

SWAG¹: Approximate Bayesian Inference Using SGD Trajectory

Wesley Maddox* **Timur Garipov*** Pavel Izmailov*
Dmitry Vetrov Andrew Gordon Wilson

*equal contirbution

February 15, 2019

¹Wesley Maddox et al. (2019). “A Simple Baseline for Bayesian Uncertainty in Deep Learning”. In: *arXiv pre-print*

Uncertainty Estimation and Bayesian Deep Learning

Uncertainty estimation in Deep Learning

- Machine learning models are used to make decisions.
- High predictive accuracy alone is not enough.
- Representing uncertainty is crucial for decision making in applications.
- Deep neural networks:
 - lack representation of uncertainty;
 - provide overconfident and miscalibrated predictions;
 - fail to generalize on out-of-sample data.

- Posterior distribution $p(\theta \mid \mathcal{D})$ over DNN weights

$$p(\theta \mid \mathcal{D}) = \frac{\overbrace{p(\mathcal{D} \mid \theta)}^{\text{likelihood}} \cdot \overbrace{p(\theta)}^{\text{prior}}}{p(\mathcal{D})}$$

- **Training** = fitting tractable posterior approximation

$$q(\theta) \approx p(\theta \mid \mathcal{D}_{\text{train}})$$

- **Inference** = Bayesian model averaging

$$\begin{aligned} p(\mathcal{D}_{\text{test}} \mid \mathcal{D}_{\text{train}}) &= \mathbb{E}_{p(\theta \mid \mathcal{D}_{\text{train}})} [p(\mathcal{D}_{\text{test}} \mid \theta)] \\ &\approx \frac{1}{N} \sum_{i=1}^N p(\mathcal{D}_{\text{test}} \mid \hat{\theta}_i), \quad \hat{\theta}_i \sim q(\theta) \end{aligned}$$

Recent Approaches to Uncertainty in Deep Learning

- Bayesian
 - (SG)MCMC (Chen et al. 2014)
 - Variational Inference (Blundell et al. 2015)
 - MC-Dropout (Gal et al. 2016)
 - Laplace approximation (diagonal/KFAC) (Ritter et al. 2018)
- Non-Bayesian
 - Deep Ensembles (Lakshminarayanan et al. 2017)
 - Temperature scaling (Guo et al. 2017)

Background

SGD as Approximate Bayesian Inference
(Mandt et al. 2017)

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^N \ell_n(\theta), \quad g(\theta) \equiv \nabla_{\theta} \mathcal{L}(\theta)$$

$$\ell_n(\theta) = -\log p(x_n \mid \theta) - \frac{1}{N} \log p(\theta)$$

\mathcal{S} — minibatch of size S

$$\hat{\mathcal{L}}_S(\theta) = \frac{1}{S} \sum_{n \in \mathcal{S}} \ell_n(\theta), \quad \hat{g}_S(\theta) = \nabla_{\theta} \hat{\mathcal{L}}_S(\theta)$$

$$\theta(t+1) = \theta(t) - \epsilon \hat{g}_S(\theta(t))$$

SGD as Approximate Bayesian Inference (Mandt et al. 2017)

Assumption 1 Gradient noise is Gaussian with covariance $\frac{1}{S}C(\theta)$

$$\hat{g}_S(\theta) \approx g(\theta) + \frac{1}{\sqrt{S}}\Delta g(\theta), \quad \Delta g(\theta) \sim \mathcal{N}(0, C(\theta))$$

Assumption 2 The covariance matrix $C(\theta)$ is approximately constant with respect to θ and full rank.

$$C(\theta) \approx C = BB^\top$$

$$\Delta\theta(t) = \theta(t+1) - \theta(t) = -\epsilon g(\theta(t)) + \frac{\epsilon}{\sqrt{S}}B\Delta W, \quad \Delta W \sim \mathcal{N}(0, I)$$

SGD is a finite-difference approximation of the continuous SDE

$$d\theta(t) = -\epsilon g(\theta)dt + \frac{\epsilon}{\sqrt{S}}BdW(t)$$

SGD as Approximate Bayesian Inference (Mandt et al. 2017)

Assumption 3 The finite-difference equation can be approximated by the stochastic differential equation.

This assumption is justified if either the gradients or the learning rates are small enough.

Assumption 4 SGD iterates are constrained to a region where the loss is well approximated by a quadratic function

$$\mathcal{L}(\theta) = \frac{1}{2}\theta^\top A\theta, \quad A > 0$$

A1-A4 result in the multivariate Ornstein-Uhlenbeck process

$$d\theta(t) = -\epsilon A\theta(t)dt + \frac{1}{\sqrt{S}}\epsilon B dW(t)$$

The Ornstein-Uhlenbeck process

$$d\theta(t) = -\epsilon A\theta(t)dt + \frac{1}{\sqrt{S}}\epsilon B dW(t)$$

has an analytic stationary distribution $q(\theta)$ that is Gaussian

$$q(\theta) \approx \exp \left\{ -\frac{1}{2} \theta^\top \Sigma^{-1} \theta \right\}.$$

The covariance Σ satisfies

$$\Sigma A + A \Sigma = \frac{\epsilon}{S} B B^\top$$

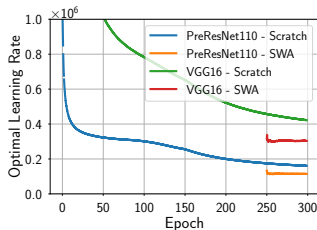
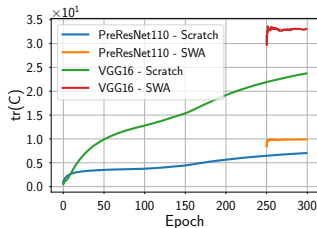
Theorem 1 (Mandt et al. 2017) Under assumptions A1-A4, the constant learning rate that minimizes KL divergence from the stationary distribution of constant SGD to the posterior is

$$\epsilon^* = 2 \frac{S}{N} \frac{D}{\text{Tr}(BB^\top)},$$

where D is the dimension of θ .

Do the assumptions hold for DNNs?

- **A2 Covariance noise.**



$$\epsilon^* = 2 \frac{S}{N} \frac{D}{\text{Tr}(C)},$$

- **A4 Hessian Eigenvalues.**

The minimum and maximum eigenvalue were estimated for PreResNet-164 on CIFAR-100 using Lanczos method.

$$\lambda_{\min} \approx -272 \quad \lambda_{\max} \approx 3580$$

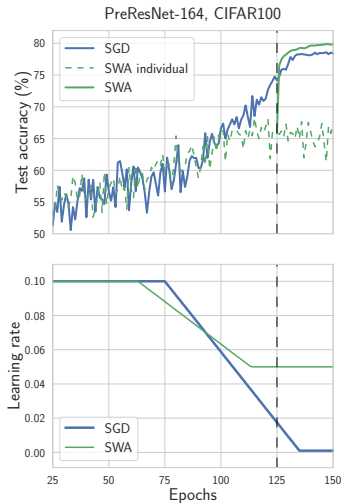
Background

Stochastic Weight Averaging (SWA)
(Izmailov et al. 2018)

Stochastic Weight Averaging (SWA) (Izmailov et al. 2018)

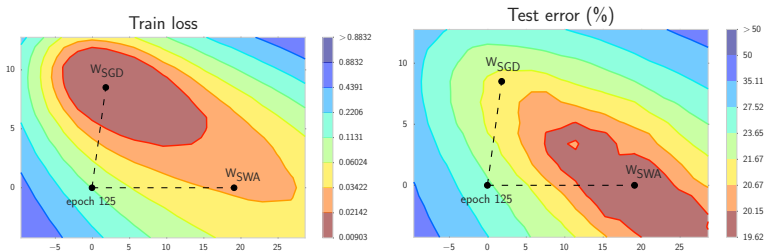
Stochastic Weight Averaging

1. Start with a pre-trained point θ_0
2. Run SGD with a constant learning rate
3. Collect SGD iterates with some frequency
4. Average collected weights $\theta_{\text{SWA}} = \frac{1}{T} \sum_{i=1}^T \theta_i$
5. Update Batch-Norm statistics for θ_{SWA} with one pass over the train set
6. Use θ_{SWA} at inference time



Stochastic Weight Averaging (SWA) (Izmailov et al. 2018)

2D cross-sections of loss/error surfaces
along the plane containing w_0 , w_{SGD} , w_{SWA}
PreResNet-164, CIFAR-100

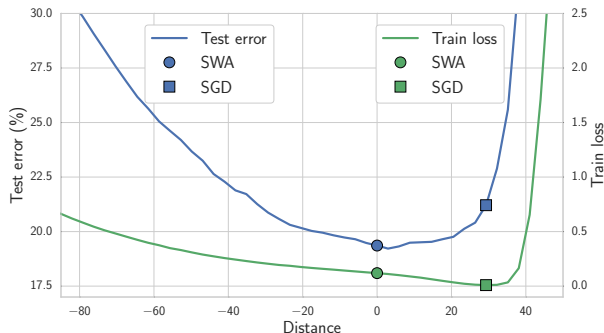


$$\psi(t_1, t_2) = \mathcal{L}(w_0 + t_1 v_1 + t_2 v_2)$$

plane: $w_0 + \text{span}(v_1, v_2)$

Stochastic Weight Averaging (SWA) (Izmailov et al. 2018)

1D cross-sections of loss/error surfaces
along the line connecting w_{SGD} and w_{SWA}
PreResNet-164, CIFAR-100



$$\phi(t) = \mathcal{L}(t \cdot w_{\text{SGD}} + (1 - t) \cdot w_{\text{SWA}})$$

SWA-Gaussian (SWAG)

- Estimate the 1st and the 2nd moments of the SGD trajectory

$$\theta_{\text{SWA}} = \frac{1}{T} \sum_{i=1}^T \theta_i, \quad \overline{\theta^2} = \frac{1}{T} \sum_{i=1}^T \theta_i^2$$

- Form Gaussian approximation with diagonal covariance matrix

$$q(\theta) = \mathcal{N}(\theta \mid \theta_{\text{SWA}}, \Sigma_{\text{diag}}), \quad \Sigma_{\text{diag}} = \text{diag}(\overline{\theta^2} - \theta_{\text{SWA}}^2)$$

- Sample weights at inference time

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

- Low-rank covariance estimation with last K SGD iterates

$$\bar{\theta}_i = \frac{1}{i} \sum_{j=1}^i \theta_j, \quad D_i = \theta_i - \bar{\theta}_i$$

$$\hat{D} = [D_{T-K+1}; D_{T-K+2}; \dots; D_T]$$

$$\Sigma_{\text{low-rank}} = \frac{1}{K-1} \hat{D} \hat{D}^\top$$

- Combination of diagonal and low-rank approximations

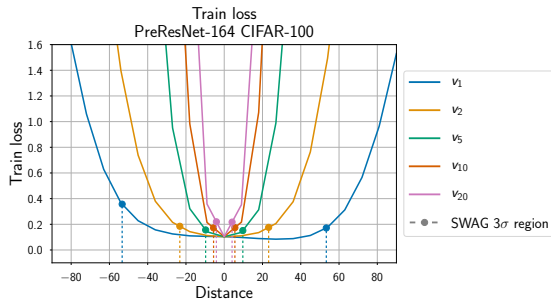
$$q(\theta) = \mathcal{N}(\theta \mid \theta_{\text{SWA}}, \frac{1}{2} \Sigma_{\text{low-rank}} + \frac{1}{2} \Sigma_{\text{diag}})$$

$$\tilde{\theta} = \theta_{\text{SWA}} + \frac{1}{\sqrt{2}} \Sigma_{\text{diag}}^{\frac{1}{2}} z_1 + \frac{1}{\sqrt{2(K-1)}} \hat{D} z_2$$

$$z_1 \sim \mathcal{N}(0, I_d) \quad z_2 \sim \mathcal{N}(0, I_K)$$

SWAG and Loss Landscapes

1D cross-sections of loss/error surfaces
along the eigenvectors of $\Sigma_{\text{low-rank}}$

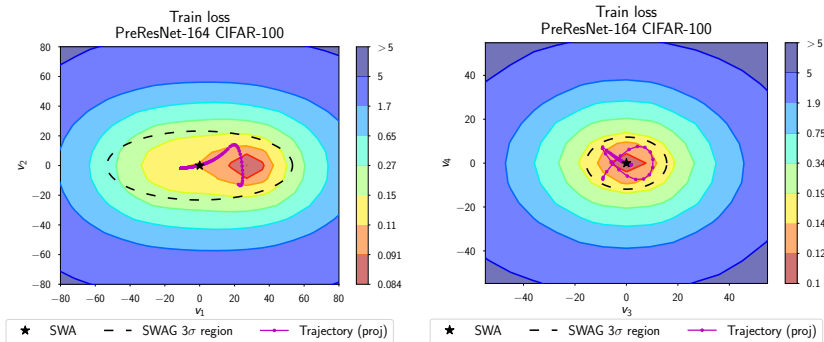


$$\phi(t) = \mathcal{L}(\theta_{\text{SWA}} + t \cdot \frac{v_i}{\|v_i\|})$$

$$\Sigma_{\text{low-rank}} = \frac{1}{K-1} \hat{D} \hat{D}^\top = V \Lambda V^T$$

SWAG and Loss Landscapes

2D cross-sections of loss/error surfaces
along the planes spanned by eigenvectors of $\Sigma_{\text{low-rank}}$



$$\psi(t_1, t_2) = \mathcal{L}(\theta_{\text{SWA}} + t_1 \cdot \frac{v_i}{\|v_i\|} + t_2 \cdot \frac{v_j}{\|v_j\|})$$

Batch normalization

Training:

$$\hat{x} = \gamma \frac{x - \mu(x)}{\sigma(x) + \epsilon} + \beta$$

Evaluation:

$$\hat{x} = \gamma \frac{x - \tilde{\mu}}{\tilde{\sigma} + \epsilon} + \beta$$

- During training $\mu(x)$ and $\sigma(x)$ are estimated from a mini-batch.
- At the evaluation stage running statistics $\tilde{\mu}$ and $\tilde{\sigma}$ are used which are estimated from the training data.
- **Problem:** BN statistics have to be recomputed before evaluation of the networks if the weights have been changed.
- **Workaround:** Recompute BN statistics for each sample of weights by doing forward-pass on the training data.

Experimental results

Measuring Uncertainty

- Negative Log-Likelihood: $\text{NLL} = -\frac{1}{N} \sum_{i=1}^N \log p_{\text{model}}(y_{\text{test}}^i \mid x_{\text{test}}^i)$.
- Calibration — a measure of difference between confidence and accuracy of a model.

$$\mathbb{P}(\hat{Y} = Y \mid \hat{P} = p) = p, \quad \forall p \in [0, 1].$$

Expected Calibration Error (ECE):

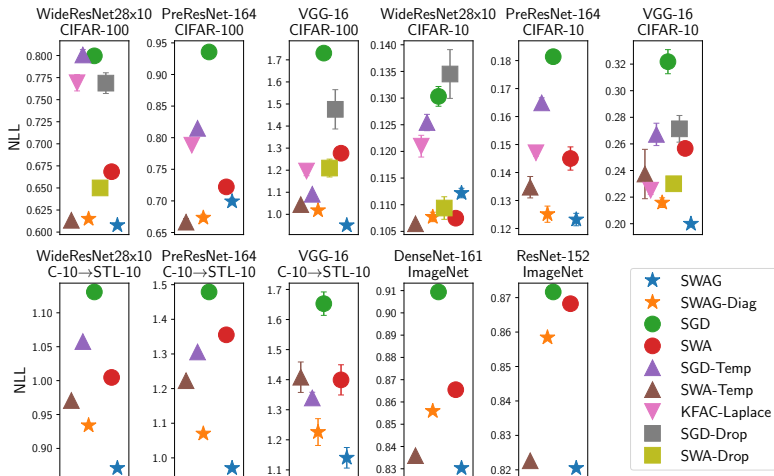
$$\text{ECE} = \mathbb{E}_{\hat{P}} \left[\left| \mathbb{P}(\hat{Y} = Y \mid \hat{P} = p) - p \right| \right]$$

- Predictive uncertainty evaluation on out-of-sample points.

Accuracy (%)

Dataset	Model	SGD	SWA	SWAG-Diag	SWAG	KFAC-Laplace	SWA-Dropout	SWA-Temp
CIFAR-10	VGG-16	93.17	93.61	93.66	93.60	92.65	93.23	93.61
CIFAR-10	PreResNet-164	95.49	96.09	96.03	96.03	95.49	96.18	96.09
CIFAR-10	WideResNet28x10	96.41	96.46	96.41	96.32	96.17	96.39	96.46
CIFAR-100	VGG-16	73.15	74.30	74.68	74.77	72.38	72.50	74.30
CIFAR-100	PreResNet-164	78.50	80.19	80.18	79.90	78.51		80.19
CIFAR-100	WideResNet28x10	80.76	82.40	82.40	82.23	80.94	82.30	82.40
ImageNet	DenseNet-161	77.79	78.60	78.59	78.59			78.60
ImageNet	ResNet-152	78.39	78.92	78.96	79.08			78.92
CIFAR10 → STL10	VGG-16	72.42	71.92	72.09	72.19		71.45	71.92
CIFAR10 → STL10	PreResNet-164	75.56	76.02	75.95	75.88			76.02
CIFAR10 → STL10	WideResNet28x10	76.75	77.50	77.26	77.09		76.91	77.50

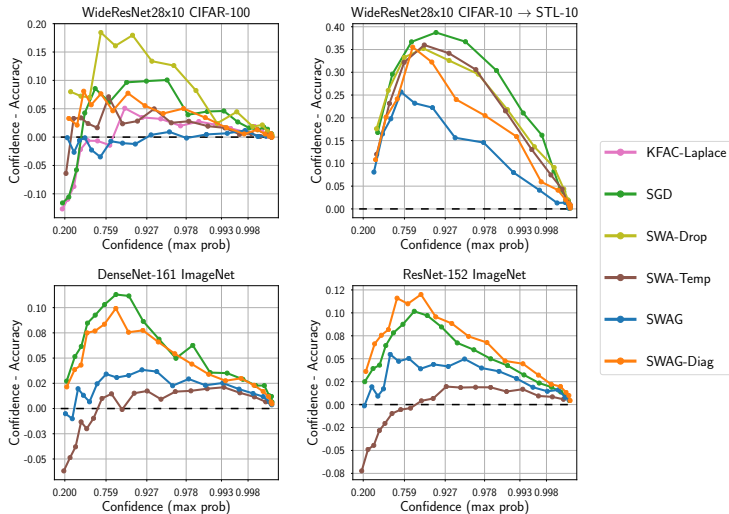
Test Set Negative Log-Likelihood



Test Set Negative Log-Likelihood

Dataset	Model	SGD	SWA	SWAG-Diag	SWAG	KFAC-Laplace	SWA-Dropout	SWA-Temp
CIFAR-10	VGG-16	0.3285	0.2621	0.2200	0.2016	0.2252	0.2328	0.2481
CIFAR-10	PreResNet-164	0.1814	0.1450	0.1251	0.1232	0.1471	0.1270	0.1347
CIFAR-10	WideResNet28x10	0.1294	0.1075	0.1077	0.1122	0.1210	0.1094	0.1064
CIFAR-100	VGG-16	1.7308	1.2780	1.0163	0.9480	1.1915	1.1872	1.0386
CIFAR-100	PreResNet-164	0.9465	0.7370	0.6837	0.7081	0.7881		0.6770
CIFAR-100	WideResNet28x10	0.7958	0.6684	0.6150	0.6078	0.7692	0.6500	0.6134
ImageNet	DenseNet-161	0.9094	0.8655	0.8559	0.8303			0.8359
ImageNet	ResNet-152	0.8716	0.8682	0.8584	0.8205			0.8226
CIFAR10 → STL10	VGG-16	1.6528	1.3993	1.2258	1.1402		1.3133	1.4082
CIFAR10 → STL10	PreResNet-164	1.4790	1.3552	1.0700	0.9706			1.2228
CIFAR10 → STL10	WideResNet28x10	1.1308	1.0047	0.9340	0.8710		0.9914	0.9706

Calibration

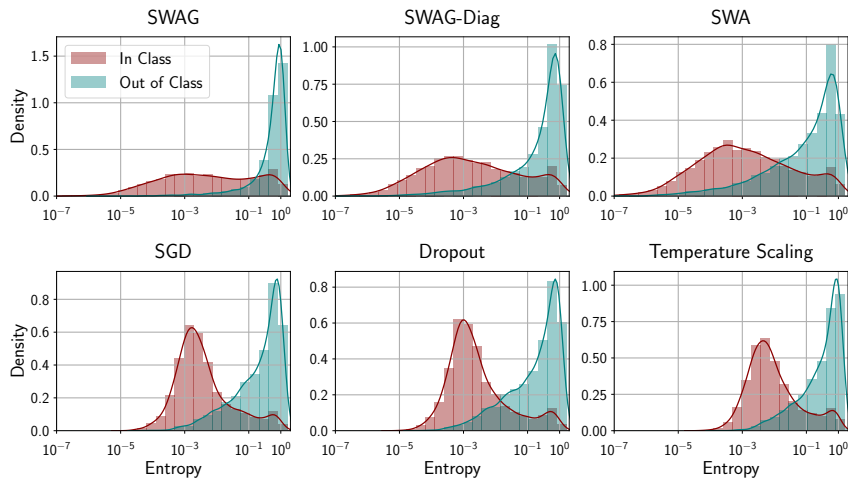


Expected Calibration Error (ECE)

Dataset	Model	SGD	SWA	SWAG-Diag	SWAG	KFAC-Laplace	SWA-Dropout	SWA-Temp
CIFAR-10	VGG-16	0.0483	0.0408	0.0267	0.0158	0.0094	0.0284	0.0366
CIFAR-10	PreResNet-164	0.0255	0.0203	0.0082	0.0053	0.0092	0.0162	0.0172
CIFAR-10	WideResNet28x10	0.0166	0.0087	0.0047	0.0088	0.0060	0.0094	0.0080
CIFAR-100	VGG-16	0.1870	0.1514	0.0819	0.0395	0.0778	0.1108	0.0291
CIFAR-100	PreResNet-164	0.1012	0.0700	0.0239	0.0587	0.0158		0.0175
CIFAR-100	WideResNet28x10	0.0479	0.0684	0.0322	0.0113	0.0379	0.0574	0.0220
ImageNet	DenseNet-161	0.0545	0.0509	0.0459	0.0204			0.0190
ImageNet	ResNet-152	0.0478	0.0605	0.0566	0.0279			0.0183
CIFAR10 → STL10	VGG-16	0.2149	0.2082	0.1719	0.1463		0.1803	0.2089
CIFAR10 → STL10	PreResNet-164	0.1758	0.1739	0.1312	0.1110			0.1646
CIFAR10 → STL10	WideResNet28x10	0.1561	0.1413	0.1241	0.1017		0.1421	0.1371

Out-of-sample image detection

WideResNet, CIFAR-(5+5)



Out-of-sample image detection

WideResNet, CIFAR-(5+5)
Symmetrized, discretized KL divergence
between the distributions of predictive entropies.

Method	KL-Distance
SGD (Baseline)	3.14
SGD + Temp. Scaling	2.98
MC Dropout	3.04
SWA	1.68
SWAG-Diag	2.27
SWAG	3.31

Paper:

- A Simple Baseline for Bayesian Uncertainty in Deep Learning

<https://arxiv.org/abs/1902.02476>

Code:

- SWAG in PyTorch

github.com/wjmaddox/swa_gaussian



Blundell, Charles et al. (2015). “Weight Uncertainty in Neural Networks”. In: *International Conference on Machine Learning*. arXiv: 1505.05424. URL: <http://arxiv.org/abs/1505.05424> (visited on 01/21/2019).








Chen, Tianqi et al. (2014). “Stochastic Gradient Hamiltonian Monte Carlo”. en. In: *International Conference on Machine Learning*. arXiv: 1402.4102. URL: <http://arxiv.org/abs/1402.4102> (visited on 12/11/2018).



Gal, Yarin et al. (2016). “Dropout as a Bayesian Approximation”. In: *International Conference on Machine Learning*. URL: <http://proceedings.mlr.press/v48/gal16.pdf> (visited on 02/22/2018).



Guo, Chuan et al. (2017). “On Calibration of Modern Neural Networks”. In: *International Conference on Machine Learning*. arXiv: 1706.04599. URL: <http://arxiv.org/abs/1706.04599> (visited on 04/20/2018).

-  Izmailov, Pavel et al. (2018). “Averaging Weights Leads to Wider Optima and Better Generalization”. In: *Uncertainty in Artificial Intelligence*. arXiv: 1803.05407. 2018. URL: <http://arxiv.org/abs/1803.05407> (visited on 03/28/2018).
-  Lakshminarayanan, Balaji et al. (2017). “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *Advances in Neural Information Processing Systems*.
-  Maddox, Wesley et al. (2019). “A Simple Baseline for Bayesian Uncertainty in Deep Learning”. In: *arXiv pre-print*.
-  Mandt, Stephan et al. (2017). “Stochastic Gradient Descent as Approximate Bayesian Inference”. en. In: *JMLR* 18, pp. 1–35.
-  Ritter, Hippolyt et al. (2018). “A Scalable Laplace Approximation for Neural Networks”. In: *International Conference on Learning Representations*.