Uncertainty in Deep Learning:

From Bayesian Neural Networks to Dropout

Samar Hadou

Department of Electrical and Systems Engineering
University of Pennsylvania

April 14, 2022

This talk is based on Yarin Gal's PhD work:

- Y. Gal. **Uncertainty in Deep Learning**. PhD Thesis, University of Cambridge, 2016.
- Y. Gal and Z. Ghahramani. **Dropout as a Bayesian approximate: Representing Model Uncertainty in Deep Learning**. ICML, 2016.
- A. Kendall and Y. Gal. **What uncertainties do we need in Bayesian deep learning for computer vision?** NeurIPS, 2017.

Standard deep learning
(CNNs, RNNs, ..., etc.)

- Imagine that we trained a CNN to classify **dog breeds**
- What happens when executed on this one?

Bayesian deep learning

- Gives the ability to models to say "I am not sure!"
- Learns distributions over the weights to tell how likely a model fits the data, and
- Provides uncertainty estimates

Yarin's question:
Do we need to replace our standard models?

Standard deep learning
(CNNs, RNNs, ..., etc.)

- Imagine that we trained a CNN to classify **dog breeds**
- What happens when executed on this one?

Bayesian deep learning

- Gives the ability to models to say "I am not sure!"
- Learns distributions over the weights to tell how likely a model fits the data, and
- Provides uncertainty estimates

Yarin's question:
Do we need to replace our standard models?

Standard deep learning
(CNNs, RNNs, ..., etc.)

- Imagine that we trained a CNN to classify **dog breeds**
- What happens when executed on this one?

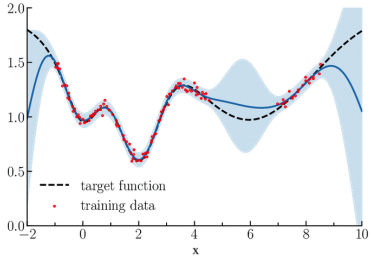
Bayesian deep learning

- Gives the ability to models to say "I am not sure!"
- Learns distributions over the weights to tell how likely a model fits the data, and
- Provides uncertainty estimates

Yarin's question:
Do we need to replace our standard models?

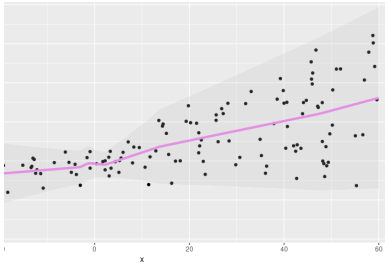
I. Epistemic Uncertainty

- Uncertainty that we can mitigate by adding more data



II. Aleatoric Uncertainty

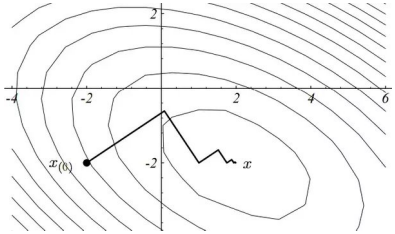
- Adding more data would not decrease uncertainty



Bayesian Deep Learning

Standard deep learning

- Finds a point estimate (model) that **minimizes** some loss function



Bayesian deep learning

- Evaluates predictive distributions via **marginalization**

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) d\boldsymbol{\omega}$$

- Assumption: $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\mathbf{y}; f^{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I})$, with τ is said to be the model precision. (We'll revisit this assumption.)

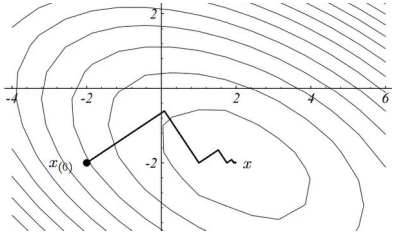
- However, the posterior distribution

$$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) p(\boldsymbol{\omega})}{p(\mathbf{Y}|\mathbf{X})}$$

is intractable.

Standard deep learning

- Finds a point estimate (model) that **minimizes** some loss function



Bayesian deep learning

- Evaluates predictive distributions via **marginalization**

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) d\boldsymbol{\omega}$$

- Assumption: $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\mathbf{y}; f^{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I})$, with τ is said to be the model precision. (We'll revisit this assumption.)

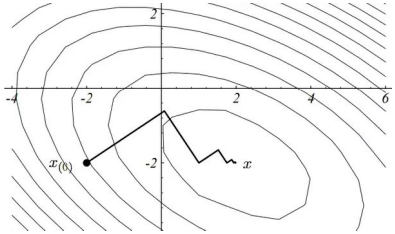
- However, the posterior distribution

$$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) p(\boldsymbol{\omega})}{p(\mathbf{Y}|\mathbf{X})}$$

is intractable.

Standard deep learning

- Finds a point estimate (model) that **minimizes** some loss function



Bayesian deep learning

- Evaluates predictive distributions via **marginalization**

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) d\boldsymbol{\omega}$$

- Assumption: $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\mathbf{y}; f^{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I})$, with τ is said to be the model precision. (We'll revisit this assumption.)

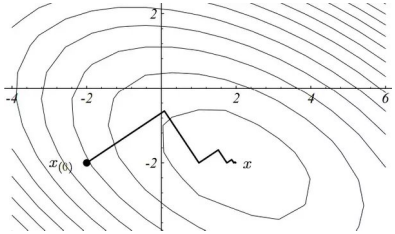
- However, the posterior distribution

$$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) p(\boldsymbol{\omega})}{p(\mathbf{Y}|\mathbf{X})}$$

is intractable.

Standard deep learning

- Finds a point estimate (model) that **minimizes** some loss function



Bayesian deep learning

- Evaluates predictive distributions via **marginalization**

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) d\boldsymbol{\omega}$$

- Assumption: $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\mathbf{y}; f^{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I})$, with τ is said to be the model precision. (We'll revisit this assumption.)

- However, the posterior distribution

$$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) p(\boldsymbol{\omega})}{p(\mathbf{Y}|\mathbf{X})}$$

is intractable.

- Approximate the posterior $p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$ with a **parameterized** distribution $q_\theta(\boldsymbol{\omega})$
- The goal is to solve

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})\big)$$

- Finding an equivalent problem

$$\begin{aligned}
\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})\big) &= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})} d\boldsymbol{\omega} \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega}) p(\mathbf{Y}|\mathbf{X})}{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) p(\boldsymbol{\omega})} d\boldsymbol{\omega} \qquad (Bayes' rule) \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega})} d\boldsymbol{\omega} - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega} + \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}) d\boldsymbol{\omega} \\
&= \underbrace{\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega}}_{-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})} + \underbrace{\log p(\mathbf{Y}|\mathbf{X})}_{\text{log evidence}}
\end{aligned}$$

$-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})$ = −ELBO: Evidence Lower BOund

- Approximate the posterior $p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$ with a **parameterized** distribution $q_\theta(\boldsymbol{\omega})$
- The goal is to solve

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})\big)$$

- Finding an equivalent problem

$$\begin{aligned}
\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})\big) &= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})} d\boldsymbol{\omega} \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega}) p(\mathbf{Y}|\mathbf{X})}{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) p(\boldsymbol{\omega})} d\boldsymbol{\omega} \qquad (Bayes' rule) \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega})} d\boldsymbol{\omega} - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega} + \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}) d\boldsymbol{\omega} \\
&= \underbrace{\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega}}_{-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})} + \underbrace{\log p(\mathbf{Y}|\mathbf{X})}_{\text{log evidence}}
\end{aligned}$$

$-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})$ = −ELBO: Evidence Lower BOund

- Approximate the posterior $p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})$ with a **parameterized** distribution $q_\theta(\boldsymbol{\omega})$
- The goal is to solve

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\big)$$

- Finding an equivalent problem

$$\begin{aligned}
\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\big) &= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})} d\boldsymbol{\omega} \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega}) p(\mathbf{Y}|\mathbf{X})}{p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) p(\boldsymbol{\omega})} d\boldsymbol{\omega} \qquad (Bayes' rule) \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega})} d\boldsymbol{\omega} - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) d\boldsymbol{\omega} + \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}) d\boldsymbol{\omega} \\
&= \underbrace{\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) d\boldsymbol{\omega}}_{-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})} + \underbrace{\log p(\mathbf{Y}|\mathbf{X})}_{\text{log evidence}}
\end{aligned}$$

$-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})$ = −ELBO: Evidence Lower BOund

- Approximate the posterior $p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})$ with a **parameterized** distribution $q_\theta(\boldsymbol{\omega})$
- The goal is to solve

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\big)$$

- Finding an equivalent problem

$$\begin{aligned}
\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\big) &= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})} d\boldsymbol{\omega} \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega}) p(\mathbf{Y}|\mathbf{X})}{p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) p(\boldsymbol{\omega})} d\boldsymbol{\omega} \qquad (Bayes' rule) \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega})} d\boldsymbol{\omega} - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) d\boldsymbol{\omega} + \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}) d\boldsymbol{\omega} \\
&= \underbrace{\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) d\boldsymbol{\omega}}_{-\mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta})} + \underbrace{\log p(\mathbf{Y}|\mathbf{X})}_{\text{log evidence}} \\
&= -\text{ELBO: Evidence Lower BOund}
\end{aligned}$$

- Approximate the posterior $p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})$ with a **parameterized** distribution $q_\theta(\boldsymbol{\omega})$
- The goal is to solve

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\text{argmin}} \quad \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\big)$$

- Finding an equivalent problem

$$\begin{aligned}
\text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\big) &= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})} d\boldsymbol{\omega} \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega}) p(\mathbf{Y}|\mathbf{X})}{p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) p(\boldsymbol{\omega})} d\boldsymbol{\omega} \qquad (Bayes' rule) \\
&= \int q_\theta(\boldsymbol{\omega}) \log \frac{q_\theta(\boldsymbol{\omega})}{p(\boldsymbol{\omega})} d\boldsymbol{\omega} - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) d\boldsymbol{\omega} + \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}) d\boldsymbol{\omega} \\
&= \underbrace{\text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) - \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\omega}) d\boldsymbol{\omega}}_{-\mathcal{L}_{\text{VI}}(\boldsymbol{\theta})} + \underbrace{\log p(\mathbf{Y}|\mathbf{X})}_{\text{log evidence}}
\end{aligned}$$

$=-$ELBO: Evidence Lower BOund

- The variational inference problem is translated into ELBO maximization

$$\begin{aligned}\boldsymbol{\theta}^* &\in \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \quad \mathcal{L}_{\mathrm{VI}}(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \quad \underbrace{\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega}}_{\text{Expected log likelihood}} - \mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \quad \sum_{i=1}^{N} \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i | f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} - \mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big),\end{aligned}$$

  assuming that $(\mathbf{x}_i, \mathbf{y}_i)$ are drawn independently from the data distribution
  - $f^{\boldsymbol{\omega}}(\mathbf{x}_i)$ is the model's stochastic output

- However, the integral is still **intractable** for models with more than one hidden layer

- The variational inference problem is translated into ELBO maximization

$$\begin{aligned}\boldsymbol{\theta}^* &\in \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \mathcal{L}_{\text{VI}}(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \underbrace{\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega}}_{\text{Expected log likelihood}} - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \sum_{i=1}^{N} \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i | f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big),\end{aligned}$$

assuming that $(\mathbf{x}_i, \mathbf{y}_i)$ are drawn independently from the data distribution

- $f^{\boldsymbol{\omega}}(\mathbf{x}_i)$ is the model's stochastic output

- However, the integral is still **intractable** for models with more than one hidden layer

- The variational inference problem is translated into ELBO maximization

$$\begin{aligned}\boldsymbol{\theta}^* &\in \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \mathcal{L}_{\text{VI}}(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \underbrace{\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega}}_{\text{Expected log likelihood}} - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \sum_{i=1}^{N} \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i | f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big),\end{aligned}$$

  assuming that $(\mathbf{x}_i, \mathbf{y}_i)$ are drawn independently from the data distribution
  - $f^{\boldsymbol{\omega}}(\mathbf{x}_i)$ is the model's stochastic output

- However, the integral is still **intractable** for models with more than one hidden layer

- The variational inference problem is translated into ELBO maximization

$$\begin{aligned}\boldsymbol{\theta}^* &\in \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \mathcal{L}_{\text{VI}}(\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \underbrace{\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) d\boldsymbol{\omega}}_{\text{Expected log likelihood}} - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \sum_{i=1}^{N} \int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i | f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big),\end{aligned}$$

  assuming that $(\mathbf{x}_i, \mathbf{y}_i)$ are drawn independently from the data distribution
  - $f^{\boldsymbol{\omega}}(\mathbf{x}_i)$ is the model's stochastic output

- However, the integral is still **intractable** for models with more than one hidden layer

- **Monte-Carlo integration**: Sample T realizations of the weights from the distribution $q_\theta(\boldsymbol{\omega})$

$$\frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{y}_i|f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_i)) \overset{T\to\infty}{\longrightarrow} \int q_\theta(\boldsymbol{\omega})\log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i))d\boldsymbol{\omega}$$

with $\hat{\boldsymbol{\omega}}_t \sim q_\theta(\boldsymbol{\omega})$.[1] However, the distribution $q_\theta(\boldsymbol{\omega})$ is not known yet

- **Re-parametrize** $q_\theta(\boldsymbol{\omega})$: Let $\boldsymbol{\omega} = \{\mathbf{W}_\ell\}_{\ell=1}^{L}$, and re-write each column in $\mathbf{W}_\ell$ as

$$\mathbf{W}_{\ell,k} = g(\boldsymbol{\theta}_{\ell,k}, \boldsymbol{\epsilon}_{\ell,k})$$

with some distribution $p(\boldsymbol{\epsilon}_{\ell,k})$ that is parameter free

Ex: If $q_\theta(\omega) = \mathcal{N}(\omega; \mu, \sigma^2)$, then we can define $\omega = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$

[1] The hat denotes a realization of a random variable

- **Monte-Carlo integration**: Sample T realizations of the weights from the distribution $q_\theta(\boldsymbol{\omega})$

$$\frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{y}_i|f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_i)) \overset{T\to\infty}{\longrightarrow} \int q_\theta(\boldsymbol{\omega})\log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i))d\boldsymbol{\omega}$$

with $\hat{\boldsymbol{\omega}}_t \sim q_\theta(\boldsymbol{\omega})$.[1] However, the distribution $q_\theta(\boldsymbol{\omega})$ is not known yet

- **Re-parametrize** $q_\theta(\boldsymbol{\omega})$: Let $\boldsymbol{\omega} = \{\mathbf{W}_\ell\}_{\ell=1}^{L}$, and re-write each column in $\mathbf{W}_\ell$ as

$$\mathbf{W}_{\ell,k} = g(\boldsymbol{\theta}_{\ell,k}, \boldsymbol{\epsilon}_{\ell,k})$$

with some distribution $p(\boldsymbol{\epsilon}_{\ell,k})$ that is parameter free

Ex: If $q_\theta(\omega) = \mathcal{N}(\omega; \mu, \sigma^2)$, then we can define $\omega = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$

[1] The hat denotes a realization of a random variable

- **Monte-Carlo integration**: Sample T realizations of the weights from the distribution $q_\theta(\boldsymbol{\omega})$

$$\frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{y}_i|f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_i)) \overset{T\to\infty}{\longrightarrow} \int q_\theta(\boldsymbol{\omega})\log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i))d\boldsymbol{\omega}$$

with $\hat{\boldsymbol{\omega}}_t \sim q_\theta(\boldsymbol{\omega})$.[1] However, the distribution $q_\theta(\boldsymbol{\omega})$ is not known yet

- **Re-parametrize** $q_\theta(\boldsymbol{\omega})$: Let $\boldsymbol{\omega} = \{\mathbf{W}_\ell\}_{\ell=1}^{L}$, and re-write each column in $\mathbf{W}_\ell$ as

$$\mathbf{W}_{\ell,k} = g(\boldsymbol{\theta}_{\ell,k}, \boldsymbol{\epsilon}_{\ell,k})$$

with some distribution $p(\boldsymbol{\epsilon}_{\ell,k})$ that is parameter free

Ex: If $q_\theta(\omega) = \mathcal{N}(\omega; \mu, \sigma^2)$, then we can define $\omega = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$

[1] The hat denotes a realization of a random variable

- **Change of variables**: $\boldsymbol{\omega} = g(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \iff q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) = \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right)$

- Re-formulating our integral

$$\begin{aligned}\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} &= \int\int q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} d\boldsymbol{\epsilon} \qquad \text{(total probability)} \\ &= \int p(\boldsymbol{\epsilon}) \left(\int \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right) \log p\left(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)\right) d\boldsymbol{\omega} \right) d\boldsymbol{\epsilon} \\ &= \int p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{g(\boldsymbol{\theta}, \boldsymbol{\epsilon})}(\mathbf{x}_i)) d\boldsymbol{\epsilon}\end{aligned}$$

with $p(\boldsymbol{\epsilon}) = \prod_{\ell,k} p(\boldsymbol{\epsilon}_{\ell,k})$

- Next step is to use Monte-Carlo integration with **a single sample**, i.e., $T = 1$

- **Change of variables**: $\boldsymbol{\omega} = g(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \iff q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) = \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right)$

- Re-formulating our integral

$$\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} = \int \int q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} d\boldsymbol{\epsilon} \qquad \text{(total probability)}$$

$$= \int p(\epsilon) \left(\int \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right) \log p\left(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)\right) d\boldsymbol{\omega} \right) d\boldsymbol{\epsilon}$$

$$= \int p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{g(\boldsymbol{\theta}, \boldsymbol{\epsilon})}(\mathbf{x}_i)) d\boldsymbol{\epsilon}$$

with $p(\boldsymbol{\epsilon}) = \prod_{\ell,k} p(\boldsymbol{\epsilon}_{\ell,k})$

- Next step is to use Monte-Carlo integration with **a single sample**, i.e., $T = 1$

- **Change of variables**: $\boldsymbol{\omega} = g(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \iff q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) = \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right)$

- Re-formulating our integral

$$\begin{aligned}\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} &= \int \int q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} d\boldsymbol{\epsilon} \qquad \text{(total probability)} \\ &= \int p(\boldsymbol{\epsilon}) \left(\int \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right) \log p\left(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)\right) d\boldsymbol{\omega} \right) d\boldsymbol{\epsilon} \\ &= \int p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{g(\boldsymbol{\theta}, \boldsymbol{\epsilon})}(\mathbf{x}_i)) d\boldsymbol{\epsilon}\end{aligned}$$

with $p(\boldsymbol{\epsilon}) = \prod_{\ell,k} p(\boldsymbol{\epsilon}_{\ell,k})$

- Next step is to use Monte-Carlo integration with **a single sample**, i.e., $T = 1$

- **Change of variables**: $\boldsymbol{\omega} = g(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \iff q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) = \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right)$

- Re-formulating our integral

$$\begin{aligned}
\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} &= \int \int q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} d\boldsymbol{\epsilon} \qquad \text{(total probability)} \\
&= \int p(\boldsymbol{\epsilon}) \left(\int \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right) \log p\left(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)\right) d\boldsymbol{\omega} \right) d\boldsymbol{\epsilon} \\
&= \int p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{g(\boldsymbol{\theta}, \boldsymbol{\epsilon})}(\mathbf{x}_i)) d\boldsymbol{\epsilon}
\end{aligned}$$

with $p(\boldsymbol{\epsilon}) = \prod_{\ell,k} p(\boldsymbol{\epsilon}_{\ell,k})$

- Next step is to use Monte-Carlo integration with **a single sample**, i.e., $T = 1$

- **Change of variables**: $\boldsymbol{\omega} = g(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \iff q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) = \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right)$

- Re-formulating our integral

$$\begin{aligned}
\int q_\theta(\boldsymbol{\omega}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} &= \int \int q_\theta(\boldsymbol{\omega}|\boldsymbol{\epsilon}) p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)) d\boldsymbol{\omega} d\boldsymbol{\epsilon} \qquad \text{(total probability)} \\
&= \int p(\boldsymbol{\epsilon}) \left(\int \delta\left(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon})\right) \log p\left(\mathbf{y}_i|f^{\boldsymbol{\omega}}(\mathbf{x}_i)\right) d\boldsymbol{\omega} \right) d\boldsymbol{\epsilon} \\
&= \int p(\boldsymbol{\epsilon}) \log p(\mathbf{y}_i|f^{g(\boldsymbol{\theta}, \boldsymbol{\epsilon})}(\mathbf{x}_i)) d\boldsymbol{\epsilon}
\end{aligned}$$

with $p(\boldsymbol{\epsilon}) = \prod_{\ell,k} p(\boldsymbol{\epsilon}_{\ell,k})$

- Next step is to use Monte-Carlo integration with **a single sample**, i.e., $T = 1$

- The ELBO optimization problem becomes

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \sum_{i=1}^{N} \log p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big)$$

  - For regression tasks: If we assume that $p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) = \mathcal{N}\left(\mathbf{y}_i; f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i), \tau^{-1}\mathbf{I}\right)$, then

$$\log p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) = -\frac{\tau}{2}\left\|\mathbf{y}_i - f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right\|^2 + \text{const}$$

- Looks familiar, Huh?

- The ELBO optimization problem becomes

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \sum_{i=1}^{N} \log p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big)$$

  - For regression tasks: If we assume that $p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) = \mathcal{N}\left(\mathbf{y}_i; f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i), \tau^{-1}\mathbf{I}\right)$, then

$$\log p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) = -\frac{\tau}{2}\left\|\mathbf{y}_i - f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right\|^2 + \text{const}$$

- Looks familiar, Huh?

- The ELBO optimization problem becomes

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\text{argmax}} \quad \sum_{i=1}^{N} \log p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) - \text{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big)$$

  - For regression tasks: If we assume that $p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) = \mathcal{N}\left(\mathbf{y}_i; f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i), \tau^{-1}\mathbf{I}\right)$, then

$$\log p\left(\mathbf{y}_i | f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right) = -\frac{\tau}{2}\left\|\mathbf{y}_i - f^{g(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)\right\|^2 + \text{const}$$

- Looks familiar, Huh?

- The output of a feed-forward NN (without dropout) can be written as

$$f^{\mathbf{M}_1,\ldots,\mathbf{M}_L}(\mathbf{x}) = \sigma\Big(\ldots\sigma\big(\mathbf{M}_2\underbrace{\sigma\left(\mathbf{M}_1\mathbf{x}\right)}_{\mathbf{h}_2}\big)\ldots\Big)$$

  where $\mathbf{M}_\ell$ is a deterministic weight matrix and $\sigma(.)$ is an activation function

- Dropout injects stochastic noise in the feature space $\{\mathbf{x}, \mathbf{h}_2, \ldots\}$, i.e.,

$$\mathbf{h}_{\ell+1} = \sigma(\mathbf{M}_\ell(\mathbf{h}_\ell \odot \hat{\boldsymbol{\epsilon}}_\ell)) = \sigma(\mathbf{M}_\ell \text{diag}(\hat{\boldsymbol{\epsilon}}_\ell)\mathbf{h}_\ell) = \sigma(\widehat{\mathbf{W}}_\ell \mathbf{h}_\ell)$$

  with $\boldsymbol{\epsilon}_\ell \sim \text{Ber}(p_\ell), \ell = 1, \ldots, L$ and $\mathbf{W}_{\ell,k} = \epsilon_\ell \mathbf{M}_{\ell,k}$

$$\underset{\mathbf{M}_1,\ldots,\mathbf{M}_L}{\text{argmin}} \quad \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_i - f^{\widehat{\mathbf{W}}_1,\ldots,\widehat{\mathbf{W}}_L}(\mathbf{x}_i)\right\|^2 + \sum_{i=1}^{L}\lambda_i\|\mathbf{M}_i\|^2$$

- The output of a feed-forward NN (without dropout) can be written as

$$f^{\mathbf{M}_1,\ldots,\mathbf{M}_L}(\mathbf{x}) = \sigma\Big(\ldots\sigma\big(\mathbf{M}_2\underbrace{\sigma\left(\mathbf{M}_1\mathbf{x}\right)}_{\mathbf{h}_2}\big)\ldots\Big)$$

  where $\mathbf{M}_\ell$ is a deterministic weight matrix and $\sigma(.)$ is an activation function

- Dropout injects stochastic noise in the feature space $\{\mathbf{x}, \mathbf{h}_2, \ldots\}$, i.e.,

$$\mathbf{h}_{\ell+1} = \sigma(\mathbf{M}_\ell(\mathbf{h}_\ell \odot \hat{\boldsymbol{\epsilon}}_\ell)) = \sigma(\mathbf{M}_\ell \text{diag}(\hat{\boldsymbol{\epsilon}}_\ell)\mathbf{h}_\ell) = \sigma(\widehat{\mathbf{W}}_\ell \mathbf{h}_\ell)$$

  with $\boldsymbol{\epsilon}_\ell \sim \text{Ber}(p_\ell), \ell = 1, \ldots, L$ and $\mathbf{W}_{\ell,k} = \epsilon_\ell \mathbf{M}_{\ell,k}$

$$\underset{\mathbf{M}_1,\ldots,\mathbf{M}_L}{\text{argmin}} \quad \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_i - f^{\widehat{\mathbf{W}}_1,\ldots,\widehat{\mathbf{W}}_L}(\mathbf{x}_i)\right\|^2 + \sum_{i=1}^{L}\lambda_i\|\mathbf{M}_i\|^2$$

- The output of a feed-forward NN (without dropout) can be written as

$$f^{\mathbf{M}_1,\ldots,\mathbf{M}_L}(\mathbf{x}) = \sigma\Big(\ldots\sigma\big(\mathbf{M}_2\underbrace{\sigma\left(\mathbf{M}_1\mathbf{x}\right)}_{\mathbf{h}_2}\big)\ldots\Big)$$

  where $\mathbf{M}_\ell$ is a deterministic weight matrix and $\sigma(.)$ is an activation function

- Dropout injects stochastic noise in the feature space $\{\mathbf{x}, \mathbf{h}_2, \ldots\}$, i.e.,

$$\mathbf{h}_{\ell+1} = \sigma(\mathbf{M}_\ell(\mathbf{h}_\ell \odot \hat{\boldsymbol{\epsilon}}_\ell)) = \sigma(\mathbf{M}_\ell \text{diag}(\hat{\boldsymbol{\epsilon}}_\ell)\mathbf{h}_\ell) = \sigma(\widehat{\mathbf{W}}_\ell \mathbf{h}_\ell)$$

  with $\boldsymbol{\epsilon}_\ell \sim \text{Ber}(p_\ell), \ell = 1, \ldots, L$ and $\mathbf{W}_{\ell,k} = \epsilon_\ell \mathbf{M}_{\ell,k}$

$$\underset{\mathbf{M}_1,\ldots,\mathbf{M}_L}{\text{argmin}} \quad \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_i - f^{\widehat{\mathbf{W}}_1,\ldots,\widehat{\mathbf{W}}_L}(\mathbf{x}_i)\right\|^2 + \sum_{i=1}^{L}\lambda_i\|\mathbf{M}_i\|^2$$

- The two problems are equivalent when we pick a prior distribution $p(\boldsymbol{\omega})$ and a family of distributions $q_\theta(\boldsymbol{\omega})$ that satisfy

$$\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \;||\; p(\boldsymbol{\omega})\big) = \frac{N\tau}{2}\sum_{i=1}^{L}\lambda_i\|\mathbf{M}_i\|^2$$

- **When is it achievable?**
  - Under a prior distribution

$$p(\boldsymbol{\omega}) = \prod_{\ell=1}^{L} p(\mathbf{W}_\ell) = \prod_{\ell=1}^{L}\mathcal{N}(\mathbf{0}, \mathbf{I}/l_\ell^2),$$

    with $l_\ell^2 = \frac{2N\tau\lambda_\ell}{p_\ell}$, and
  - High dimensional random vectors $\mathbf{W}_{\ell,k}, \forall \ell, k$, which means the number of neurons at each layer should be sufficiently large

Sketch of the proof:

1. Under the above assumptions, $q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k})$ is a **mixture of two Gaussian distributions**:

   - In Dropout, $\mathbf{W}_{\ell,k} = \epsilon_\ell \mathbf{M}_{\ell,k} \iff p(\mathbf{W}_{\ell,k}|\epsilon_\ell) = \delta(\mathbf{W}_{\ell,k} - \epsilon_\ell \mathbf{M}_{\ell,k})$

   $$q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) = \sum_{\epsilon_\ell=0}^{1} p(\mathbf{W}_{\ell,k}|\epsilon_\ell)p(\epsilon_\ell) = p_\ell \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{M}_{\ell,k}, \sigma^2\mathbf{I}) + (1-p_\ell) \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{0}, \sigma^2\mathbf{I})$$

2. KL divergence between $q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k})$ and $p(\mathbf{W}_{\ell,k})$ is

   $$\mathrm{KL}\big(q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) \,||\, p(\mathbf{W}_{\ell,k})\big) = l_\ell^2 \frac{p_\ell}{2} \|\mathbf{M}_{\ell,k}\|^2 + \text{const}$$

   since the mixture components do not overlap in high dimension spaces

3. Total KL divergence is

   $$\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \,||\, p(\boldsymbol{\omega})\big) = \sum_{\ell,k} \mathrm{KL}\big(q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) \,||\, p(\mathbf{W}_{\ell,k})\big)$$

Sketch of the proof:

1. Under the above assumptions, $q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k})$ is a **mixture of two Gaussian distributions**:

   - In Dropout, $\mathbf{W}_{\ell,k} = \epsilon_\ell \mathbf{M}_{\ell,k} \iff p(\mathbf{W}_{\ell,k}|\epsilon_\ell) = \delta(\mathbf{W}_{\ell,k} - \epsilon_\ell \mathbf{M}_{\ell,k})$

   $$q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) = \sum_{\epsilon_\ell=0}^{1} p(\mathbf{W}_{\ell,k}|\epsilon_\ell) p(\epsilon_\ell) = p_\ell \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{M}_{\ell,k}, \sigma^2 \mathbf{I}) + (1 - p_\ell) \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{0}, \sigma^2 \mathbf{I})$$

2. KL divergence between $q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k})$ and $p(\mathbf{W}_{\ell,k})$ is

   $$\mathrm{KL}\big(q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) \,||\, p(\mathbf{W}_{\ell,k})\big) = l_\ell^2 \frac{p_\ell}{2} \|\mathbf{M}_{\ell,k}\|^2 + \text{const}$$

   since the mixture components do not overlap in high dimension spaces

3. Total KL divergence is

   $$\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \,||\, p(\boldsymbol{\omega})\big) = \sum_{\ell,k} \mathrm{KL}\big(q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) \,||\, p(\mathbf{W}_{\ell,k})\big)$$

Sketch of the proof:

1. Under the above assumptions, $q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k})$ is a **mixture of two Gaussian distributions**:

   - In Dropout, $\mathbf{W}_{\ell,k} = \epsilon_\ell \mathbf{M}_{\ell,k} \iff p(\mathbf{W}_{\ell,k}|\epsilon_\ell) = \delta(\mathbf{W}_{\ell,k} - \epsilon_\ell \mathbf{M}_{\ell,k})$

   $$q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) = \sum_{\epsilon_\ell=0}^{1} p(\mathbf{W}_{\ell,k}|\epsilon_\ell) p(\epsilon_\ell) = p_\ell \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{M}_{\ell,k}, \sigma^2 \mathbf{I}) + (1 - p_\ell) \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{0}, \sigma^2 \mathbf{I})$$

2. KL divergence between $q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k})$ and $p(\mathbf{W}_{\ell,k})$ is

   $$\mathrm{KL}\big(q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) \,||\, p(\mathbf{W}_{\ell,k})\big) = l_\ell^2 \frac{p_\ell}{2} \|\mathbf{M}_{\ell,k}\|^2 + \text{const}$$

   since the mixture components do not overlap in high dimension spaces

3. Total KL divergence is

   $$\mathrm{KL}\big(q_\theta(\boldsymbol{\omega}) \,||\, p(\boldsymbol{\omega})\big) = \sum_{\ell,k} \mathrm{KL}\big(q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) \,||\, p(\mathbf{W}_{\ell,k})\big)$$

- Gal and his co-authors showed that a standard NN trained with dropout is equivalent to a Bayesian NN.
- A standard NN learns a model that minimizes a loss function, while BNNs learn a distribution over the models that maximizes an expected loglikelihood (plus regularization terms)
- Under the above conditions, the optimal weights $\mathbf{M}_\ell$ in a standard NN = the optimal parameter $\boldsymbol{\theta}$ in a Bayesian NN

Uncertainty Estimates

- Recall that the predictive distribution is

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*))p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})d\boldsymbol{\omega}$$

$$\text{replaced with} \quad q_\theta(\mathbf{y}_*|\mathbf{x}_*) = \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*))q_\theta(\boldsymbol{\omega})d\boldsymbol{\omega},$$

with $p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*)) = \mathcal{N}(\mathbf{y}_*; f^{\boldsymbol{\omega}}(\mathbf{x}_*), \tau^{-1}\mathbf{I})$

- **Predictive Mean**:

$$\mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*)}[\mathbf{y}_*] = \int \mathbf{y}_* q_\theta(\mathbf{y}_*|\mathbf{x}_*)d\mathbf{y}_* = \int \left(\int \mathbf{y}_* p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*))d\mathbf{y}_*\right) q_\theta(\boldsymbol{\omega})d\boldsymbol{\omega}$$

$$= \int f^{\boldsymbol{\omega}}(\mathbf{x}_*)q_\theta(\boldsymbol{\omega})d\boldsymbol{\omega} \qquad \text{(approximated by Monte-Carlo integration)}$$

- In testing, sample T realization of the model (using dropout) to have $\widetilde{\mathbf{E}}[\mathbf{y}_*] = \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)$

- Recall that the predictive distribution is

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*))p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})d\boldsymbol{\omega}$$

$$\text{replaced with} \quad q_\theta(\mathbf{y}_*|\mathbf{x}_*) = \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*))q_\theta(\boldsymbol{\omega})d\boldsymbol{\omega},$$

with $p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*)) = \mathcal{N}(\mathbf{y}_*; f^{\boldsymbol{\omega}}(\mathbf{x}_*), \tau^{-1}\mathbf{I})$

- **Predictive Mean**:

$$\mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*)}[\mathbf{y}_*] = \int \mathbf{y}_* q_\theta(\mathbf{y}_*|\mathbf{x}_*)d\mathbf{y}_* = \int \left(\int \mathbf{y}_* p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*))d\mathbf{y}_* \right) q_\theta(\boldsymbol{\omega})d\boldsymbol{\omega}$$

$$= \int f^{\boldsymbol{\omega}}(\mathbf{x}_*)q_\theta(\boldsymbol{\omega})d\boldsymbol{\omega} \qquad \text{(approximated by Monte-Carlo integration)}$$

- In testing, sample T realization of the model (using dropout) to have $\widetilde{\mathbf{E}}[\mathbf{y}_*] = \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)$

- **Predictive variance** as a measure of uncertainty

$$\widetilde{\text{Var}}[\mathbf{y}_*] = \underbrace{\tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*) f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)^\top}_{\text{second moment}} - \widetilde{\mathbf{E}}[\mathbf{y}_*]\widetilde{\mathbf{E}}[\mathbf{y}_*]^\top$$

- **Predictive log likelihood** as a measure of uncertainty

$$\log q_\theta(\mathbf{y}_*|\mathbf{x}_*) = \log \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*)) q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega}$$

$$\implies \widetilde{\log} q_\theta(\mathbf{y}_*|\mathbf{x}_*) = \log\left(\frac{1}{T}\sum_{t=1}^{T} p(\mathbf{y}_*|f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*))\right)$$

$$= \text{logsumexp}\left(\frac{-\tau}{2}\left\|\mathbf{y}_* - f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)\right\|^2\right) + \frac{1}{2}\log\tau + \text{const}$$

  - High uncertainty = low τ = high penalty from the second term
  - Over-confident model (high τ) with poor mean estimation = high penalty from the first term

- **Predictive variance** as a measure of uncertainty

$$\widetilde{\text{Var}}[\mathbf{y}_*] = \underbrace{\tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*) f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)^\top}_{\text{second moment}} - \widetilde{\mathbf{E}}[\mathbf{y}_*]\widetilde{\mathbf{E}}[\mathbf{y}_*]^\top$$

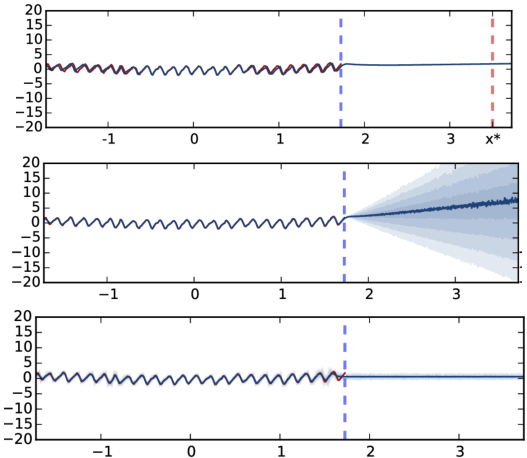
- **Predictive log likelihood** as a measure of uncertainty

$$\log q_\theta(\mathbf{y}_*|\mathbf{x}_*) = \log \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*)) q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega}$$

$$\implies \widetilde{\log} q_\theta(\mathbf{y}_*|\mathbf{x}_*) = \log\left(\frac{1}{T}\sum_{t=1}^{T} p(\mathbf{y}_*|f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*))\right)$$

$$= \text{logsumexp}\left(\frac{-\tau}{2}\left\|\mathbf{y}_* - f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)\right\|^2\right) + \frac{1}{2}\log\tau + \text{const}$$

  - High uncertainty = low τ = high penalty from the second term
  - Over-confident model (high τ) with poor mean estimation = high penalty from the first term



CO2 concentrations dataset. (Top) Standard Dropout, (Middle) MC dropout with Relu, and (Bottom) with Tanh. Different shades of blue represent half a standard deviation.

- Add a softmax layer, $\text{Softmax}(f^{\omega}(\mathbf{x}))$, to predict the likelihood of each class
- Sample T realizations of the model and let the prediction be

$$c^* = \underset{c=1,\dots,C}{\text{argmax}} \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t = c]$$

- **Predictive Entropy** as an uncertainty estimate

$$\mathbb{H}[y_*|\mathbf{x}_*] = -\sum_c q_\theta(y_* = c|\mathbf{x}_*) \log\left(q_\theta(y_* = c|\mathbf{x}_*)\right)$$

where we approximate the predictive distribution with

$$q_\theta(y_* = c|\mathbf{x}_*) = \int \underbrace{p(y_* = c|\mathbf{x}_*, \boldsymbol{\omega})}_{\text{Softmax outputs}} q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega} \approx \frac{1}{T} \sum_{t=1}^{T} p(y_* = c|\mathbf{x}_*, \hat{\boldsymbol{\omega}}_t)$$

- $\exists c, q_\theta(y_* = c|\mathbf{x}_*) = 1 \implies \mathbb{H}[y_*|\mathbf{x}_*] \downarrow$, and $q_\theta(y_* = c|\mathbf{x}_*) \sim \text{Unif} \implies \mathbb{H}[y_*|\mathbf{x}_*] \uparrow$

- Add a softmax layer, $\text{Softmax}(f^{\omega}(\mathbf{x}))$, to predict the likelihood of each class
- Sample T realizations of the model and let the prediction be

$$c^* = \underset{c=1,\dots,C}{\text{argmax}} \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t = c]$$

- **Predictive Entropy** as an uncertainty estimate

$$\mathbb{H}[y_*|\mathbf{x}_*] = -\sum_c q_\theta(y_* = c|\mathbf{x}_*) \log\left(q_\theta(y_* = c|\mathbf{x}_*)\right)$$

where we approximate the predictive distribution with

$$q_\theta(y_* = c|\mathbf{x}_*) = \int \underbrace{p(y_* = c|\mathbf{x}_*, \boldsymbol{\omega})}_{\text{Softmax outputs}} q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega} \approx \frac{1}{T} \sum_{t=1}^{T} p(y_* = c|\mathbf{x}_*, \hat{\boldsymbol{\omega}}_t)$$

- $\exists c, q_\theta(y_* = c|\mathbf{x}_*) = 1 \implies \mathbb{H}[y_*|\mathbf{x}_*] \downarrow$, and $q_\theta(y_* = c|\mathbf{x}_*) \sim \text{Unif} \implies \mathbb{H}[y_*|\mathbf{x}_*] \uparrow$

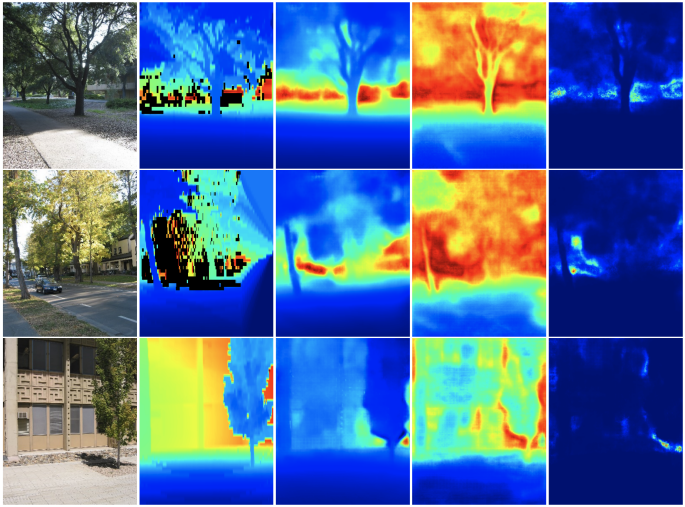
- In order to let BNNs learn aleatoric uncertainty, we parameterize τ as $\tau^{\boldsymbol{\omega}}(\mathbf{x})$

$$\mathcal{N}(\mathbf{y}; f^{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I}) \implies \mathcal{N}(\mathbf{y}; f^{\boldsymbol{\omega}}(\mathbf{x}), \tau^{\boldsymbol{\omega}}(\mathbf{x})^{-1})$$

- The goal is to learn distributions over the weights used for both $f^{\boldsymbol{\omega}}(\mathbf{x})$ and $\tau^{\boldsymbol{\omega}}(\mathbf{x})$ following the same framework

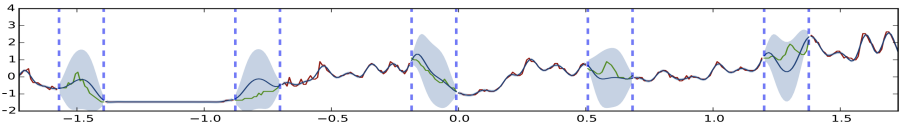
- Predictive variance is calculated as

$$\widetilde{\text{Var}}[\mathbf{y}_*] = \frac{1}{T}\sum_{t=1}^{T} \tau^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x})\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*) f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}_*)^\top - \widetilde{\mathbf{E}}[\mathbf{y}_*]\widetilde{\mathbf{E}}[\mathbf{y}_*]^\top$$

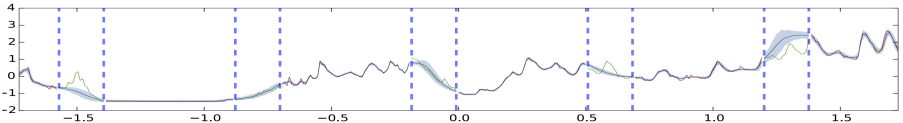


Left to Right: input image, ground truth, **depth prediction**, aleatoric uncertainty, epistemic uncertainty.
Make3D does not provide labels for depth greater than 70m

Final Remarks



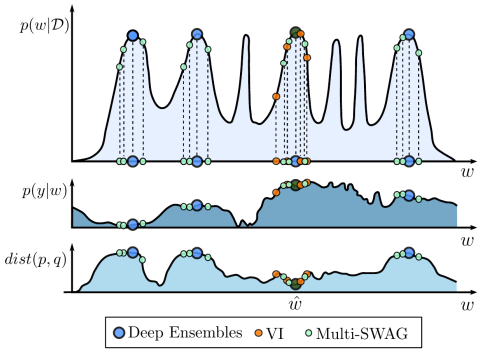
(a) Gaussian process with SE covariance function



(b) MC dropout with ReLU non-linearities

Predictive mean and uncertainties on the reconstructed solar irradiance dataset with missing segments, for the GP and MC dropout approximation. In red is the observed function and in green are the missing segments. In blue is the predictive mean plus/minus two standard deviations of the various approximations.

- Every time we train a standard NN, we reach **a** local minimum $\implies$ **multimodal posterior**



- MC dropout:

$$q_{\theta_{\ell,k}}(\mathbf{W}_{\ell,k}) = p_\ell \, \mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{M}_{\ell,k}, \sigma^2\mathbf{I}) + (1-p_\ell) \underbrace{\mathcal{N}(\mathbf{W}_{\ell,k}; \mathbf{0}, \sigma^2\mathbf{I})}_{?}$$

Large gap in computing the predictive distribution

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}_*|f^{\boldsymbol{\omega}}(\mathbf{x}_*)) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) d\boldsymbol{\omega}$$

- Deep Ensembles
  Check Rahul's presentation for more details!