UNCERTAINTY QUANTIFICATION IN SCIENTIFIC MACHINE LEARNING

Introduction to Model Uncertainties in Deep Neural Networks

Advanced School on Foundation Models for Scientific Discovery

The Abdus Salam International Centre for Theoretical Physics, Italy

DARIO COSCIA

SISSA, University of Amsterdam

LECTURE OUTLINE 1/2

- **Introduction to Uncertainty Quantification:**
  - The Need of Uncertainty Quantification in AI4Science
  - Maximum Likelihood Estimation and Bayes' Theorem

- **Modelling Uncertainty in Neural Networks with *Sampling Methods*:**
  - Introduction to Sampling Methods
  - Metropolis-Hasting algorithm, Langevein Dynamic

- **Modelling Uncertainty in Neural Networks with *Ensembles*:**
  - Introduction to Ensembles
  - Deep Ensemble, Snapshot Ensemble, Stochastic Weight Average

- ***Modelling Uncertainty in Neural Networks with Ensembles with* Variational-Inference**
  - Introduction to Variational Inference
  - Bayes-by-Backprop, Monte Carlo Dropout, Variational Dropout

SCIENTIFIC DISCOVERY: FROM EXPERIMENT TO SIMULATION

Galileo Galilei
- inventor of scientific method
15th/16th century

"Scientific method process involves observations, forming a hypotheses, making predictions, ***conducting an experiment*** *and finally analyzing the results"*

nature

Explore content | About the journal | Publish with us | Subscribe

nature > articles > article

Article | Published: 01 February 1975

Computer simulation of protein folding

Michael Levitt & Arieh Warshel

Nature **253**, 694–698 (1975) | Cite this article

7643 Accesses | **791** Citations | **55** Altmetric | Metrics

An Erratum to this article was published on 17 July 1975

Abstract

A new and very simple representation of protein conformations has been used together with energy minimisation and thermalisation to simulate protein folding. Under certain conditions, the method succeeds in 'renaturing' bovine pancreatic trypsin inhibitor from an open-chain conformation into a folded conformation close to that of the native molecule.

SCIENTIFIC DISCOVERY: FROM SIMULATION TO EMULATION

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Open access | Published: 15 July 2021

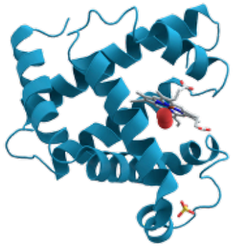
Highly accurate protein structure prediction with AlphaFold

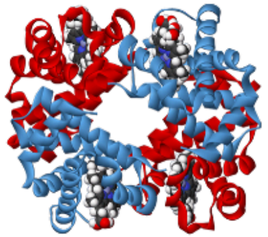
John Jumper ✉, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, … Demis Hassabis ✉ + Show authors

Nature **596**, 583–589 (2021) | Cite this article

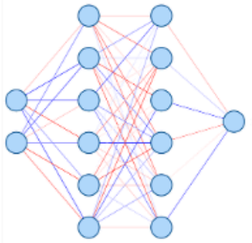
2.13m Accesses | **3977** Altmetric | Metrics

slow simulation

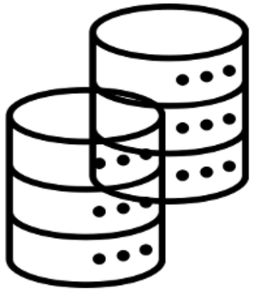




fast

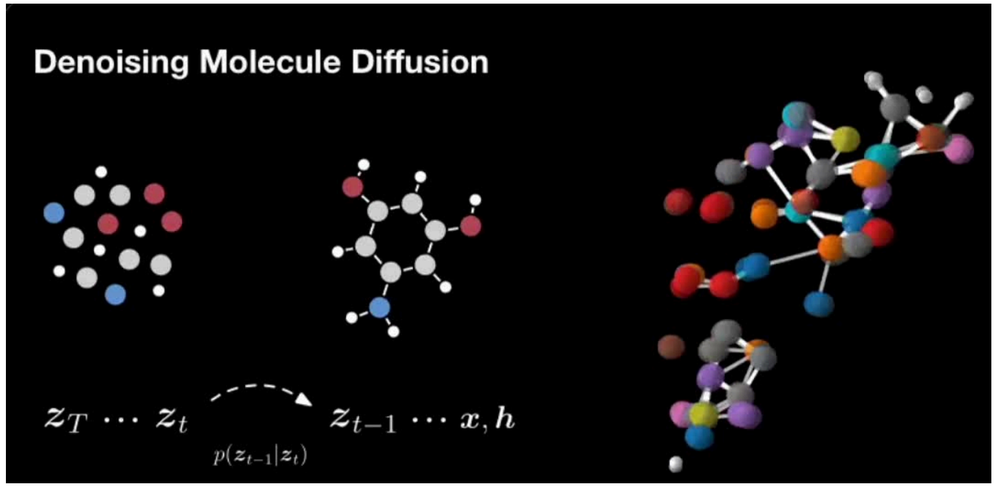


slow

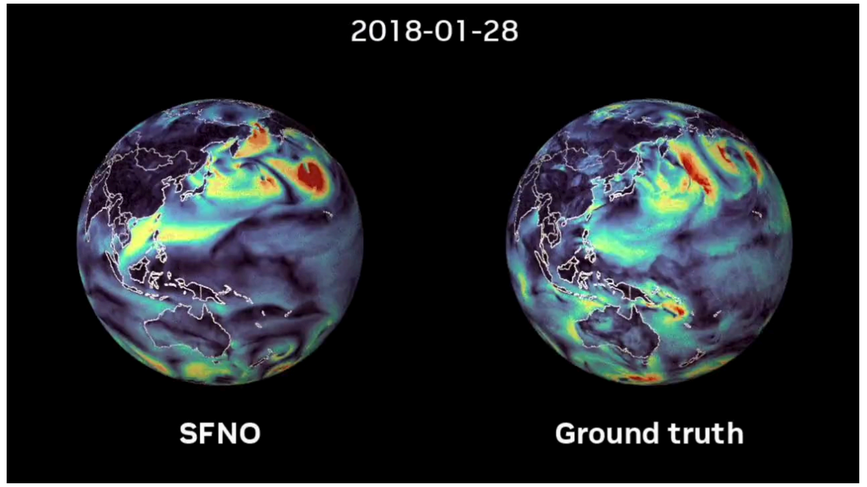


EMULATORS IN AI4SCIENCE

Neural Emulator are reaching ***standard simulator performances*** at much ***lower inference cost*** and ***higher generalizability***

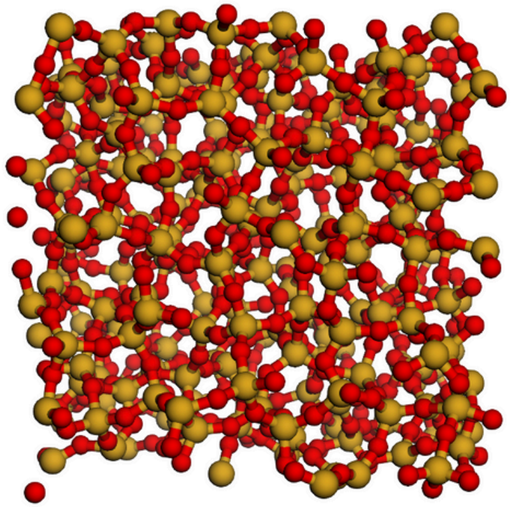


Hoogeboom, Emiel, et al. "Equivariant diffusion for Molecule Generation in 3D". ICML, 2022.



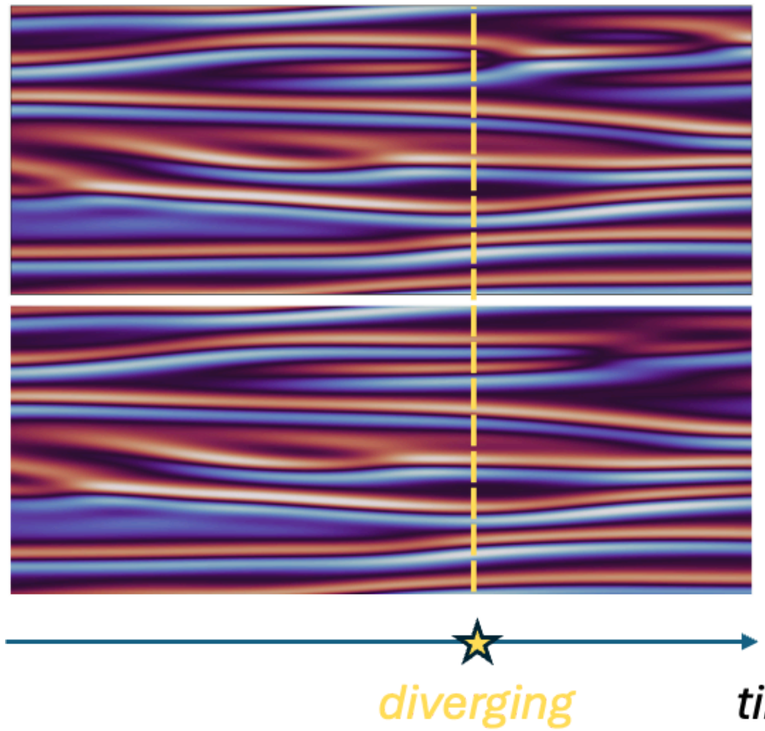
Bonev, Boris, et al. "Spherical fourier neural operators: Learning stable dynamics on the sphere". ICML, 2023

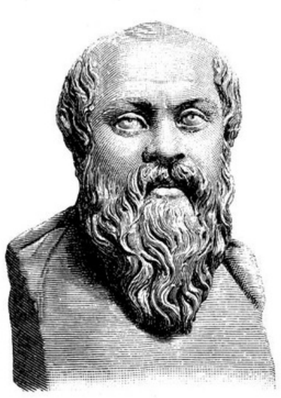
NEURAL EMULATORS CAN MAKE MISTAKES!



DFT Energy: 74.20 eV

MLFF Energy: -233.03 eV

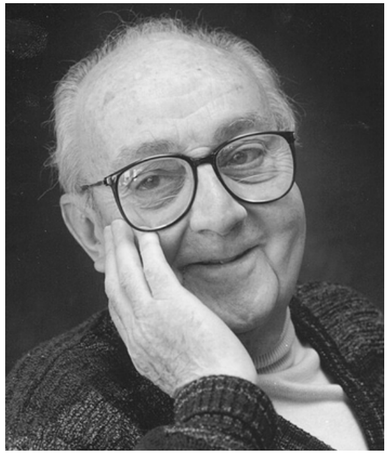




"I know that I know nothing" - Socrates

UNCERTAINTY QUANTIFICATION

Uncertainty Quantification *is the science of* ***identifying***, ***quantifying***, *and* ***reducing*** *uncertainties associated with* ***models***, ***numerical algorithms***, ***experiments*** *and* ***predicted outcomes***.



Essentially, all models are wrong, but some are useful
- George E.P. Box

References

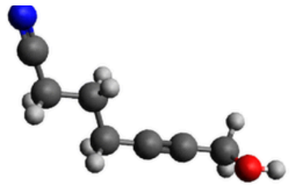
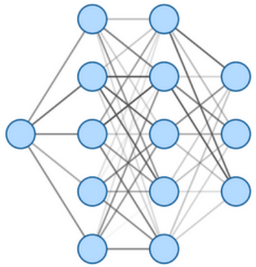
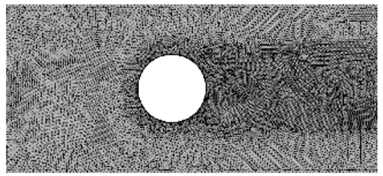
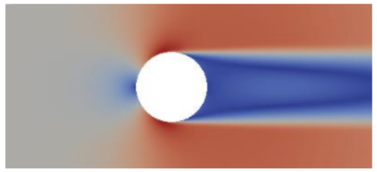
1. *Smith, R. C. (2024). Uncertainty quantification: theory, implementation, and applications. Society for Industrial and Applied Mathematics.*
2. *Psaros, A. F., Meng, X., Zou, Z., Guo, L., & Karniadakis, G. E. (2023). Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. Journal of Computational Physics, 477, 111902.*

PROBABILISTIC SETUP

Setting: We are given a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ of *N* observations

Probabilistic Parametrical Model: $p(y \mid x, \omega)$

Example:

x_i	y_i	ω
	Energy	
		

INFERENCE

Inference: Process of using data $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ to infer properties of the probabilistic model

$$\omega_{\text{best}} = \arg\max_{\omega} \mathcal{L}(\omega)$$

with,

$$\mathcal{L}(\omega) = \log p(\mathcal{D} \mid \omega)$$

Frequentist Approach

$$\omega \sim p(\omega \mid \mathcal{D})$$

with,

$$p(\omega \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \omega)p(w)}{p(\mathcal{D})}$$

Bayesian Approach

BAYESIAN APPROACH: EPISTEMIC AND ALEATORIC UNCERTAINTY

The **predictive distribution** in Bayesian models is computed by marginalising over the model parameters (sometimes called **Bayesian Model Average**)

$$p(y \mid x) = \int p(y \mid x, \omega) p(\omega \mid \mathcal{D}) \mathrm{d}\omega$$

Using the **law of total variance** we can decompose the uncertainty

$$\mathbb{V}\mathrm{ar}[y \mid x] = \mathbb{E}_{\omega \sim p(\omega \mid \mathcal{D})} \left[\mathbb{V}\mathrm{ar}[y \mid x, \omega] \right] + \mathbb{V}\mathrm{ar}_{\omega \sim p(\omega \mid \mathcal{D})} \left[\mathbb{E}[y \mid x, \omega] \right]$$

aleatoric ***epistemic***

WHAT IS THE BIG PROBLEM?

Computing the predictive distribution and the posterior is hard!

$$p(\omega \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \omega)p(w)}{p(\mathcal{D})}, \quad p(\mathcal{D}) = \int p(\mathcal{D} \mid \omega)p(w)\mathrm{d}\omega$$

$$p(y \mid x) = \int p(y \mid x, \omega)p(\omega \mid \mathcal{D})\mathrm{d}\omega$$

In modern Neural Networks the number of parameters goes from 10K to > 100B (i.e. > 180 GB using float16)

POSSIBLE SOLUTION AND METHODS FOR UQ

$$\omega \sim p(\omega \mid \mathcal{D})$$

Sampling Methods

Markov Chain Monte Carlo, Langevin Dynamics, ...

Ensamble Methods

Deep Ensemble, SWAG, Snapshot Ensemble, ...

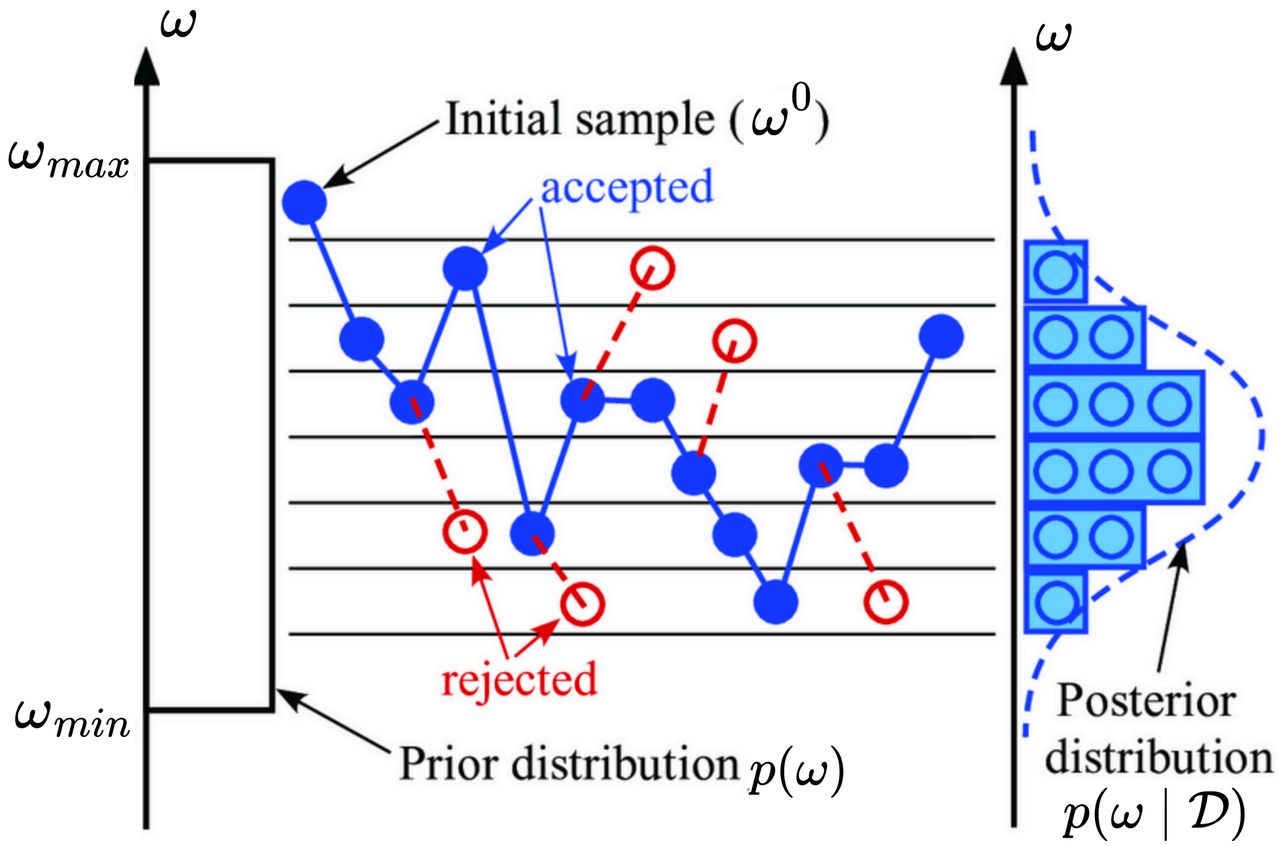
Variational Inference Methods

Bayes-By-Backprop, MC Dropout, Variational Dropout, ...

SAMPLING METHODS FOR BAYESIAN INFERENCE

*Sampling methods aim to find samples of the posterior by **only** knowing likelihood function and prior distribution*

$$p(\omega \mid \mathcal{D}) \propto p(\mathcal{D} \mid \omega)p(w)$$



References

1. *Neal RM. MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo. 2011 Mar 2;2(11):2*
2. *Welling M,TehYW. Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the 28th internationalconference on machine learning (ICML-11) 2011 (pp. 681-688).*

METROPOLIS HASTINGS ALGORITHM

1. Sample the initial state: $\omega = \omega^0 \sim p(\omega)$

2. For multiple monte carlo steps:

   - Propose a new sample: $\omega' \sim \pi(\omega' \mid \omega)$

   - Accept with probability: $\alpha = \min\left(1, \frac{p(\mathcal{D} \mid \omega)\, p(\omega)\, \pi(\omega' \mid \omega)}{p(\mathcal{D} \mid \omega')\, p(\omega')\, \pi(\omega \mid \omega')}\right)$

Very Slow Convergence, Inefficient for large Datasets

Converge to True Posterior, Many Variants (HMC, NUTS, ...)

LANGEVEIN DYNAMICS ALGORITHM

0. Set $U(\omega) = -\log(\omega) - \log(\mathcal{D} \mid \omega)$

1. Sample the initial state: $\omega = \omega^0 \sim p(\omega)$

2. For multiple steps:

   - Update with the Langevein Dynamics: $\omega_{t+1} = \omega_t - \frac{\epsilon}{2}\nabla U(\omega_t) + \eta,\ \eta \sim \mathcal{N}(0, \epsilon)$

or...

   - Stochastic Langevein Dynamics: $\omega_{t+1} = \omega_t - \frac{\epsilon}{2}\frac{N}{B}\nabla U(\omega_t) + \eta,\ \eta \sim \mathcal{N}(0, \epsilon)$

Sensitive to Hyperparameters, Slow Converge compared to HMC

Stochastic version allows some scaling to high dimensions

TAKE AWAYS FROM SAMPLING BASED METHODS

✅ Strengths:

- **Gold standard for Bayesian inference**: Asymptotically exact (under mild assumptions)
- **Flexibility**: Can handle complex, multimodal, or correlated posteriors
- **Theoretically principled**: Apply strictly Bayes theorem to compute posterior samples

⚠️ Limitations:

- **Very slow**: Computationally expensive and rarely scalable to modern deep networks.
- **Difficult to parallelize effectively**
- **Struggles with high-dimensional weight**: Unpractical for modern large-scale networks

📌 **Takeaway**: Powerful and precise, but often impractical for large-scale deep learning.

POSSIBLE SOLUTION AND METHODS FOR UQ

$$\omega \sim p(\omega \mid \mathcal{D})$$

Sampling Methods

Markov Chain Monte Carlo, Langevin Dynamics, ...

✅

Ensamble Methods

Deep Ensemble, SWAG, Snapshot Ensemble, ...

❓

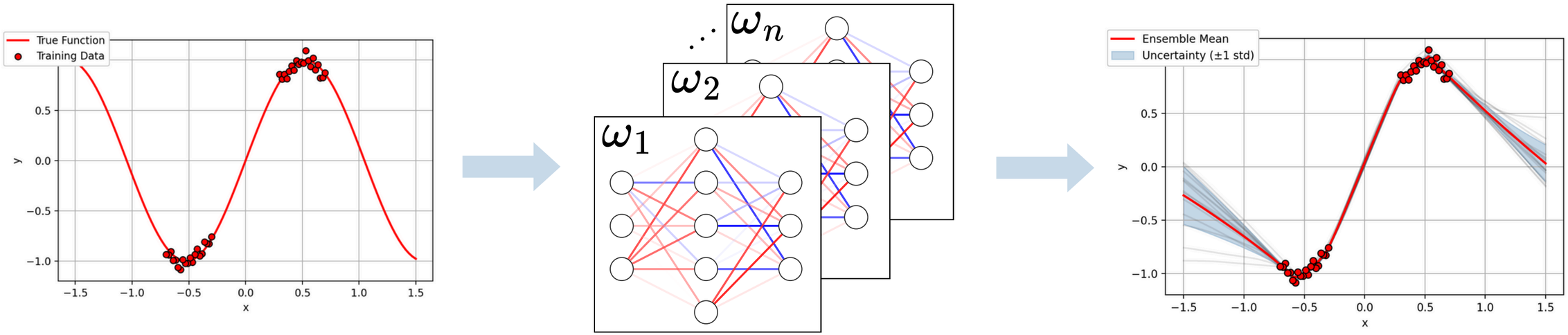
Variational Inference Methods

Bayes-By-Backprop, MC Dropout, Variational Dropout, ...

❓

ENSEMBLE METHODS

Ensembles combine ***multiple Neural Networks***, to produce ***uncertainty estimates*** by capturing variations across models weights



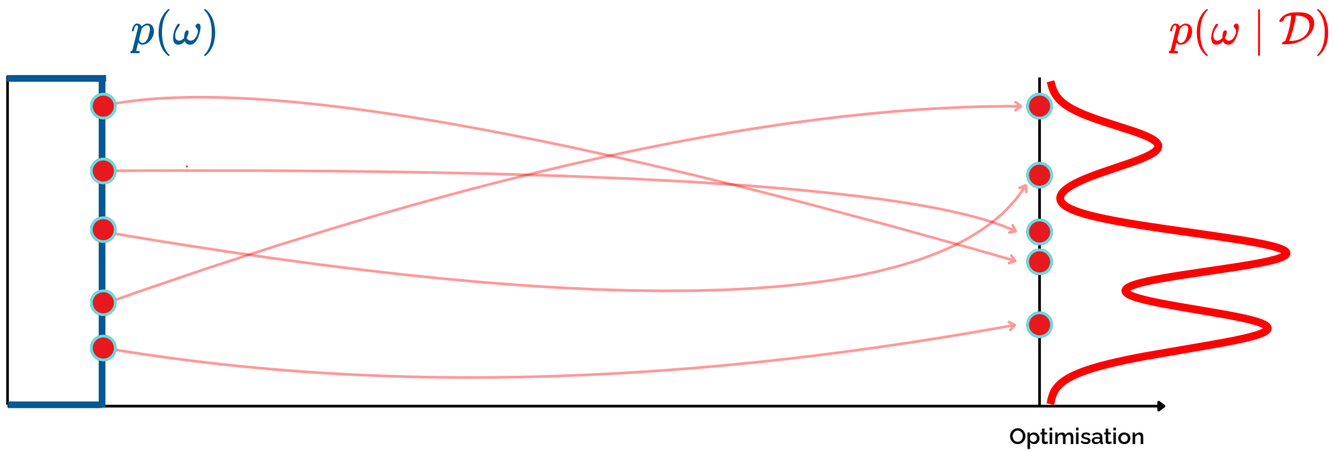
References

1. *Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles.*
2. *Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning.*

BAYESIAN INTERPRETATION OF ENSAMBLES

Bayesian model average assumes that ***one parameter setting is correct***, and ***averages over models due to an inability to distinguish between hypotheses*** given limited information, without enriching the hypothesis space

$$p(\omega \mid \mathcal{D}) \approx p_{\text{ensemble}}(\omega \mid \mathcal{D}) = \sum_{i=1}^{n} \delta(\omega - \omega_i^*)$$



References

1. *Wilson, A. G., & Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization.*
2. *Izmailov, P., Vikram, S., Hoffman, M. D., & Wilson, A. G. G. (2021, July). What are Bayesian neural network posteriors really like?*

WHAT IS A GOOD ENSEMBLE?

Suppose we have *n* models:

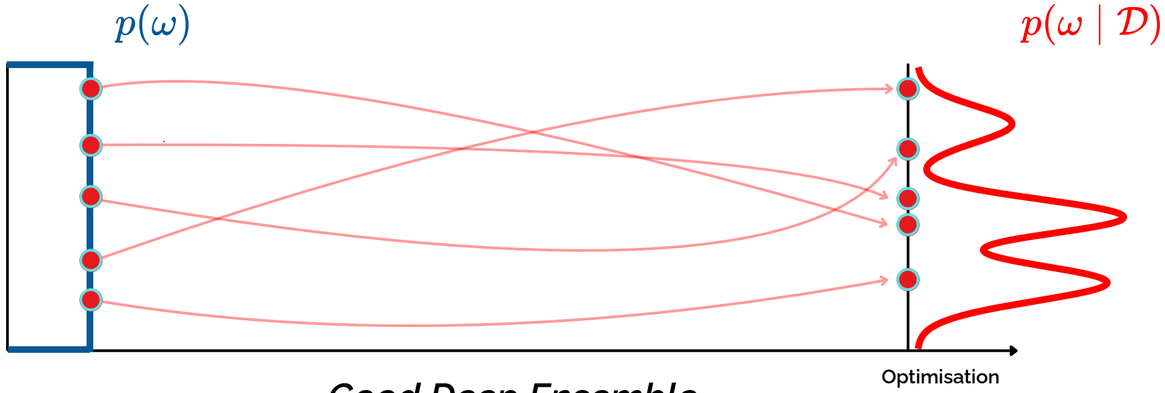
$$\{p(y \mid x, \omega_i^*)\}_{i=1}^n$$

Each model's error ϵ_i (w.r.t. ground truth) has mean zero, and:

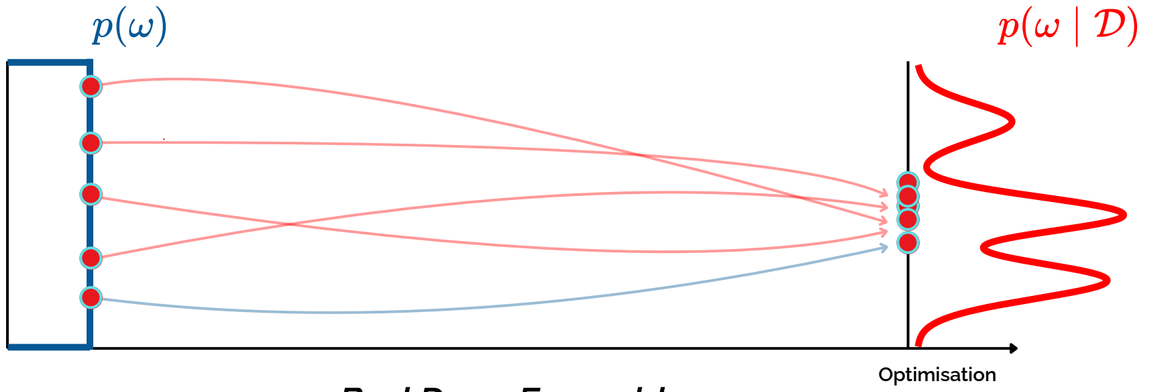
$$\mathbb{E}[\epsilon_i^2] = v, \quad \mathbb{E}[\epsilon_i \epsilon_j] = c$$

Then the ***ensemble error mean and variance*** are:

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right] = 0, \; \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right)^2\right] = \frac{1}{n}(v + (n-1)c)$$



Good Deep Ensemble



Bad Deep Ensemble

References

1. *Wilson, A. G., & Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization.*
2. *Izmailov, P., Vikram, S., Hoffman, M. D., & Wilson, A. G. G. (2021, July). What are Bayesian neural network posteriors really like?*

ENSEMBLE PREDICTIONS AND UNCERTAINTIES

Given *n* ensemble members, the Bayesian Model Average is approximated by:

$$p(y \mid x) = \int p(y \mid x, \omega) p(\omega \mid \mathcal{D}) \mathrm{d}\omega \approx \int p(y \mid x, \omega) \sum_{i=1}^{n} \delta(\omega - \omega_i^*) d\omega$$

Then the ensemble mean prediction is:

$$\mathbb{E}[y \mid x] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[y \mid x, \omega_i^*]$$

And the variance is given by:

$$\mathbb{V}\mathrm{ar}[y \mid x] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{V}\mathrm{ar}[y \mid x, \omega_i^*] + \frac{1}{n-1} \sum_{i=1}^{n} (\mathbb{E}[y \mid x] - \mathbb{E}[y \mid x, \omega_i^*])^2$$

aleatoric uncertainty ***epistemic uncertainty***

DEEP ENSAMBLE

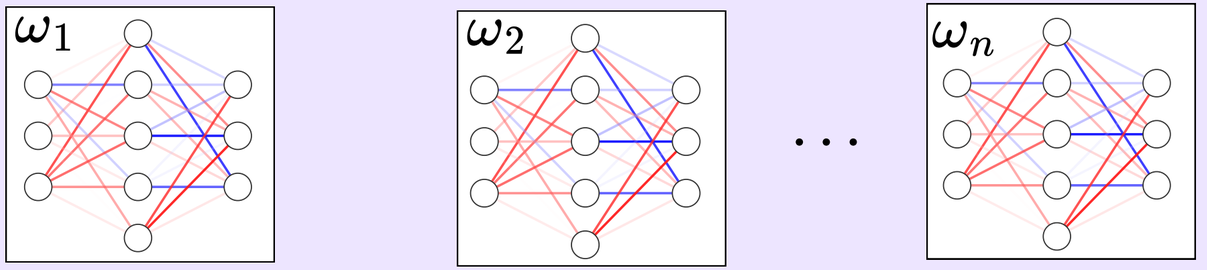
Training

1. **Initialise *n* different Neural Networks** (different initialisation, different architecture, ...)
2. Train **each Network independently**

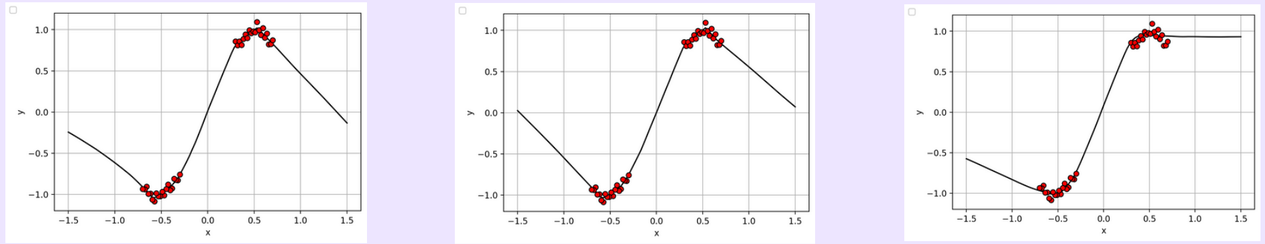
Inference

1. **For each** Network compute its output
2. Use the *n* outputs for computing statistical moments (e.g. mean, var, ...)

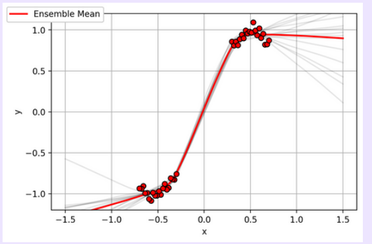
Ensemble Members



Ensemble Predictions



Statistical Estimates



DEEP ENSAMBLE

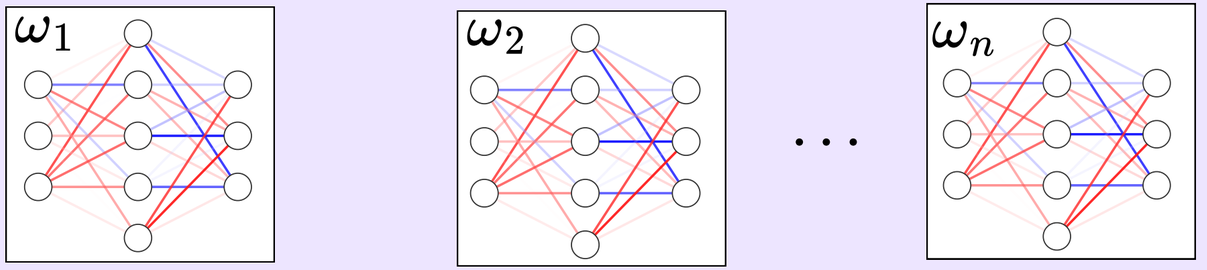
Pros

1. **Simple to implement and highly parallelizable**
2. **Robust to overfitting** and **improvement in average performance**

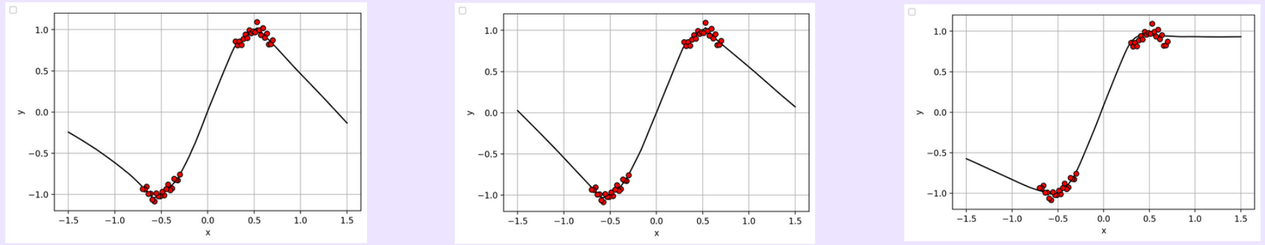
Cons

1. **Uncertainties vary a lot** depending on ensemble construction
2. **Memory heay**

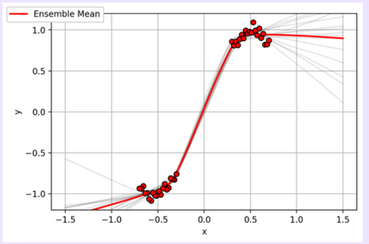
Ensemble Members



Ensemble Predictions



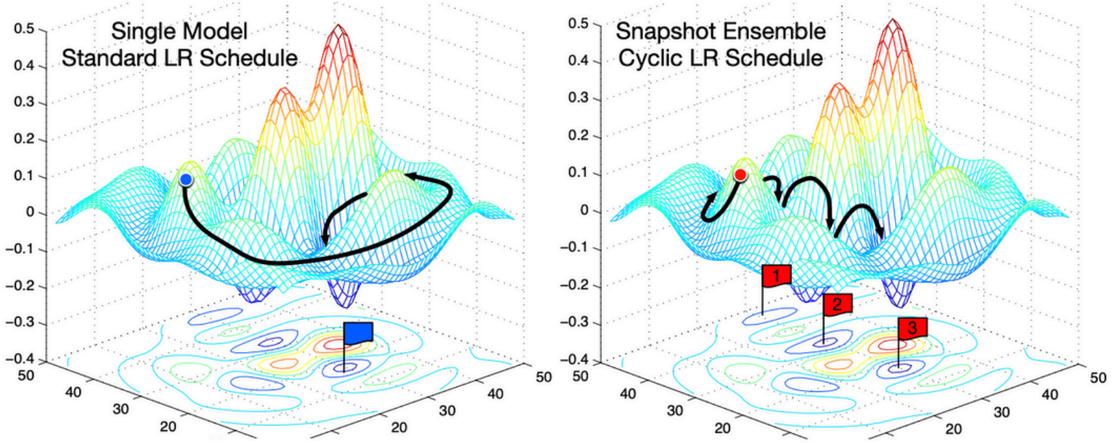
Statistical Estimates



SNAPSHOT ENSEMBLE

$$\alpha(t) = f(\mathrm{mod}(t-1, T/M))$$

Cyclic annealing schedule for T iterations updating the learning rate M times using *f* function (monotonically decreasing)



Model undergoes several learning rate annealing cycles, converging to and escaping from multiple local minima. A weight snapshot (ensemble member) is each minimum for validation loss

References

1. *Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., & Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get M for free.*

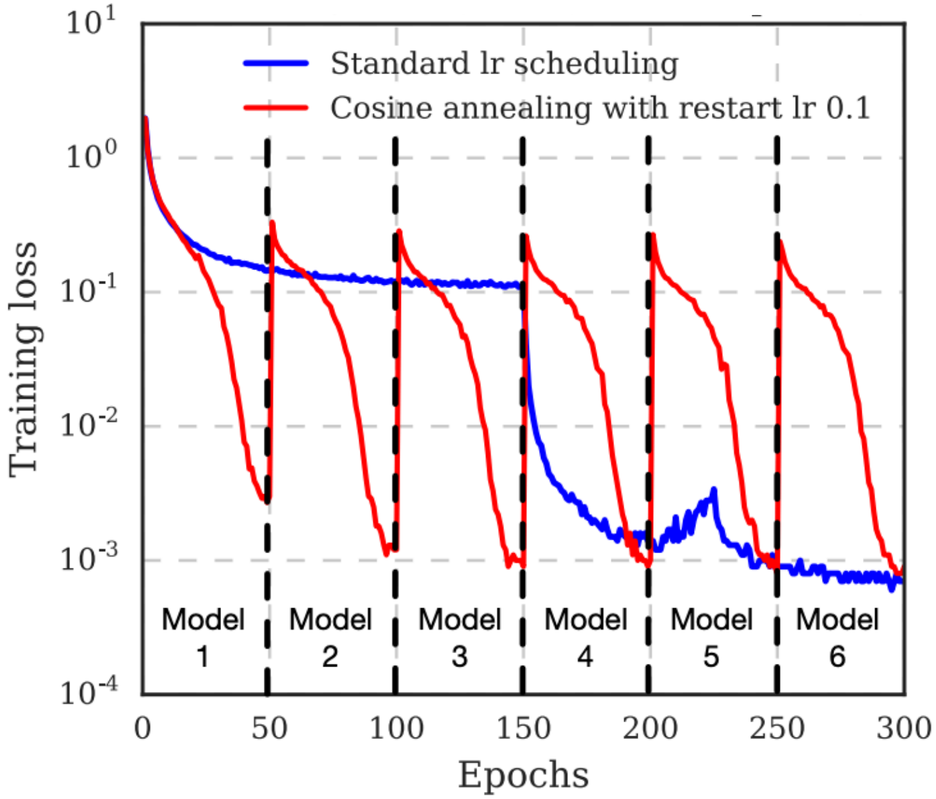
SNAPSHOT ENSEMBLE

Pros

1. **Simple to implement and requires only one Network**
2. **Robust to overfitting** and **improvement in average performance**

Cons

1. Uncertainties vary a lot depending on ensemble construction
2. **Memory heavy**



References

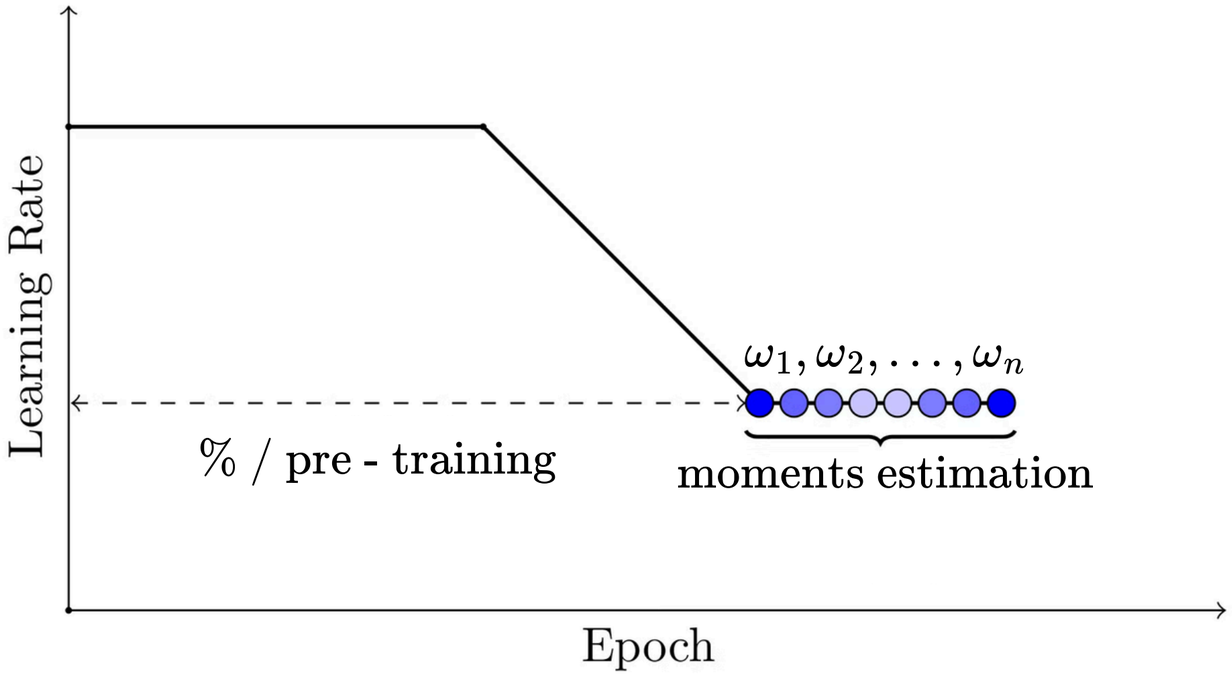
1. *Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., & Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get M for free.*

STOCHASTIC WEIGHTS AVERAGE - GAUSSIAN

SWAG uses moving averages to **approximate the posterior** distribution as a Normal Distribution

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n}\omega_i = \frac{\omega_n + (n-1)\mu_{n-1}}{n}$$

$$s_n^2 = \frac{1}{n}\sum_{i=1}^{n}\omega_i^2 = \frac{\omega_n^2 + (n-1)s_{n-1}^2}{n} \implies \sigma_n^2 = s_n^2 - (\mu_n)^2$$



References

1. *Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning.*

STOCHASTIC WEIGHTS AVERAGE - GAUSSIAN

Pros

1. **Simple to implement** and **requires only one Network**
2. **Bayesian approximation of the posterior, easy sampling**
3. **Robust to overfitting** and **improvement in average performance**
4. **Requires only an additional copy of parameters**

Cons

1. Uncertainties are not always good, and model might be miscalibrated
2. **Gaussian approximation is too restrictive** for multimodal posteriors

References

1. *Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning.*

TAKE AWAYS FROM ENSEMBLE METHODS

✅ Strengths:

- **Model uncertainty through diversity**
- **Simple to implement**
- **Reduce Overfitting**:

⚠️ Limitations:

- **Memory demanding**: Computationally expensive and rarely scalable to large ensembles
- **Not Fully Bayesian**: Heuristic or approximate may lack formal posterior interpretation.
- **Struggles with high-dimensional weight**: Unpractical for modern large-scale networks

📌 **Takeaway**: Practical and strong predictive performance with satisfactory uncertainty, especially when compute is not a bottleneck. Lack theoretical rigor or scalability.

POSSIBLE SOLUTION AND METHODS FOR UQ

$$\omega \sim p(\omega \mid \mathcal{D})$$

Sampling Methods

Markov Chain Monte Carlo, Langevin Dynamics, ...

Ensamble Methods

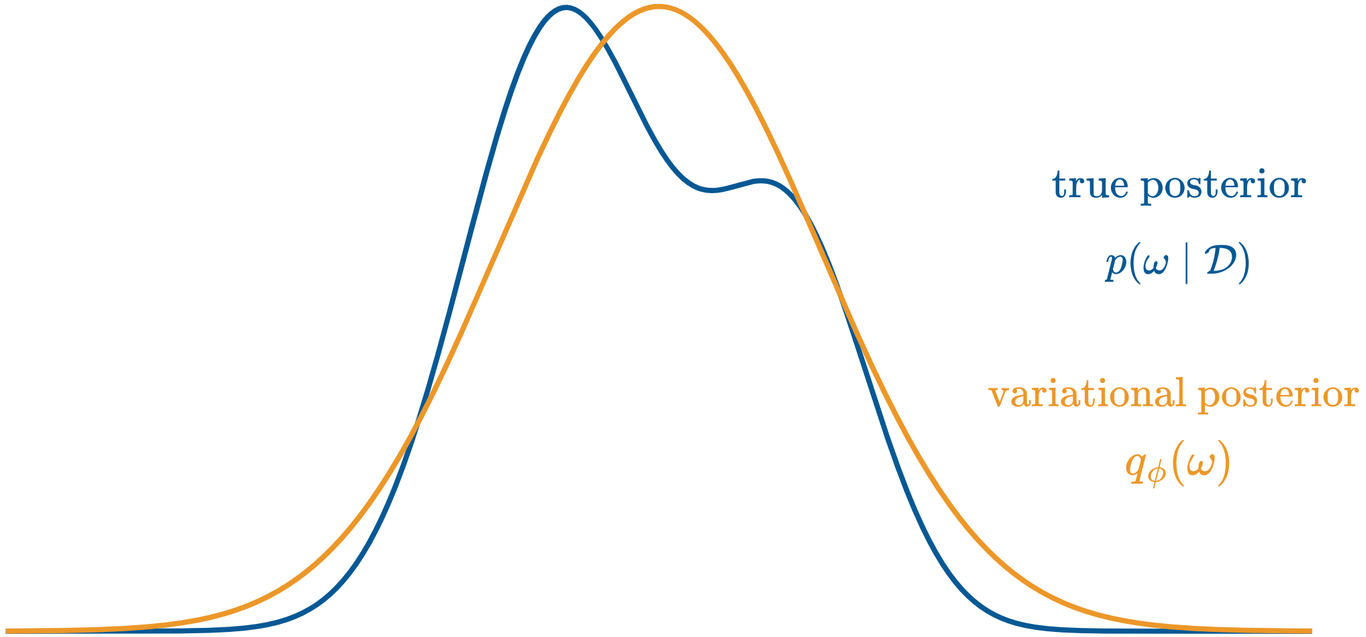
Deep Ensemble, SWAG, Snapshot Ensemble, ...

Variational Inference Methods

Bayes-By-Backprop, MC Dropout, Variational Dropout, ...

VARIATIONAL INFERENCE METHODS

Variational Inference ***optimises the parameters of some parameterized posterior***, e.g. Neural Networks, such that it is a ***close approximation*** of the true ***intractable posterior***



References

1. *Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, June). Weight uncertainty in neural network.*
2. *Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick.*
3. *Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.*

EVIDENCE LOWER BOUND OPTIMIZATION

The KL divergence between ***true posterior*** and **variational posterior** can be written as:

$$D_{\mathrm{KL}}[q_\phi(\omega) \mid p(\omega \mid \mathcal{D})] = \log p(\mathcal{D}) - \mathbb{E}_{q_\phi(\omega)}[\log p(\mathcal{D}, \omega) - \log q_\phi(\omega)] \geq 0$$

The ***variational posterior*** perfectly approximates the **true posterior** when the KL is zero. Since the evidence is a constant, we seek to maximise the **E**vidence **L**ower **BO**und:

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q_\phi(\omega)}[\log p(\mathcal{D}, \omega) - \log q_\phi(\omega)]$$

Which is equivalent to:

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q_\phi(\omega)}[\log p(\mathcal{D} \mid \omega)] - D_{\mathrm{KL}}[q_\phi(\omega) \mid p(\omega)]$$

VARIATIONAL-INFERENCE PREDICTIONS AND UNCERTAINTIES

Given a variational distribution, the Bayesian Model Average is approximated by:

$$p(y \mid x) = \int p(y \mid x, \omega) p(\omega \mid \mathcal{D}) \mathrm{d}\omega \approx \int p(y \mid x, \omega) q(\omega) d\omega$$

Then the ensemble mean prediction is:

$$\mathbb{E}[y \mid x] = \frac{1}{n} \sum_{i=1}^{S} \mathbb{E}[y \mid x, \omega_i], \quad \omega_i \sim q(\omega) \text{ for } i = 1, \ldots, S \text{ samples}$$

And the variance is given by:

$$\mathbb{V}\mathrm{ar}[y \mid x] = \frac{1}{n} \sum_{i=1}^{S} \mathbb{V}\mathrm{ar}[y \mid x, \omega_i] + \frac{1}{n-1} \sum_{i=1}^{S} (\mathbb{E}[y \mid x] - \mathbb{E}[y \mid x, \omega_i])^2$$

aleatoric uncertainty ***epistemic uncertainty***

BAYES BY BACK-PROP

Factorise Gaussian distribution variational posterior over Network layers

$$q(\omega) = \prod_{l \geq 1} q(\omega_l) = \prod_{l \geq 1} \mathcal{N}(\mu_l, \sigma_l^2 \mathbb{I}) \implies \omega_l = \mu_l + \sigma_l \epsilon, \ \epsilon \sim \mathcal{N}(0, \mathbb{I})$$

Scale mixture prior over the Network weights

$$p(\omega) = \prod_{l \geq 1} p(\omega_l) = \prod_{l \geq 1} p\mathcal{N}(0, \sigma_1^2 \mathbb{I}) + (1-p)\mathcal{N}(0, \sigma_2^2 \mathbb{I})$$

Network Optimisation in almost-analytical, and uses (usually) one sample per interation

$$\min_{\mu_1, \ldots, \mu_L, \sigma_1, \ldots, \sigma_L} \frac{1}{N} \sum_{i=1}^{N} \left[-\log p\left(y_i \mid x_i, \omega^{(s)}\right) + \sum_{l=1}^{L} \left[\log q\left(\omega_l^{(s)}\right) - \log p\left(\omega_l^{(s)}\right) \right] \right]$$

with, $\omega_l^{(s)} \sim q(\omega_l) \ \forall l$

References

1. *Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, June). Weight uncertainty in neural network.*

BAYES-BY-BACKPROP

Pros

1. **Principled Bayesian approximation of the posterior, easy sampling**
2. **Robust to overfitting**
3. **Easy to implement**

Cons

1. Requires **hyperparameters tuning**
2. **Doubles the network parameters**
3. **Gaussian posterior might be restrictive for multimodal true posterior**

References

1. *Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, June). Weight uncertainty in neural network.*

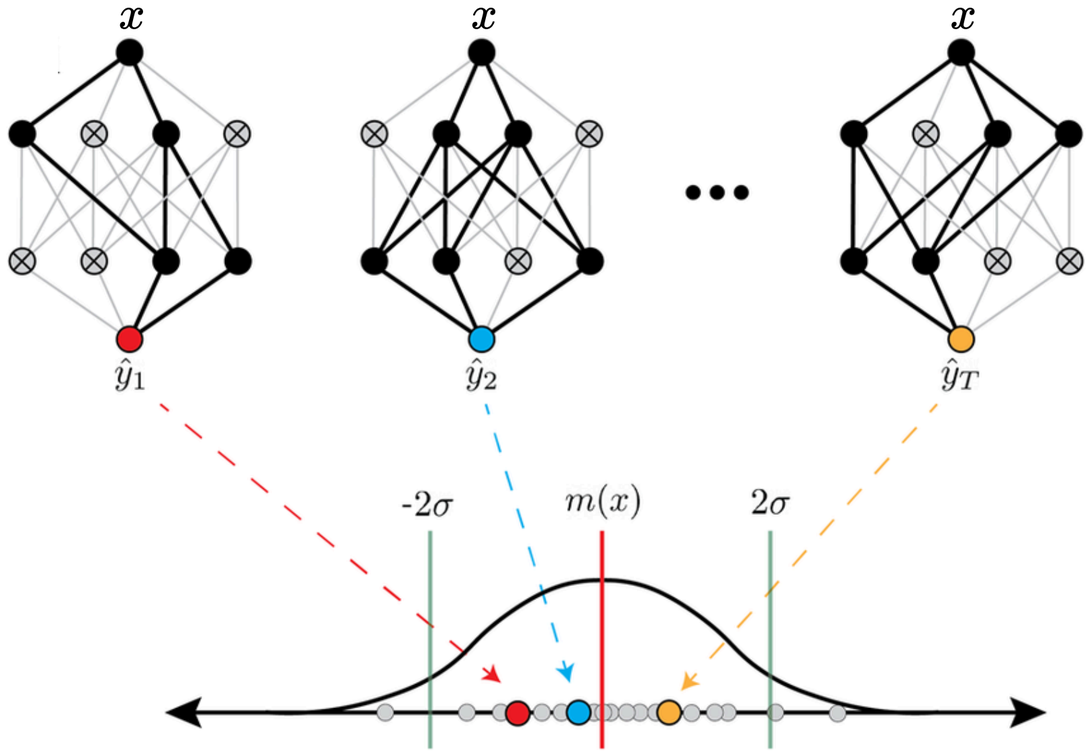
MONTE CARLO DROPOUT

Variational Posterior for Dropout

$$q(\omega) = \prod_{l \geq 1} q(\omega_l), \ \omega_l = \theta_l \odot \eta_l, \ \eta_l \sim \text{Bernulli}(p_l)$$

Simple Isotropic Gaussian prior

$$p(\omega) = \prod_{l \geq 1} p(\omega_l) = \prod_{l \geq 1} \mathcal{N}(0, \sigma^2 \mathbb{I})$$



References

1. *Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.*

MONTE CARLO DROPOUT

Given the prior and the posterior:

$$q(\omega) = \prod_{l \geq 1} q(\omega_l),\ \omega_l = \theta_l \odot \eta_l,\ \eta_l \sim \text{Bernulli}(p_l) \qquad p(\omega) = \prod_{l \geq 1} p(\omega_l) = \prod_{l \geq 1} \mathcal{N}(0, \sigma^2 \mathbb{I})$$

The derivative of the KL wrt. the variational parameter is equivalent to weight L2 regularization (weight decay):

$$\frac{\partial}{\partial \theta_l} D_{\text{KL}}[q(\omega_l) \mid p(\omega_l)] = \frac{(1 - p_l)}{2\sigma^2} \|\theta_l\|^2$$

This is usually the weight decay factor!
We just need to add dropout to our model (before each weight layer) and we have Bayesian Variational Inference for free

References

1. *Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.*

MC DROPOUT

Pros

1. **Principled Bayesian approximation of the posterior, easy sampling**
2. **Robust to overfitting**
3. **Very Very Easy to implement**

Cons

1. Requires **hyperparameters tuning**
2. **Might be miscalibrated (simple posterior)**

References

1. *Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.*

VARIATIONAL DROPOUT

Variational Posterior for Variational Dropout
(**the *p's* are jointly optimized**)

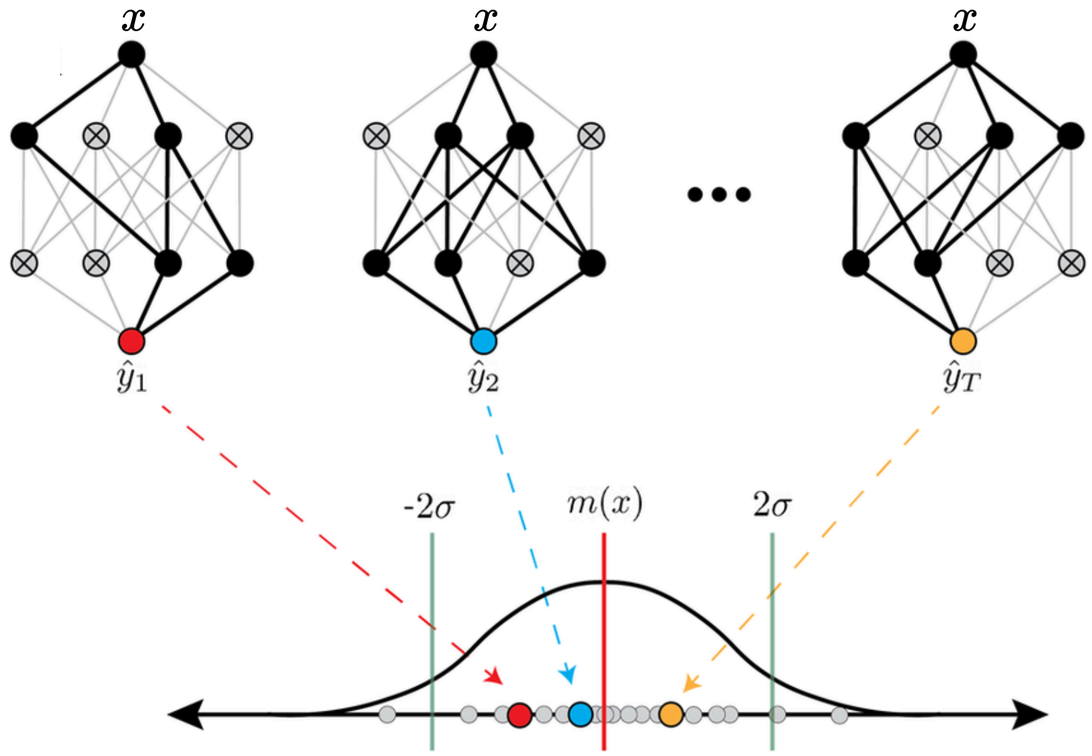
$$q(\omega) = \prod_{l \geq 1} q(\omega_l),\ \omega_l = \theta_l \odot \eta_l,\ \eta_l \sim \mathcal{N}\left(1, \frac{p}{1-p}\right)$$

Improper log-uniform prior over the weights

$$p(\omega) = \prod_{l \geq 1} p(\omega_l) \propto \prod_{l \geq 1} \frac{1}{|\omega_l|}$$

Local reparametrization trick

$$q(\omega) = \mathcal{N}(\mu, \sigma) \implies q(y \mid x) = \mathcal{N}(x\mu, x^T \Sigma x)$$



References

1. *Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick.*

VARIATIONAL DROPOUT OPTIMIZATION

The variational parameter $\theta, \alpha = \frac{p}{1-p}$ are jointly optimised:

$$\mathrm{ELBO}(\theta, \alpha) = \mathbb{E}_{q(\alpha)}[\log p(\mathcal{D} \mid \theta, \alpha)] - D_{\mathrm{KL}}[q \mid p]$$

To be consistent with Gaussian Dropout (i.e. dropout where the mask is sampled from $\mathcal{N}(1, \alpha)$), the KL term **does not need to depend on** θ**.** The KL between posterior and prior is indeed:

$$-D_{\mathrm{KL}}[q \mid p] = \mathrm{const.} + 0.5 \log(\alpha) - \mathbb{E}_{\epsilon \sim N(1,\alpha)}[\log(\epsilon)]$$

In practice the expectation term is analytically approximated via polynomial expansion or ad hoc expression, see [1]

References

1. *Molchanov, D., Ashukha, A., & Vetrov, D. (2017, July). Variational dropout sparsifies deep neural networks*

VARIATIONAL DROPOUT

Pros

1. **Principled Bayesian approximation of the posterior, easy sampling**
2. **Robust to overfitting**
3. **Easy to implement and no hyperparameters**
4. **Gives usually better uncertainty than MC Dropout**

Cons

1. Oversparsification of the Network might ruin uncertainties

References

1. *Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.*

TAKE AWAYS FROM VARIATIONAL METHODS

✅ **Strengths**:

- **Scalable:** Efficient and compatible with modern deep learning architectures
- **Easy Integration:** Minimal code changes
- **Uncertainty Estimates:** Principled uncertainty quantification via approximate posterior

⚠️ **Limitations**:

- **Approximate Inference:** Quality depends on variational family and assumptions
- **Sensitive to Hyperparameters: Performance and uncertainty quality vary widely**

📌 **Takeaway**: Practical and scalable Bayesian approximation, well-suited for large models. However, uncertainty might be sub-optimal due to crude posterior approximations.

POSSIBLE SOLUTION AND METHODS FOR UQ

$$\omega \sim p(\omega \mid \mathcal{D})$$

Sampling Methods

Markov Chain Monte Carlo, Langevin Dynamics, ...

Ensamble Methods

Deep Ensemble, SWAG, Snapshot Ensemble, ...

Variational Inference Methods

Bayes-By-Backprop, MC Dropout, Variational Dropout, ...