A review of uncertainty quantification in deep learning*

Paper Presentation, CSE 8803-IUQ

Ashish Dhiman

Georgia Tech

*M Abdar et al (2021)

Agenda

- **Uncertainty Quantification in Deep Learning**
  - What , Why and How ?
- **How to quantify Uncertainty in Deep Learning**
  - Bayesian Methods
    - Monte Carlo Dropout
    - MCMC
    - Variational Inference
    - Laplace Approximation
  - Ensemble methods
    - Deep Ensembles
    - Evidential Deep Learning
- **Gaps and future work**

Georgia Tech

What is Uncertainty ?

- Uncertainty Quantification:
  - Return a distribution over predictions rather than a single prediction.
    - ***Classification***: Output label along with its confidence.
    - ***Regression***: Output mean along with its variance.

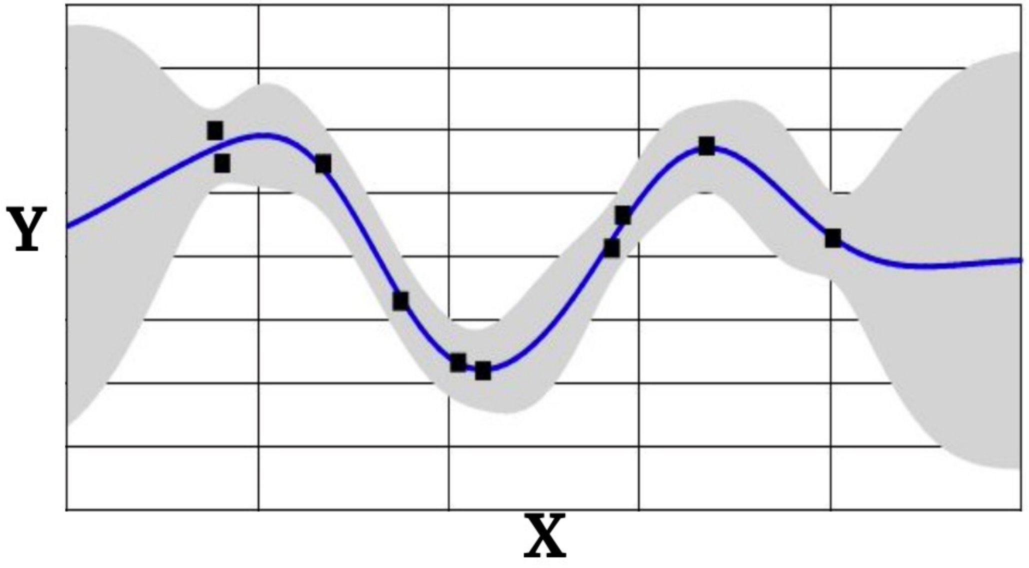
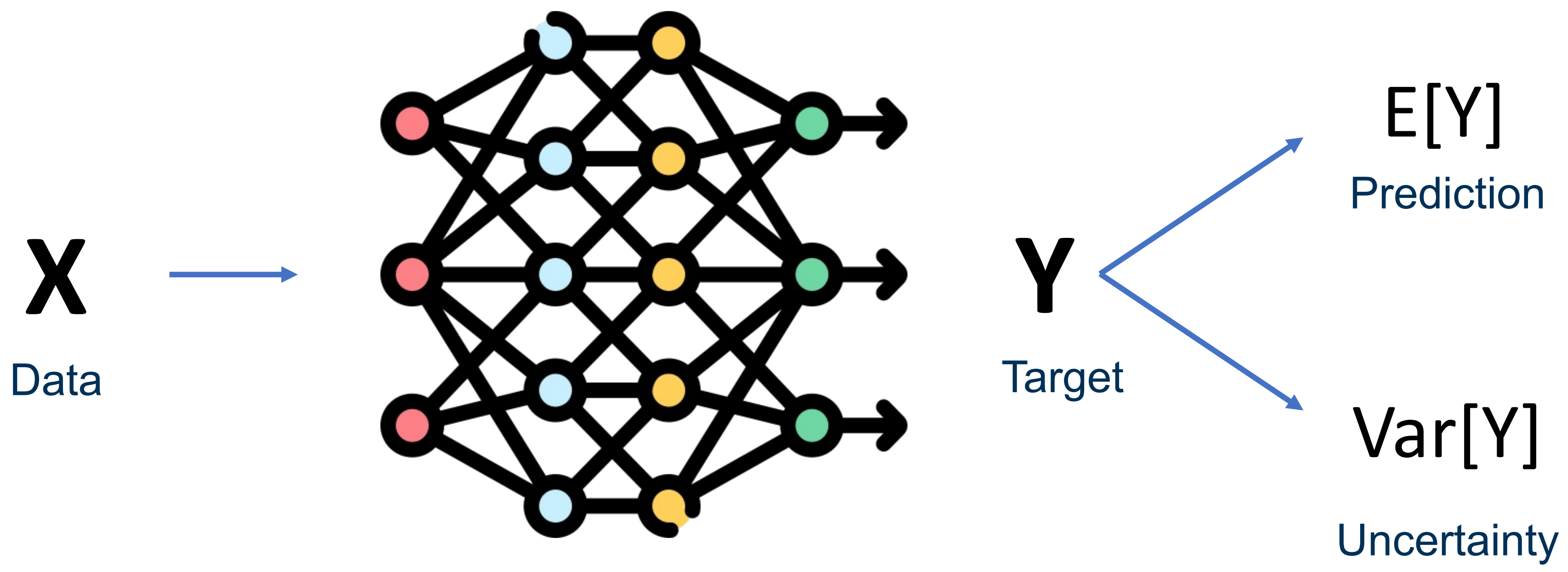


Image credit: Eric Nalisnick

Georgia Tech

What is UQ for Deep Learning?

- Typical Deep Learning methods only provide with a point prediction.
- UQ in Deep Learning provides **prediction** and ***confidence in the prediction***
- *Two types of Uncertainty: Aleatoric (data) and Epistemic (model)*



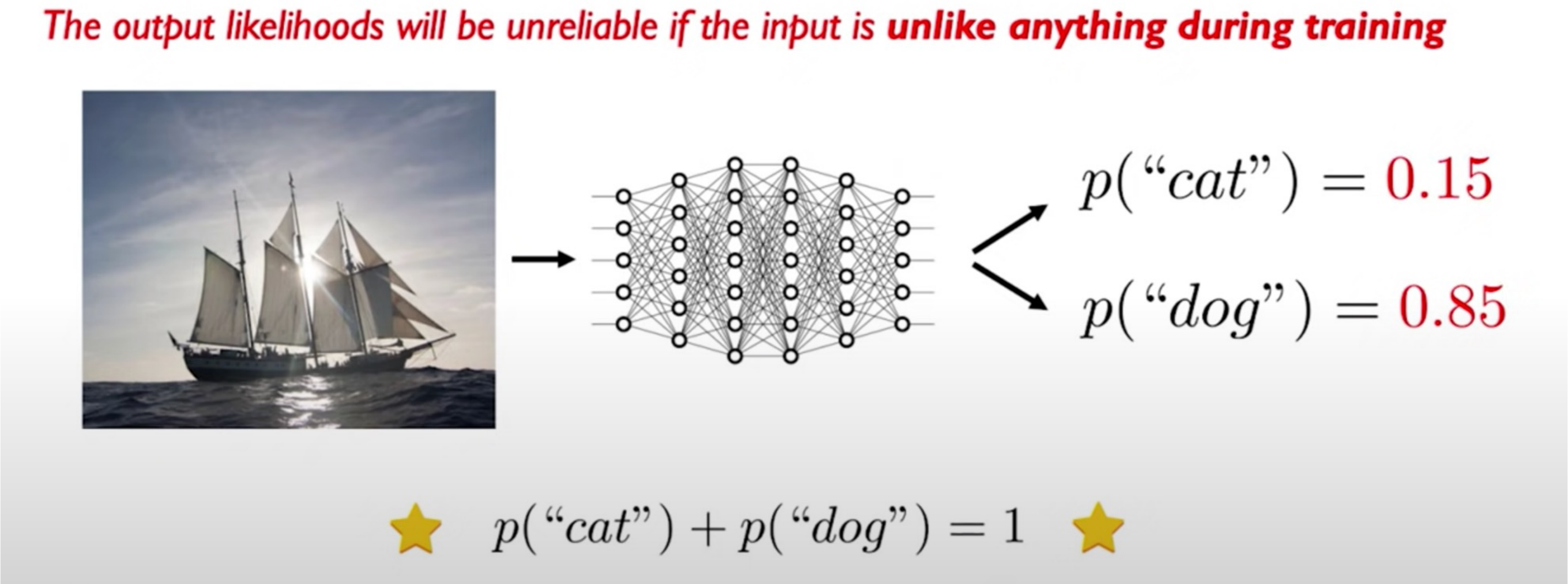
Georgia Tech

Why is UQ important for Deep Learning?

- Deep Learning models today used in numerous high stakes decisions.
- Often there is a huge disparity between the data used for training Deep Learning models, and the data they are finally used on.

$$P_{train}(x,y) \neq P_{test}(x,y)$$

OOD Distribution Shift



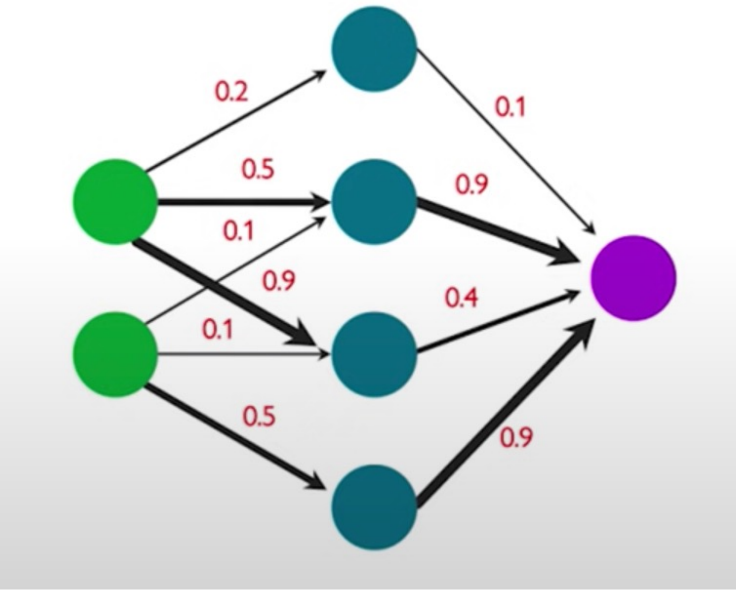
[*image source](#)

UQ methods for Deep Learning

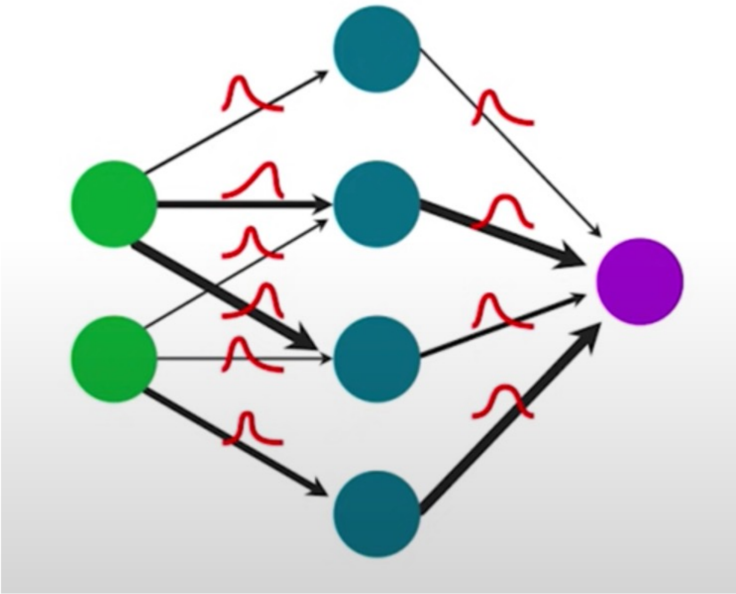
- Bayesian Methods
  - Monte Carlo Dropout
  - MCMC
  - Variational Inference
  - Laplace Approximation
- Ensemble Methods
  - Deep Ensembles
  - Evidential Deep Learning

UQ for Deep Learning: Bayesian Methods

- Instead of point estimates of NN weights, establish a distribution over the NN parameters



Deterministic NN



Bayesian NN

*image source

Georgia Tech

UQ for Deep Learning: Bayesian Methods

- Model a joint distribution over target (y) and parameters (θ): P(y, θ | x)
- Use it to calculate Posterior of parameters given data
- Use the same posterior for inference

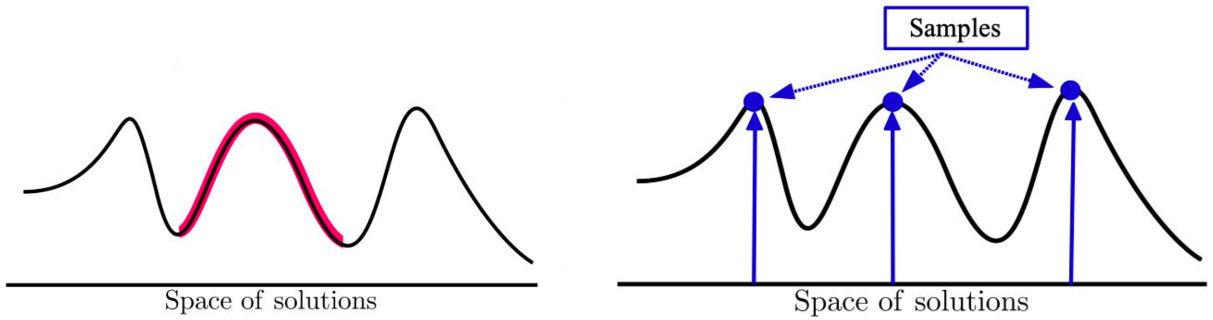
Intractable Lik Prior

Train $$P(\theta|D) = \frac{P(D|\theta) * P(\theta)}{P(D)} \propto P(D|\theta) * P(\theta)$$

Test $$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) = \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D})\, \mathrm{d}\boldsymbol{\theta}$$

Approximating the posterior

$p(\boldsymbol{\theta} \mid \mathcal{D})$ is multimodal and complex, so how do we estimate and represent it?



Local approximations **Sampling**

- Sampling: Monte Carlo Dropout, MCMC
- Local approximation: VI, Laplace approx.

*image source

Bayesian Methods: Monte Carlo Dropout

- Typically Dropout used in training as Regularization method
  - Randomly drop neurons in hidden layers, do forward and backward passes
- Monte Carlo Dropout is when Dropout used in testing/inference
  - Neurons are randomly sampled ‘n’ times basis a Bernoulli
  - We arrive at n predictions for target which can be used to get uncertainty

Literature shows Monte Carlo Dropout might be better for Deep BNNs

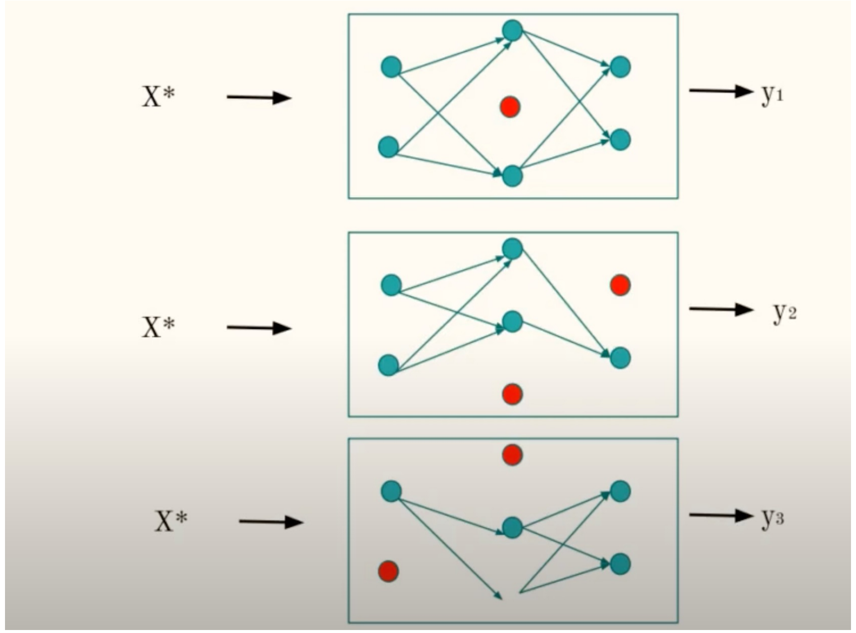


*image source

Bayesian Methods: MCMC (1/3)

- Set up a Markov Chain with stationary distribution that converges to Posterior P(θ | D)
  - Need to define Likelihood P(D| θ), and
  - Prior beliefs about parameters P(θ)

$$P(\theta|D) = \frac{P(D|\theta) * P(\theta)}{P(D)} \propto P(D|\theta) * P(\theta)$$

- NN training with Metropolis Hastings
  - Randomly initialize NN parameters θ
  - Run a MCMC chain with a proposal distribution J and acceptance ratio A
    - A = min(1,r)

$$r = \frac{P(\theta_*|D)}{P(\theta_{n-1}|D)}$$

$$J(\theta_*|\theta_{n-1}) = N(\theta_*|\theta_{n-1}, \alpha I)$$

(Need to chose step size α)

(Symmetric proposal J)

SG-MCMC approximate r with only a sample of points

*image source

Georgia Tech

Bayesian Methods: MCMC (2/3)

- Vanilla Metropolis Hastings involves a lot of random walk and hence useless steps
- Achieve quicker convergence by taking more informed steps

Langevin Diffusion

Let $U(\theta) = -\log P(D|\theta) - \log P(\theta)$

$$\Longrightarrow P(\theta|D) \propto \exp(-U(\theta))$$

We can define the Langevin diffusion, which is a stochastic differential equation:

$$\theta(t) = -\frac{1}{2}\nabla_U \theta(t)\,dt + dB_t \text{ (B is Brownian motion)} \longleftrightarrow \partial_t\theta = \nabla_\theta \log\pi(\theta)/2 + \partial_t B_t$$

$$\theta^{(t+1)} = \theta^{(t)} + \sigma^2\nabla_\theta \log\pi(\theta^{(t)})/2 + \sigma\xi_t$$

(Gradient Descent with Noise)

N(0,I)

SG-LD MCMC: Gradient approximated with a sample of points

$$\hat{\nabla}U(\theta) = \frac{N}{|S|}\sum_{i\in S}\nabla U_i(\theta)$$

*image source

Cyclical SG-MCMC

Cyclical SG-MCMC algorithm works as the SGLD algorithm above, but using varying step sizes. The algorithm starts with a large step size (exploration mode) to quickly move towards an interesting mode

Idea Similar to numerous MCMC chains that start at distant locations

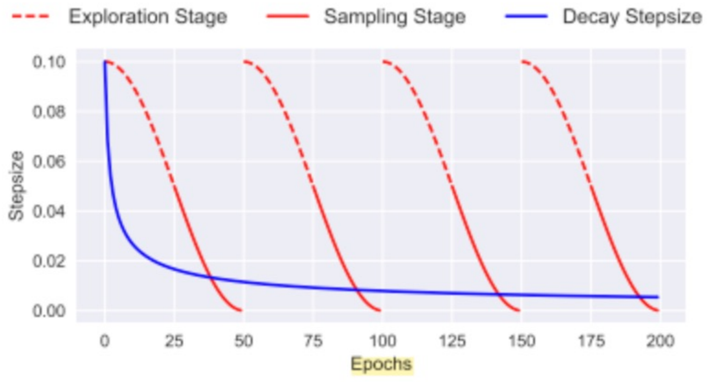


Figure 1: Illustration of the proposed cyclical stepsize schedule (red) and the traditional decreasing stepsize schedule (blue) for SG-MCMC algorithms.

Bayesian Methods: MCMC (3/3)

- HMC is usually the most accurate but takes the longest to run
- Circular versions good for multi modal posterior distributions

MCMC for ResNet-20 on CIFAR-10

Metric	HMC (Reference)	SGMCMC			
		SGLD	SGHMC	SGHMC CLR	SGHMC CLR-Prec
Accuracy	89.64 ±0.25	89.32 ±0.23	89.38 ±0.32	**89.63 ±0.37**	87.46 ±0.21
Agreement	94.01 ±0.25	91.54 ±0.15	91.98 ±0.35	**92.67 ±0.52**	90.96 ±0.24
Total Var	0.074 ±0.003	0.110 ±0.001	0.109 ±0.001	**0.099 ±0.006**	0.111 ±0.002

(a) Target (b) SGLD (c) cSGLD

Figure 2: Sampling from a mixture of 25 Gaussians shown in (a) for the parallel setting. With a budget of $50\text{k} \times 4 = 200\text{k}$ samples, traditional SGLD in (b) has only discovered 4 of the 25 modes, while our cSGLD in (c) has fully explored the distribution.

*image source1 | image source2

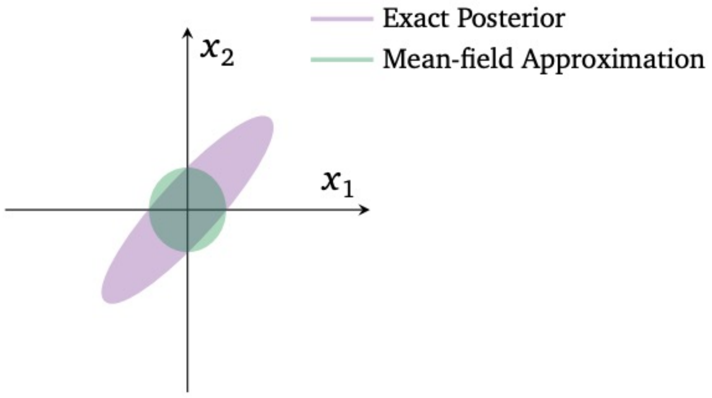
Bayesian Methods: Variational Inference (1/2)

- Approximate Posterior with simpler distributions

$$P(\theta|\mathcal{D}) \sim q(\theta;\phi)$$

Parameters of q

Requires factorization



Over layers:

$$\mathrm{q}\left(\mathrm{W}_1,\ldots,\mathrm{W}_L;\phi\right)=\prod_{l=1}^{L}\mathrm{q}\left(\mathrm{W}_l;\phi_l\right)$$

Over weights ("mean-field"):

$$=\prod_{l=1}^{L}\prod_{d=1}^{D_l}\mathrm{q}\left(\mathrm{w}_{l,d};\phi_{l,d}\right)$$

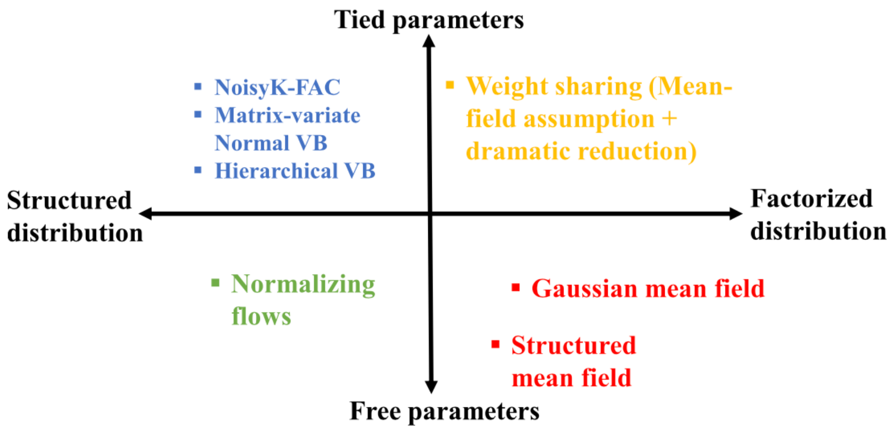


*image 1 source | *image 2 from M Abdar et al (2021) | *theta and w are used interchangeably for NN params*

Bayesian Methods: Variational Inference (2/2)

- VI is set up as a Optimization problem with the goal to find best approximation

$$P(\theta|\mathcal{D}) \sim q(\theta;\phi)$$

$$\phi^* = \operatorname{argmin}_\phi \mathrm{KLD}\left[q\left(w;\phi\right)||p\left(w|y,x\right)\right]$$

$$\mathrm{KLD}\left[q\left(w;\phi\right)||p\left(w|y,x\right)\right] =$$

$$\mathbb{E}_{q_\phi}\left[-\log p\left(y|x,w\right)\right] + \mathrm{KLD}\left[q(w;\phi)||p(w)\right] + \text{const}.$$

Bayes by backprop:
Use this as cost function and take gradients to train NN

$$= \mathbb{E}_\eta\left[-\log p\left(y|x, w = g(\eta;\phi)\right)\right]$$

This term can be reparametrized

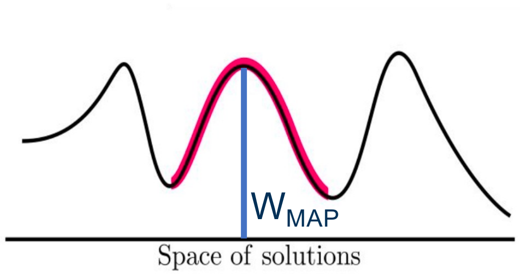
*image source | Bayes by backprop Blundell et al (2015) | *theta and w are used interchangeably for NN params*

Bayesian Methods: Laplace Approximation

- Approximate the posterior with a Normal surrogate centered at the MAP estimate of Posterior
  - Can be used on a pre trained model
  - Hessian can be numerically unstable

Laplace: Fit a quadratic at the mode, using the Hessian or Fisher information

$$p\left(w \mid y, x\right) \approx N\left(\hat{w}_{MAP}, \bar{H}^{-1}(\hat{w}_{MAP})\right)$$



Hessian of Lik at W_{MAP}

$$\hat{w}_{MAP} = \underset{w}{\operatorname{argmax}} P(w|x, y)$$

$$p\left(W_1, \ldots, W_L \mid y, x\right) \propto$$

$$\log p\left(y \mid x, W_1, \ldots, W_L\right) + \sum_{l=1}^{L} \log p(W_l)$$

(Normal Prior = Ridge Regularization)

*image source | *theta and w are used interchangeably for NN params*

UQ for Deep learning: Ensemble Methods

- Bayesian methods are:
  - Slow – We need to run the network ‘N’ times
  - Memory intensive: Store “N” copies of network
  - Calibration: Require proper tuning of Prior beliefs

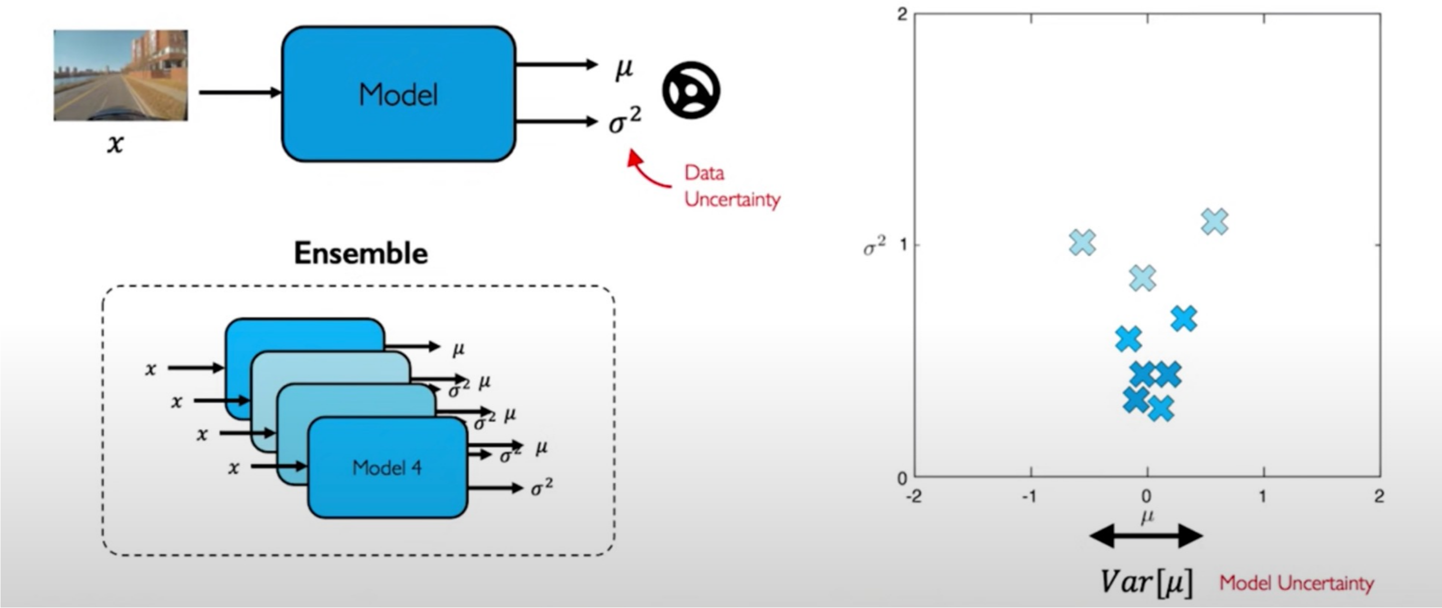
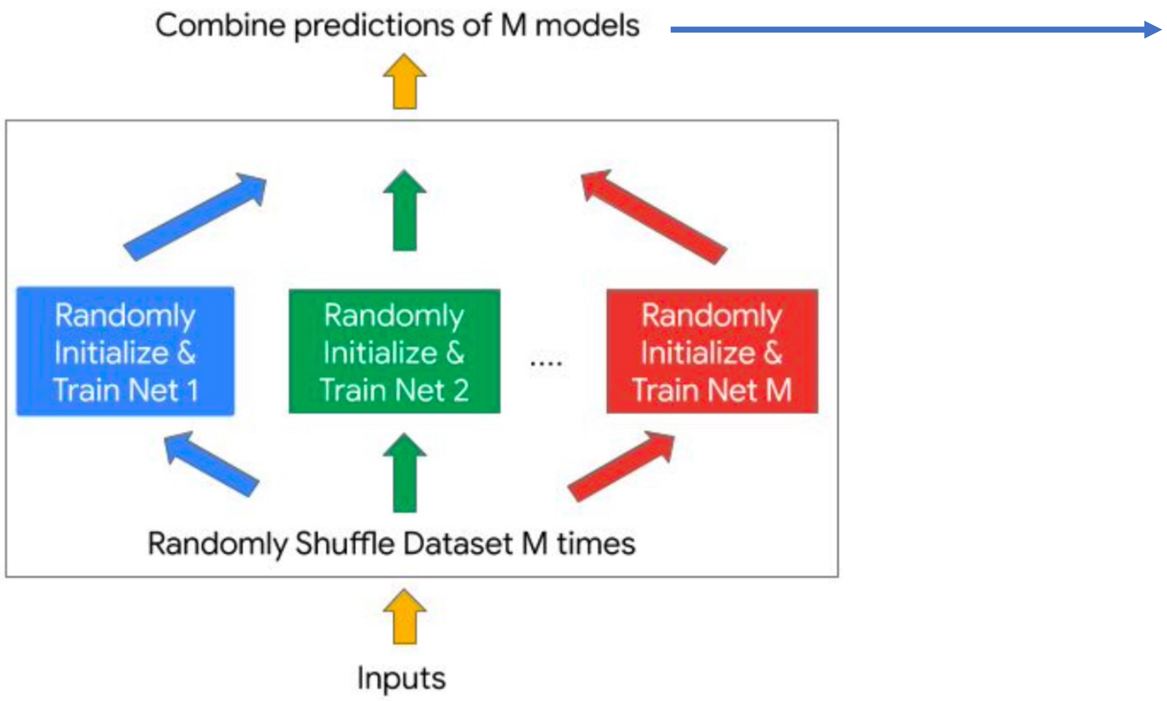


*image source

Ensemble Methods: Deep Ensembles (1/2)

- Re-run the standard SGD training but with different random seeds and average the predictions.

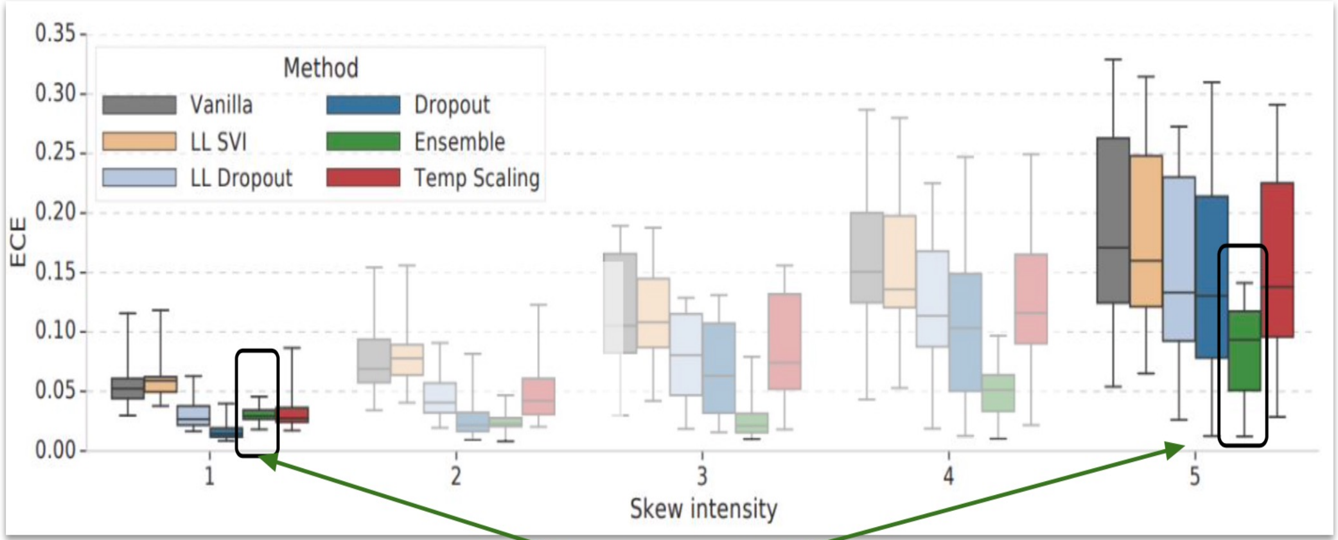


Compute Variance with the predictions

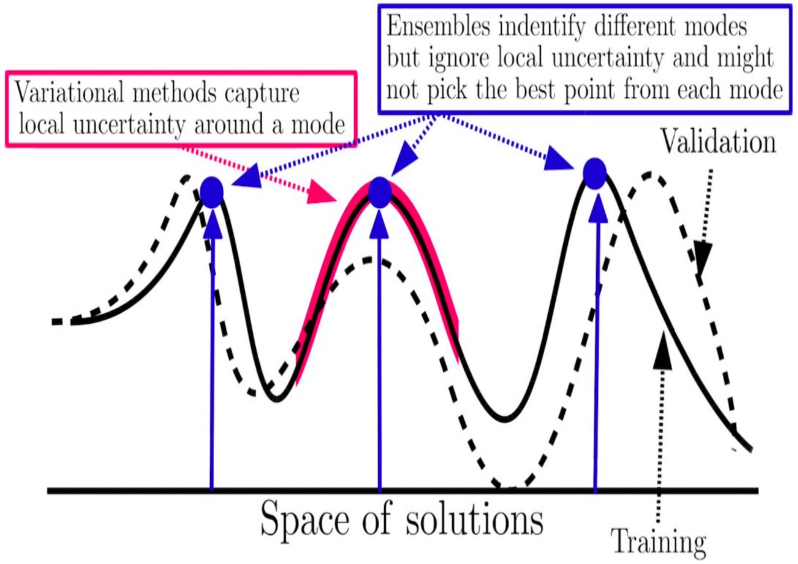
*image source

Ensemble Methods: Deep Ensembles (2/2)

- Deep Ensembles work surprisingly well in practice
  - Like Bayesian methods though they too require time and memory



Deep Ensembles are consistently among the best performing methods, especially under dataset shift



[*image source](#) | [Evaluating Predictive Uncertainty Under Dataset Shift, Ovadia et al](#)

Ensemble Methods: Evidential Deep Learning

Model parameters as generated from higher order evidential distribution

- Deep Evidence Classification
- Deep Evidence Regression

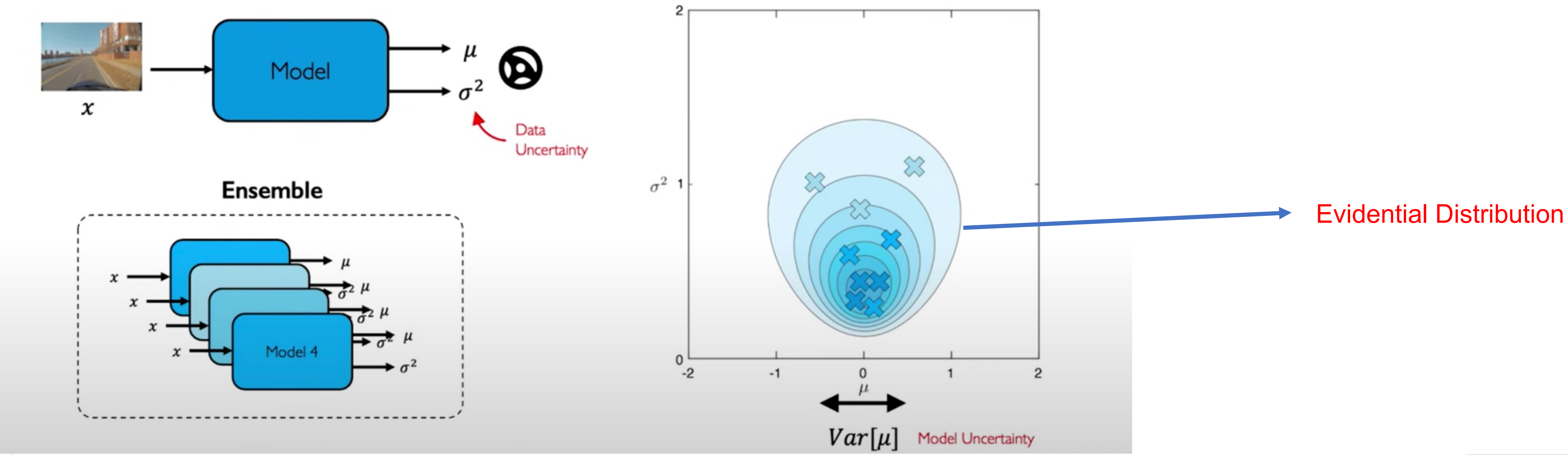


*image source. | Amini et al (NeurIPS 2020)

UQ for Deep Learning: Review

Advantages and disadvantages of UQ methods.

Method		Advantage	Disadvantage
Bayesian	MC	(1) No need to change the model training process, (2) Low training complexity, (3) Easy to implement.	(1) Not very reliable for OoD data, (2) Needs multiple samplings during inference.
	MCMC	(1) Computationally more intensive compared to VI, (2) Asymptotically guarantees of producing exact samples.	(1) Very slow, (2) Fail to find poor convergence, (3) High MC error.
	VI	(1) Very fast (faster than MCMC), (2) Benefiting from stochastic optimization methods, (3) Suited to big datasets.	(1) Heavily depend on the starting point, (2) Very complicated calculations.
Ensemble	DE	(1) Robust prediction, (2) Can be considered as base learners, (3) Limiting the dispensable sensitivity of particular training data, (4) Robust uncertainty estimates.	(1) More resource consuming, (2) Time consuming, (3) Weak performance on smaller problems.

[* image from M Abdar et al (2021))]()

Georgia Tech

UQ for Deep Learning: Gaps and future work

- Relatively limited work for unsupervised methods
- Limited work for UQ for conventional ML models
- Efficient Ensembles
- Deep Gaussian Processes
  - Very wide NN with gaussian priors has a multivariate gaussian joint density

* M Abdar et al (2021))

Georgia Tech

References

- *A review of uncertainty quantification in deep learning: Techniques, applications and challenges, M Abdar et al (2021)*
- *Uncertainty & Robustness in Deep Learning (ICML Balaji et al 2021)*
- *https://www.gatsby.ucl.ac.uk/~balaji/balaji-uncertainty-talk-cifar-dlrl.pdf*
- Weight Uncertainty in Neural Networks, Blundell et al (2015)
- Bayesian NN (Eric Nalisnick)
- BNN with MCMC tutorial
- MIT 6.S191: Evidential Deep Learning and Uncertainty
- MIT 6.S191: Uncertainty in Deep Learning
- MCMC Training of Bayesian Neural Networks
- Monte Carlo Dropout

Georgia Tech