Bayesian Deep Learning

Yarin Gal

Research Fellow, University of Cambridge
Research Fellow, The Alan Turing Institute
yg279@cam.ac.uk

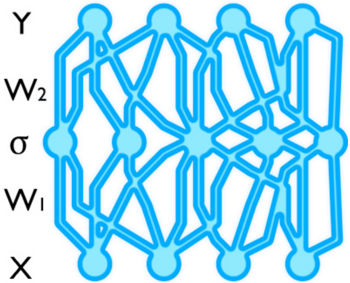
Pillar I: Deep learning

Conceptually simple models

Data: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$
Model: given matrices $\mathbf{W}$ and non-linear func. $\sigma(\cdot)$, define "network"

$$\tilde{\mathbf{y}}_i(\mathbf{x}_i) = \mathbf{W}_2 \cdot \sigma\big(\mathbf{W}_1 \mathbf{x}_i\big)$$

Objective: find $\mathbf{W}$ for which $\tilde{\mathbf{y}}_i(\mathbf{x}_i)$ is close to $\mathbf{y}_i$ for all $i \leq N$.



Conceptually simple models

Data: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$
Model: given matrices $\mathbf{W}$ and non-linear func. $\sigma(\cdot)$, define "network"

$$\tilde{\mathbf{y}}_i(\mathbf{x}_i) = \mathbf{W}_2 \cdot \sigma\big(\mathbf{W}_1 \mathbf{x}_i\big)$$

Objective: find $\mathbf{W}$ for which $\tilde{\mathbf{y}}_i(\mathbf{x}_i)$ is close to $\mathbf{y}_i$ for all $i \leq N$.

Deep learning is awesome ✓

- Simple and modular
- Huge attention from practitioners and engineers
- Great software tools
- Scales with data and compute
- Real-world impact

... but has many issues ✗

- What does a model not know?
- Uninterpretable black-boxes
- Easily fooled (AI safety)
- Lacks solid mathematical foundations (mostly ad hoc)
- Crucially relies on big data

Pillar I: Deep learning

UNIVERSITY OF CAMBRIDGE

Conceptually simple models

Data: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$
Model: given matrices $\mathbf{W}$ and non-linear func. $\sigma(\cdot)$, define "network"

$$\tilde{\mathbf{y}}_i(\mathbf{x}_i) = \mathbf{W}_2 \cdot \sigma\big(\mathbf{W}_1 \mathbf{x}_i\big)$$

Objective: find $\mathbf{W}$ for which $\tilde{\mathbf{y}}_i(\mathbf{x}_i)$ is close to $\mathbf{y}_i$ for all $i \leq N$.

Deep learning is awesome ✓

- Simple and modular
- Huge attention from practitioners and engineers
- Great software tools
- Scales with data and compute
- Real-world impact

… but has many issues ✗

- What does a model not know?
- Uninterpretable black-boxes
- Easily fooled (AI safety)
- Lacks solid mathematical foundations (mostly ad hoc)
- Crucially relies on big data

No uncertainty!

UNIVERSITY OF CAMBRIDGE

- We need a way to tell **what our model knows** and what not.
  - We train a model to recognise dog breeds

Why should I care about uncertainty?

- We need a way to tell **what our model knows** and what not.
  - We train a model to recognise dog breeds
  - And are given a cat to classify

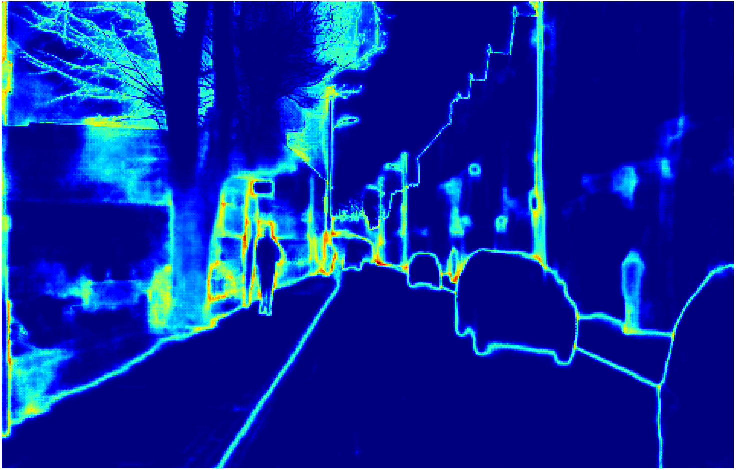
Why should I care about uncertainty?

- We need a way to tell **what our model knows** and what not.
  - We train a model to recognise dog breeds
  - And are given a cat to classify
  - What would you want your model to do?

- We need a way to tell **what our model knows** and what not.
  - We train a model to recognise dog breeds
  - And are given a cat to classify
  - What would you want your model to do?
  - Similar problems in *decision making*, *physics*, *life science*, etc.

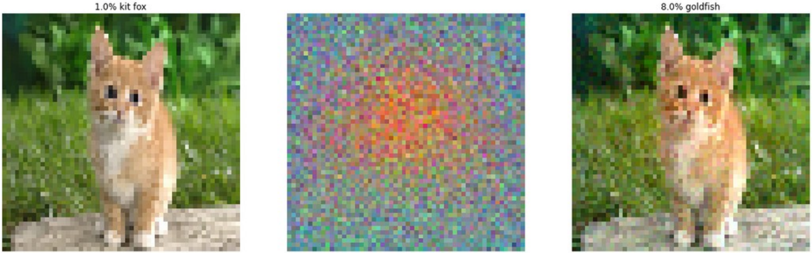
- We need a way to tell **what our model knows** and what not.

- Uncertainty gives insights into the black-box when it fails —where am I not certain?



Why should I care about uncertainty?

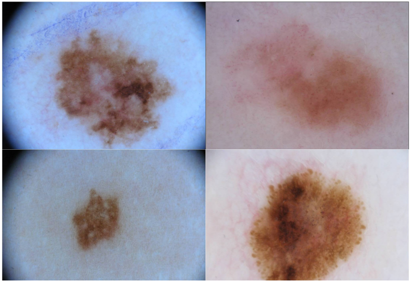
- We need a way to tell **what our model knows** and what not.
- Uncertainty gives insights into the black-box when it fails —where am I not certain?
- Uncertainty might even be useful to identify when attacked with adversarial examples!



- Lastly, need less data if label only where **model is uncertain**: wear-and-tear in robotics, expert time in medical analysis

Why should I care about uncertainty?

- We need a way to tell **what our model knows** and what not.
- Uncertainty gives insights into the black-box when it fails —where am I not certain?
- Uncertainty might even be useful to identify when attacked with adversarial examples!
- Lastly, need less data if label only where **model is uncertain**: wear-and-tear in robotics, expert time in medical analysis

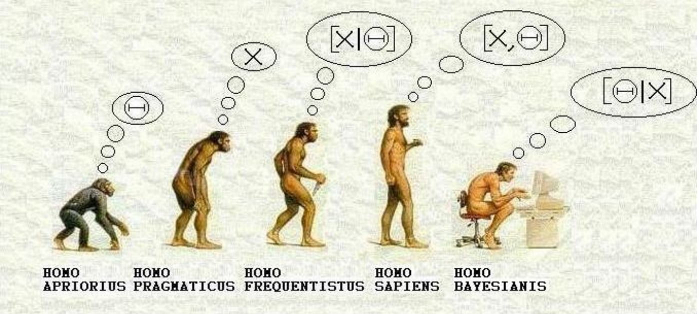
Pillar II: Bayes

The language of uncertainty

- Probability theory
- Specifically *Bayesian probability theory* (1750!)

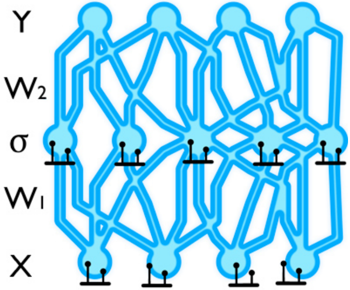
When applied to *Information Engineering*...

- Bayesian modelling



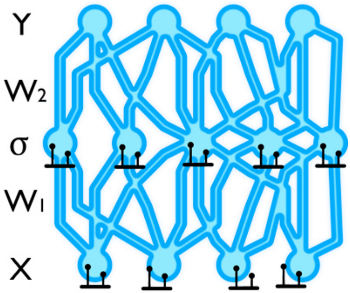
- Built on solid mathematical foundations
- Orthogonal to deep learning...

- “Dropout”: a popular method in deep learning, cited hundreds and hundreds of times
- Works by randomly setting network units to zero
- This **somehow** improves performance and reduces over-fitting
- Used in almost **all** modern deep learning models

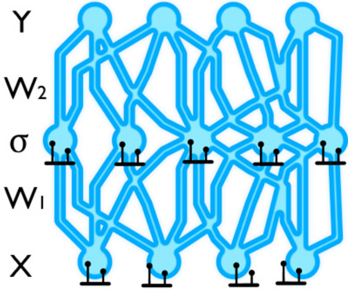


A simple way to tie the two pillars together

- "Dropout": a popular method in deep learning, cited hundreds and hundreds of times
- Works by randomly setting network units to zero
- This **somehow** improves performance and reduces over-fitting
- Used in almost **all** modern deep learning models

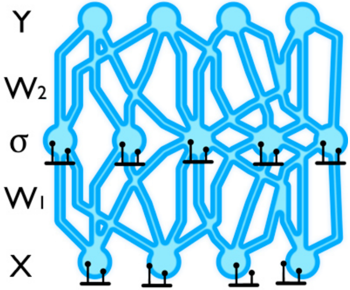


- "Dropout": a popular method in deep learning, cited hundreds and hundreds of times
- Works by randomly setting network units to zero
- This **somehow** improves performance and reduces over-fitting
- Used in almost **all** modern deep learning models



A simple way to tie the two pillars together

- "Dropout": a popular method in deep learning, cited hundreds and hundreds of times
- Works by randomly setting network units to zero
- This **somehow** improves performance and reduces over-fitting
- Used in almost **all** modern deep learning models

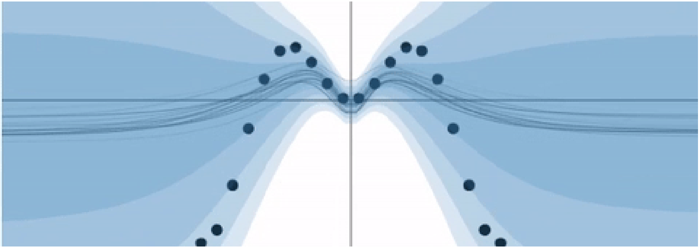


A simple way to tie the two pillars together

- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],

- Connecting **Deep Learning to Bayesian probability theory**.

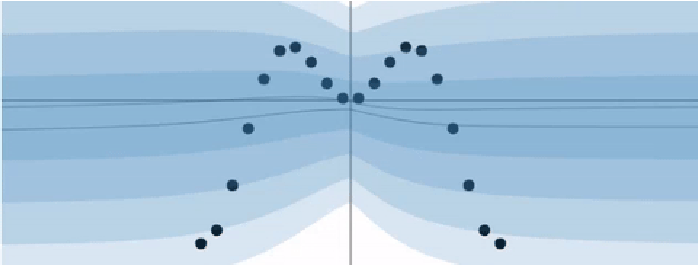
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains



A simple way to tie the two pillars together

UNIVERSITY OF CAMBRIDGE

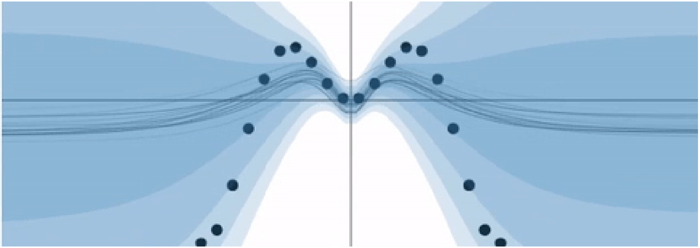
- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],
- Connecting **Deep Learning to Bayesian probability theory**.
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains



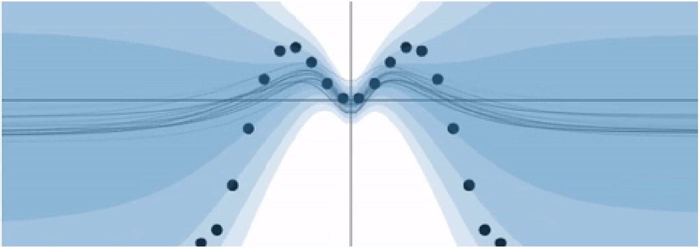
A simple way to tie the two pillars together

UNIVERSITY OF CAMBRIDGE

- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],
- Connecting **Deep Learning to Bayesian probability theory**.
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains

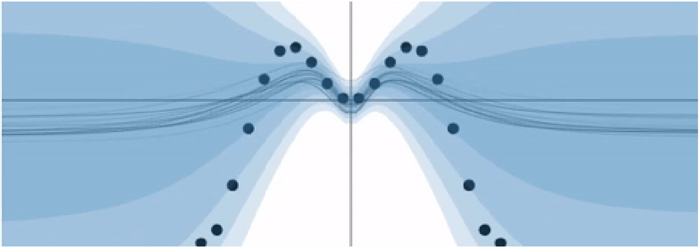


- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],
- Connecting **Deep Learning to Bayesian probability theory**.
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains



A simple way to tie the two pillars together

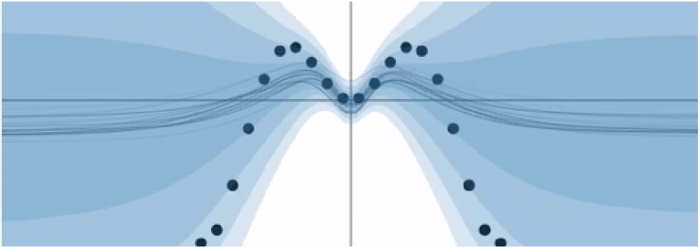
- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],
- Connecting **Deep Learning to Bayesian probability theory**.
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains



A simple way to tie the two pillars together

UNIVERSITY OF CAMBRIDGE

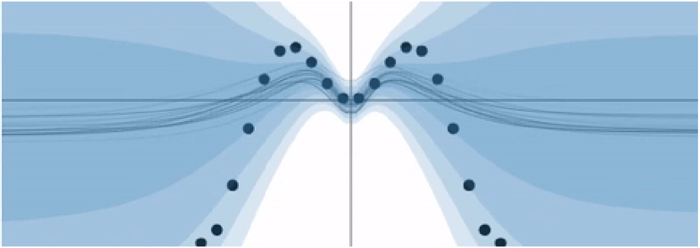
- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],
- Connecting **Deep Learning to Bayesian probability theory**.
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains



A simple way to tie the two pillars together

UNIVERSITY OF CAMBRIDGE

- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],
- Connecting **Deep Learning to Bayesian probability theory**.
- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains

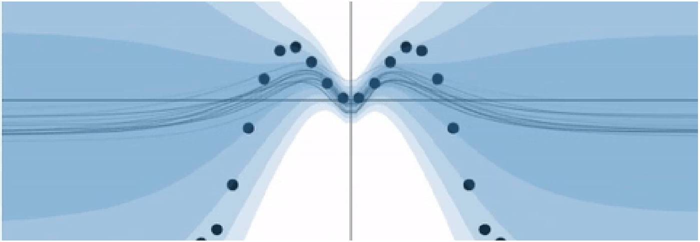


- Can be shown that dropout training is identical to *approximate inference in Bayesian modelling* [Gal, 2016],

- Connecting **Deep Learning to Bayesian probability theory**.

- The **mathematically grounded** connection gives a treasure trove of new research opportunities:
  - **uncertainty** in deep learning, e.g. interpretability and AI safety
  - **principled extensions** to deep learning
  - enable deep learning in **small data** domains

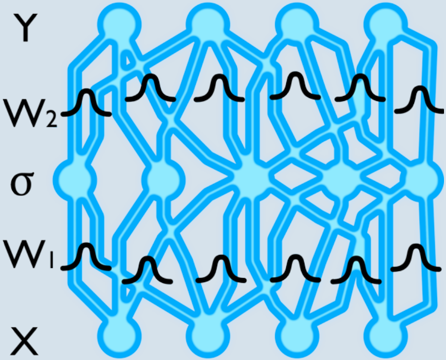
More in a second. First, some **theory**.



UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s



- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p\left(\mathbf{W}|\mathbf{X}, \mathbf{Y}\right)$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate
$$q_{\mathbf{M}}(\mathbf{W}) \approx p\left(\mathbf{W}|\mathbf{X}, \mathbf{Y}\right)$$
- This is called **approximate variational inference**.

From Bayesian neural networks to Dropout

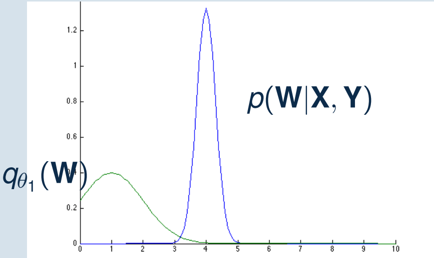
- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big)$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate
$$q_{\mathbf{M}}(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big)$$
- This is called **approximate variational inference**.

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate

$$q_{\mathbf{M}}(\mathbf{W}) \approx p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$$

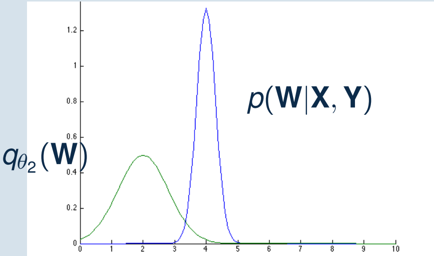


UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate

$$q_{\mathbf{M}}(\mathbf{W}) \approx p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$$

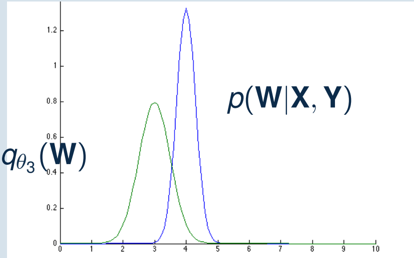


UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate

$$q_{\mathbf{M}}(\mathbf{W}) \approx p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$$

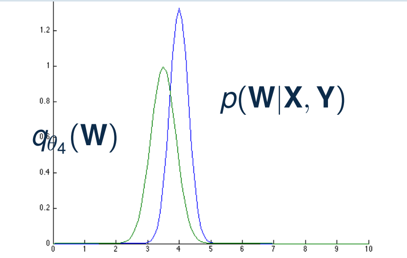


UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate

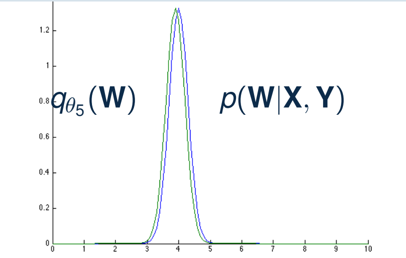
$$q_{\mathbf{M}}(\mathbf{W}) \approx p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$$



From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate

$$q_{\mathbf{M}}(\mathbf{W}) \approx p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$$



Some theory

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big)$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_{\mathbf{M}}(\cdot)$ and approximate
$$q_{\mathbf{M}}(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big)$$
- This is called **approximate variational inference**.

UNIVERSITY OF CAMBRIDGE

Theorem (Dropout as approximate variational inference)

Define

$$q_{\mathbf{M}}(\mathbf{W}) := \mathbf{M} \cdot \mathit{diag}(\mathit{Bernoulli})$$

with variational parameter $\mathbf{M}$.
The optimisation objective of (stochastic) variational inference with $q_{\mathbf{M}}(\mathbf{W})$ *is identical to the objective of a dropout neural network.*

Proof.

See Gal [2016]. □

UNIVERSITY OF CAMBRIDGE

Theorem (Dropout as approximate variational inference)

Define

$$q_{\mathbf{M}}(\mathbf{W}) := \mathbf{M} \cdot diag(Bernoulli)$$

with variational parameter $\mathbf{M}$.
The optimisation objective of (stochastic) variational inference with $q_{\mathbf{M}}(\mathbf{W})$ *is identical to the objective of a dropout neural network.*

Proof.

See Gal [2016]. □

Implementing **inference** with $q_{\mathbf{M}}(\mathbf{W})$

$=$

Implementing **dropout training**.

Line to line.

UNIVERSITY OF CAMBRIDGE

Theorem (Dropout as approximate variational inference)

Define

$$q_{\mathbf{M}}(\mathbf{W}) := \mathbf{M} \cdot diag(Bernoulli)$$

with variational parameter $\mathbf{M}$.
The optimisation objective of (stochastic) variational inference with $q_{\mathbf{M}}(\mathbf{W})$ *is identical to the objective of a dropout neural network.*

Corollary (Model uncertainty with dropout)

Given $p(\mathbf{y}^*|\mathbf{f}^{\mathbf{W}}(\mathbf{x}^*)) = \mathcal{N}(\mathbf{y}^*; \mathbf{f}^{\mathbf{W}}(\mathbf{x}^*), \tau^{-1}\mathbf{I})$ *for some* $\tau > 0$, *the model's predictive variance can be estimated with the unbiased estimator:*

$$\widetilde{Var}[\mathbf{y}^*] := \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} \mathbf{f}^{\widehat{\mathbf{W}}_t}(\mathbf{x}^*)^T \mathbf{f}^{\widehat{\mathbf{W}}_t}(\mathbf{x}^*) - \widetilde{\mathbb{E}}[\mathbf{y}^*]^T \widetilde{\mathbb{E}}[\mathbf{y}^*]$$

with $\widehat{\mathbf{W}}_t \sim q^*_{\mathbf{M}}(\mathbf{W})$.

Some code, just for fun

In practical terms[1], given point x:

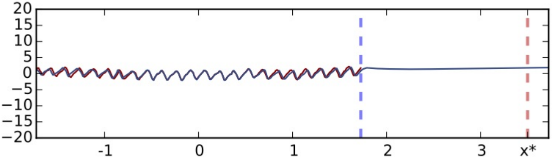
- drop units **at test time**
- **repeat 10 times**
- and look at **mean and sample variance**.
- Or in Python:

```
y = []
for _ in xrange(10):
    y.append(model.output(x, dropout=True))
y_mean = numpy.mean(y)
y_var = numpy.var(y)
```

[1] Friendly introduction given in `yarin.co/blog`

What would be the CO_2 concentration level in Mauna Loa, Hawaii, in 20 years' time?

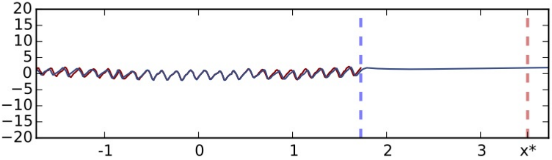
- Normal dropout:



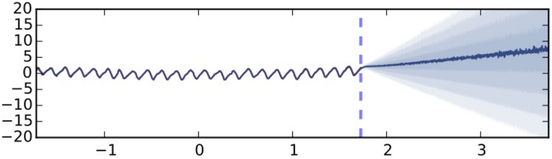
- Same network, Bayesian perspective:

What would be the CO_2 concentration level in Mauna Loa, Hawaii, in 20 years' time?

- Normal dropout:

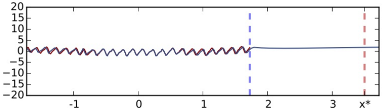


- Same network, Bayesian perspective:

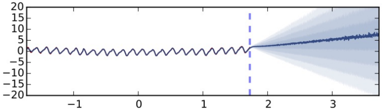


What would be the CO_2 concentration level in Mauna Loa, Hawaii, in 20 years' time?

Normal dropout:



Bayesian perspective:



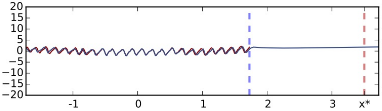
What can we do with this?

- Interpretability & AI safety
- Principled deep learning extensions
- Deep learning in small data domains

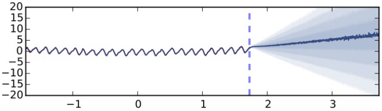
Example uncertainty in deep learning

What would be the CO_2 concentration level in Mauna Loa, Hawaii, in 20 years' time?

Normal dropout:



Bayesian perspective:

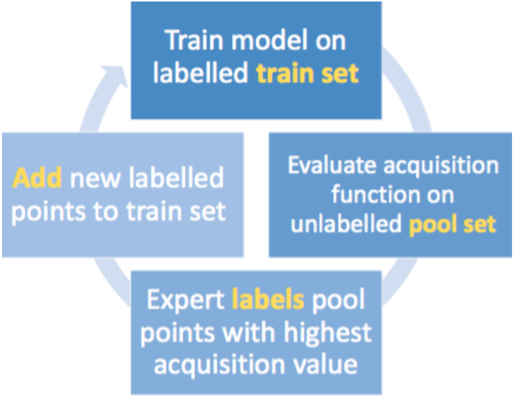


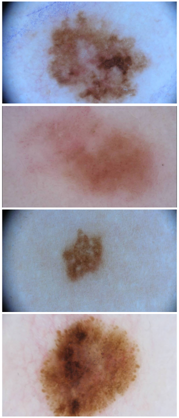
What can we do with this?

- Interpretability & AI safety
- Principled deep learning extensions
- **Deep learning in small data domains**
  - **Cancer diagnosis**

UNIVERSITY OF CAMBRIDGE

Active learning of images [Gal, Islam & Ghahramani, 2017]
E.g. diagnose melanoma with a handful of images.





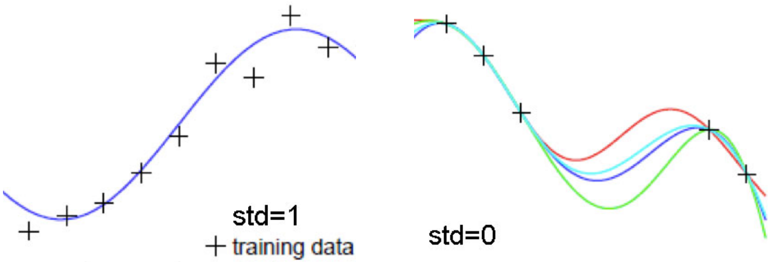
Active Learning acquisition functions

Choose x^* that maximises **acquisition functions** $a(\mathbf{x})$:

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} a(\mathbf{x})$$

E.g. points that maximise uncertainty. But, **which uncertainty?**

- *Aleatoric uncertainty* captures noise inherent in the data
- *Epistemic uncertainty* captures model's lack of knowledge
- *Predictive uncertainty* captures the sum of the two



Figures adapted from Hanna M. Wallach (Cambridge, UMassAmherst)

Acquisition functions for classification

Choose x^* that maximises **acquisition functions** $a(\mathbf{x})$:

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} a(\mathbf{x})$$

Possible **measures of uncertainty** in classification:

- Predictive entropy ($\mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}]$)

$$a_{\text{PE}}(\mathbf{x}) = -\sum_c p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}) \log p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}})$$

- Information gained about the model parameters ($\mathbb{I}[y, \mathbf{W}|\mathbf{x}, \mathcal{D}_{\text{train}}]$)

$$a_{\text{MI}}(\mathbf{x}) = \mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] - \mathbb{E}_{p(\mathbf{W}|\mathcal{D}_{\text{train}})}\big[\mathbb{H}[y|\mathbf{x}, \mathbf{W}]\big]$$

- *Variation ratios*

$$a_{\text{VR}}(\mathbf{x}) = 1 - \max_y p(y|\mathbf{x}, \mathcal{D}_{\text{train}})$$

- *Random* acquisition (baseline): $a_{\text{U}}(\mathbf{x}) = \text{unif}()$

Acquisition functions intuition

Want to classify dogs vs. cats given image **x** with models $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$

- Stochastic forward passes give **probability vectors** for each model:
  1. $(1, 0), \ldots, (1, 0)$
  2. $(0.5, 0.5), \ldots, (0.5, 0.5)$, and
  3. $(1, 0), (0, 1), (1, 0), \ldots, (0, 1)$

Acquisition functions intuition

Want to classify dogs vs. cats given image **x** with models $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$

- Stochastic forward passes give **probability vectors** for each model:
  1. $(1, 0), ..., (1, 0)$
  2. $(0.5, 0.5), ..., (0.5, 0.5)$, and
  3. $(1, 0), (0, 1), (1, 0), ..., (0, 1)$

What's the epistemic uncertainty for each model?
What's the predictive uncertainty for each model?

Acquisition functions intuition

Want to classify dogs vs. cats given image **x** with models $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$

- Stochastic forward passes give **probability vectors** for each model:
    1. $(1, 0), ..., (1, 0)$
    2. $(0.5, 0.5), ..., (0.5, 0.5)$, and
    3. $(1, 0), (0, 1), (1, 0), ..., (0, 1)$

What's the epistemic uncertainty? models $\mathcal{M}_1$ and $\mathcal{M}_2$ are confident about the output. Model $\mathcal{M}_3$ is uncertain.
What's the predictive uncertainty? $\mathcal{M}_1$ has low uncertainty, $\mathcal{M}_2$ and $\mathcal{M}_3$ have high uncertainty.

Acquisition functions intuition

Want to classify dogs vs. cats given image $\mathbf{x}$ with models $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$

- Stochastic forward passes give **probability vectors** for each model:
  1. $(1, 0), ..., (1, 0)$
  2. $(0.5, 0.5), ..., (0.5, 0.5)$, and
  3. $(1, 0), (0, 1), (1, 0), ..., (0, 1)$

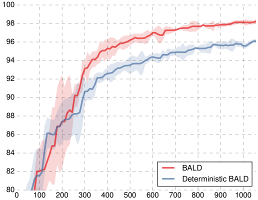
What's the epistemic uncertainty? models $\mathcal{M}_1$ and $\mathcal{M}_2$ are confident about the output. Model $\mathcal{M}_3$ is uncertain.
What's the predictive uncertainty? $\mathcal{M}_1$ has low uncertainty, $\mathcal{M}_2$ and $\mathcal{M}_3$ have high uncertainty.

Acquisition functions intuition:

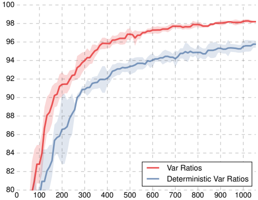
- $\mathcal{M}_1$: all acquisition functions give **low** uncertainty
- $\mathcal{M}_2$: variation ratios and predictive entropy give **high** uncertainty; mutual information gives **low** uncertainty.
- $\mathcal{M}_3$: all acquisition functions give **high** uncertainty

MNIST experiments

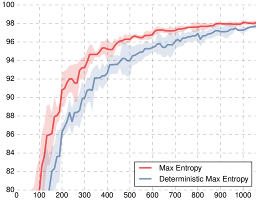
Test accuracy as a function of number of acquired images (up to 1K):



BALD



Var Ratios



Max Entropy

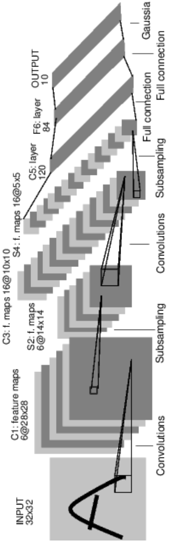
using both a **Bayesian CNN (red) and a deterministic CNN (blue)**

Number of acquired images **to get to model error of %**:

% error	BALD	Var Ratios	Max Ent	Random
10%	145	120	165	255
5%	335	295	355	835

Test error on MNIST with 1000 labelled training samples, for active learning (using simple LeNet) vs. **semi-supervised techniques**:

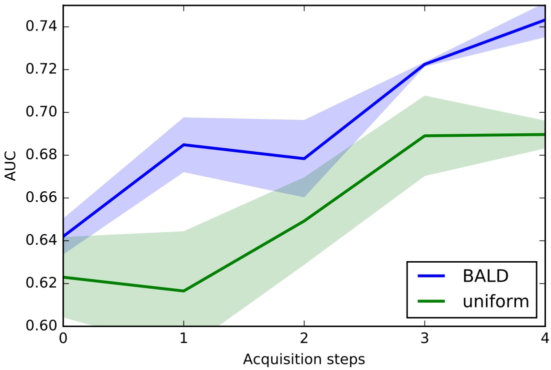
Technique	Test error
Semi-supervised:	
SS Embedding (Weston et al., 2012)	5.73%
DGN (Kingma et al., 2014)	2.40%
Γ Ladder Network (Rasmus et al., 2015)	1.53%
Virtual Adversarial (Miyato et al., 2015)	1.32%
Active learning with various acquisitions:	
Random	4.66%
BALD	1.80%
Max Entropy	1.74%
Var Ratios	1.64%



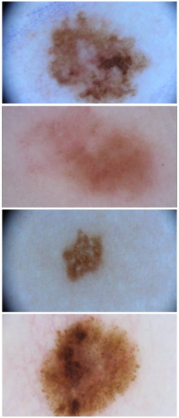
Medical analysis with Active Learning

Active learning of images [Gal, Islam & Ghahramani, 2017]

E.g. diagnose melanoma with a handful of images:

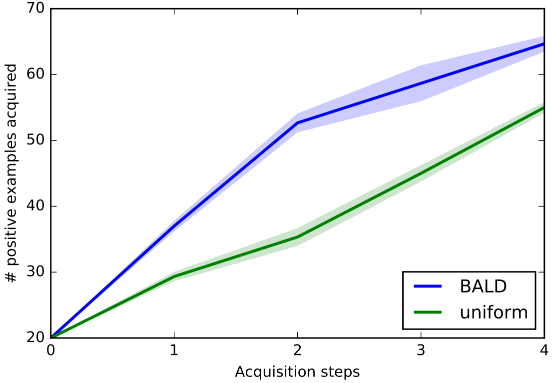


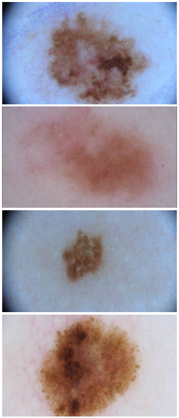
Performance vs. acquisition



Active learning of images [Gal, Islam & Ghahramani, 2017]

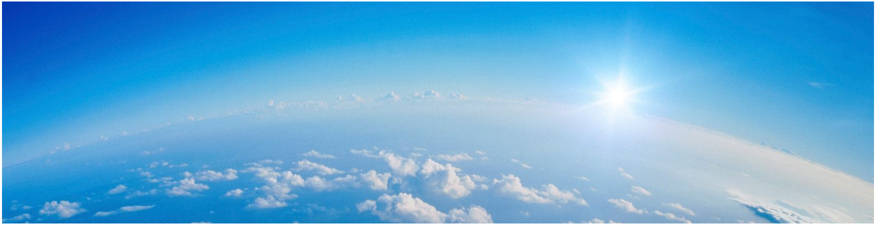
E.g. diagnose melanoma with a handful of images:





acquired positive examples vs. acquisition

New horizons

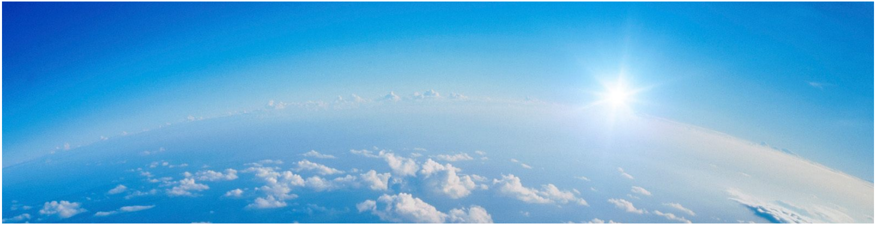


Most exciting is work to come:

- What is *interesting* data to **label**? (when model is uncertain)
- Active learning in real-world **medical applications**

and much, much, more.

New horizons



Most exciting is work to come:

- What is *interesting* data to **label**? (when model is uncertain)
- Active learning in real-world **medical applications**

and much, much, more.

Thank you for listening.

UNIVERSITY OF CAMBRIDGE

- **Y Gal, R Turner**, "Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs", ICML (2015)
- **Y Gal, Z Ghahramani**, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", ICML (2016)
- **Y Gal, Z Ghahramani**, "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks", NIPS (2016)
- **Y Gal, R McAllister, C Rasmussen**, "Improving PILCO with Bayesian Neural Network Dynamics Models", DEML workshop, ICML (2016)
- **Y Gal, R Islam, Z Ghahramani**, "Deep Bayesian Active Learning with Image Data", ICML (2017)
- **Y Li, Y Gal**, "Dropout Inference in Bayesian Neural Networks with Alpha-divergences", ICML (2017)
- **A Kendall, Y Gal**, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?", arXiv preprint, arXiv:1703.04977 (2017)
- **A Shah, Y Gal**, "Invertible Transformations for Bayesian Neural Network Inference" (2017)
- and more...