Imperial College London

Aalto University
School of Science

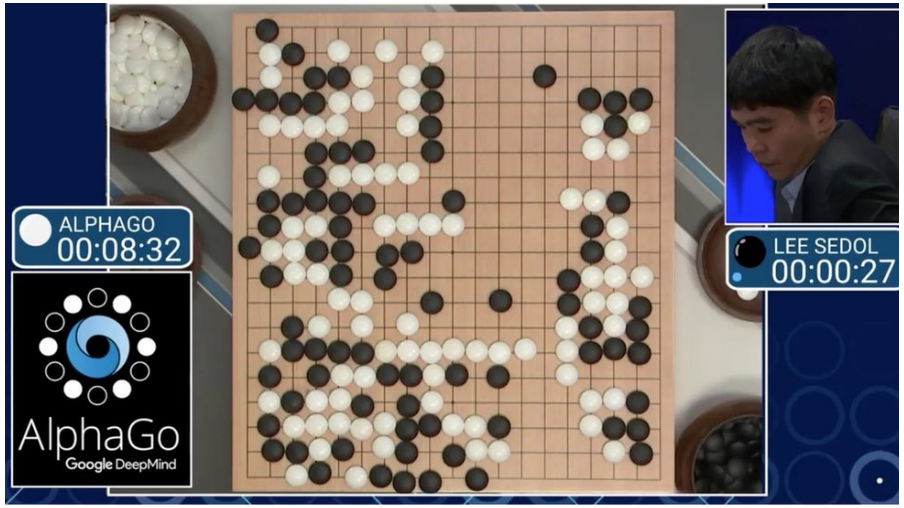
Priors in Bayesian Deep Learning

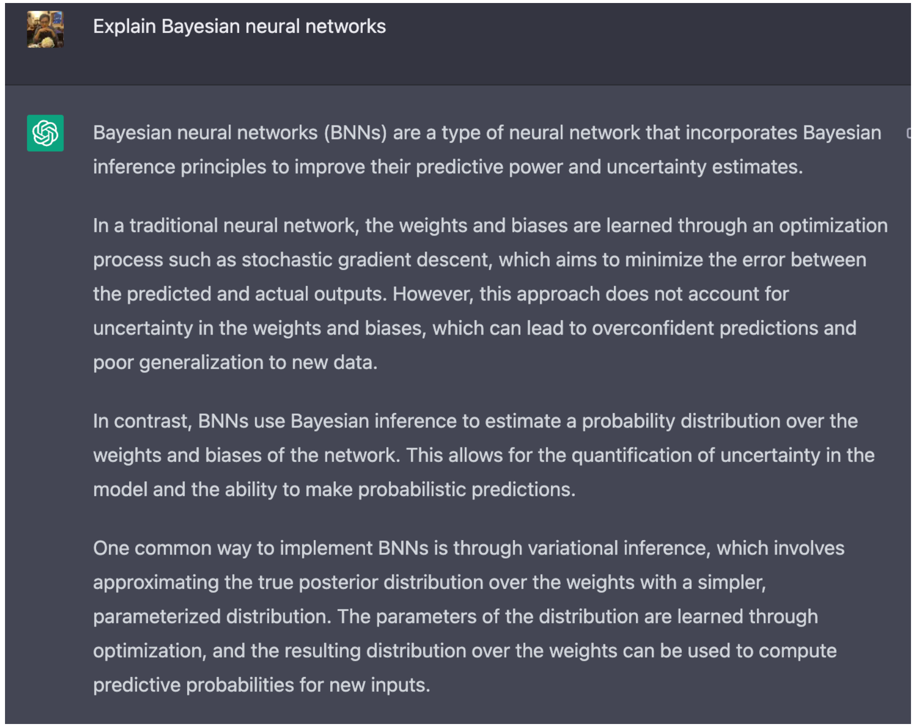
Tianyu Cui

Imperial College London

t.cui23@imperial.ac.uk

Advanced machine learning for Innovative Drug Discovery





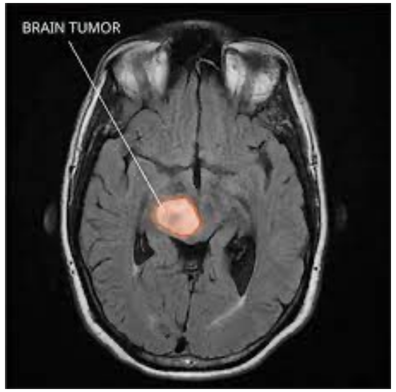
Explain Bayesian neural networks

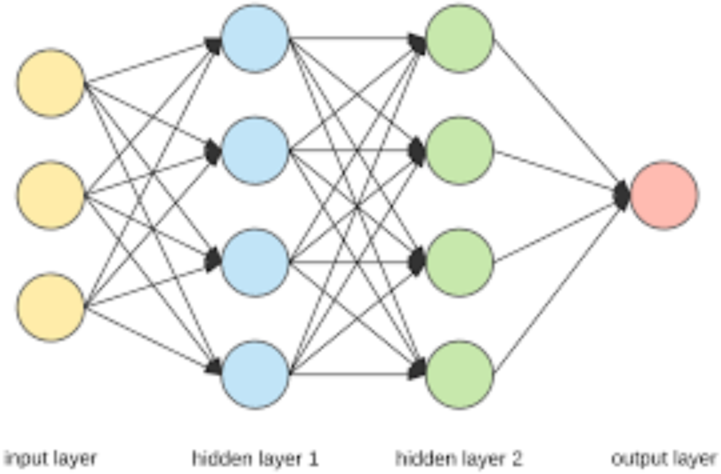
Bayesian neural networks (BNNs) are a type of neural network that incorporates Bayesian inference principles to improve their predictive power and uncertainty estimates.

In a traditional neural network, the weights and biases are learned through an optimization process such as stochastic gradient descent, which aims to minimize the error between the predicted and actual outputs. However, this approach does not account for uncertainty in the weights and biases, which can lead to overconfident predictions and poor generalization to new data.

In contrast, BNNs use Bayesian inference to estimate a probability distribution over the weights and biases of the network. This allows for the quantification of uncertainty in the model and the ability to make probabilistic predictions.

One common way to implement BNNs is through variational inference, which involves approximating the true posterior distribution over the weights with a simpler, parameterized distribution. The parameters of the distribution are learned through optimization, and the resulting distribution over the weights can be used to compute predictive probabilities for new inputs.





Yes, there is a tumor

Is the model sure about that?

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Bayesian inference

Bayesian inference

- Bayes' rule:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} = \frac{P(A)P(B|A)}{\int P(A)P(B|A)\,dA}$$

Bayesian inference

- $P(\theta|data) = \frac{P(\theta)P(data|\theta)}{P(data)} = \frac{P(\theta)P(data|\theta)}{\int P(\theta)P(data|\theta)d\theta}$

- Prior: $P(\theta)$;
- Likelihood of $data$ given θ: $P(data|\theta)$;
- Posterior: $P(\theta|data)$.

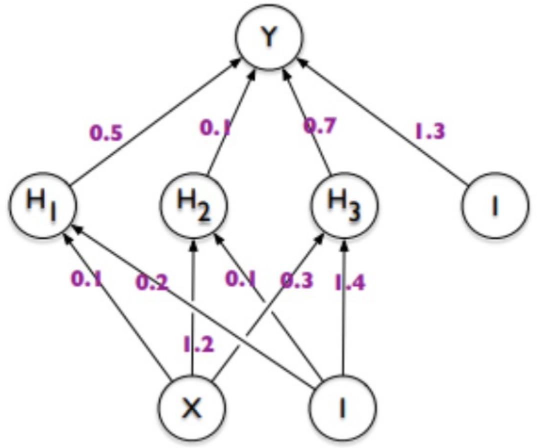
Introduction to BNNs

Bayesian deep learning

Bayesian deep learning: definition

- Deep neural network: $f(x; \theta)$;
- Likelihood:

$$p(\mathcal{D}|\theta) = \prod_i p(y_i|f(x_i; \theta)), \text{ with } \mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$$



- Standard neural network:
  - Maximize likelihood:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_i \log p(y_i|f(x_i; \theta))$$

  - Prediction on x^*:

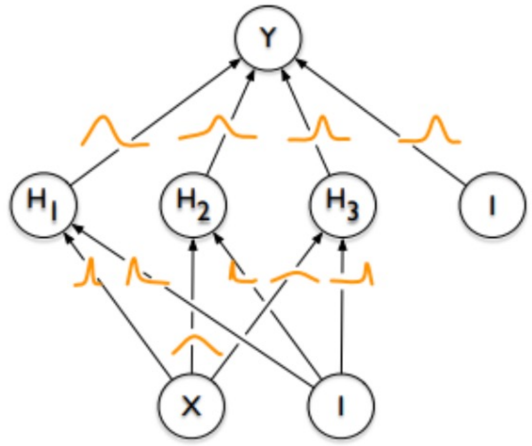
$$p(y^*|f(x^*; \theta^*))$$

Blundell, Charles, et al. "Weight uncertainty in neural network." *ICML*, 2015.

Bayesian deep learning: definition

- Deep neural network: $f(x;\theta)$;
- Likelihood:

$$p(\mathcal{D}|\theta) = \prod_i p(y_i|f(x_i;\theta)), \text{ with } \mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$$



- Bayesian neural network:
  - Bayesian inference:

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

Intractable!

  - Prediction on x^* (Bayesian model averaging):

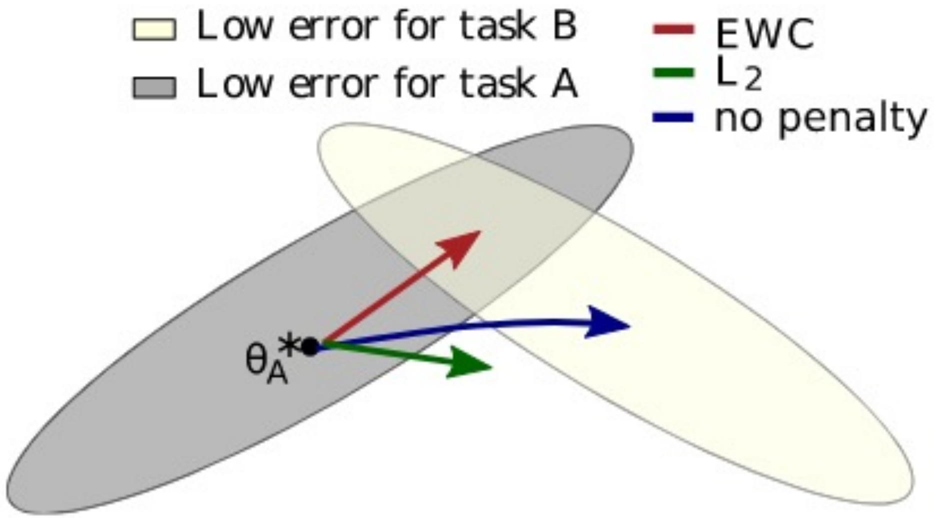
$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|f(x^*;\theta))p(\theta|\mathcal{D})\, d\theta$$

Intractable!

Blundell, Charles, et al. "Weight uncertainty in neural network." *ICML*, 2015.

Bayesian deep learning: application

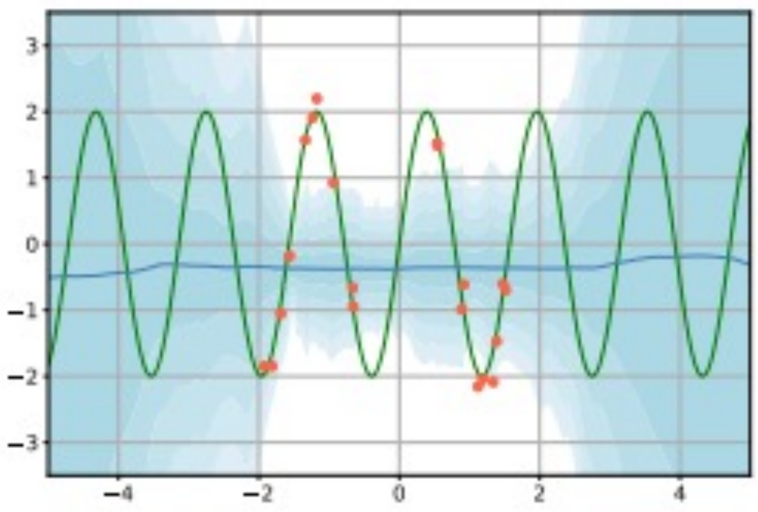
- Incorporate prior knowledge
  - Overcoming catastrophic forgetting in transfer/continual learning.

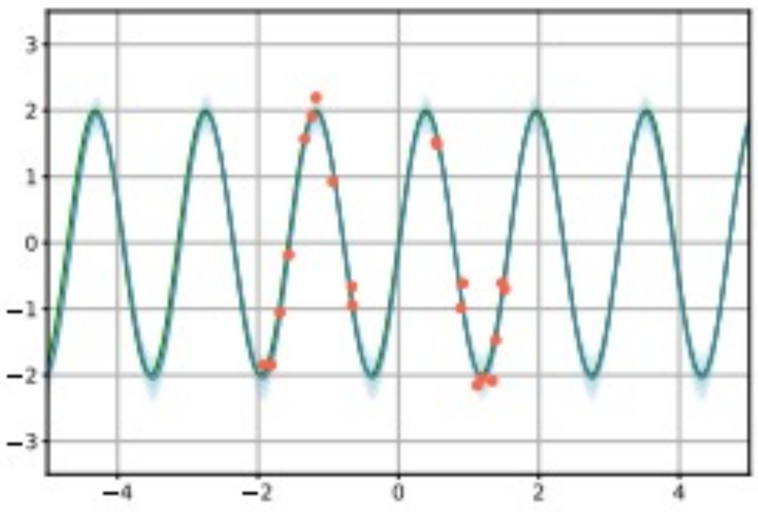


Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.

Bayesian deep learning: application

- Incorporate prior knowledge
  - Improve out-of-distribution generalization.

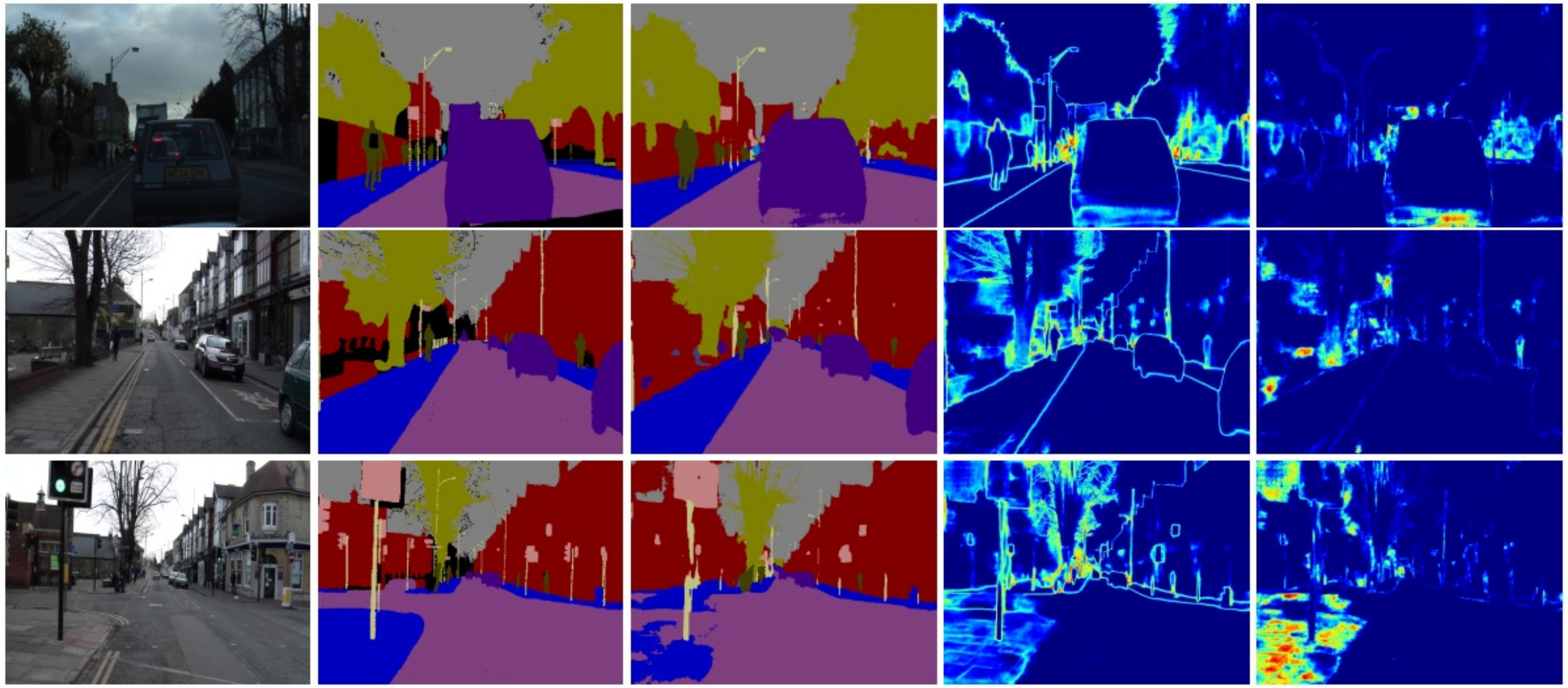




Sun, Shengyang, et al. "Functional variational Bayesian neural networks." *ICLR*, 2019.

Bayesian deep learning: application

- Provide predictive uncertainty



(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

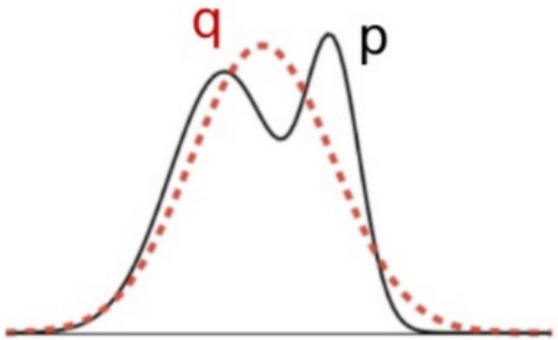
Kendall, Alex, and Yarin Gal. "What uncertainties do we need in Bayesian deep learning for computer vision?." *NeurIPS*, 2017.

Approximation in BNNs

Approximate inference in BNNs

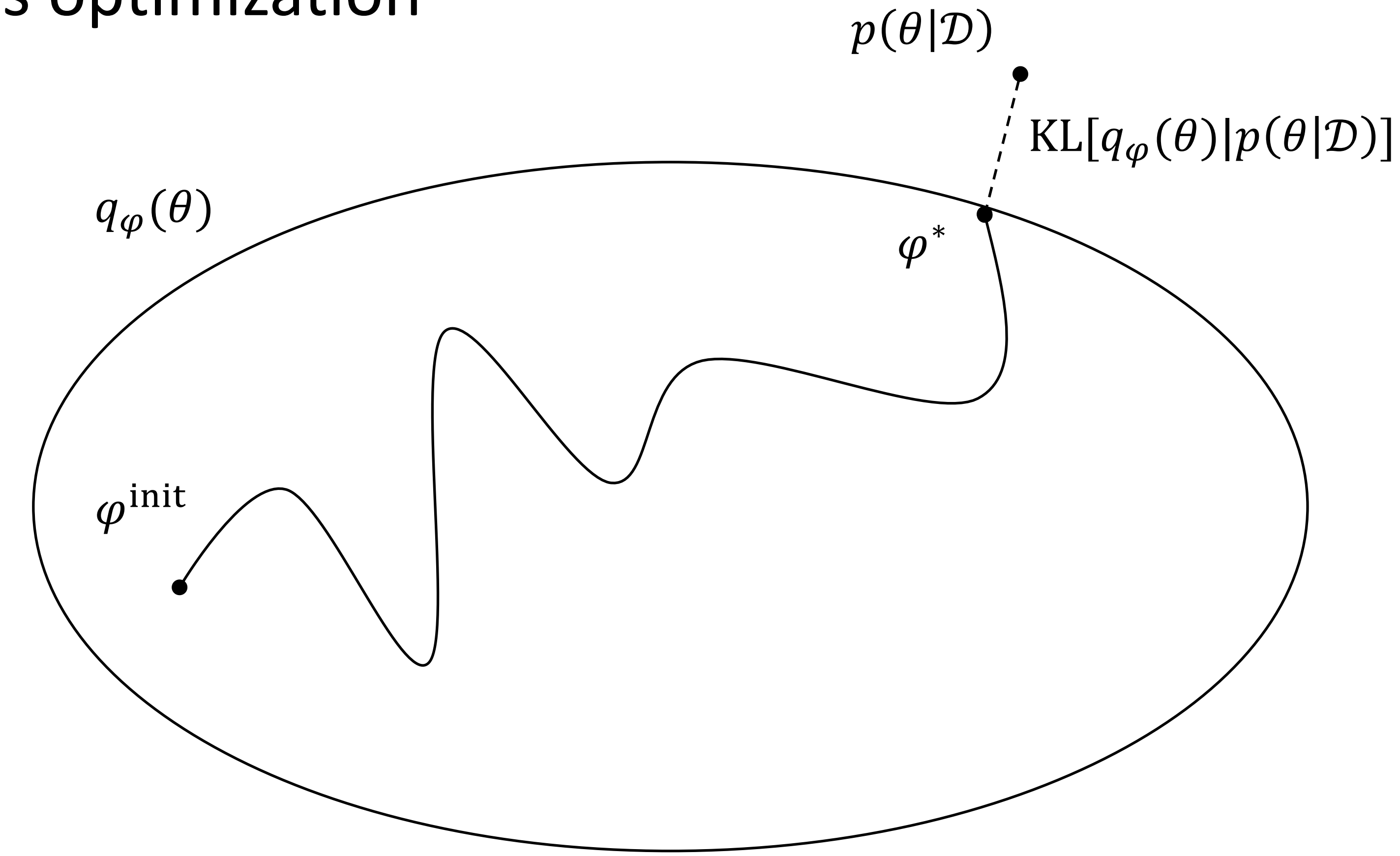
- True posterior $p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{\int p(\theta)p(\mathcal{D}|\theta)d\theta}$:
  - Approximate $p(\theta|\mathcal{D})$ with $q_\varphi(\theta)$:
    - Choose $q_\varphi(\theta)$ from a simple distribution family, e.g., mean-field Gaussian;
    - $q_\varphi(\theta)$ is parametrized by variational parameters φ, e.g., mean and std in Gaussian;
    - Minimize the dissimilarity between $q_\varphi(\theta)$ and $p(\theta|\mathcal{D})$.
- Predictive distribution $p(y^*|x^*, \mathcal{D}) = \int p(y^*|f(x^*;\theta))p(\theta|\mathcal{D})\, d\theta$:
  - Monte Carlo approximation:

$$p(y^*|x^*, \mathcal{D}) \approx \int p(y^*|f(x^*;\theta))q_\varphi(\theta)\, d\theta$$
$$\approx \frac{1}{K}\sum_{k=1}^{K} p(y^*|f(x^*;\theta_k)), \theta_k \sim q_\varphi(\theta)$$



Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Inference as optimization



Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. "Variational inference: A review for statisticians." *JASA*, 2017.

Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Kullback-Leibler Divergence

$$\mathrm{KL}\left[q_\varphi(\theta)\middle|p(\theta|\mathcal{D})\right] = \mathbb{E}_{q_\varphi(\theta)}[\log \frac{q_\varphi(\theta)}{p(\theta|\mathcal{D})}]$$

  - Measures how similar are $q_\varphi(\theta)$ and $p(\theta|\mathcal{D})$;
  - $\mathrm{KL}\left[q_\varphi(\theta)\middle|p(\theta|\mathcal{D})\right] \geq 0$;
  - $\mathrm{KL}\left[q_\varphi(\theta)\middle|p(\theta|\mathcal{D})\right] = 0$ when $q_\varphi(\theta) = p(\theta|\mathcal{D})$;
  - $\mathrm{KL}\left[q_\varphi(\theta)\middle|p(\theta|\mathcal{D})\right] \neq \mathrm{KL}\left[p(\theta|\mathcal{D})\middle|q_\varphi(\theta)\right]$.

Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Derive the Evidence Lower BOund (ELBO) from KL Divergence

always ≥ 0

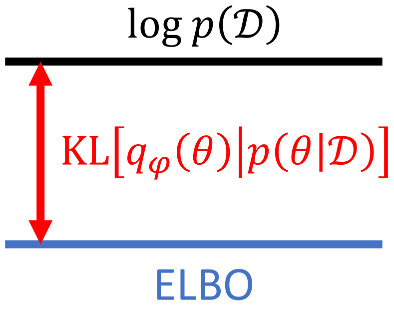
$$\mathrm{KL}\left[q_\varphi(\theta)\middle|p(\theta|\mathcal{D})\right] = \mathbb{E}_{q_\varphi(\theta)}\left[\log\frac{q_\varphi(\theta)}{p(\theta|\mathcal{D})}\right]$$

$$= \mathbb{E}_{q_\varphi(\theta)}\left[\log\frac{q_\varphi(\theta)p(\mathcal{D})}{p(\theta)p(\mathcal{D}|\theta)}\right] = \log p(\mathcal{D}) + \mathbb{E}_{q_\varphi(\theta)}\left[\log\frac{q_\varphi(\theta)}{p(\theta)p(\mathcal{D}|\theta)}\right]$$

$$= \log p(\mathcal{D}) - \mathbb{E}_{q_\varphi(\theta)}\left[\log\frac{p(\theta)p(\mathcal{D}|\theta)}{q_\varphi(\theta)}\right]$$

Model Evidence

ELBO

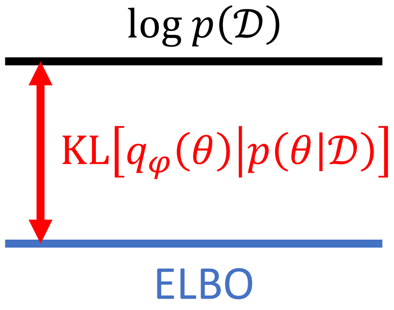


Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Minimize $\mathrm{KL}\left[q_\varphi(\theta) \middle| p(\theta|\mathcal{D})\right]$ is equivalent to maximize ELBO.
  - $\mathrm{KL}\left[q_\varphi(\theta) \middle| p(\theta|\mathcal{D})\right]$ is intractable but ELBO is tractable.

- Rewrite the ELBO ($L(\varphi)$):

$$L(\varphi) = \mathbb{E}_{q_\varphi(\theta)}\left[\log \frac{p(\theta)p(\mathcal{D}|\theta)}{q_\varphi(\theta)}\right] = \mathbb{E}_{q_\varphi(\theta)}[\log p(\mathcal{D}|\theta)] - \mathrm{KL}[q_\varphi(\theta)|p(\theta)]$$



Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Minimize $\mathrm{KL}\left[q_\varphi(\theta) \middle| p(\theta|\mathcal{D})\right]$ is equivalent to maximize ELBO.
  - $\mathrm{KL}\left[q_\varphi(\theta) \middle| p(\theta|\mathcal{D})\right]$ is intractable but ELBO is tractable.

- Rewrite the ELBO ($L(\varphi)$):

$$L(\varphi) = \mathbb{E}_{q_\varphi(\theta)}\left[\log \frac{p(\theta)p(\mathcal{D}|\theta)}{q_\varphi(\theta)}\right] = \mathbb{E}_{q_\varphi(\theta)}[\log p(\mathcal{D}|\theta)] - \mathrm{KL}[q_\varphi(\theta)|p(\theta)]$$

Data fitness term

Regularization term

Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Rewrite the ELBO ($L(\varphi)$):

$$L(\varphi) = \mathbb{E}_{q_\varphi(\theta)}\left[\log \frac{p(\theta)p(\mathcal{D}|\theta)}{q_\varphi(\theta)}\right] = \mathbb{E}_{q_\varphi(\theta)}[\log p(\mathcal{D}|\theta)] - \mathrm{KL}[q_\varphi(\theta)|p(\theta)]$$

Data fitness term

Regularization term

- Data fitness term: expected log likelihood
    - Similar with the standard DL loss that we use for training;
    - Except that now the network's weights are sampled from $q_\varphi(\theta)$;
    - $\mathbb{E}_{q_\varphi(\theta)}[\log p(\mathcal{D}|\theta)] \approx \sum_i \log p(y_i|f(x_i;\theta_j)),\ \theta_j \sim q_\varphi(\theta)$.

Variational Inference: $q_\varphi(\theta) \approx p(\theta|\mathcal{D})$

- Rewrite the ELBO ($L(\varphi)$):

$$L(\varphi) = \mathbb{E}_{q_\varphi(\theta)}\left[\log \frac{p(\theta)p(\mathcal{D}|\theta)}{q_\varphi(\theta)}\right] = \mathbb{E}_{q_\varphi(\theta)}[\log p(\mathcal{D}|\theta)] - \mathrm{KL}[q_\varphi(\theta)|p(\theta)]$$

Data fitness term

Regularization term

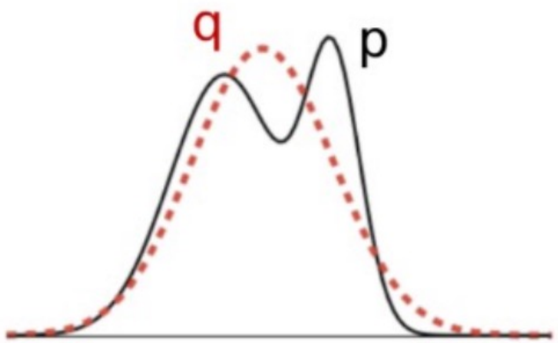
- Regularization term: KL between the posterior and prior
  - Make the posterior distribution closer to the prior;
  - Often analytically tractable.

Recap: approximate inference in BNNs

- True posterior $p(\theta|\mathcal{D})$ is intractable:
  - Approximate $p(\theta|\mathcal{D})$ with variational inference:
    - Choose $q_\varphi(\theta)$ from a simple distribution family, e.g., mean-field Gaussian;
    - Maximize the 'ELBO' w.r.t. φ to fit $q_\varphi(\theta)$.

- Predictive distribution $p(y^*|x^*, \mathcal{D})$ is intractable:
  - Monte Carlo approximation:

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|f(x^*;\theta))q_\varphi(\theta)\, d\theta$$
$$\approx \frac{1}{K}\sum_{k=1}^{K} p(y^*|f(x^*;\theta_k)),\ \theta_k \sim q_\varphi(\theta)$$



Example: mean-field BNNs

- Fully factorized Gaussian prior:

$$p(\theta) = \prod_{i,j,l} p(\theta_{ij}^{(l)}),\ \ p(\theta_{ij}^{(l)}) = \mathcal{N}(0, \sigma_0^2)$$

- Fully factorized Gaussian approximated posterior:

$$q_{\mu,\sigma}(\theta) = \prod_{i,j,l} q_{\mu_{ij}^{(l)},\sigma_{ij}^{(l)}}\left(\theta_{ij}^{(l)}\right),\ \ q_{\mu_{ij}^{(l)},\sigma_{ij}^{(l)}}\left(\theta_{ij}^{(l)}\right) = \mathcal{N}(\mu_{ij}^{(l)}, \sigma_{ij}^{(l)2})$$

- Likelihood: $p(\mathcal{D}|\theta) = \prod_i p(y_i|f(x_i;\theta))$, with i.i.d. assumption of data
  - Regression: $p(y_i|f(x_i;\theta)) = \mathcal{N}(f(x_i;\theta), \sigma_\epsilon^2)$.

Example: mean-field BNNs

- Fully factorized Gaussian prior:

$$p(\theta) = \prod_{i,j,l} p(\theta_{ij}^{(l)}), \;\; p(\theta_{ij}^{(l)}) = \mathcal{N}(0, \sigma_0^2)$$

- Fully factorized Gaussian approximated posterior:

$$q_{\mu,\sigma}(\theta) = \prod_{i,j,l} q_{\mu_{ij}^{(l)}, \sigma_{ij}^{(l)}}\left(\theta_{ij}^{(l)}\right), \;\; q_{\mu_{ij}^{(l)}, \sigma_{ij}^{(l)}}\left(\theta_{ij}^{(l)}\right) = \mathcal{N}(\mu_{ij}^{(l)}, \sigma_{ij}^{(l)2})$$

- Likelihood: $p(\mathcal{D}|\theta) = \prod_i p(y_i|f(x_i;\theta))$, with i.i.d. assumption of data
  - Classification: $p(y_i|f(x_i;\theta)) = \text{Categorical}(\text{logit} = f(x_i;\theta))$.

Example: mean-field BNNs

- Revisit the ELBO ($L(\mu, \sigma)$):

$$L(\mu, \sigma) = \sum_{i=1}^{N} \mathbb{E}_{q_{\mu,\sigma}(\theta)}[\log p(y_i | f(x_i; \theta)] - \mathrm{KL}\left[q_{\mu,\sigma}(\theta) \middle| p(\theta)\right]$$

- Expected loglikelihood:
  - Mini-batch training with $\{(x_m, y_m)\}_{m=1}^{M} \sim \mathcal{D}$.

$$\sum_{i=1}^{N} \mathbb{E}_{q_{\mu,\sigma}(\theta)}[\log p(y_i | f(x_i; \theta)] \approx \frac{N}{M} \sum_{m=1}^{M} \mathbb{E}_{q_{\mu,\sigma}(\theta)}[\log p(y_m | f(x_m; \theta)]$$

  - Monte Carlo: $\mathbb{E}_{q_{\mu,\sigma}(\theta)}[\log p(y_m | f(x_m; \theta)] \approx \log p(y_m | f(x_m; \theta_k), \ \theta_k \sim q_{\mu,\sigma}(\theta)$
    - Reparameterization trick for mean-field BNNs: $\theta_k \sim q_{\mu,\sigma}(\theta) \Leftrightarrow \theta_k = \mu + \sigma \odot \varepsilon_k, \ \varepsilon_k \sim \mathcal{N}(0,1)$

Example: mean-field BNNs

- Revisit the ELBO ($L(\mu, \sigma)$):

$$L(\mu, \sigma) = \sum_{i=1}^{N} \mathbb{E}_{q_{\mu,\sigma}(\theta)}[\log p(y_i | f(x_i; \theta)] - \mathrm{KL}\left[q_{\mu,\sigma}(\theta) \middle| p(\theta)\right]$$

- Regularization:
  - Analytically tractable for two Gaussian distributions:

$$\mathrm{KL}\left[q_{\mu,\sigma}(\theta) \middle| p(\theta)\right] = \sum_{i,j,l} \left[\log \frac{\sigma_0}{\sigma_{ij}^{(l)}} + \frac{\sigma_{ij}^{(l)2} + \mu_{ij}^{(l)2}}{\sigma_0^2} - \frac{1}{2}\right]$$

Example: mean-field BNNs

- Revisit the ELBO ($L(\mu, \sigma)$):

$$L(\mu, \sigma) = \sum_{i=1}^{N} \mathbb{E}_{q_{\mu,\sigma}(\theta)}[\log p(y_i | f(x_i; \theta)] - \mathrm{KL}\left[q_{\mu,\sigma}(\theta) \middle| p(\theta)\right]$$

- Maximize the ELBO w.r.t. μ, σ using SGD or Adam

Priors in BNNs

Mean-field prior is not good enough

- Fully factorized Gaussian prior:

$$p(\theta) = \prod_{i,j,l} p(\theta_{ij}^{(l)}),\ p(\theta) = \mathcal{N}(0, \sigma_0^2 I)$$

- It is hard to encode any prior knowledge into such prior.
  - θ is high-dimensional parameters of a nonlinear deep NN.

- Can we use better priors?

Pretrain priors on relevant tasks

- Two tasks:
  - Task A (source task) with dataset $\mathcal{D}_A$, e.g., ImageNet;
  - Task B (target task) with dataset $\mathcal{D}_B$, e.g., CIFAR-10.

- Standard transfer/continual learning:
  - Pretrain the NN on task A $\rightarrow \theta_A^*$
  - Finetune the NN on task B with θ_A^* as the initialization $\rightarrow \theta^*$
  - Catastrophic forgetting: θ^*no longer works on $\mathcal{D}_A$
  - Suboptimal transfer learning

Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.

Pretrain priors on relevant tasks

- Two tasks:
  - Task A (source task) with dataset $\mathcal{D}_A$, e.g., ImageNet;
  - Task B (target task) with dataset $\mathcal{D}_B$, e.g., CIFAR-10.

- Bayesian transfer/continual learning:
  - Pretrain the BNN on task A → $\log p(\theta|\mathcal{D}_A)$

$$\log p(\theta|\mathcal{D}_A) = \log p(\mathcal{D}_A|\theta) + \log p(\theta) - \log p(\mathcal{D}_A)$$

  - Finetune the BNN on task B with $\log p(\theta|\mathcal{D}_A)$ as the prior → $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$

$$\log p(\theta|\mathcal{D}_A, \mathcal{D}_B) = \log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A) - \log p(\mathcal{D}_B)$$

Prior used in $\mathcal{D}_B$ is the posterior in $\mathcal{D}_A$, e.g., a pretrained prior.

Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.

Pretrain priors on relevant tasks

- Two tasks:
  - Task A (source task) with dataset $\mathcal{D}_A$, e.g., ImageNet;
  - Task B (target task) with dataset $\mathcal{D}_B$, e.g., CIFAR-10.

- Bayesian transfer/continual learning:
  - Pretrain the BNN on task A → $\log p(\theta|\mathcal{D}_A)$

$$\log p(\theta|\mathcal{D}_A) = \log p(\mathcal{D}_A|\theta) + \log p(\theta) - \log p(\mathcal{D}_A)$$

  - Finetune the BNN on task B with $\log p(\theta|\mathcal{D}_A)$ as the prior → $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$

$$\log p(\theta|\mathcal{D}_A, \mathcal{D}_B) = \log p(\mathcal{D}_B|\theta) + \lambda \log p(\theta|\mathcal{D}_A) - \log p_\lambda(\mathcal{D}_B)$$

Scaling hyperparameter

Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.

Pretrain priors on relevant tasks

- Two tasks:
  - Task A (source task) with dataset $\mathcal{D}_A$, e.g., ImageNet;
  - Task B (target task) with dataset $\mathcal{D}_B$, e.g., CIFAR-10.

- Bayesian transfer/continual learning:
  - Pretrain the BNN on task A → $\log p(\theta|\mathcal{D}_A)$

$$\log p(\theta|\mathcal{D}_A) = \log p(\mathcal{D}_A|\theta) + \log p(\theta) - \log p(\mathcal{D}_A)$$

  - Finetune the BNN on task B with $\log p(\theta|\mathcal{D}_A)$ as the prior → $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$

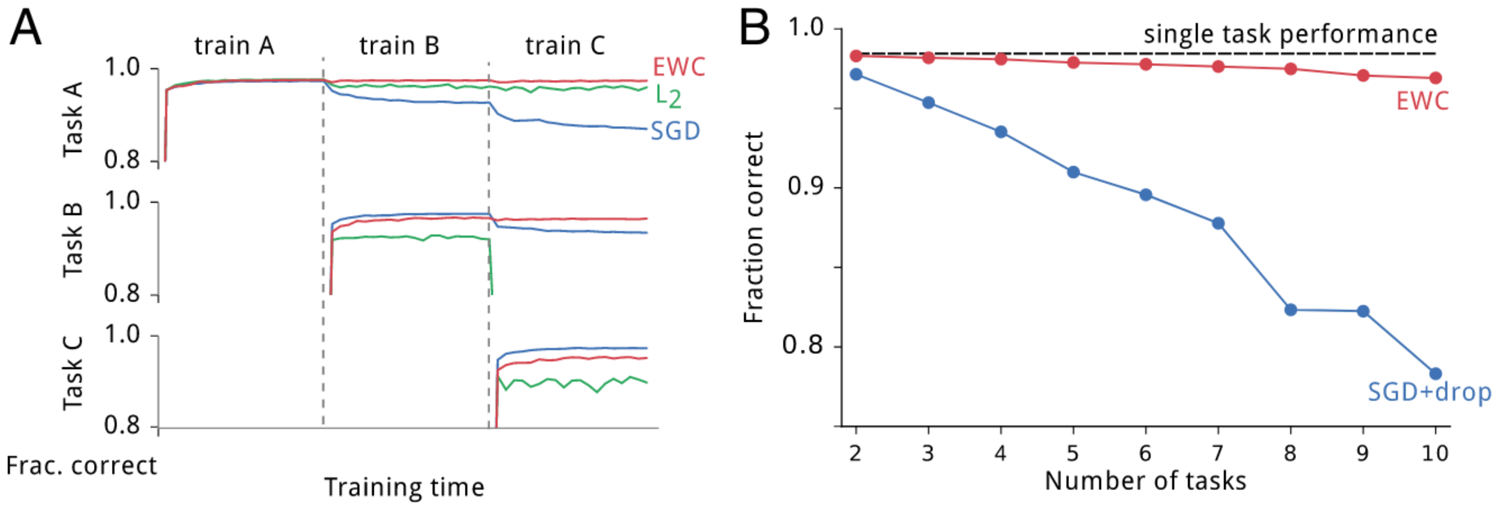
$$\log p(\theta|\mathcal{D}_A, \mathcal{D}_B) = \log p(\mathcal{D}_B|\theta) + \lambda \log p(\theta|\mathcal{D}_A) - \log p_\lambda(\mathcal{D}_B)$$

  - Catastrophic forgetting resolved: $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$ works on both $\mathcal{D}_A$ and $\mathcal{D}_B$
  - Better performance on $\mathcal{D}_B$ in transfer learning.

Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.

Pretrain priors on relevant tasks

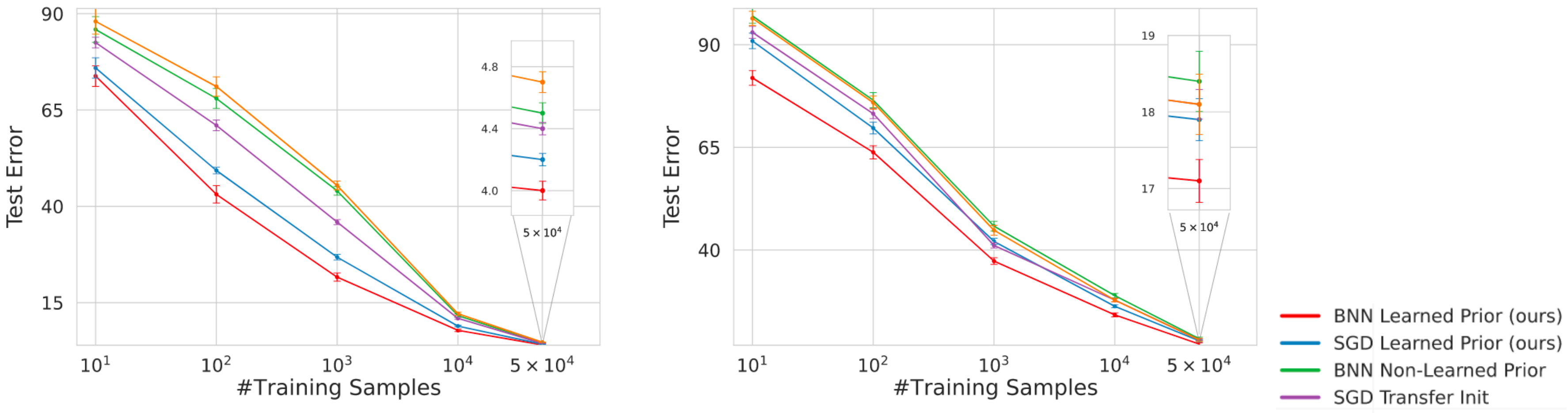
- Elastic Weight Consolidation (EWC)
  - Approximate $\log p(\theta|\mathcal{D}_A)$ by Laplace approximation
    - Similar with mean-field variational inference
  - Find the Maximum a Posteriori (MAP) estimation of $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$
    - Ignores the uncertainty of $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$



Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.

Pretrain priors on relevant tasks

- Pretrained prior can come from self-supervised learning!
  - Approximate $\log p(\theta|\mathcal{D}_A)$ by SWAG with a SimCLR loss
    - SWAG can estimate the covariance which is ignored in VI and Laplace approximation
  - Approximate the $\log p(\theta|\mathcal{D}_A, \mathcal{D}_B)$ with SGHMC



(a) CIFAR-10 (b) CIFAR-100

Shwartz-Ziv, Ravid, et al. "Pre-train your loss: Easy Bayesian transfer learning with informative priors." *NeurIPS, 2022.*

Pretrain priors with functional information

- Fully factorized Gaussian prior:

$$p(\theta) = \prod_{i,j,l} p(\theta_{ij}^{(l)}),\ p(\theta_{ij}^{(l)}) = \mathcal{N}(0, \sigma_0^2)$$

- A more flexible fully factorized hierarchical Gaussian prior:

$$p(\theta) = \prod_{i,j,l} p(\theta_{ij}^{(l)}),\ p(\theta_{ij}^{(l)}) = \mathcal{N}(0, \sigma_0^2),\ p(\sigma_0^2) = \Gamma^{-1}(\alpha_0, \beta_0)$$

- Optimize α_0, β_0 to match the functional information at hand.

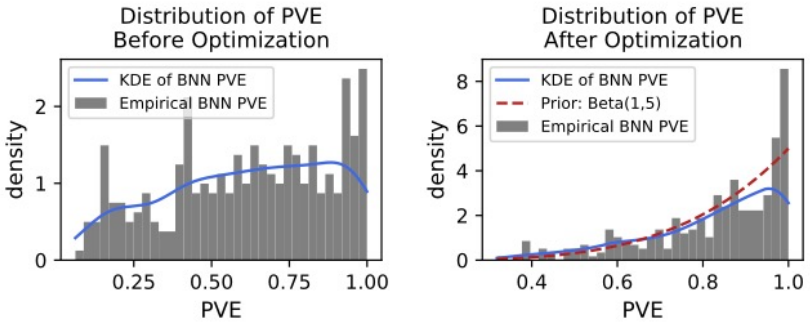
Pretrain priors with functional information

- Suppose we know that features can only explain 80% of the target:
  - I.e., the proportion of variance explained (PVE) is 0.8.

- By definition, PVE of a function $f(x;\theta)$ is:

$$\mathrm{PVE}(\theta) = \frac{\mathrm{Var}_x(f(x;\theta))}{\mathrm{Var}_x\big(f(x;\theta)\big) + \sigma_\epsilon^2}$$

  - $p(\theta;\alpha_0,\beta_0)$ defines the a prior over model $\mathrm{PVE}(\theta;\alpha_0,\beta_0)$, i.e., $p(\mathrm{PVE}_{\alpha_0,\beta_0})$.
  - If we have a prior belief on PVE, i.e., $p(\mathrm{PVE})$:
    - Beta distribution with mode equals to 0.8
  - Optimize α_0,β_0 such that $p(\mathrm{PVE}_{\alpha_0,\beta_0}) \approx p(\mathrm{PVE})$
    - $\alpha_0^*,\beta_0^* = \underset{\alpha_0,\beta_0}{\mathrm{argmin}}\,\mathrm{KL}\big[p\big(\mathrm{PVE}_{\alpha_0,\beta_0}\big)|p(\mathrm{PVE})\,\big]$



Cui, Tianyu, et al. "Informative Bayesian neural network priors for weak signals." *Bayesian Analysis,* 2022.

Pretrain priors with functional information

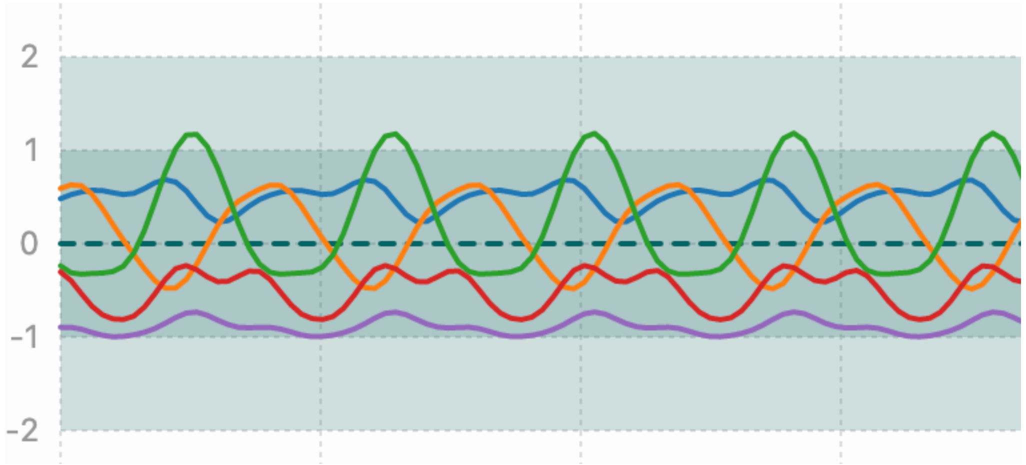
- When the prior knowledge about PVE is unavailable:
  - A noninformative prior over PVE ($U[0,1]$) improves when the data is noisy;
  - A noninformative prior over weights leads to overfitting.

Periods	7 days		14 days		21 days		28 days	
Metrics	MSE	PVE	MSE	PVE	MSE	PVE	MSE	PVE
MF+CV	0.582 (0.016)	0.278 (0.013)	0.615 (0.017)	0.164 (0.031)	0.686 (0.031)	0.118 (0.015)	0.701 (0.043)	0.095 (0.011)
HS	0.500 (0.008)	0.301 (0.006)	0.556 (0.011)	0.189 (0.010)	0.652 (0.054)	0.120 (0.041)	0.629 (0.019)	0.101 (0.013)
HMF	**0.481 (0.012)**	**0.322 (0.009)**	0.589 (0.022)	0.179 (0.023)	0.660 (0.047)	0.085 (0.072)	0.664 (0.043)	0.066 (0.051)
HMF+PVE	0.482 (0.013)	0.320 (0.010)	**0.546 (0.014)**	**0.227 (0.011)**	**0.613 (0.014)**	**0.138 (0.013)**	**0.622 (0.017)**	**0.109 (0.011)**

Cui, Tianyu, et al. “Informative Bayesian neural network priors for weak signals.” *Bayesian Analysis,* 2022.

Pretrain priors with functional information

- Suppose we know the neural network functions come from a Gaussian process $p(f)$
  - Gaussian processes are distributions over functions f



Tran, Ba-Hien, et al. "All you need is a good functional prior for Bayesian deep learning." *JMLR*, 2022.

Pretrain priors with functional information

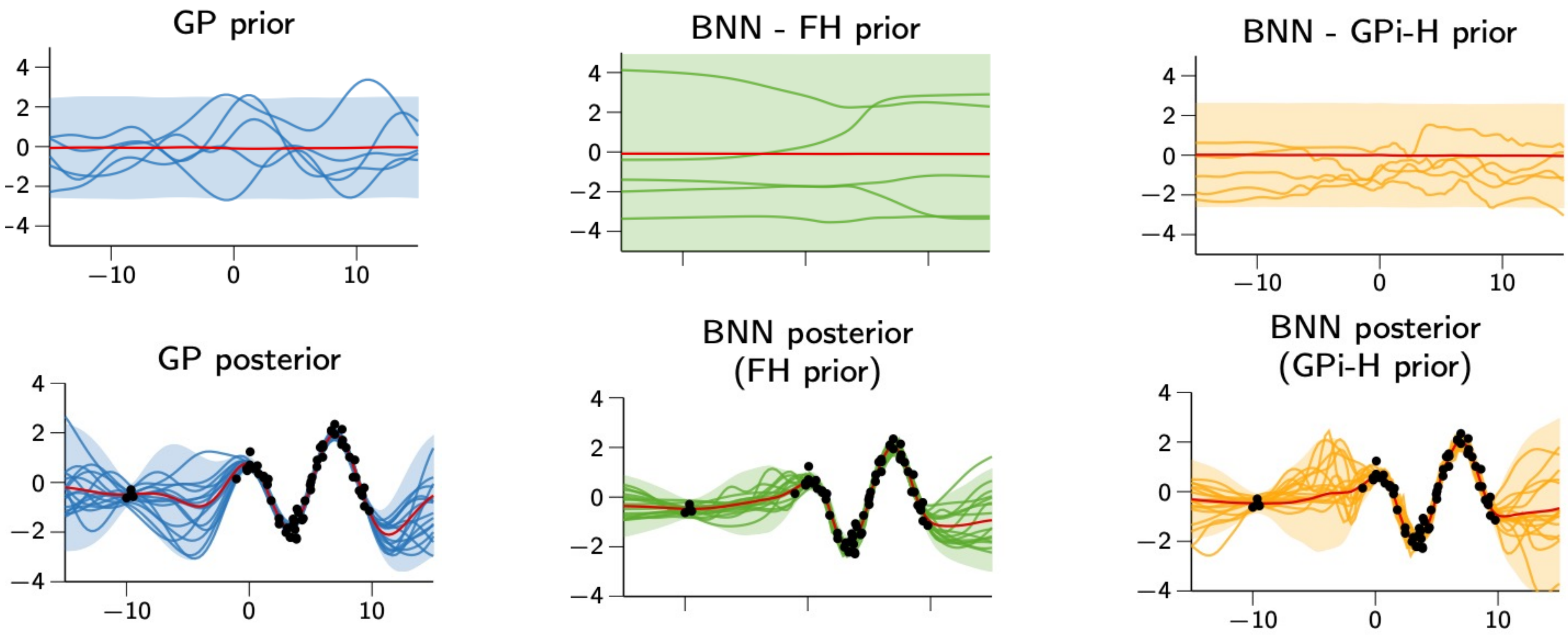
- Suppose we know the neural network functions come from a Gaussian process $p(f)$
  - Gaussian processes are distributions over functions f
- For a fully factorized hierarchical Gaussian prior over weights θ:

$$p(\theta) = \prod_{i,j,l} p(\theta_{ij}^{(l)}),\ p(\theta_{ij}^{(l)}) = \mathcal{N}\left(0, \sigma_l^2\right),\ p\left(\sigma_l^2\right) = \Gamma^{-1}(\alpha_l, \beta_l)$$

  - It defines a prior over function: $p(f; \alpha, \beta) = \int p(f|\theta)p(\theta; \alpha, \beta)d\theta$
- Minimize the Wasserstein distance between $p(f; \alpha, \beta)$ and $p(f)$

Tran, Ba-Hien, et al. "All you need is a good functional prior for Bayesian deep learning." *JMLR*, 2022.

Pretrain priors with functional information



Tran, Ba-Hien, et al. "All you need is a good functional prior for Bayesian deep learning." *JMLR*, 2022.

Recap: informative weight space prior

- Pretrain the weight space prior on relevant tasks
  - Resolve the catastrophic forgetting in continual learning;
  - Improve the prediction performance on target tasks;

- Pretrain the weight space prior with functional information
  - Prior over PVE improves the model prediction on noisy data;
  - Gaussian processes prior improves the model uncertainty on OOD data.

References

Articles:

1. Blundell, Charles, et al. "Weight uncertainty in neural network." *ICML*, 2015.
2. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS*, 2017.
3. Sun, Shengyang, et al. "Functional variational Bayesian neural networks." *ICLR*, 2019.
4. Kendall, Alex, et al. "What uncertainties do we need in bayesian deep learning for computer vision?." *NeurIPS*, 2017.
5. Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. "Variational inference: A review for statisticians." *JASA*, 2017.
6. Shwartz-Ziv, Ravid, et al. "Pre-train your loss: Easy Bayesian transfer learning with informative priors." *NeurIPS, 2022.*
7. Cui, Tianyu, et al. “Informative Bayesian neural network priors for weak signals.” *Bayesian Analysis,* 2022.
8. Tran, Ba-Hien, et al. "All you need is a good functional prior for Bayesian deep learning." *JMLR*, 2022.

Tutorials:

1. Maddox, Wesley. “Bayesian neural networks: a tutorial” *CILVR Lab Tutorial*, 2020.
2. Li, Yingzhen. “An Introduction to Bayesian Neural Networks” *ProbAI*, 2022.