Friedrich-Alexander-Universität
Technische Fakultät

# Advanced Deep Learning

## Bayesian Deep Learning

**K. Breininger, V. Christlein**

Artificial Intelligence in Medical Imaging + Pattern Recognition Lab,

Friedrich-Alexander-Universität Erlangen-Nürnberg SoSe 2023

# 1. Introduction

## 2. Bayesian Neural Networks

2.1 Bayes by Backprop

2.2 Monte Carlo Dropout

## 3. Ensemble Methods

# Overconfident Networks

Sources:
https://i2.cdn.turner.com/money/dam/assets/160726155524-tesla-truck-crash-780x439.jpg
https://edition.cnn.com/2015/07/02/tech/google-image-recognition-gorillas-tag/index.html

# Overconfident Networks


Sources:
https://i2.cdn.turner.com/money/dam/assets/160726155524-tesla-truck-crash-780x439.jpg
https://edition.cnn.com/2015/07/02/tech/google-image-recognition-gorillas-tag/index.html

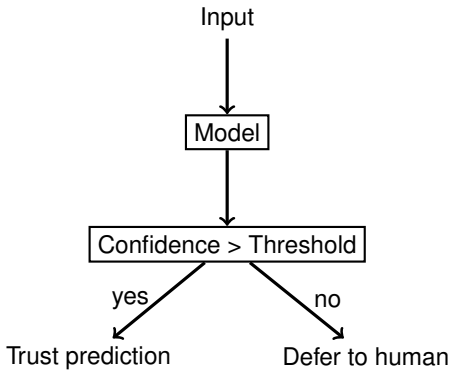
# Why Uncertainty?

## Uncertainty estimation

- Confidence along with prediction
  “the model knows what it doesn’t know”
- Beyond accuracy towards model calibration
- Possible: OOD classification

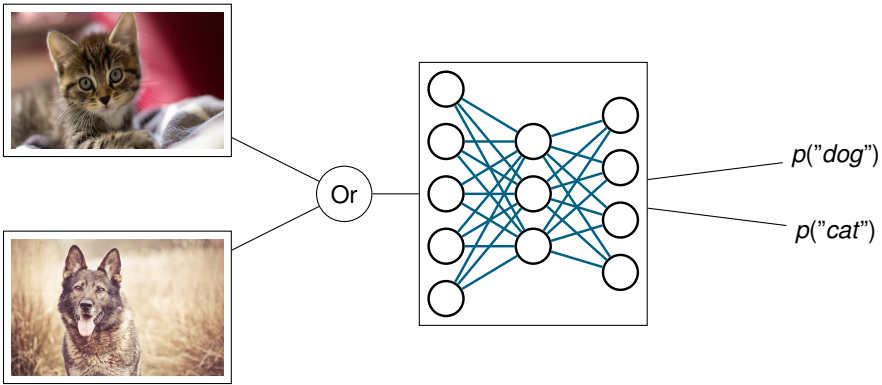
## Applications

- Safety/trustworthy systems
  - Autonomous driving
  - Medical diagnosis
- Active learning, continual learning, reinforcement learning, Bayesian optimization, decision making



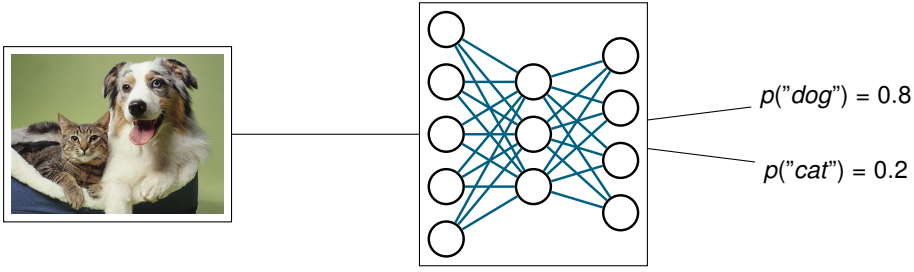

Source: https://sonpeter.github.io/slides/Uncertainty&BDL.pdf

# Likelihood vs. Confidence




Source: Cat: https://getwallpapers.com/wallpaper/full/0/d/4/884248-best-cute-kitten-wallpaper-1920x1080-tablet.jpg
Dog: https://wallup.net/animals-dog-tongues/

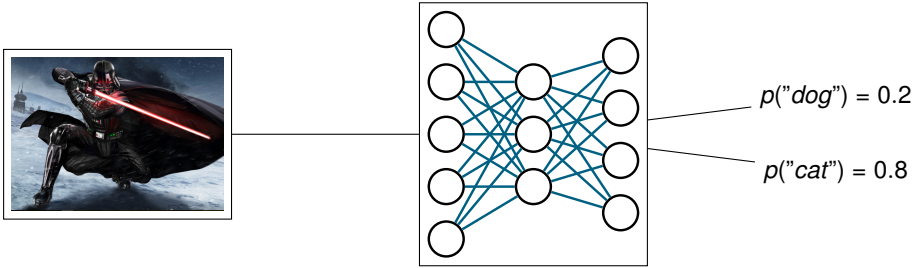
# Likelihood vs. Confidence




Softmax Layer: p("cat") + p("dog") = 1 !

Source: https://www.nydailynews.com/resizer/q9yptCGJ7mpurnOPeudjD3NzFgs=/1200x0/arc-anglerfish-arc2-prod-tronc.s3.amazonaws.com/public/CEYBMESWDXM7UQNGPTU273WXH4.jpg

# Likelihood vs. Confidence




Softmax Layer: p("cat") + p("dog") = 1 !

Source: https://cutewallpaper.org/21/darth-vader-iphone-wallpaper-hd/view-page-21.html

# Empirical uncertainty estimation

Activations: 2.2e-12, 4e-11

Activations: 0.745, 0.435

- Low softmax activation $\rightarrow$ High uncertainty
- Rejection class

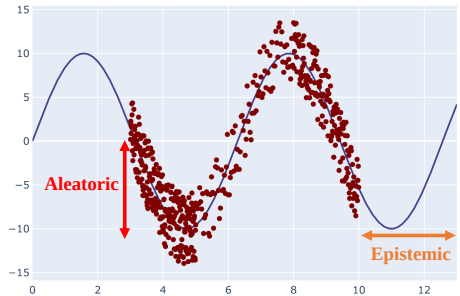
Source:
https://www.nydailynews.com/resizer/q9yptCGJ7mpurnOPeudjD3NzFgs=/1200x0/arc-anglerfish-arc2-prod-tronc.s3.amazonaws.com/public/CEYBMESWDXM7UQNGPTU273WXH4.jpg
https://cutewallpaper.org/21/darth-vader-iphone-wallpaper-hd/view-page-21.html

# Sources of Uncertainty

## Data and Model Uncertainty

### Data uncertainty (aleotoric uncertainty)

- Captures noise inherent in the data
- Caused by inherent noise, ambiguous/missing data, human bias
- **Irreducible** with more data – reducible by better data




Source: Abdar et al. 2021 [2]

# Sources of Uncertainty

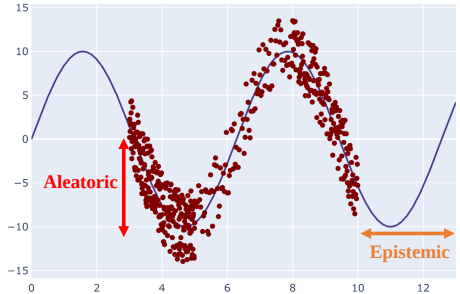
## Data and Model Uncertainty

### Data uncertainty (aleotoric uncertainty)

- Captures noise inherent in the data
- Caused by inherent noise, ambiguous/missing data, human bias
- **Irreducible** with more data – reducible by better data

### Model uncertainty (epistemic uncertainty)

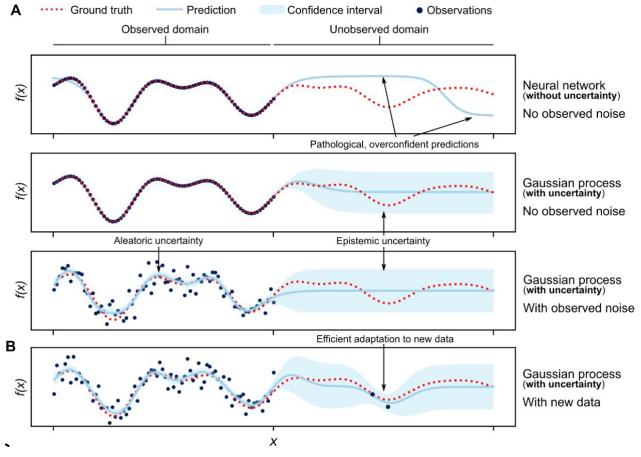
- Prediction confidence
- Captures ignorance which model generated our collected data
- Incurred by lack of training data, imbalanced/sparse data, out-of-distribution data
- **Reducible** with more data




Source: Abdar et al. 2021 [2]

# Sources of Uncertainty

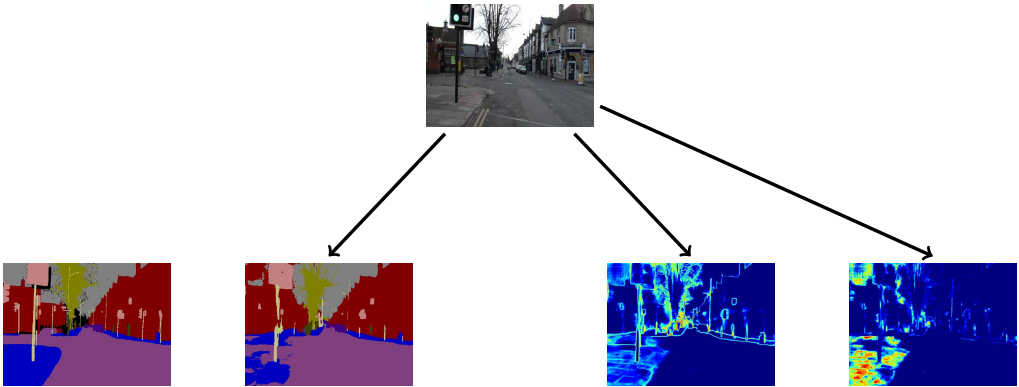
## Data and Model Uncertainty




Source: Hie et al. 2020 [3]

# Sources of Uncertainty

## Data and Model Uncertainty




Source: Kendall and Gal 2017 [1]

# Bayesian Learning

Given: observed dataset $\mathcal{D} = \{x_i, y_i | i = 1, \ldots, N\}$ and model $\mathcal{M}$ with weights $\boldsymbol{\theta}$

## Frequentists

- Maximize likelihood $p(\mathcal{D}|\boldsymbol{\theta})$
- Learning: Find optimal $\boldsymbol{\theta}$ using optimization through differentiation

## Bayesianists

- Instead of point estimation, infer posterior distribution over $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{\overbrace{p(\mathcal{D}|\boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathcal{D})}_{\text{evidence}}} = \frac{p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\mathcal{D}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')\, d\boldsymbol{\theta}'}$$

# Bayesian Learning

Given: observed dataset $\mathcal{D} = \{x_i, y_i | i = 1, \ldots, N\}$ and model $\mathcal{M}$ with weights $\boldsymbol{\theta}$

## Frequentists

- Maximize likelihood $p(\mathcal{D}|\boldsymbol{\theta})$
- Learning: Find optimal $\boldsymbol{\theta}$ using optimization through differentiation

## Bayesianists

- Instead of point estimation, infer posterior distribution over $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{\overbrace{p(\mathcal{D}|\boldsymbol{\theta})}^{\text{likelihood}}\ \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathcal{D})}_{\text{evidence}}} = \frac{p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\mathcal{D}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')\, d\boldsymbol{\theta}'}$$

$p(\mathcal{D}|\boldsymbol{\theta})$ : encodes data uncertainty
$p(\boldsymbol{\theta}|\mathcal{D})$ : encodes model uncertainty

# Bayesian Learning (cont.)

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\mathcal{D}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')\, d\boldsymbol{\theta}'}$$

- Learning: "(Exact) Bayesian inference", finding posterior distribution

☞ For full posterior: marginalize over the whole parameter space → intractable!

→ Main problem of Bayesian learning

- At test time: predictive distribution is approximated via MC sampling:

$$p(y|x, \mathcal{D}) = \int p(y|x, \boldsymbol{\theta}')p(\boldsymbol{\theta}'|\mathcal{D})\, d\boldsymbol{\theta}' = \frac{1}{S}\sum_{s=1}^{S} p\left(y|x, \boldsymbol{\theta}^s\right)$$

# Bayesian Learning

**Goal: posterior distribution $p(\theta|\mathcal{D})$ intractable → approximate inference**

## 1. Gradient-based stochastic approximation

- Energy-based perspective
- Simulate dynamical systems whose stationary distribution is desired target distribution
- True posterior sampling generated via discretizing differential equations describing those dynamics

## Goal: posterior distribution $p(\theta|\mathcal{D})$ intractable → approximate inference

### 1. Gradient-based stochastic approximation

- Energy-based perspective
- Simulate dynamical systems whose stationary distribution is desired target distribution
- True posterior sampling generated via discretizing differential equations describing those dynamics
- Methods:
  - Hamiltonian Monte Carlo (HMC): gold standard
  - Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [4]
  - Stochastic Gradient Langevin Dynamics (SGLD) [5]

\+ high fidelity approximation

– high complexity, many potential biases

**Goal: posterior distribution $p(\theta|\mathcal{D})$ intractable → approximate inference**

## 2. Deterministic approximation: local approximation

- Laplace approximation [6]
- Variational inference: employ parametric variational distribution $q_\phi(\boldsymbol{\theta})$
  → minimize $\mathbb{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}))$
  → equivalent to maximizing variational lower bound (ELBO):

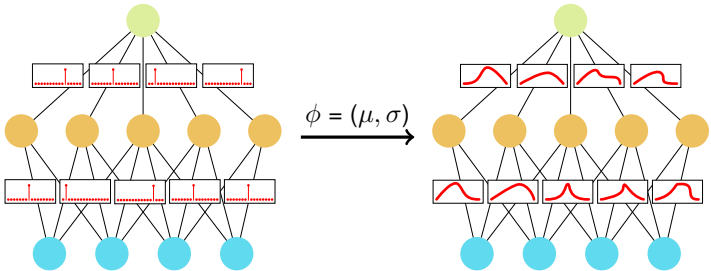
$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\boldsymbol{\theta})} \log p(\mathcal{D}|\boldsymbol{\theta}) - \mathbb{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{\theta})||p(\boldsymbol{\theta}))$$

  - Mean-field VI: $q_\phi(\boldsymbol{\theta})$ is factorized distribution (e.g. diagonal Gaussian), examples: Bayes by backprop
  - Dropout inference: Monte-Carlo Dropout

# Bayes by Backprop

# Bayes by Backprop

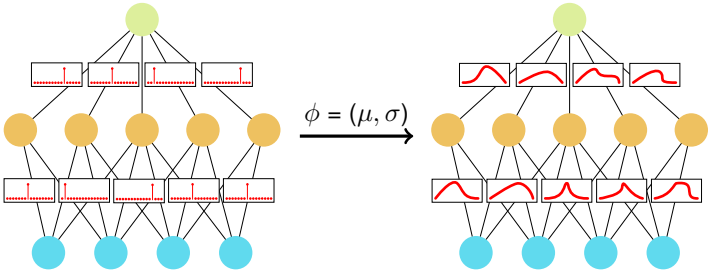
## Weight Uncertainty in Neural Network [7]




Source: Jospin et al. 2022 [8]

# Bayes by Backprop

## Weight Uncertainty in Neural Network [7]



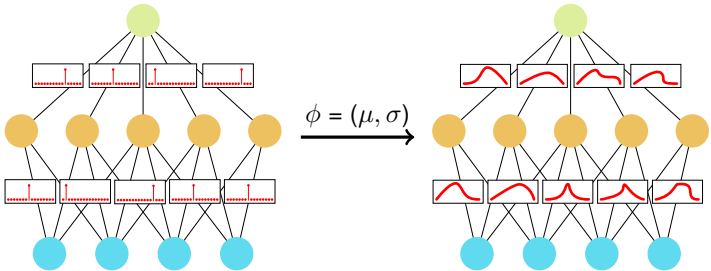

In practice: use re-parametrization trick!

- $w_i = \mu_i + \sigma_i * \epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- NN with 2x #parameters

Source: Jospin et al. 2022 [8]

# Bayes by Backprop

## Weight Uncertainty in Neural Network [7]




In practice: use re-parametrization trick!

- $w_i = \mu_i + \sigma_i * \epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- NN with 2x #parameters

☞ Careful: many people refer to this architecture as “Bayes Neural Network”

- BNN: any stochastic NN trained using Bayesian inference

Source: Jospin et al. 2022 [8]

# Monte Carlo Dropout

# Monte Carlo Dropout for VI

## Dropout as Bernoulli Sampling [14]

- Recap: Dropout:

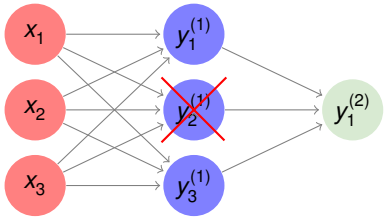
$$z_i \sim \text{Bernoulli}(1-p)$$
$$\hat{\mathbf{W}}_i = \boldsymbol{W}_i \,\text{diag}(\mathbf{z}_i)$$

with:

$p$ : dropout propability
$\mathbf{W}_i$ : weight matrix of layer $i$

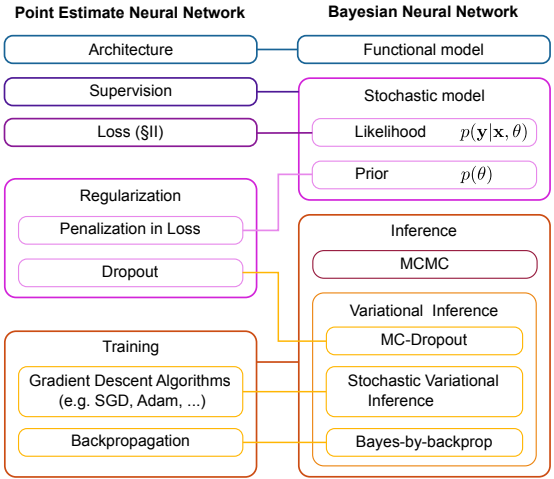
- Apply dropout during Bayesian inference
- Should not be seen as regularization

→ Add other regularization, e.g., $\ell^2$



# Bayesian Neural Networks

## Summary




Source: Jospin et al. 2022 [8]

# A word of caution...

> For the good of science, please stop calling deep ensembles a “non-Bayesian” competitor to standard approximate Bayesian inference procedures.
>
> Wilson and Izmailov, https://cims.nyu.edu/~andrewgw/deepensembles/

[1] is a closer approximation to the Bayesian posterior predictive distribution

# A word of caution...

> For the good of science, please stop calling deep ensembles a “non-Bayesian” competitor to standard approximate Bayesian inference procedures.
>
> Wilson and Izmailov, https://cims.nyu.edu/~andrewgw/deepensembles/

- The posterior for NNs is typically intractable ➔ all methods in Bayesian deep learning provide approximate inference ➔ are “non-Bayesian” in various ways

[1] is a closer approximation to the Bayesian posterior predictive distribution

# A word of caution...

> For the good of science, please stop calling deep ensembles a “non-Bayesian” competitor to standard approximate Bayesian inference procedures.
>
> Wilson and Izmailov, https://cims.nyu.edu/~andrewgw/deepensembles/

- The posterior for NNs is typically intractable ➔ all methods in Bayesian deep learning provide approximate inference ➔ are “non-Bayesian” in various ways
- Bear with me:
  If **method A** is called **“non-Bayesian”**, and **method B “Bayesian”**, but method A behaves more Bayesian[1] than method B, the success of method A cannot serve as evidence that Bayesian methods don’t work very well

[1] is a closer approximation to the Bayesian posterior predictive distribution

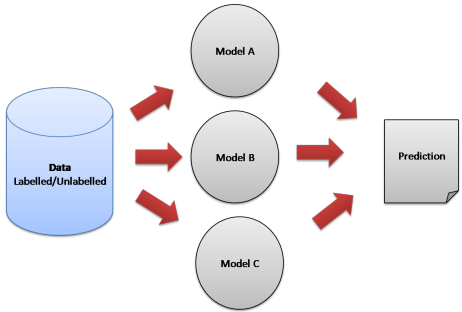
# Approaches for Ensemble-based Bayesian Learning

- Deep Ensembles [9]
- SWAG [10] & MultiSWAG [11]
- Hyperparameter ensembles [13]
- Batch ensembles [12]

Source: Image by rawpixel.com</a> on Freepik

# Deep Ensembles [9]

- Core idea: Train *M* different models with different initializations (same architecture)

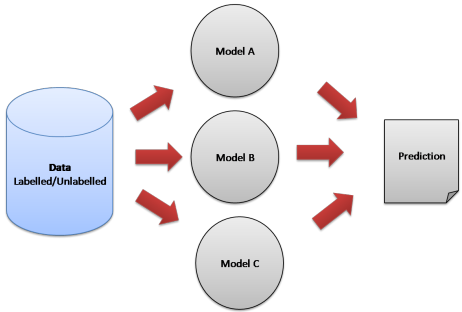



Source: https://hub.packtpub.com/what-is-ensemble-learning/

# Deep Ensembles [9]

- Core idea: Train $M$ different models with different initializations (same architecture)
- Approximate posterior predictive distribution as (uniformly weighted **mixture model**):

$$p(\mathbf{y}|\mathbf{x}, D) \approx \frac{1}{M} \sum_i p(\mathbf{y}|x, \boldsymbol{\theta}_i) \qquad (1)$$



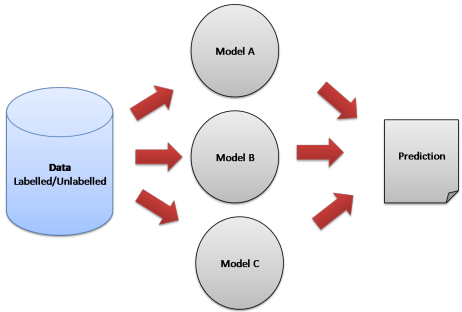

Source: https://hub.packtpub.com/what-is-ensemble-learning/

# Deep Ensembles [9]

- Core idea: Train $M$ different models with different initializations (same architecture)
- Approximate posterior predictive distribution as (uniformly weighted **mixture model**):

$$p(\mathbf{y}|\mathbf{x}, D) \approx \frac{1}{M} \sum_i p(\mathbf{y}|x, \boldsymbol{\theta}_i) \tag{1}$$

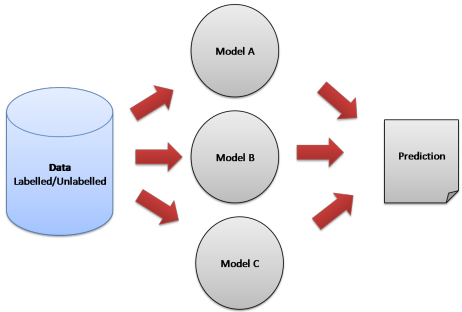
- Loss function should be proper *scoring rule* (e.g., softmax-cross entropy, squared error)




Source: https://hub.packtpub.com/what-is-ensemble-learning/

# Deep Ensembles [9](cont.)

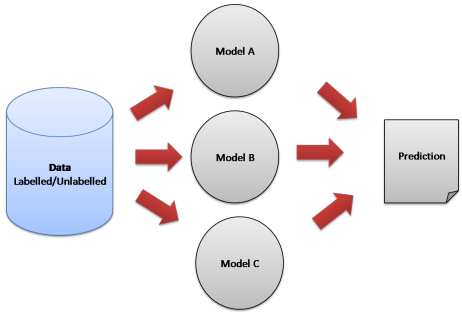
- Additional tricks:
  - Training with adversarial examples [16] to smooth distributions
  - Let network estimate mean and variance, then sample from a Gaussian distribution




Source: https://hub.packtpub.com/what-is-ensemble-learning/

# Deep Ensembles [9](cont.)

- Additional tricks:
  - Training with adversarial examples [16] to smooth distributions
  - Let network estimate mean and variance, then sample from a Gaussian distribution
- For less “random” models, it can make sense to work also with random subsets of the data [9].

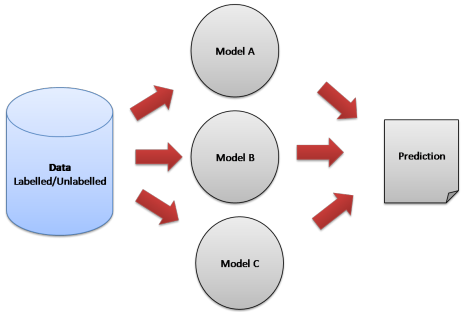



Source: https://hub.packtpub.com/what-is-ensemble-learning/

# Deep Ensembles [9](cont.)

- Additional tricks:
  - Training with adversarial examples [16] to smooth distributions
  - Let network estimate mean and variance, then sample from a Gaussian distribution
- For less “random” models, it can make sense to work also with random subsets of the data [9].

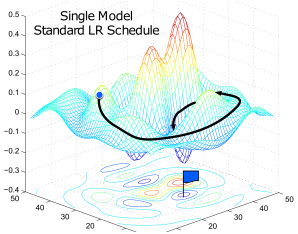
☺ Performance boost compared to MC-dropout
☹ Need to train *M* models




Source: https://hub.packtpub.com/what-is-ensemble-learning/

# Better Self-ensembling

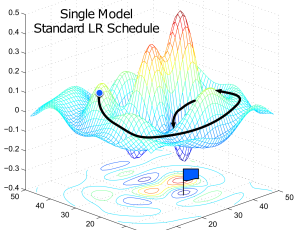
- Previously: Separate models trained
- “Self-ensembling”: Re-use same network
  - MC Dropout
  - Reuse different learning stages: Snapshop, Fast Geometric & Averaged




Source: Huang et al. 2017 [17]

# Better Self-ensembling

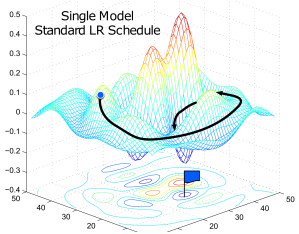
- Previously: Separate models trained
- “Self-ensembling”: Re-use same network
  - MC Dropout
  - Reuse different learning stages: Snapshop, Fast Geometric & Averaged
- Stochastic Weight Averaging (SWA) [17]




Source: Huang et al. 2017 [17]

# Better Self-ensembling

- Previously: Separate models trained
- “Self-ensembling”: Re-use same network
  - MC Dropout
  - Reuse different learning stages: Snapshop, Fast Geometric & Averaged
- Stochastic Weight Averaging (SWA) [17]
  - Uses a cyclic learning rate

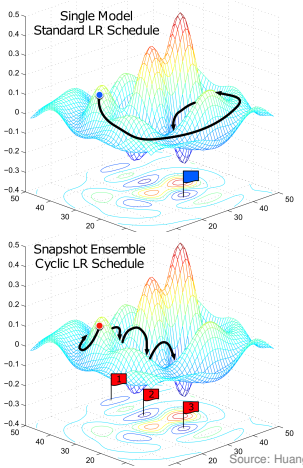



Source: Huang et al. 2017 [17]

# Better Self-ensembling

- Previously: Separate models trained
- “Self-ensembling”: Re-use same network
  - MC Dropout
  - Reuse different learning stages: Snapshop, Fast Geometric & Averaged
- Stochastic Weight Averaging (SWA) [17]
  - Uses a cyclic learning rate
  - Two models:
    **running average** ($\theta_{\text{SWA}}$ and **“normal”** $\theta$)

$$\theta_{SWA} \leftarrow \frac{\theta_{\text{SWA}} \cdot m + \theta}{m + 1} \qquad (2)$$



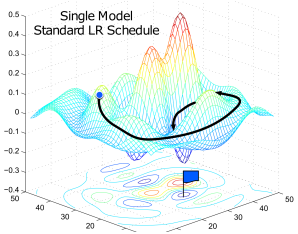

Source: Huang et al. 2017 [17]

# Better Self-ensembling

- Previously: Separate models trained
- “Self-ensembling”: Re-use same network
  - MC Dropout
  - Reuse different learning stages: Snapshop, Fast Geometric & Averaged
- Stochastic Weight Averaging (SWA) [17]
  - Uses a cyclic learning rate
  - Two models:
    **running average** ($\theta_{\text{SWA}}$ and **“normal”** $\theta$)

$$\theta_{SWA} \leftarrow \frac{\theta_{\text{SWA}} \cdot m + \theta}{m + 1} \qquad (2)$$

  - ➔ **Single model as ensemble**




Source: Huang et al. 2017 [17]

# SWAG [10] & MultiSWAG [11]

Let's leverage this for uncertainty:

- Stochastic Weight Averaging with Gaussian distributions (SWAG)

➔ MultiSwag: Combine "true" and SWAG-Ensembling

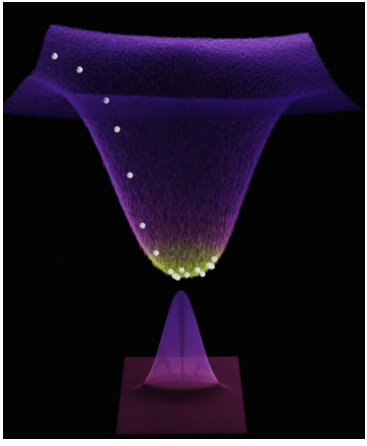
Source: https://www.youtube.com/watch?v=5WOj_ZZJ2wM

# SWAG [10] & MultiSWAG [11]

Let's leverage this for uncertainty:

- Stochastic Weight Averaging with Gaussian distributions (SWAG)
- Idea similar to before:
  Fit Gaussian using SWA solution as mean and low rank (+ diagonal) as covariance

→ MultiSwag: Combine "true" and SWAG-Ensembling



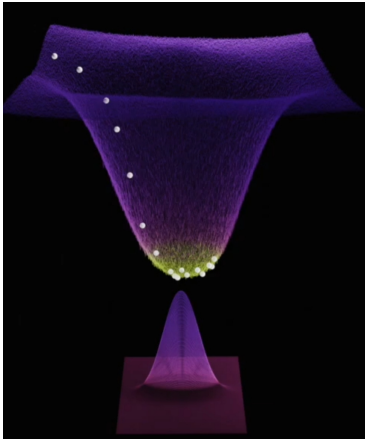
Source: https://www.youtube.com/watch?v=5WOj_ZZJ2wM

# SWAG [10] & MultiSWAG [11]

Let's leverage this for uncertainty:

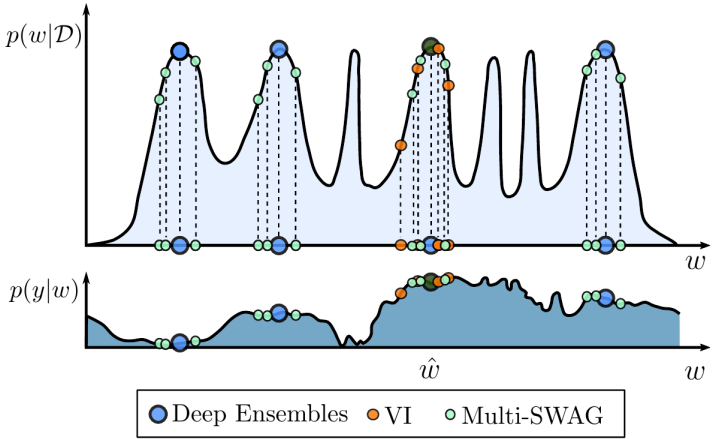
- Stochastic Weight Averaging with Gaussian distributions (SWAG)
- Idea similar to before:
  Fit Gaussian using SWA solution as mean and low rank (+ diagonal) as covariance
- Sample from Gaussian to perform Bayesian Model Averaging

➔ MultiSwag: Combine "true" and SWAG-Ensembling



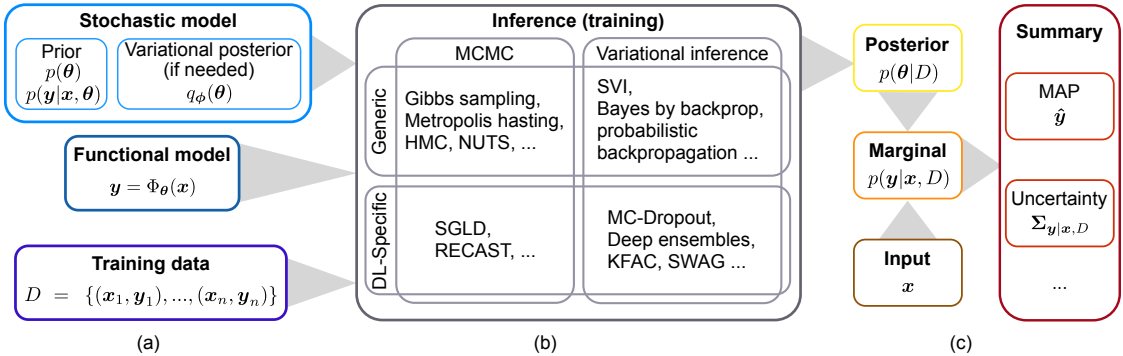
Source: https://www.youtube.com/watch?v=5WOj_ZZJ2wM

# Sampling the Posteriors




Source: Adapted from https://cims.nyu.edu/~andrewgw/deepensembles/

# Summary



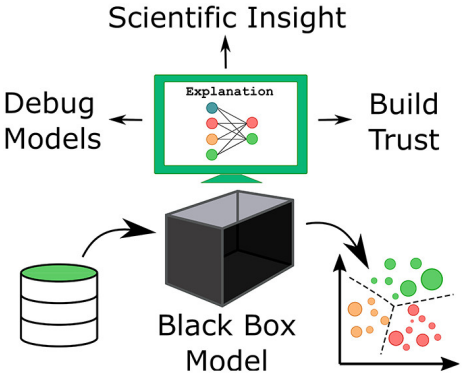

Source: Jospin et al. 2022 [8]

# NEXT TIME

ADVANCED

ON DEEP LEARNING

# Interpretable DL & Causal DL




Source: Oviedo et al. 22 [15]

# Comprehensive Questions

- Which types/sources of uncertainties exist?
- Why can we not use a normal NN for uncertainty classification?
- What is the main problem of Bayesian learning?
- What is the predictive distribution?
- Which approximation methods for the posterior exist?
- How does Bayes by backprop work?
- How does Dropout work?
- How does ensembling work?
- How does SWAG work?

# Further Reading

- Blog about Bayesian inference
  https://www.cs.toronto.edu/~duvenaud/distill_bayes_net/public/
- Another blog
  https://jorisbaan.nl/2021/03/02/introduction-to-bayesian-deep-learning.html
- Argument for Ensembles as Bayesian Approaches:
  https://cims.nyu.edu/~andrewgw/deepensembles/
- Yarin Gal's lecture on Bayesian DLL: https://www.youtube.com/watch?v=G6tUZRHnJYc
- Nice blog post on Stochastic Weight Averaging:
  https://pechyonkin.me/stochastic-weight-averaging/
- Jospin et al. "Hands-on Bayesian Neural Networks – A Tutorial for Deep Learning Users" [8]

# References

# References I


[1] Alex Kendall and Yarin Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[2] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. “A review of uncertainty quantification in deep learning: Techniques, applications and challenges”. In: *Information Fusion* 76 (2021), pp. 243–297.

[3] Brian Hie, Bryan D. Bryson, and Bonnie Berger. “Leveraging Uncertainty in Machine Learning Accelerates Biological Discovery and Design”. In: *Cell Systems* 11.5 (Nov. 2020), 461–477.e9.

[4] Tianqi Chen, Emily Fox, and Carlos Guestrin. “Stochastic Gradient Hamiltonian Monte Carlo”. In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1683–1691.

# References II


[5] Max Welling and Yee Whye Teh. “Bayesian Learning via Stochastic Gradient Langevin Dynamics”. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML’11. Bellevue, Washington, USA: Omnipress, 2011, pp. 681–688.

[6] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. “Laplace Redux - Effortless Bayesian Deep Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20089–20103.

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. “Weight Uncertainty in Neural Network”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1613–1622.

[8] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. “Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users”. In: *IEEE Computational Intelligence Magazine* 17.2 (May 2022), pp. 29–48.

# References III


[9] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. 2017. arXiv: 1612.01474 [stat.ML].

[10] Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. *A Simple Baseline for Bayesian Uncertainty in Deep Learning*. 2019. arXiv: 1902.02476 [cs.LG].

[11] Andrew Gordon Wilson and Pavel Izmailov. "Bayesian Deep Learning and a Probabilistic Perspective of Generalization". In: *CoRR* abs/2002.08791 (2020). arXiv: 2002.08791.

[12] Yeming Wen, Dustin Tran, and Jimmy Ba. "BatchEnsemble: An Alternative Approach to Efficient Ensemble and Lifelong Learning". In: *CoRR* abs/2002.06715 (2020). arXiv: 2002.06715.

[13] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. "Hyperparameter Ensembles for Robustness and Uncertainty Quantification". In: *CoRR* abs/2006.13570 (2020). arXiv: 2006.13570.

# References IV


[14] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059.

[15] Felipe Oviedo, Juan Lavista Ferres, Tonio Buonassisi, and Keith T. Butler. “Interpretable and Explainable Machine Learning for Materials Science and Chemistry”. In: *Accounts of Materials Research* 3.6 (2022), pp. 597–607. eprint: https://doi.org/10.1021/accountsmr.1c00244.

[16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[17] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. “Snapshot Ensembles: Train 1, get M for free”. In: *CoRR* abs/1704.00109 (2017). arXiv: 1704.00109.