An Ensemble of Neural Nets

Rahul Ramesh

STAT-991

Problem Setting

We consider the supervised learning problem. The training data is denoted by

$$D = \{(x_i, y_i)\}_{i=1}^{n}.$$

Problem Setting

We consider the supervised learning problem. The training data is denoted by

$$D = \{(x_i, y_i)\}_{i=1}^n.$$

We seek a model $h = \mathcal{A}(D)$ that has low generalization error

$$e(h) = \mathbb{P}[h(x) \neq y],$$

Problem Setting

We consider the supervised learning problem. The training data is denoted by

$$D = \{(x_i, y_i)\}_{i=1}^{n}.$$

We seek a model $h = \mathcal{A}(D)$ that has low generalization error

$$e(h) = \mathbb{P}[h(x) \neq y],$$

and is well-calibrated.

$$\gamma = \mathbb{P}(Y|p_h(Y|X) = q)$$
$$c(h) = \mathop{\mathbb{E}}_{q}\left[d(\gamma, q)\right]$$

Bayesian Deep Learning

Why should we estimate the Bayes posterior for a neural network?

Bayesian Deep Learning

Why should we estimate the Bayes posterior for a neural network?

- Accuracy

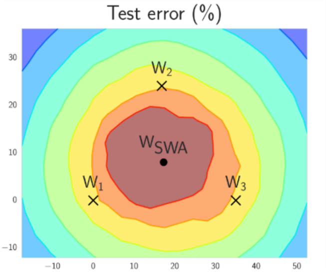


Figure: See Izmailov et al. (2018)

Bayesian Deep Learning

Why should we estimate the Bayes posterior for a neural network?

- Accuracy
- Calibration

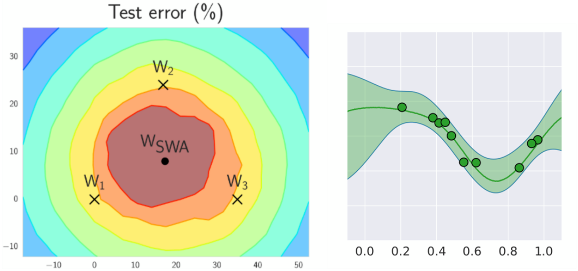


Figure: See Izmailov et al. (2018) and Ovadia et al. (2019)

Outline

We discuss some popular Bayesian methods for deep learning.

Outline

We discuss some popular Bayesian methods for deep learning.

Ensembles have emerged as useful approximations of the Bayes posterior.

Outline

We discuss some popular Bayesian methods for deep learning.

Ensembles have emerged as useful approximations of the Bayes posterior.

We explore some recent results that make use of ensembles.

Bayesian Deep Learning

The Bayes posterior

Consider a dataset $D = \{x_i, y_i\}_{i=1}^n$ and the negative log-likelihood

$$U(\theta) = -\log p(\theta) - \sum_{i=1}^{n} \log p(y_i | x_i, \theta).$$

The Bayes posterior

Consider a dataset $D = \{x_i, y_i\}_{i=1}^n$ and the negative log-likelihood

$$U(\theta) = -\log p(\theta) - \sum_{i=1}^{n} \log p(y_i | x_i, \theta).$$

Usually, we train the parameters to minimize this function

$$\theta_{\text{map}} = \underset{\theta}{\text{argmin}}\, U(\theta)$$

The Bayes posterior

Consider a dataset $D = \{x_i, y_i\}_{i=1}^n$ and the negative log-likelihood

$$U(\theta) = -\log p(\theta) - \sum_{i=1}^{n} \log p(y_i|x_i, \theta).$$

Usually, we train the parameters to minimize this function

$$\theta_{\text{map}} = \underset{\theta}{\text{argmin}}\, U(\theta)$$

Instead, we hedge our bets and consider a distribution over parameters

$$p(\theta|D) \propto p(D|\theta)p(\theta) = \exp(-U(\theta))$$

How do we make predictions?

How do we make predictions?

We marginalize over the entire posterior to obtain the predictive distribution

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D) \ \mathrm{d}\theta.$$

How do we make predictions?

We marginalize over the entire posterior to obtain the predictive distribution

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D)\ \mathrm{d}\theta.$$

In practice, we use a Monte Carlo approximation. First sample

$$\theta_1, \theta_2, \cdots \theta_k \sim p(\theta|D),$$

How do we make predictions?

We marginalize over the entire posterior to obtain the predictive distribution

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D) \ \mathrm{d}\theta.$$

In practice, we use a Monte Carlo approximation. First sample

$$\theta_1, \theta_2, \cdots \theta_k \sim p(\theta|D),$$

and use them to make predictions.

$$p(y|x, D) \approx \frac{1}{k} \sum_{i=1}^{k} p(y|x, \theta_i)$$

How do we train a model?

We are interested in estimating

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D)\ \mathrm{d}\theta.$$

Existing methods:

How do we train a model?

We are interested in estimating

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D) \; \mathrm{d}\theta.$$

Existing methods:

1. Variational approximation ($q_w(\theta) \approx p(\theta|D)$)

How do we train a model?

We are interested in estimating

$$p(y|x, D) = \int p(y|x, \theta) \color{darkred}{p(\theta|D)} \, \mathrm{d}\theta.$$

Existing methods:

1. Variational approximation ($q_w(\theta) \approx p(\theta|D)$)
2. Markov-chain Monte-carlo (MCMC)

Variational Approximation

The ELBO forms the basis of VI methods (Blundell et al., 2015; Wen et al., 2018; Louizos and Welling, 2016)

$$\log p(y|x) \geq \mathbb{E}_{q_w(\theta)}[\log p(y|x,\theta)] - \mathbb{E}_{q_w(\theta)}\left[\log \frac{q_w(\theta)}{p(\theta)}\right]$$

Variational Approximation

The ELBO forms the basis of VI methods (Blundell et al., 2015; Wen et al., 2018; Louizos and Welling, 2016)

$$\log p(y|x) \geq \mathbb{E}_{q_w(\theta)}[\log p(y|x,\theta)] - \mathbb{E}_{q_w(\theta)}\left[\log \frac{q_w(\theta)}{p(\theta)}\right]$$

For example, assume Σ is diagonal and

$$\theta \sim \mathcal{N}(\mu, \Sigma)$$

Variational Approximation

The ELBO forms the basis of VI methods (Blundell et al., 2015; Wen et al., 2018; Louizos and Welling, 2016)

$$\log p(y|x) \geq \mathbb{E}_{q_w(\theta)}[\log p(y|x,\theta)] - \mathbb{E}_{q_w(\theta)}\left[\log \frac{q_w(\theta)}{p(\theta)}\right]$$

For example, assume Σ is diagonal and

$$\theta \sim \mathcal{N}(\mu, \Sigma)$$

$w = (\mu, \Sigma)$ are now the learnable parameters.

Variational Approximation

The ELBO forms the basis of VI methods (Blundell et al., 2015; Wen et al., 2018; Louizos and Welling, 2016)

$$\log p(y|x) \geq \mathbb{E}_{q_w(\theta)}[\log p(y|x,\theta)] - \mathbb{E}_{q_w(\theta)}\left[\log \frac{q_w(\theta)}{p(\theta)}\right]$$

For example, assume Σ is diagonal and

$$\theta \sim \mathcal{N}(\mu, \Sigma)$$

$w = (\mu, \Sigma)$ are now the learnable parameters. We assume $p(\theta) = \mathcal{N}(0, \mathbb{I})$.

Variational Approximation - Dropout

Consider dropout (Gal and Ghahramani, 2016), which randomly sets some parameters to 0 during training.

Variational Approximation - Dropout

Consider dropout (Gal and Ghahramani, 2016), which randomly sets some parameters to 0 during training.

Let the variation family be

$$q_\mu(\theta) = \begin{cases} \mu & \text{with probability } d \\ 0 & \text{otherwise} \end{cases}$$

where d is fixed.

Variational Approximation - Dropout

Consider dropout (Gal and Ghahramani, 2016), which randomly sets some parameters to 0 during training.

Let the variation family be

$$q_\mu(\theta) = \begin{cases} \mu & \text{with probability } d \\ 0 & \text{otherwise} \end{cases}$$

where d is fixed.
The first term

$$\mathbb{E}_{\theta \sim q_\mu}[\log p(y|x, \theta)]$$

is the gradient with dropout.

Variational Approximation - Dropout

Consider dropout (Gal and Ghahramani, 2016), which randomly sets some parameters to 0 during training.

Let the variation family be

$$q_\mu(\theta) = \begin{cases} \mu & \text{with probability } d \\ 0 & \text{otherwise} \end{cases}$$

where d is fixed.
The first term

$$\mathbb{E}_{\theta \sim q_\mu}[\log p(y|x, \theta)]$$

is the gradient with dropout.
The second term is usually absent in an implementation of dropout

$$\mathbb{E}_{q_\mu(\theta)}[\log \frac{q_\mu(\theta)}{p(\theta)}]$$

but approximately corresponds to the L2-penalty.

MCMC

Before describing the details of the algorithm, we revisit $p(\theta|D)$.

MCMC

Before describing the details of the algorithm, we revisit $p(\theta|D)$.

Consider the Gibbs distribution

$$p(\omega) \propto \exp\left(-\frac{\epsilon(\omega)}{kT}\right).$$

MCMC

Before describing the details of the algorithm, we revisit $p(\theta|D)$.

Consider the Gibbs distribution

$$p(\omega) \propto \exp\left(-\frac{\epsilon(\omega)}{kT}\right).$$

Similarly, we introduce the temperature parameter T:

$$\hat{p}(\theta|D) \propto \exp\left(-\frac{U(\theta)}{T}\right).$$

MCMC

Before describing the details of the algorithm, we revisit $p(\theta|D)$.

Consider the Gibbs distribution

$$p(\omega) \propto \exp\left(-\frac{\epsilon(\omega)}{kT}\right).$$

Similarly, we introduce the temperature parameter T:

$$\hat{p}(\theta|D) \propto \exp\left(-\frac{U(\theta)}{T}\right).$$

$T = 1$ corresponds to the Bayes posterior.

MCMC

Before describing the details of the algorithm, we revisit $p(\theta|D)$.

Consider the Gibbs distribution

$$p(\omega) \propto \exp\left(-\frac{\epsilon(\omega)}{kT}\right).$$

Similarly, we introduce the temperature parameter T:

$$\hat{p}(\theta|D) \propto \exp\left(-\frac{U(\theta)}{T}\right).$$

$T = 1$ corresponds to the Bayes posterior.

Parameters with low losses are sampled more frequently.

MCMC - Langevin Equation

Consider the stochastic differential equation (SDE)

$$d\theta(t) = -\nabla U(\theta(t))\,\mathrm{d}t + \sqrt{2T}\,\mathrm{d}B_t$$

MCMC - Langevin Equation

Consider the stochastic differential equation (SDE)

$$d\theta(t) = -\nabla U(\theta(t))\,\mathrm{d}t + \sqrt{2T}\,\mathrm{d}B_t$$

This distribution converges to a stationary distribution where

$$p^*(\theta) \propto \exp\left(\frac{-U(\theta)}{T}\right)$$

MCMC - Langevin Equation

Consider the stochastic differential equation (SDE)

$$d\theta(t) = -\nabla U(\theta(t))\,\mathrm{d}t + \sqrt{2T}\,\mathrm{d}B_t$$

This distribution converges to a stationary distribution where

$$p^*(\theta) \propto \exp\left(\frac{-U(\theta)}{T}\right)$$

Hence, if we simulate the SDE, we will converge to the stationary distribution

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{} = -\nabla U(\theta(t))\underbrace{\mathrm{d}t}_{} + \sqrt{2T}\underbrace{\mathrm{d}B_t}_{}$$

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t} + \sqrt{2T} \underbrace{\mathrm{d}B_t}$$

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t}_{\eta} + \sqrt{2T}\, \underbrace{\mathrm{d}B_t}$$

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t}_{\eta} + \sqrt{2T} \underbrace{\mathrm{d}B_t}_{\mathcal{N}(0,\eta)}$$

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t}_{\eta} + \sqrt{2T} \underbrace{\mathrm{d}B_t}_{\mathcal{N}(0,\eta)}$$

$$\theta_{t+1} - \theta_t = -\eta \nabla U(\theta(t)) + \sqrt{2T}\xi_t$$

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t}_{\eta} + \sqrt{2T} \underbrace{\mathrm{d}B_t}_{\mathcal{N}(0,\eta)}$$

$$\theta_{t+1} - \theta_t = -\eta \nabla U(\theta(t)) + \sqrt{2T}\xi_t$$

Simulating this equation will give us samples from the distribution.

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t}_{\eta} + \sqrt{2T} \underbrace{\mathrm{d}B_t}_{\mathcal{N}(0,\eta)}$$

$$\theta_{t+1} - \theta_t = -\eta \nabla U(\theta(t)) + \sqrt{2T}\xi_t$$

Simulating this equation will give us samples from the distribution.

We can use mini-batch version of this update (Welling and Teh, 2011)

MCMC - Langevin Equation

Start from a random θ_0.

$$\underbrace{d\theta(t)}_{\theta_{t+1}-\theta_t} = -\nabla U(\theta(t)) \underbrace{\mathrm{d}t}_{\eta} + \sqrt{2T} \underbrace{\mathrm{d}B_t}_{\mathcal{N}(0,\eta)}$$

$$\theta_{t+1} - \theta_t = -\eta \nabla U(\theta(t)) + \sqrt{2T}\xi_t$$

Simulating this equation will give us samples from the distribution.

We can use mini-batch version of this update (Welling and Teh, 2011)

$$\theta_{t+1} = \theta_t - \eta \nabla [U(\theta(t))]_i + \sqrt{2T}\xi_t$$

MCMC - Hamiltonian Monte Carlo

Hamiltonian Monte-carlo is the ”gold-standard” for the MCMC methods.

MCMC - Hamiltonian Monte Carlo

Hamiltonian Monte-carlo is the ”gold-standard” for the MCMC methods.

$$
\begin{aligned}
\mathrm{d}\theta &= v\,\mathrm{d}t \\
\mathrm{d}v &= -\nabla U(\theta)\,\mathrm{d}t - \gamma v\,\mathrm{d}t + \sqrt{2\gamma T}\,\mathrm{d}B_t
\end{aligned}
$$

MCMC - Hamiltonian Monte Carlo

Hamiltonian Monte-carlo is the ”gold-standard” for the MCMC methods.

$$\begin{aligned} \mathrm{d}\theta &= v\,\mathrm{d}t \\ \mathrm{d}v &= -\nabla U(\theta)\,\mathrm{d}t - \gamma v\,\mathrm{d}t + \sqrt{2\gamma T}\,\mathrm{d}B_t \end{aligned}$$

Introduce a ”velocity” variable and attempt to conserve the Hamiltonian of the system.

A closer look at Bayesian posteriors

How Good is the Bayes Posterior in Deep Neural Networks Really?

Florian Wenzel[* 1] **Kevin Roth**[* + 2] **Bastiaan S. Veeling**[* + 3 1] **Jakub Świątkowski**[4 +] **Linh Tran**[5 +]
Stephan Mandt[6 +] **Jasper Snoek**[1] **Tim Salimans**[1] **Rodolphe Jenatton**[1] **Sebastian Nowozin**[7 +]

What Are Bayesian Neural Network Posteriors Really Like?

Pavel Izmailov[1] **Sharad Vikram**[2] **Matthew D. Hoffman**[2] **Andrew Gordon Wilson**[1]

A quick recap

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

A quick recap

- $p(\theta|D) \propto \exp(\frac{-U(\theta)}{T})$
- We can use MCMC methods to sample from this distribution

A quick recap

- $p(\theta|D) \propto \exp(\frac{-U(\theta)}{T})$
- We can use MCMC methods to sample from this distribution

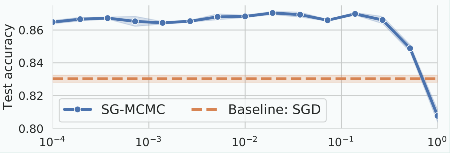
MCMC is compute intensive but accurate.

Cold posteriors

Wenzel et al. (2020) show that $T < 1$ is better.

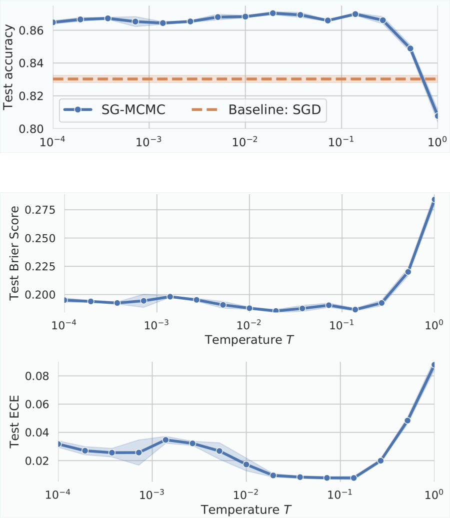
Cold posteriors

Wenzel et al. (2020) show that $T < 1$ is better.



Cold posteriors

Wenzel et al. (2020) show that $T < 1$ is better.



Cold posteriors

Most MCMC methods in literature use a cold-posterior.

Related work that uses $T < 1$ posteriors in SG-MCMC. The following table lists work that uses SG-MCMC on deep neural networks and tempers the posterior.[3]

Reference	Temperature T
(Li et al., 2016)	$1/\sqrt{n}$
(Leimkuhler et al., 2019)	$T < 10^{-3}$
(Heek & Kalchbrenner, 2020)	$T = 1/5$
(Zhang et al., 2020)	$T = 1/\sqrt{50000}$

Cold posteriors

Most MCMC methods in literature use a cold-posterior.

Related work that uses $T < 1$ posteriors in SG-MCMC. The following table lists work that uses SG-MCMC on deep neural networks and tempers the posterior.[3]

Reference	Temperature T
(Li et al., 2016)	$1/\sqrt{n}$
(Leimkuhler et al., 2019)	$T < 10^{-3}$
(Heek & Kalchbrenner, 2020)	$T = 1/5$
(Zhang et al., 2020)	$T = 1/\sqrt{50000}$

This is problematic since we artificially sharpen the posterior and scale the variance of the prior.

Why is $T < 1$ better?

Wenzel et al. (2020) hypothesize that cold-posteriors are better because

- $p(y|x, \theta)$ is not a likelihood function due to batch-norm, dropout and data-augmentation

Why is $T < 1$ better?

Wenzel et al. (2020) hypothesize that cold-posteriors are better because

- $p(y|x, \theta)$ is not a likelihood function due to batch-norm, dropout and data-augmentation
- Inadequate prior

Why is $T < 1$ better?

Wenzel et al. (2020) hypothesize that cold-posteriors are better because

- $p(y|x, \theta)$ is not a likelihood function due to batch-norm, dropout and data-augmentation
- Inadequate prior
- SGD does not work with Bayesian methods

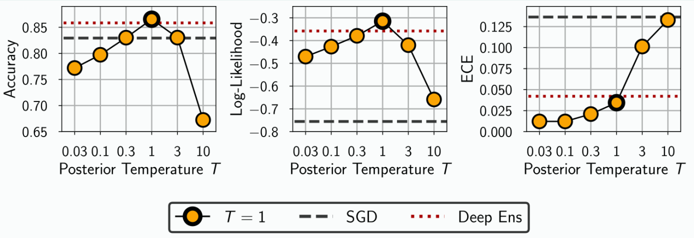
Is $T < 1$ better?

Izmailov et al. (2021) conduct large scale HMC experiments.

Is $T < 1$ better?

Izmailov et al. (2021) conduct large scale HMC experiments.

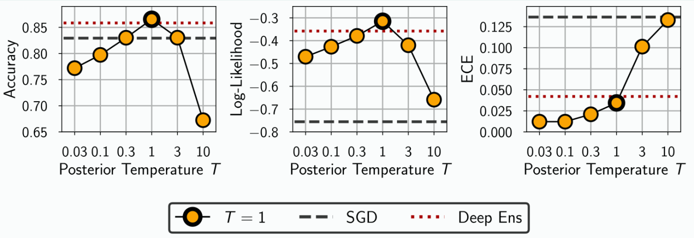
- More compute (6̃0 million epochs on CIFAR).



Is $T < 1$ better?

Izmailov et al. (2021) conduct large scale HMC experiments.

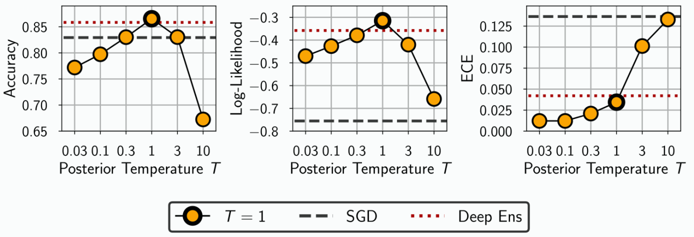
- More compute (6̃0 million epochs on CIFAR).
- No data augmentations



Is $T < 1$ better?

Izmailov et al. (2021) conduct large scale HMC experiments.

- More compute (6̃0 million epochs on CIFAR).
- No data augmentations
- Batch-norm $\rightarrow$ filter-response normalization



In summary

- Variational inference is inaccurate

In summary

- Variational inference is inaccurate
- MCMC (HMC) is useful but

In summary

- Variational inference is inaccurate
- MCMC (HMC) is useful but
    - Compute hungry

In summary

- Variational inference is inaccurate
- MCMC (HMC) is useful but
    - Compute hungry
    - Doesn't work with augmentations yet

In summary

- Variational inference is inaccurate
- MCMC (HMC) is useful but
  - Compute hungry
  - Doesn't work with augmentations yet
- We want an inexpensive and accurate model that captures the Bayes posterior

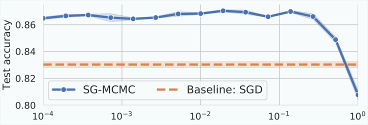
Ensembles

Emergence of Ensembles

Revisiting earlier experiments

Emergence of Ensembles

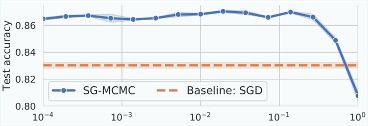
Revisiting earlier experiments

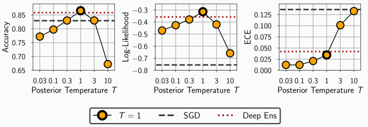


- Cold posteriors are better

Emergence of Ensembles

Revisiting earlier experiments





- Cold posteriors are better
- Deep ensembles match the accuracy and calibration of HMC.

Emergence of Ensembles

We focus on ensembles, which are finite particle approximations of the posterior

$$p(\theta|D) \approx \sum_{i=1}^{k} \frac{1}{k}\delta(\theta - \theta_i)$$

Ensembles are common in machine learning (Breiman, 1996; Schapire, 1990).

Emergence of Ensembles

We focus on ensembles, which are finite particle approximations of the posterior

$$p(\theta|D) \approx \sum_{i=1}^{k} \frac{1}{k}\delta(\theta - \theta_i)$$

Ensembles are common in machine learning (Breiman, 1996; Schapire, 1990).

Instead of training a single neural net

$$\theta$$

Emergence of Ensembles

We focus on ensembles, which are finite particle approximations of the posterior

$$p(\theta|D) \approx \sum_{i=1}^{k} \frac{1}{k}\delta(\theta - \theta_i)$$

Ensembles are common in machine learning (Breiman, 1996; Schapire, 1990).

Instead of training a single neural net

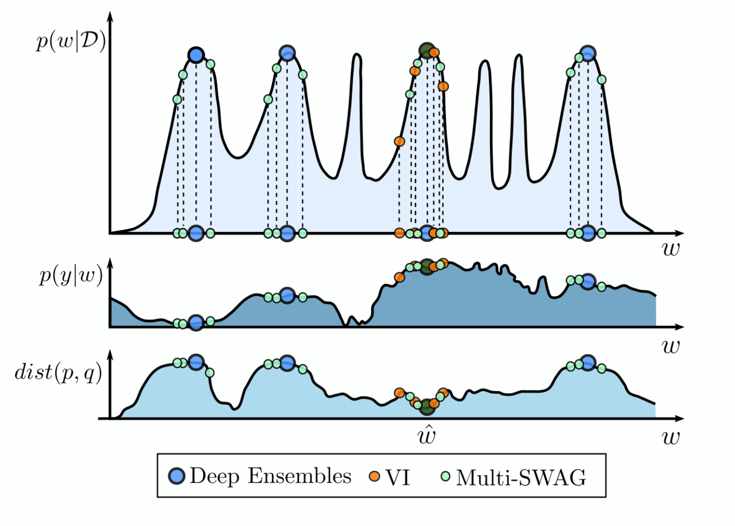
$$\theta$$

we train k copies of it, each initialized randomly:

$$\theta_1 \quad \theta_2 \quad \cdots \quad \theta_k$$

Why do they work?

Ensembles capture different modes (Wilson and Izmailov, 2020).



Why do they work?

Variational inference methods are not nearly as diverse (Fort et al., 2019).

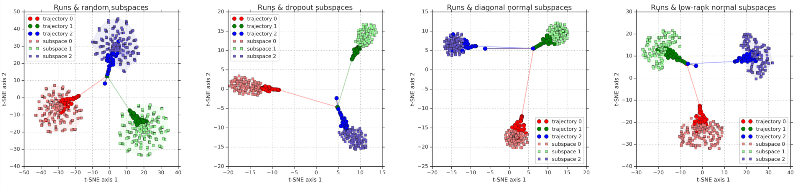


Figure: 1) Random 2) Dropout 3) Diagonal Gaussian 4) Low-rank Gaussian

Why do they work?

Variational inference methods are not nearly as diverse (Fort et al., 2019).

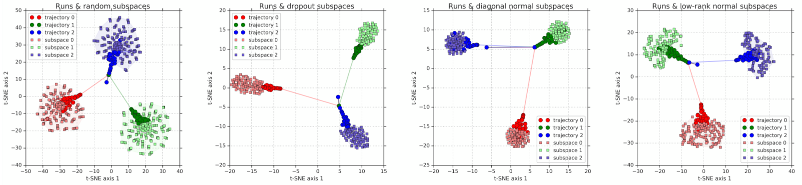


Figure: 1) Random 2) Dropout 3) Diagonal Gaussian 4) Low-rank Gaussian

Initialization influences diversity in predictions more than other factors.

Are ensembles Bayesian?

D'Angelo and Fortuin (2021) minimize the KL-divergence between

$$p(\theta|x) \propto \exp(-U(\theta))$$

Are ensembles Bayesian?

D'Angelo and Fortuin (2021) minimize the KL-divergence between

$$p(\theta|x) \propto \exp(-U(\theta))$$

and

$$\rho(\theta) = \sum_{i=1}^{k} w_i \delta(\theta - \theta_i)$$

Are ensembles Bayesian?

D'Angelo and Fortuin (2021) minimize the KL-divergence between

$$p(\theta|x) \propto \exp(-U(\theta))$$

and

$$\rho(\theta) = \sum_{i=1}^{k} w_i \delta(\theta - \theta_i)$$

to get the update equation

$$\theta_i^{t+1} = \left(\theta_i^t - \eta \nabla U(\theta_i^t)\right) - \text{Repulsion}$$

Are ensembles Bayesian?

D'Angelo and Fortuin (2021) minimize the KL-divergence between

$$p(\theta|x) \propto \exp(-U(\theta))$$

and

$$\rho(\theta) = \sum_{i=1}^{k} w_i \delta(\theta - \theta_i)$$

to get the update equation

$$\theta_i^{t+1} = \left(\theta_i^t - \eta \nabla U(\theta_i^t)\right) - \text{Repulsion}$$

In practice, we don't use the repulsion term and find random initializations to be sufficient.

Evaluating ensembles on OOD data

Ovadia et al. (2019) evaluate ensembles under distribution shift (heavy augmentations)

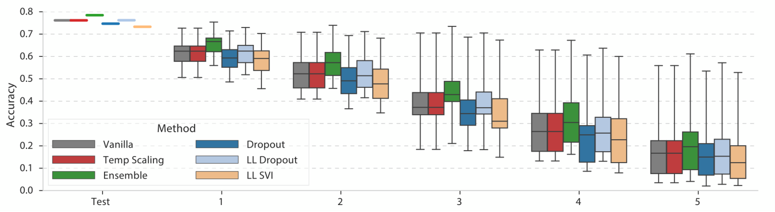
Evaluating ensembles on OOD data

Ovadia et al. (2019) evaluate ensembles under distribution shift (heavy augmentations)

We focus in evaluations in Imagenet, which used at most 10 models for ensembling.

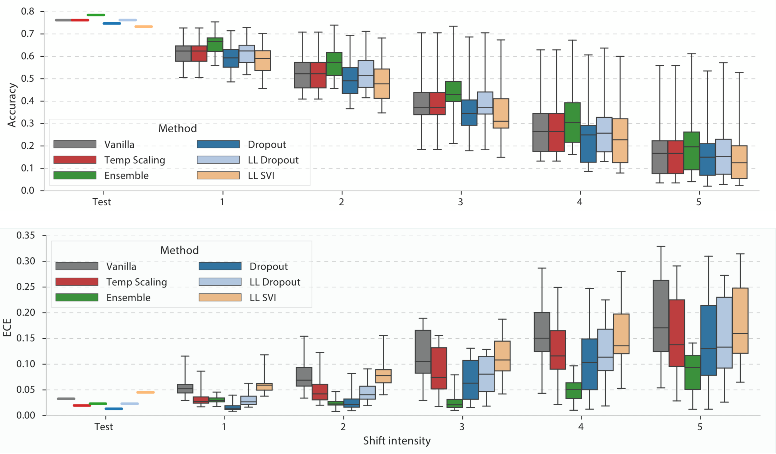
Evaluating ensembles on OOD data

Ensembles are well-calibrated and usually more accurate.



Evaluating ensembles on OOD data

Ensembles are well-calibrated and usually more accurate.



What about the prior?

The negative log-likelihood

$$U(\theta) = -\log p(\theta) + \cdots$$

includes the prior.

What about the prior?

The negative log-likelihood

$$U(\theta) = -\log p(\theta) + \cdots$$

includes the prior.

Usually, the prior is

$$p(\theta) \stackrel{d}{=} \mathcal{N}(0, I)$$

Wilson and Izmailov (2020); Wenzel et al. (2020) mention the importance of the prior but it is relatively unexplored beyond Gaussians.

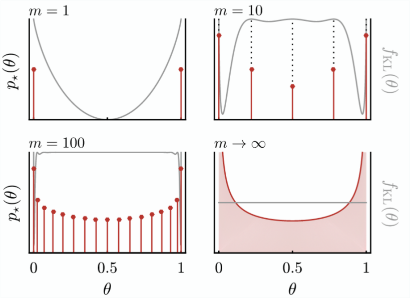
Deep Reference Priors

Gao et al. (2022) attempt to learn the prior from unlabeled data.

Deep Reference Priors

Gao et al. (2022) attempt to learn the prior from unlabeled data.

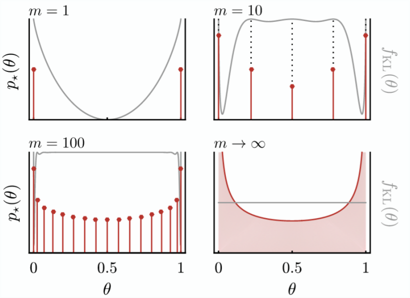
Reference priors are ”uninformative” and let the data dominate the posterior.



Deep Reference Priors

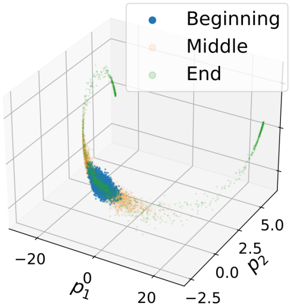
Gao et al. (2022) attempt to learn the prior from unlabeled data.

Reference priors are ”uninformative” and let the data dominate the posterior.

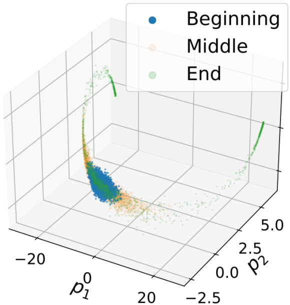


Reference priors are supported on a finite number of atoms

Deep Reference Priors



Deep Reference Priors



Method	Samples				
	50	100	250	500	1000
MixMatch	64.21	80.29	88.91	90.35	92.25
FixMatch (RA)	86.19 $\pm$ 3.37 (40)	90.12	94.93 $\pm$ 0.65	93.91	94.3
Deep Reference Prior	85.45 $\pm$ 2.12	88.53 $\pm$ 0.67	92.13 $\pm$ 0.39	92.94 $\pm$ 0.22	93.48 $\pm$ 0.24

Conclusion

We approximate the Bayes posterior using ensembles, which are effective even on large datasets.

Conclusion

We approximate the Bayes posterior using ensembles, which are effective even on large datasets.

Can we do better?

References I

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

D'Angelo, F. and Fortuin, V. (2021). Repulsive deep ensembles are bayesian. *Advances in Neural Information Processing Systems*, 34.

Fort, S., Hu, H., and Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Gao, Y., Ramesh, R., and Chaudhari, P. (2022). Deep reference priors: What is the best way to pretrain a model? *arXiv preprint arXiv:2202.00187*.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. (2021). What are bayesian neural network posteriors really like? In *International Conference on Machine Learning*, pages 4629–4640. PMLR.

Louizos, C. and Welling, M. (2016). Structured and efficient variational deep learning with matrix gaussian posteriors. In *International conference on machine learning*, pages 1708–1716. PMLR.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. (2018). Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*.

Wenzel, F., Roth, K., Veeling, B. S., Świątkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708.