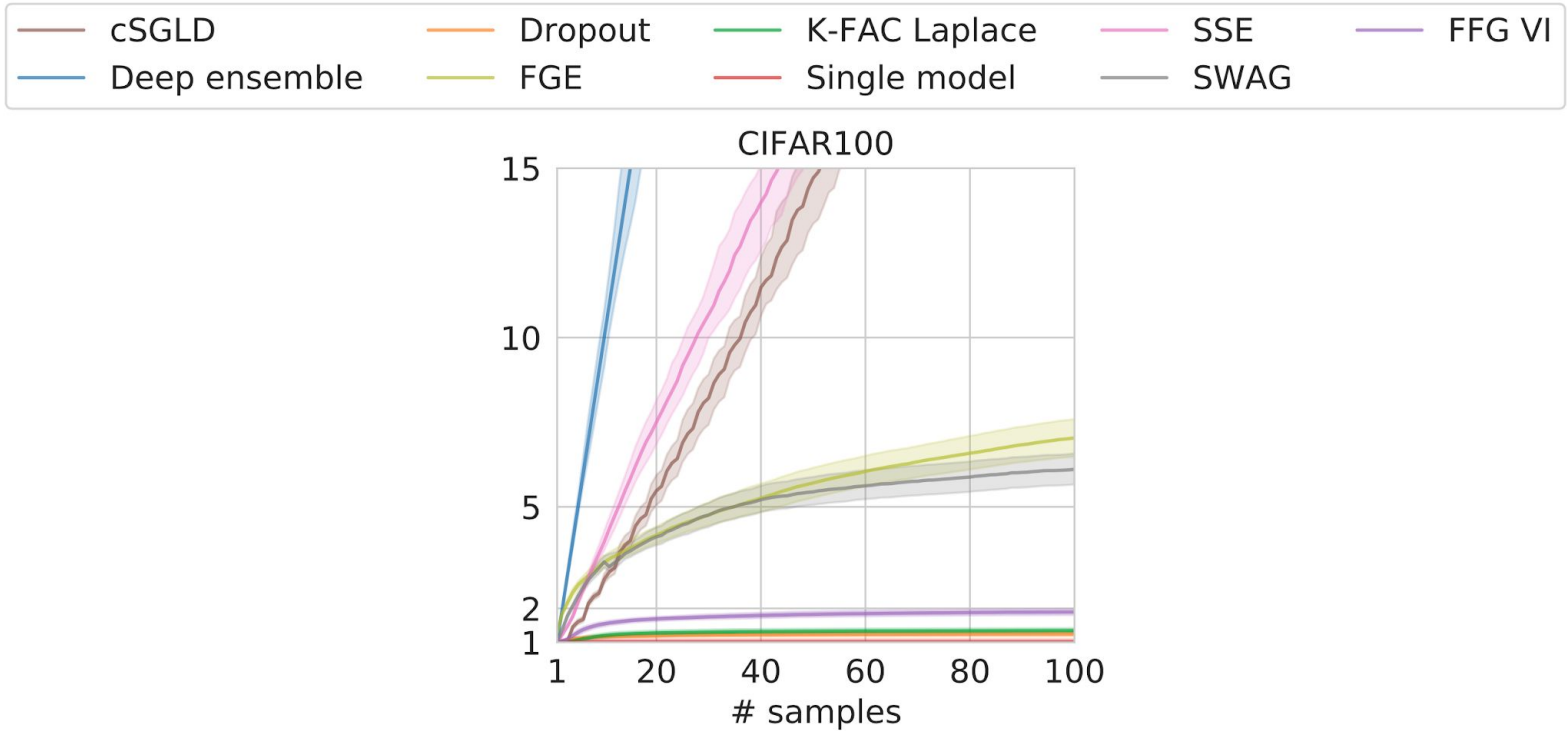
Deep Ensamble Equivalent

What number of independently trained networks combined yields the same performance as a particular ensembling method?

$$\mathrm{DEE}_m(k) = \min\left\{ l \in \mathbb{R}, l \geq 1 \,\middle|\, \mathrm{CLL}_{DE}^{\mathrm{mean}}(l) \geq \mathrm{CLL}_m^{\mathrm{mean}}(k) \right\}$$



Deep Ensamble Equivalent



DATA AUGMENTATION IMPROVES ENSEMBLES

Model	cSGLD	Deep ensemble	FGE	K-FAC-L	Single model	FFG VI
ResNet110	0.115 vs 0.111≈	0.106 vs 0.105≈	0.121 vs 0.121≈	0.147 vs 0.130↓	0.150 vs 0.129↓	0.142 vs 0.130↓
ResNet164	0.110 vs 0.108≈	0.100 vs 0.100≈	0.115 vs 0.115≈	0.142 vs 0.127↓	0.144 vs 0.124↓	0.141 vs 0.128↓
VGG16	0.147 vs 0.146≈	0.138 vs 0.139≈	0.150 vs 0.150≈	0.200 vs 0.164↓	0.229 vs 0.170↓	0.184 vs 0.160↓
WideResNet	0.099 vs 0.100≈	0.090 vs 0.094≈	0.102 vs 0.102≈	0.120 vs 0.111↓	0.124 vs 0.113↓	0.117 vs 0.117↓

Conclusion 3.1: DA improves simple ensambles on CIFARs.

DATA AUGMENTATION IMPROVES ENSEMBLES

Model (# samples)	Deep ensemble	FGE	K-FAC-L	Single model	SSE	FFG VI
ResNet50 (7 samples)	21.01 vs 20.66↓	23.59 vs 21.61↓	24.04 vs 21.97↓	23.81 vs 21.97↓	21.96 vs 21.29↓	23.76 vs 22.07↓
ResNet50 (40 samples)	20.64 vs 19.40↓	23.29 vs 20.75↓	23.80 vs 21.14↓	23.81 vs 21.08↓		23.69 vs 21.20↓
ResNet50 (50 samples)	20.63 vs 19.36↓		23.82 vs 21.04↓	23.81 vs 21.06↓		23.69 vs 21.11↓
Model (# samples)	Deep ensemble	FGE	K-FAC-L	Single model	SSE	FFG VI
ResNet50 (7 samples)	0.813 vs 0.817↑	0.918 vs 0.863↓	0.950 vs 0.877↓	0.938 vs 0.872↓	0.852 vs 0.843↓	0.925 vs 0.877↓
ResNet50 (40 samples)	0.789 vs 0.742↓	0.907 vs 0.798↓	0.937 vs 0.812↓	0.938 vs 0.808↓		0.920 vs 0.807↓
ResNet50 (50 samples)	0.788 vs 0.739↓		0.936 vs 0.808↓	0.938 vs 0.805↓		0.920 vs 0.804↓

Conclusion 3.2:
Data augmentation consistently improves all ensembles on ImageNet!

Outcomes

- Temperature scaling is a must even for ensembles.
- Many metrics for in-domain uncertainty have a lot of pitfalls.
- Many popular ensembling techniques are equivalent to a handful of independently trained models, but much less efficient.
- Single mode vs different modes of the loss landscape.
- Test-time data augmentation is a surprisingly strong baseline.