Pillar I: Deep learning

Conceptually simple models

**Data:** $X = \{x_1, x_2, \ldots, x_N\}$, $Y = \{y_1, y_2, \ldots, y_N\}$

**Model:** given matrices $W$ and non-linear func. $\sigma(\cdot)$, define “network”

$$\tilde{y}_i(x_i) = W_2 \cdot \sigma(W_1 x_i)$$

**Objective:** find $W$ for which $\tilde{y}_i(x_i)$ is close to $y_i$ for all $i \leq N$. 
# Pillar I: Deep learning

## Conceptually simple models

**Data:** $X = \{x_1, x_2, ..., x_N\}$, $Y = \{y_1, y_2, ..., y_N\}$  

**Model:** given matrices $W$ and non-linear func. $\sigma(\cdot)$, define “network”

$$\tilde{y}_i(x_i) = W_2 \cdot \sigma(W_1 x_i)$$

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

---

**Deep learning is awesome ✔️ ... but has many issues ✗**

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

[Images of various dog breeds]

- 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.
  - We train a model to recognise dog breeds
  - And are given a cat to classify

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.
  - We train a model to recognise dog breeds
  - And are given a cat to classify
  - What would you want your model to do?

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.
  - 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.
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?
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...
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

- “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

- “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

- 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 \textit{approximate inference in Bayesian modelling} [Gal, 2016],

- Connecting \textbf{Deep Learning to Bayesian probability theory}.

- The \textbf{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

- 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

- 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

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

From Bayesian neural networks to Dropout

- Place prior $p(W)$ dist. on weights, making these r.v.s

- Given dataset $X, Y$, the r.v. $W$ has a posterior: $p(W|X, Y)$
Some theory

### From Bayesian neural networks to Dropout

- Place **prior** $p(W)$ dist. on weights, making these r.v.s
- Given dataset $X, Y$, the r.v. $W$ has a **posterior**: $p(W|X, Y)$
  - Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_M(\cdot)$ and approximate
  \[ q_M(W) \approx p(W|X, Y) \]
- This is called **approximate variational inference**.
From Bayesian neural networks to Dropout

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

From Bayesian neural networks to Dropout

- Place **prior** $p(W)$ dist. on weights, making these r.v.s
- Given dataset $X, Y$, the r.v. $W$ has a **posterior**: $p(W|X, Y)$
- Which is difficult to evaluate—many great researchers tried
- Can define **simple distribution** $q_M(\cdot)$ and approximate
  $$q_M(W) \approx p(W|X, Y)$$
From Bayesian neural networks to Dropout

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

\[ q_{\theta_2}(W) \approx p(W|X, Y) \]
Some theory

From Bayesian neural networks to Dropout

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

\[ q_{\theta_3}(W) \]
\[ p(W|X, Y) \]
Some theory

From Bayesian neural networks to Dropout

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

![Graph showing approximate variational inference](image)
Some theory

From Bayesian neural networks to Dropout

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

From Bayesian neural networks to Dropout

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

Theorem (Dropout as approximate variational inference)

Define \( q_M(W) := M \cdot \text{diag}(\text{Bernoulli}) \)

with variational parameter \( M \).

The optimisation objective of (stochastic) variational inference with \( q_M(W) \) is identical to the objective of a dropout neural network.

Proof.

See Gal [2016].
Theorem (Dropout as approximate variational inference)

Define

\[ q_M(W) := M \cdot \text{diag}(\text{Bernoulli}) \]

with variational parameter \( M \).

The optimisation objective of (stochastic) variational inference with \( q_M(W) \) is identical to the objective of a dropout neural network.

Proof.

See Gal [2016].

Implementing \textit{inference} with \( q_M(W) \)

\[ = \]

Implementing \textit{dropout training}.

Line to line.
Some theory

Theorem (Dropout as approximate variational inference)

Define

\[ q_M(W) := M \cdot \text{diag}(\text{Bernoulli}) \]

with variational parameter \( M \).

The optimisation objective of (stochastic) variational inference with \( q_M(W) \) is identical to the objective of a dropout neural network.

Corollary (Model uncertainty with dropout)

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

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

with \( \widehat{W}_t \sim q^*_M(W) \).
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:

```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](http://yarin.co/blog)
What would be the CO$_2$ concentration level in Mauna Loa, Hawaii, in 20 years’ time?

Normal deep learning:  
Bayesian perspective:

What can we do with this?  
Deep learning with small data • Interpretable AI • Safe AI
Enabling Deep Learning with small data

Human-in-the-loop AI for Galaxy Zoo morphology classification

**Figure 1.** The Galaxy Zoo web interface as shown to volunteers. This screenshot shows the first question in the decision tree: is the galaxy smooth or featured?

with Lewis Smith [work done w. Chris Lintott, Zooniverse Citizen Science Project]
Interpretable AI

Bayesian deep learning for exoplanet atmospheric retrieval

with Adam Cobb [work done with NASA Goddard while at NASA FDL]
Uncertainties in computer vision

- **Aleatoric uncertainty**, capturing inherent noise in the data
- **Epistemic uncertainty**, capturing model’s lack of knowledge

with Alex Kendall
Informal settlement detection

with Tim Rudner [work done with ESA while at FDL Europe]
ML in Space (literally)

Flood detection, from space

with Lewis Smith [work done with ESA while at FDL Europe]
ML in Space (literally)

Flood detection, from space

with Lewis Smith [work done with ESA while at FDL Europe]
ML in Space (literally)

Flood detection, from space

How can we reduce the time from disaster to data?

1. How do we get data to the ground more quickly?

2. How can we accelerate/automate the image analysis process?

American Red Cross

with Lewis Smith [work done with ESA while at FDL Europe]
Flood detection, from space

Why process onboard?

560 Mbps

ESA Maspalomas, Spain

< 1 Mbps

COTS ground station (ISIS)

with Lewis Smith [work done with ESA while at FDL Europe]
ML in Space (literally)

Flood detection, from space

Project Proposal

- Demonstrate that low resolution images from CubeSats can be used to provide useful flood intelligence
- Perform flood segmentation onboard satellite to reduce downlinked data
- Deploy on ФSat-1 in 2019/2020

with Lewis Smith [work done with ESA while at FDL Europe]
Oxford Applied and Theoretical Machine Learning Group
http://oatml.ox.ac.uk
Researchers coming from academia (Oxford, Cambridge, MILA, Yale, U of Toronto, U of Amsterdam, etc) .. and industry (Google, DeepMind, Twitter, etc)