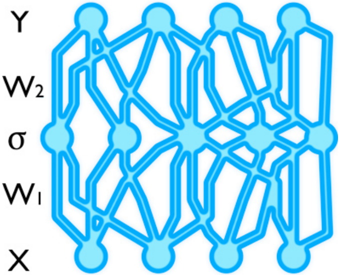
UNIVERSITY OF CAMBRIDGE

Modern Deep Learning through Bayesian Eyes

Yarin Gal

yg279@cam.ac.uk

To keep things interesting, a photo or an equation in every slide! (unless specified otherwise, photos are either original work or taken from Wikimedia, under Creative Commons license)



Conceptually simple models...

- Attracts **tremendous attention** from popular media,
- **Fundamentally affected** the way ML is used in industry,
- Driven by **pragmatic** developments...
- of **tractable** models...
- that **work** well...
- and **scale** well.

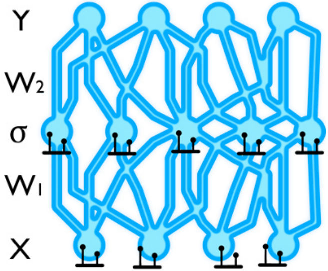
But many unanswered questions...

UNIVERSITY OF CAMBRIDGE

- **Why** does my model work

We don't understand many of the tools that we use...

E.g. stochastic reg. techniques (*dropout*, MGN[1]) are used in most deep learning models to avoid over-fitting. Why do they work?



- **What** does my model know?
- **Why** does my model predict this and not that?

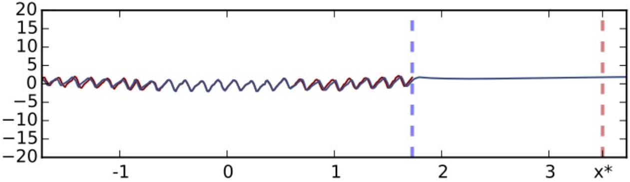
[1] Wager et al. (2013) and Baldi and Sadowski (2013) attempt to explain dropout as sparse regularisation but cannot generalise to other techniques.

But many unanswered questions...

- **Why** does my model work
- **What** does my model know?

We can't tell whether our models are certain or not...

E.g. what would be the CO_2 concentration level in Mauna Loa, Hawaii, *in 20 years' time*?



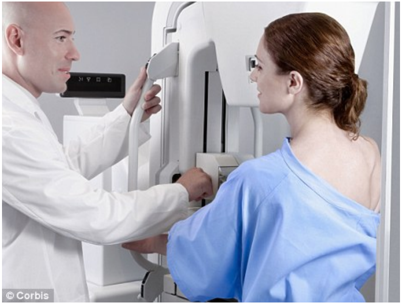
- **Why** does my model predict this and not that?

But many unanswered questions...

- **Why** does my model work
- **What** does my model know?
- **Why** does my model predict this and not that?

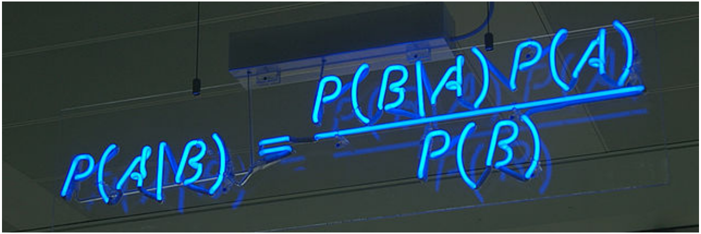
Our models are black boxes and not interpretable...

Physicians and others need to understand *why* a model predicts an output.



But many unanswered questions...

- **Why** does my model work
- **What** does my model know?
- **Why** does my model predict this and not that?



Surprisingly, we can use **Bayesian modelling** to answer the questions above

Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
- What does my model know?
- Why does my model predict this and not that, and other open problems
- Conclusions

Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
  - Bayesian modelling and neural networks
  - Modern deep learning as approximate inference
  - Real-world implications
- What does my model know?
- Why does my model predict this and not that, and other open problems
- Conclusions

- Observed inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and outputs $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$
- Capture stochastic process believed to have generated outputs
- Def. ω model parameters as r.v.
- Prior dist. over ω: $p(\omega)$
- Likelihood: $p(\mathbf{Y}|\omega, \mathbf{X})$
- Posterior: $p(\omega|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\omega,\mathbf{X})p(\omega)}{p(\mathbf{Y}|\mathbf{X})}$ (Bayes' theorem)
- Predictive distribution given new input $\mathbf{x}^*$

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega) \underbrace{\boxed{p(\omega|\mathbf{X}, \mathbf{Y})}}_{\text{posterior}} \mathrm{d}\omega$$

- But... $p(\omega|\mathbf{X}, \mathbf{Y})$ is often intractable

- Observed inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and outputs $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$
- Capture stochastic process believed to have generated outputs
- Def. ω model parameters as r.v.
- Prior dist. over ω: $p(\omega)$
- Likelihood: $p(\mathbf{Y}|\omega, \mathbf{X})$
- Posterior: $p(\omega|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\omega,\mathbf{X})p(\omega)}{p(\mathbf{Y}|\mathbf{X})}$ (Bayes' theorem)
- Predictive distribution given new input $\mathbf{x}^*$

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega) \underbrace{\boxed{p(\omega|\mathbf{X}, \mathbf{Y})}}_{\text{posterior}} d\omega$$

- But... $p(\omega|\mathbf{X}, \mathbf{Y})$ is often intractable

- Observed inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and outputs $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$
- Capture stochastic process believed to have generated outputs
- Def. ω model parameters as r.v.
- Prior dist. over ω: $p(\omega)$
- Likelihood: $p(\mathbf{Y}|\omega, \mathbf{X})$
- Posterior: $p(\omega|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\omega,\mathbf{X})p(\omega)}{p(\mathbf{Y}|\mathbf{X})}$ (Bayes' theorem)
- Predictive distribution given new input $\mathbf{x}^*$

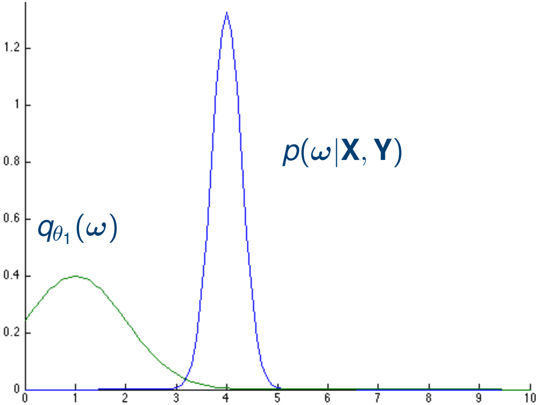
$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega) \underbrace{\boxed{p(\omega|\mathbf{X}, \mathbf{Y})}}_{\text{posterior}} \mathrm{d}\omega$$

- But... $p(\omega|\mathbf{X}, \mathbf{Y})$ is often intractable

UNIVERSITY OF CAMBRIDGE

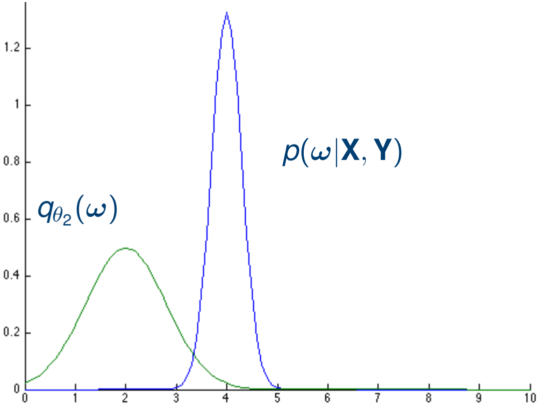
- Approximate $p(\omega|\mathbf{X},\mathbf{Y})$ with simple dist. $q_\theta(\omega)$
- Minimise divergence from posterior w.r.t. θ

$$\mathrm{KL}(q_\theta(\omega) \mid\mid p(\omega|\mathbf{X},\mathbf{Y}))$$



- Approximate $p(\omega|\mathbf{X}, \mathbf{Y})$ with simple dist. $q_\theta(\omega)$
- Minimise divergence from posterior w.r.t. θ

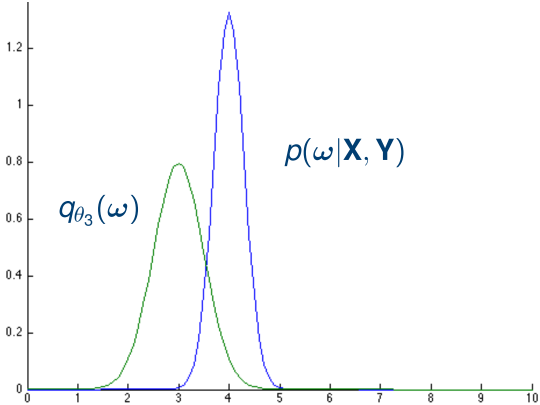
$$\mathrm{KL}(q_\theta(\omega) \,||\, p(\omega|\mathbf{X}, \mathbf{Y}))$$



Approximate inference

- Approximate $p(\omega|\mathbf{X}, \mathbf{Y})$ with simple dist. $q_\theta(\omega)$
- Minimise divergence from posterior w.r.t. θ

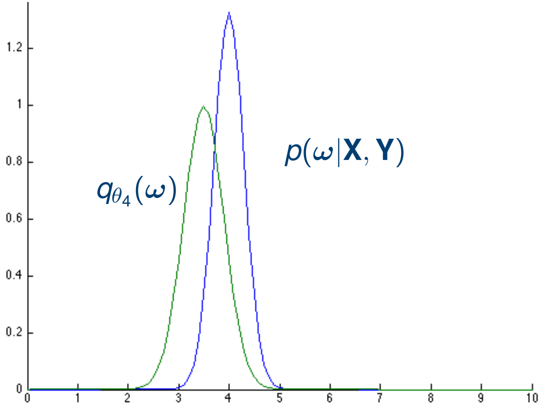
$$\text{KL}(q_\theta(\omega) \;||\; p(\omega|\mathbf{X}, \mathbf{Y}))$$



UNIVERSITY OF CAMBRIDGE

- Approximate $p(\omega|\mathbf{X},\mathbf{Y})$ with simple dist. $q_\theta(\omega)$
- Minimise divergence from posterior w.r.t. θ

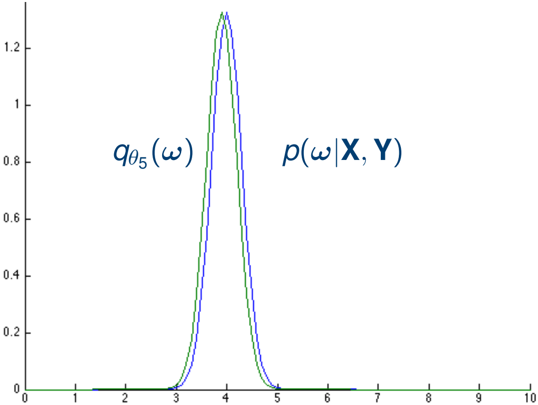
$$\mathrm{KL}(q_\theta(\omega) \,||\, p(\omega|\mathbf{X},\mathbf{Y}))$$



UNIVERSITY OF CAMBRIDGE

- Approximate $p(\omega|\mathbf{X}, \mathbf{Y})$ with simple dist. $q_\theta(\omega)$
- Minimise divergence from posterior w.r.t. θ

$$\text{KL}(q_\theta(\omega) \,||\, p(\omega|\mathbf{X}, \mathbf{Y}))$$



$q_{\theta_5}(\omega)$ $p(\omega|\mathbf{X}, \mathbf{Y})$

Approximate inference

- Approximate $p(\omega|\mathbf{X}, \mathbf{Y})$ with simple dist. $q_\theta(\omega)$
- Minimise divergence from posterior w.r.t. θ

$$\mathrm{KL}(q_\theta(\omega) \,||\, p(\omega|\mathbf{X}, \mathbf{Y}))$$

- Identical to minimising

$$\mathcal{L}_{\mathrm{VI}}(\theta) := -\int q_\theta(\omega) \log \overbrace{p(\mathbf{Y}|\mathbf{X}, \omega)}^{\text{likelihood}} \mathrm{d}\omega + \mathrm{KL}(q_\theta(\omega) || \overbrace{p(\omega)}^{\text{prior}})$$

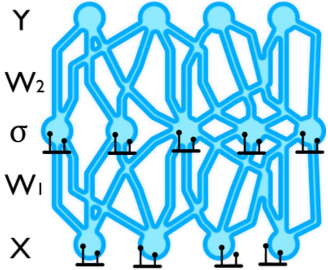
- We can approximate the **predictive distribution**

$$q_\theta(\mathbf{y}^*|\mathbf{x}^*) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega) q_\theta(\omega) \mathrm{d}\omega.$$

What to this and deep learning?

We'll look at dropout specifically:

- Used in **most modern deep learning models**



- It somehow circumvents **over-fitting**
- And improves **performance**

With Bayesian modelling we can explain **why**

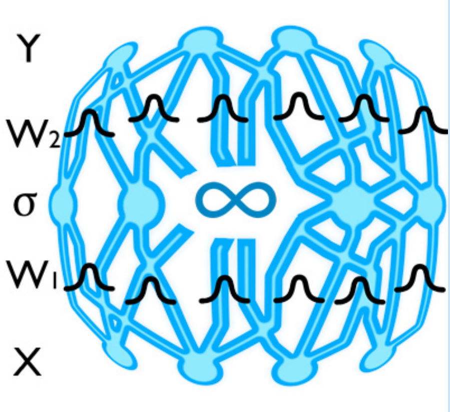
The link —

Bayesian neural networks

- Place prior $p(\mathbf{W}_i)$:

$$\mathbf{W}_i \sim \mathcal{N}(0, \mathbf{I})$$

for $i \leq L$ (and write $\omega := \{\mathbf{W}_i\}_{i=1}^{L}$).



UNIVERSITY OF CAMBRIDGE

Bayesian neural networks

- Place prior $p(\mathbf{W}_i)$:
$$\mathbf{W}_i \sim \mathcal{N}(0, \mathbf{I})$$
for $i \leq L$ (and write $\omega := \{\mathbf{W}_i\}_{i=1}^L$).
- Output is a r.v. $\mathbf{f}\big(\mathbf{x}, \omega\big) = \mathbf{W}_L \sigma\big(...\mathbf{W}_2 \sigma\big(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1\big)...\big)$.
- Softmax likelihood for class.: $p\big(y|\mathbf{x}, \omega\big) = \text{softmax}\left(\mathbf{f}\big(\mathbf{x}, \omega\big)\right)$ or a Gaussian for regression: $p\big(\mathbf{y}|\mathbf{x}, \omega\big) = \mathcal{N}\left(\mathbf{y}; \mathbf{f}\big(\mathbf{x}, \omega\big), \tau^{-1}\mathbf{I}\right)$.
- But difficult to evaluate posterior
$$p\big(\omega|\mathbf{X}, \mathbf{Y}\big).$$

Many have tried...

UNIVERSITY OF CAMBRIDGE

Bayesian neural networks

- Place prior $p(\mathbf{W}_i)$:

$$\mathbf{W}_i \sim \mathcal{N}(0, \mathbf{I})$$

  for $i \leq L$ (and write $\omega := \{\mathbf{W}_i\}_{i=1}^{L}$).

- Output is a r.v. $\mathbf{f}\big(\mathbf{x}, \omega\big) = \mathbf{W}_L \sigma\big(...\mathbf{W}_2 \sigma\big(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1\big)...\big)$.

- Softmax likelihood for class.: $p\big(y|\mathbf{x}, \omega\big) = \text{softmax}\left(\mathbf{f}\big(\mathbf{x}, \omega\big)\right)$
  or a Gaussian for regression: $p\big(\mathbf{y}|\mathbf{x}, \omega\big) = \mathcal{N}\left(\mathbf{y}; \mathbf{f}\big(\mathbf{x}, \omega\big), \tau^{-1}\mathbf{I}\right)$.

- But difficult to evaluate posterior

$$p\big(\omega|\mathbf{X}, \mathbf{Y}\big).$$

Many have tried...

UNIVERSITY OF CAMBRIDGE

Bayesian neural networks

- Place prior $p(\mathbf{W}_i)$:

$$\mathbf{W}_i \sim \mathcal{N}(0, \mathbf{I})$$

  for $i \leq L$ (and write $\omega := \{\mathbf{W}_i\}_{i=1}^L$).

- Output is a r.v. $\mathbf{f}(\mathbf{x}, \omega) = \mathbf{W}_L \sigma(...\mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)...)$.

- Softmax likelihood for class.: $p(y|\mathbf{x}, \omega) = \text{softmax}(\mathbf{f}(\mathbf{x}, \omega))$
  or a Gaussian for regression: $p(\mathbf{y}|\mathbf{x}, \omega) = \mathcal{N}(\mathbf{y}; \mathbf{f}(\mathbf{x}, \omega), \tau^{-1}\mathbf{I})$.

- But difficult to evaluate posterior

$$p(\omega|\mathbf{X}, \mathbf{Y}).$$

Many have tried...

Long history[1]

- Denker, Schwartz, Wittner, Solla, Howard, Jackel, Hopfield (1987)
- Denker and LeCun (1991)
- MacKay (1992)
- Hinton and van Camp (1993)
- Neal (1995)
- Barber and Bishop (1998)

And more recently...

- **Graves (2011)**
- Blundell, Cornebise, Kavukcuoglu, and Wierstra (2015)
- **Hernandez-Lobato and Adam (2015)**

But we don't use these... do we?

[1] Complete references at end of slides

Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
  - Bayesian modelling and neural networks
  - Modern deep learning as approximate inference
  - Real-world implications
- What does my model know?
- Why does my model predict this and not that, and other open problems
- Conclusions

UNIVERSITY OF CAMBRIDGE

Approximate inference in Bayesian NNs

- Def $q_\theta(\omega)$ to approximate posterior $p(\omega|\mathbf{X}, \mathbf{Y})$
- KL divergence to minimise:

$$\begin{aligned}
&\mathrm{KL}\big(q_\theta(\omega) \,||\, p(\omega|\mathbf{X}, \mathbf{Y})\big) \\
&\quad \propto \boxed{-\int q_\theta(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega) \mathrm{d}\omega} + \mathrm{KL}\big(q_\theta(\omega) \,||\, p(\omega)\big) \\
&\quad =: \mathcal{L}(\theta)
\end{aligned}$$

- Approximate the integral with MC integration $\widehat{\omega} \sim q_\theta(\omega)$:

$$\widehat{\mathcal{L}}(\theta) := -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \mathrm{KL}\big(q_\theta(\omega) \,||\, p(\omega)\big)$$

UNIVERSITY OF CAMBRIDGE

Approximate inference in Bayesian NNs

- Def $q_\theta(\omega)$ to approximate posterior $p(\omega|\mathbf{X}, \mathbf{Y})$
- KL divergence to minimise:

$$\mathrm{KL}(q_\theta(\omega) \,||\, p(\omega|\mathbf{X}, \mathbf{Y}))$$

$$\propto \boxed{-\int q_\theta(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega) \mathrm{d}\omega} + \mathrm{KL}(q_\theta(\omega) \,||\, p(\omega))$$

$$=: \mathcal{L}(\theta)$$

- Approximate the integral with MC integration $\widehat{\omega} \sim q_\theta(\omega)$:

$$\widehat{\mathcal{L}}(\theta) := -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \mathrm{KL}(q_\theta(\omega) \,||\, p(\omega))$$

UNIVERSITY OF CAMBRIDGE

Approximate inference in Bayesian NNs

- Def $q_\theta(\omega)$ to approximate posterior $p(\omega|\mathbf{X}, \mathbf{Y})$
- KL divergence to minimise:

$$\mathrm{KL}(q_\theta(\omega) \,||\, p(\omega|\mathbf{X}, \mathbf{Y}))$$

$$\propto \boxed{-\int q_\theta(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega) \mathrm{d}\omega} + \mathrm{KL}(q_\theta(\omega) \,||\, p(\omega))$$

$$=: \mathcal{L}(\theta)$$

- Approximate the integral with MC integration $\widehat{\omega} \sim q_\theta(\omega)$:

$$\widehat{\mathcal{L}}(\theta) := -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \mathrm{KL}(q_\theta(\omega) \,||\, p(\omega))$$

UNIVERSITY OF CAMBRIDGE

Stochastic approx. inference in Bayesian NNs

- Unbiased estimator:

$$E_{\widehat{\omega} \sim q_\theta(\omega)}\big(\widehat{\mathcal{L}}(\theta)\big) = \mathcal{L}(\theta)$$

- Converges to the same optima as $\mathcal{L}(\theta)$
- For inference, repeat:
  - Sample $\widehat{\omega} \sim q_\theta(\omega)$
  - And minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \text{KL}\big(q_\theta(\omega) \,||\, p(\omega)\big)$$

w.r.t. θ.

UNIVERSITY OF CAMBRIDGE

Stochastic approx. inference in Bayesian NNs

- Unbiased estimator:

$$E_{\widehat{\omega} \sim q_\theta(\omega)}\big(\widehat{\mathcal{L}}(\theta)\big) = \mathcal{L}(\theta)$$

- Converges to the same optima as $\mathcal{L}(\theta)$
- For inference, repeat:
  - Sample $\widehat{\omega} \sim q_\theta(\omega)$
  - And minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \mathrm{KL}\big(q_\theta(\omega) \,||\, p(\omega)\big)$$

w.r.t. θ.

UNIVERSITY OF CAMBRIDGE

Stochastic approx. inference in Bayesian NNs

- Unbiased estimator:

$$E_{\widehat{\omega} \sim q_\theta(\omega)}\big(\widehat{\mathcal{L}}(\theta)\big) = \mathcal{L}(\theta)$$

- Converges to the same optima as $\mathcal{L}(\theta)$
- For inference, repeat:
  - Sample $\widehat{\omega} \sim q_\theta(\omega)$
  - And minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p\big(\mathbf{Y}|\mathbf{X}, \widehat{\omega}\big) + \mathrm{KL}\big(q_\theta(\omega) \; || \; p(\omega)\big)$$

w.r.t. θ.

Specifying $q_\theta(\cdot)$

- Given $\mathbf{z}_{i,j}$ Bernoulli r.v. and variational parameters $\theta = \{\mathbf{M}_i\}_{i=1}^L$ (set of matrices):

$$\mathbf{z}_{i,j} \sim \text{Bernoulli}(p_i) \text{ for } i = 1, ..., L,\ j = 1, ..., K_{i-1}$$

$$\mathbf{W}_i = \mathbf{M}_i \cdot \text{diag}([\mathbf{z}_{i,j}]_{j=1}^{K_i})$$

$$q_\theta(\omega) = \prod q_{\mathbf{M}_i}(\mathbf{W}_i)$$

In summary:

Minimise divergence between $q_\theta(\omega)$ and $p(\omega|\mathbf{X}, \mathbf{Y})$:

- Repeat:
  - Sample $\widehat{\mathbf{z}}_{i,j} \sim \text{Bernoulli}(p_i)$ and set

$$\widehat{\mathbf{W}}_i = \mathbf{M}_i \cdot \text{diag}([\widehat{\mathbf{z}}_{i,j}]_{j=1}^{K_i})$$
$$\widehat{\omega} = \{\widehat{\mathbf{W}}_i\}_{i=1}^{L}$$

  - Minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p\big(\mathbf{Y}|\mathbf{X}, \widehat{\omega}\big) + \text{KL}\big(q_\theta(\omega) \mid\mid p(\omega)\big)$$

w.r.t. $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (set of matrices).

UNIVERSITY OF CAMBRIDGE

In summary:

Minimise divergence between $q_\theta(\omega)$ and $p(\omega|\mathbf{X}, \mathbf{Y})$:

- Repeat:
  - = Randomly set columns of $\mathbf{M}_i$ to zero
  - Minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p\big(\mathbf{Y}|\mathbf{X}, \widehat{\omega}\big) + \mathrm{KL}\big(q_\theta(\omega) \;||\; p(\omega)\big)$$

w.r.t. $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (set of matrices).

In summary:

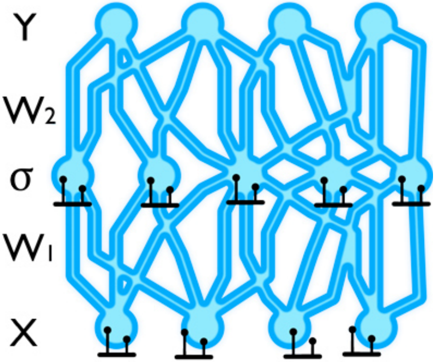
Minimise divergence between $q_\theta(\omega)$ and $p(\omega|\mathbf{X}, \mathbf{Y})$:

- Repeat:
  - = Randomly set units of the network to zero
  - Minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p\big(\mathbf{Y}|\mathbf{X}, \widehat{\omega}\big) + \mathrm{KL}\big(q_\theta(\omega) \,||\, p(\omega)\big)$$

w.r.t. $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (set of matrices).

Sounds familiar?[2]



$$\widehat{\mathcal{L}}(\theta) = \overbrace{-\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega})}^{=\text{ loss}} + \overbrace{\mathrm{KL}(q_\theta(\omega) \,||\, p(\omega))}^{=\, L_2 \text{ reg}}$$

[2]For more details see appendix of Gal and Ghahramani (2015) – yarin.co/dropout

Now we can answer: "Why does dropout work?"

- ~~It adds noise~~
- ~~Sexual reproduction[3]~~

> **2. Motivation**
>
> A motivation for dropout comes from a theory of the role of sex in evolution (Livnat et al., 2010). Sexual reproduction involves taking half the genes of one parent and half of the

- Because it approximately **integrates** over model parameters
- The **noise is a side-effect** of approx. integration
- Explains model **over specification**, "adaptive model capacity"
- We **fit the process** that generated our data

[3] Srivastava et al. (2014)

Now we can answer: “Why does dropout work?”

- ~~It adds noise~~
- ~~Sexual reproduction[3]~~

> **2. Motivation**
>
> A motivation for dropout comes from a theory of the role of sex in evolution (Livnat et al., 2010). Sexual reproduction involves taking half the genes of one parent and half of the

- Because it approximately **integrates** over model parameters
- The **noise is a side-effect** of approx. integration
- Explains model **over specification**, “adaptive model capacity”
- We **fit the process** that generated our data

[3] Srivastava et al. (2014)

UNIVERSITY OF CAMBRIDGE

Now we can answer: "Why does dropout work?"

- ~~It adds noise~~
- ~~Sexual reproduction[3]~~

> **2. Motivation**
>
> A motivation for dropout comes from a theory of the role of sex in evolution (Livnat et al., 2010). Sexual reproduction involves taking half the genes of one parent and half of the

- Because it approximately **integrates** over model parameters
- The **noise is a side-effect** of approx. integration
- Explains model **over specification**, "adaptive model capacity"
- We **fit the process** that generated our data

[3] Srivastava et al. (2014)

Now we can answer: "Why does dropout work?"

- ~~It adds noise~~
- ~~Sexual reproduction[3]~~

> **2. Motivation**
>
> A motivation for dropout comes from a theory of the role of sex in evolution (Livnat et al., 2010). Sexual reproduction involves taking half the genes of one parent and half of the

- Because it approximately **integrates** over model parameters
- The **noise is a side-effect** of approx. integration
- Explains model **over specification**, "adaptive model capacity"
- We **fit the process** that generated our data

[3] Srivastava et al. (2014)

Why does my model work?

Now we can answer: “Why does dropout work?”

- ~~It adds noise~~
- ~~Sexual reproduction[3]~~

> **2. Motivation**
>
> A motivation for dropout comes from a theory of the role of sex in evolution (Livnat et al., 2010). Sexual reproduction involves taking half the genes of one parent and half of the

- Because it approximately **integrates** over model parameters
- The **noise is a side-effect** of approx. integration
- Explains model **over specification**, “adaptive model capacity”
- We **fit the process** that generated our data

[3] Srivastava et al. (2014)

Diving deeper into dropout

- "Why this $q_\theta(\cdot)$?"
  - Bernoullis are cheap
  - Dropout at test time ≈ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Constrains the weights to near the origin:
    - Posterior uncertainty decreases with more data
    - $\text{Var}(\mathbf{W}_i) = \mathbf{M}_i \mathbf{M}_i^T (p_i - p_i^2)$
    - For fixed p_i, to decrease uncertainty must decrease $||\mathbf{M}_i||$
  - Smallest $||\mathbf{M}_i||$ = strongest reg. at $p_i = 0.5$.

- "Why this $q_\theta(\cdot)$?"
  - Bernoullis are cheap
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i\mathbf{M}_i$
  - Constrains the weights to near the origin:
    - Posterior uncertainty decreases with more data
    - $\text{Var}(\mathbf{W}_i) = \mathbf{M}_i\mathbf{M}_i^T(p_i - p_i^2)$
    - For fixed p_i, to decrease uncertainty must decrease $||\mathbf{M}_i||$
  - Smallest $||\mathbf{M}_i||$ = strongest reg. at $p_i = 0.5$.

UNIVERSITY OF CAMBRIDGE

- "Why this $q_\theta(\cdot)$?"
  - Bernoullis are cheap
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Constrains the weights to near the origin:
    - Posterior uncertainty decreases with more data
    - $\text{Var}(\mathbf{W}_i) = \mathbf{M}_i \mathbf{M}_i^T (p_i - p_i^2)$
    - For fixed p_i, to decrease uncertainty must decrease $||\mathbf{M}_i||$
  - Smallest $||\mathbf{M}_i||$ = strongest reg. at $p_i = 0.5$.

UNIVERSITY OF CAMBRIDGE

- "Why this $q_\theta(\cdot)$?"
  - Bernoullis are cheap
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Constrains the weights to near the origin:
    - Posterior uncertainty decreases with more data
    - $\text{Var}(\mathbf{W}_i) = \mathbf{M}_i \mathbf{M}_i^T (p_i - p_i^2)$
    - For fixed p_i, to decrease uncertainty must decrease $||\mathbf{M}_i||$
  - Smallest $||\mathbf{M}_i||$ = strongest reg. at $p_i = 0.5$.

- Multiplicative Gaussian noise (Srivastava et al. 2014) –
- Multiply network units by $\mathcal{N}(1, 1)$
- Same performance as dropout

$$\Updownarrow$$

Multiplicative Gaussian noise as approximate inference[4]

$$\mathbf{z}_{i,j} \sim \mathcal{N}(1, 1) \text{ for } i = 1, ..., L,\ j = 1, ..., K_{i-1}$$

$$\mathbf{W}_i = \mathbf{M}_i \cdot \text{diag}([\mathbf{z}_{i,j}]_{j=1}^{K_i})$$

$$q_\theta(\boldsymbol{\omega}) = \prod q_{\mathbf{M}_i}(\mathbf{W}_i)$$

Similarly for **drop-connect** (Wan et al., 2013), **hashed neural networks** (Chen et al., 2015)

[4]See Gal and Ghahramani (2015) and Kingma et al. (2015)

Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
  - Bayesian modelling and neural networks
  - Modern deep learning as approximate inference
  - Real-world implications
- What does my model know?
- Why does my model predict this and not that, and other open problems
- Conclusions

Real world implications

"A theory is worth nothing if you can't use it to make better code."

– DeadMG Jun 10 '12, stackexchange

- Better use of dropout
- Model structure selection
  - (No time: use Bayesian statistics to understand model architecture)

How do we use dropout with convolutional neural networks (convnets)?

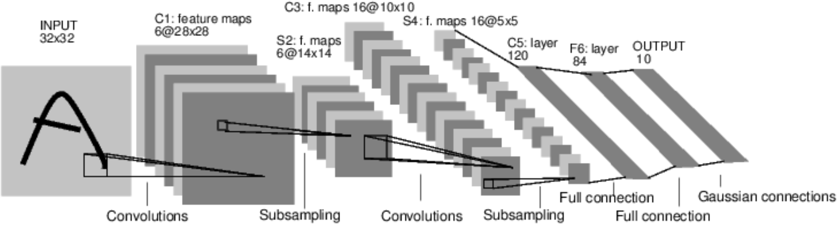


Figure : LeNet convnet structure

Image Source: LeCun et al. (1998)

How do we use dropout with convolutional neural networks (convnets)?

Dropout ↓

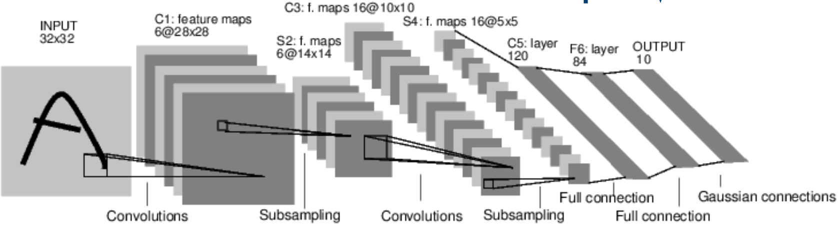


Figure : LeNet convnet structure

Image Source: LeCun et al. (1998)

Better use of dropout

How do we use dropout with convolutional neural networks (convnets)?

No dropout ↓ **Dropout ↓**

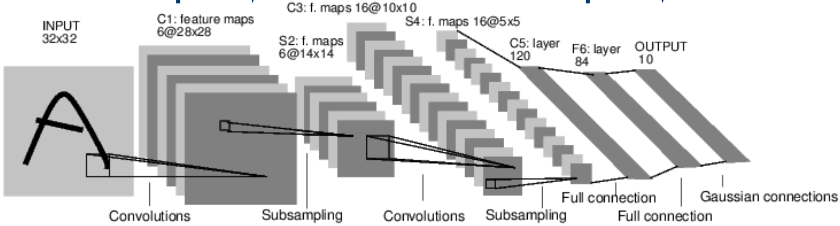


Figure : LeNet convnet structure

Image Source: LeCun et al. (1998)

Why not use dropout et al. with convolutions?

- ~~It doesn't work~~
- ~~Low co-adaptation in convolutions~~
- Because it's not used correctly
  - Standard dropout averages **weights** at test time

Why not use dropout et al. with convolutions?

- ~~It doesn't work~~
- ~~Low co-adaptation in convolutions~~
- Because it's not used correctly
  - Standard dropout averages **weights** at test time

UNIVERSITY OF CAMBRIDGE

Instead, **predictive mean**, approx. with MC integration:

$$\mathbb{E}_{q_\theta(\mathbf{y}^*|\mathbf{x}^*)}\big(\mathbf{y}^*\big) = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t).$$

with $\widehat{\omega}_t \sim q_\theta(\omega)$.

- In practice, **average stochastic forward passes through the network** (referred to as "MC dropout").[5]
- Dropout after convolutions and averaging forward passes = **approximate inference in Bayesian convnets.**[6]

[5] Also suggested in Srivastava et al. (2014) as *model averaging*.
[6] See `yarin.co/bcnn` for more details

Instead, **predictive mean**, approx. with MC integration:

$$\mathbb{E}_{q_\theta(\mathbf{y}^*|\mathbf{x}^*)}\big(\mathbf{y}^*\big) = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t).$$

with $\widehat{\omega}_t \sim q_\theta\big(\omega\big)$.

- In practice, **average stochastic forward passes through the network** (referred to as "MC dropout").[5]
- Dropout after convolutions and averaging forward passes = **approximate inference in Bayesian convnets.**[6]

[5] Also suggested in Srivastava et al. (2014) as *model averaging*.

[6] See `yarin.co/bcnn` for more details

Better use of dropout

UNIVERSITY OF CAMBRIDGE

Instead, **predictive mean**, approx. with MC integration:

$$\mathbb{E}_{q_\theta(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t).$$

with $\widehat{\omega}_t \sim q_\theta(\omega)$.

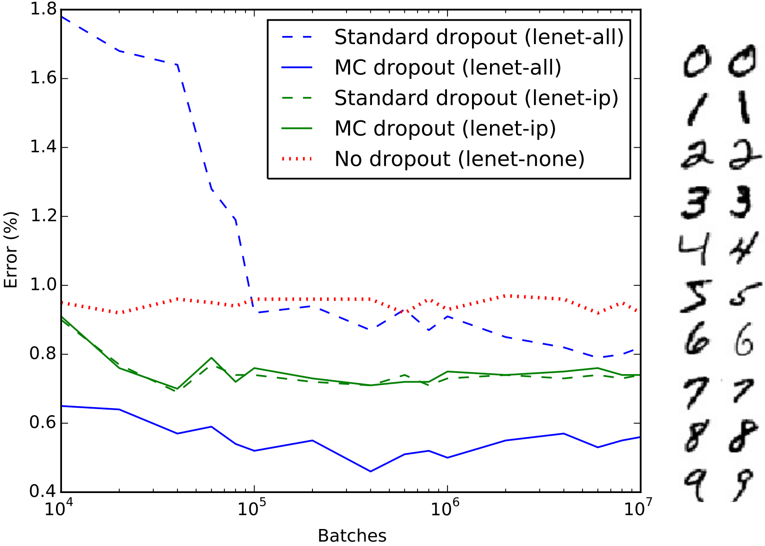
- In practice, **average stochastic forward passes through the network** (referred to as "MC dropout").[5]
- Dropout after convolutions and averaging forward passes = **approximate inference in Bayesian convnets.**[6]

---

[5]Also suggested in Srivastava et al. (2014) as *model averaging*.

[6]See `yarin.co/bcnn` for more details

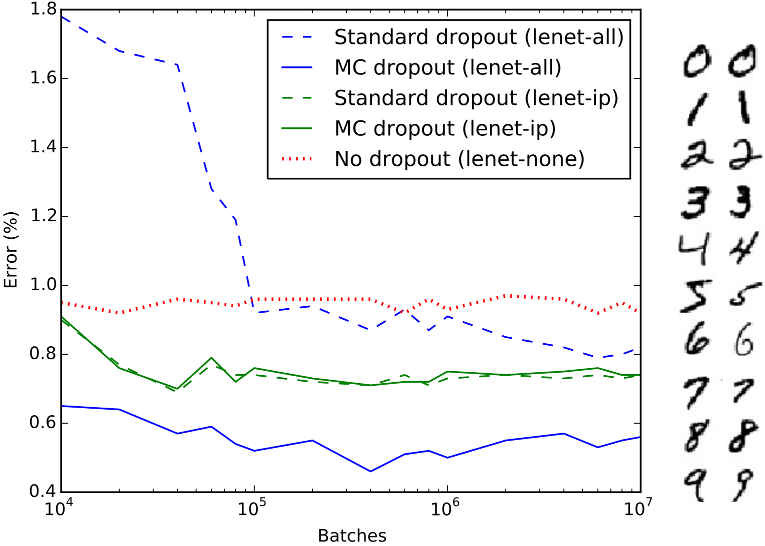
Huge improvement (MNIST)



Red: standard LeNet (no dropout)

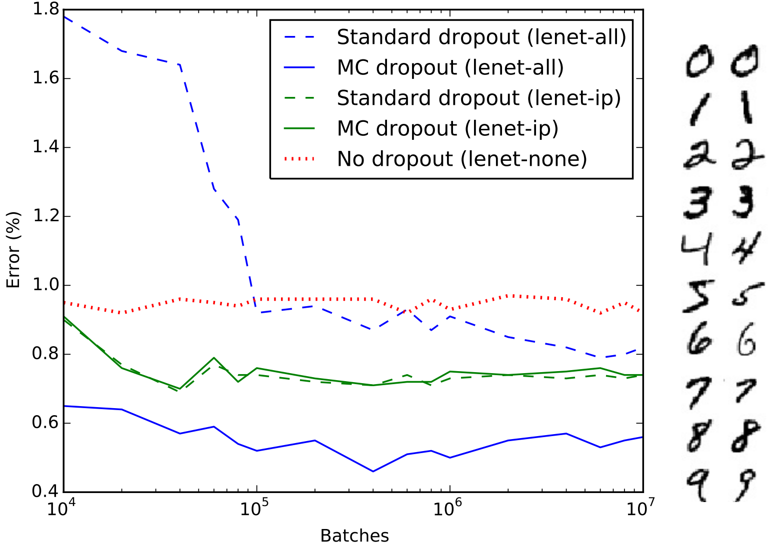
Huge improvement (MNIST)

UNIVERSITY OF CAMBRIDGE



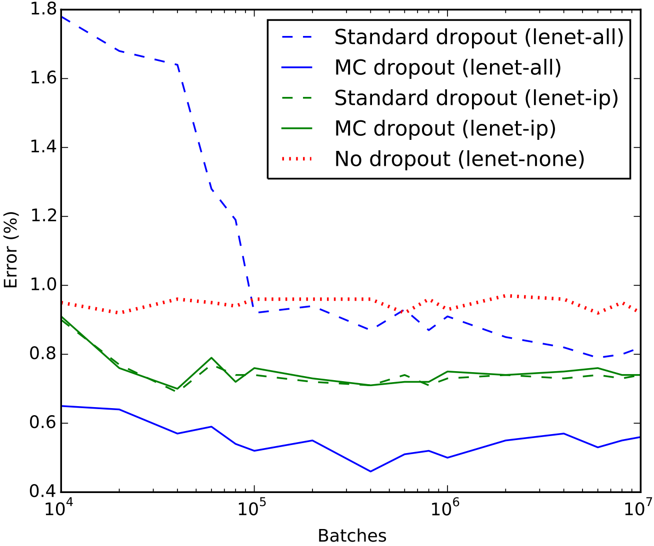
Green: standard dropout LeNet (dropout at the end)

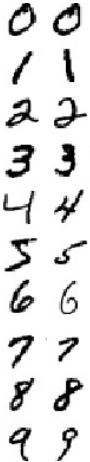
Huge improvement (MNIST)



Dashed blue: Bayesian LeNet (weight averaging – FAIL)

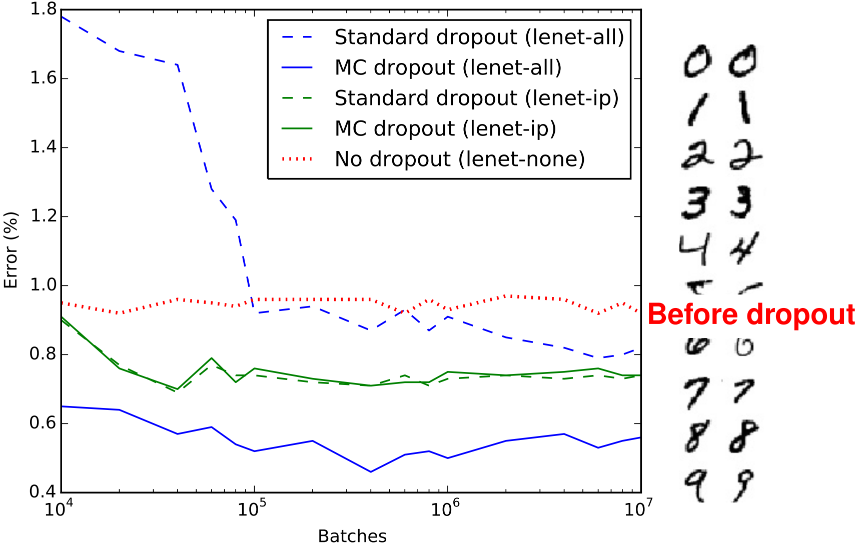
Huge improvement (MNIST)



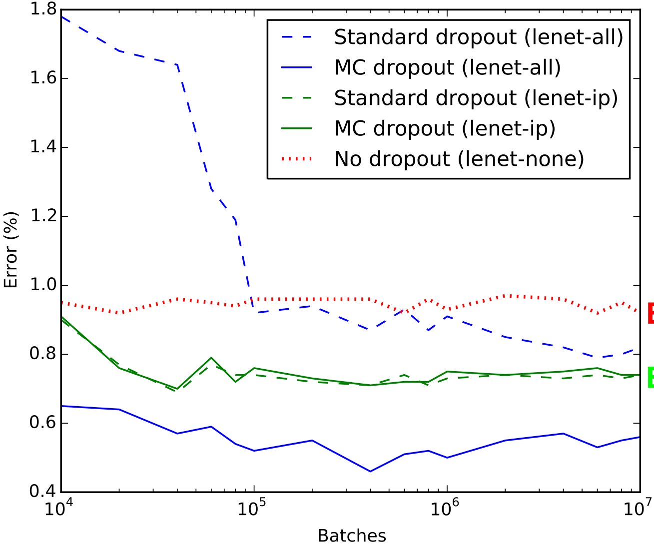


Solid blue: Bayesian LeNet (MC dropout)

Huge improvement (MNIST)



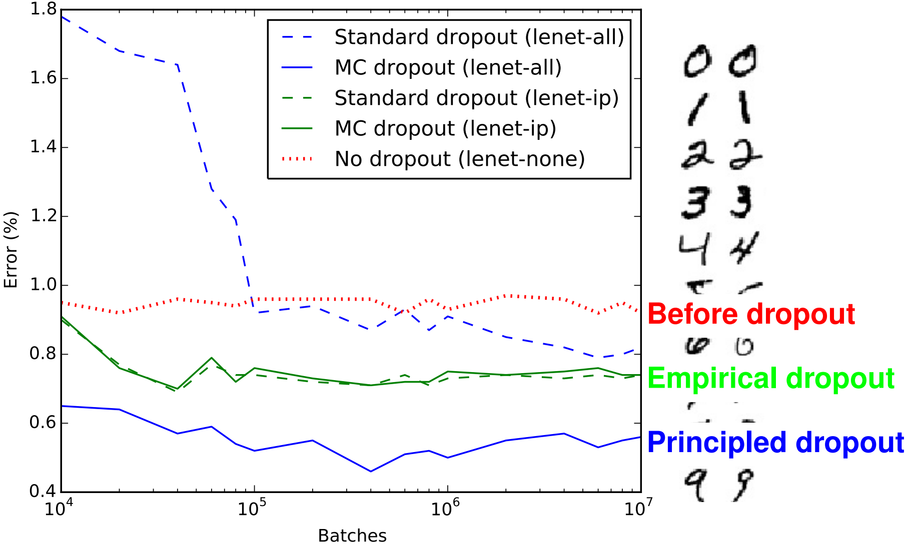
Huge improvement (MNIST)



Before dropout

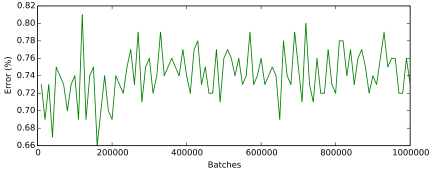
Empirical dropout

Huge improvement (MNIST)

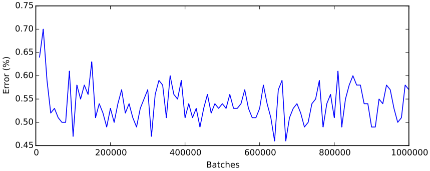


Over-fitting on small data

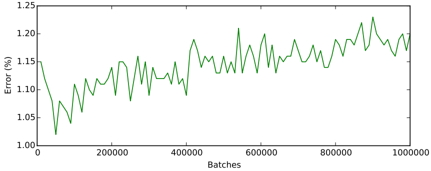
- Robustness to over-fitting on smaller datasets:



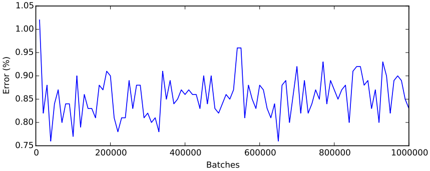
(a) Entire MNIST
Standard dropout convnet



(b) Entire MNIST
Bayesian convnet



(c) $1/4$ of MNIST
Standard dropout convnet

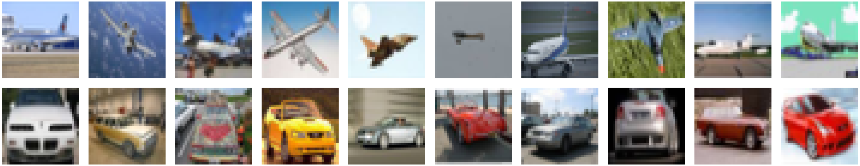


(d) $1/4$ of MNIST
Bayesian convnet

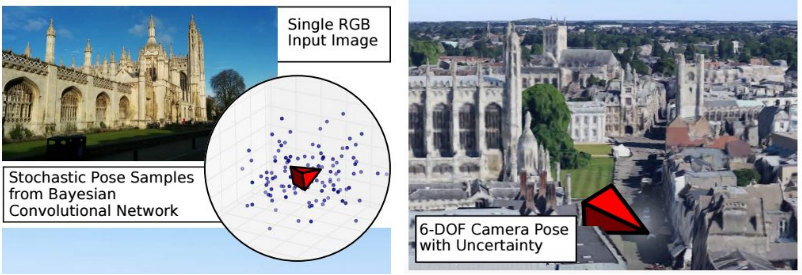
State-of-the-art on CIFAR-10

Model	CIFAR Test Error (and Std.)	
	Standard Dropout	**MC Dropout**
NIN	10.43 (Lin et al., 2013)	**10.27** $\pm$ **0.05**
DSN	9.37 (Lee et al., 2014)	**9.32** $\pm$ **0.02**
Augmented-DSN	7.95 (Lee et al., 2014)	**7.71** $\pm$ **0.09**

Table : Bayesian techniques (MC dropout) with existing state-of-the-art



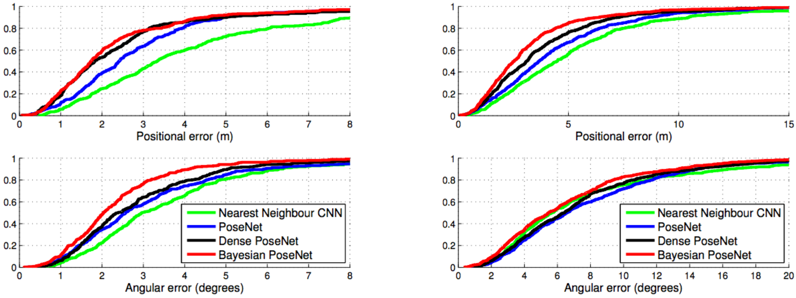
- Find the location from which a picture was taken[7]



- Kendall and Cipolla (2015) show **10–15%** improvement on **state-of-the-art** with Bayesian convnets

[7] Figures used with author permission

- Find the location from which a picture was taken[7]
- Kendall and Cipolla (2015) show **10–15%** improvement on **state-of-the-art** with Bayesian convnets



(a) King's College (b) St Mary's Church

Localisation accuracy for different error thresholds

[7]Figures used with author permission

Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
- What does my model know?
  - Why should I care about uncertainty?
  - How can I get uncertainty in deep learning?
  - What does this uncertainty look like?
  - Real-world implications
- Why does my model predict this and not that, and other open problems
- Conclusions

Why should I care about uncertainty?

- We train a model to recognise dog breeds

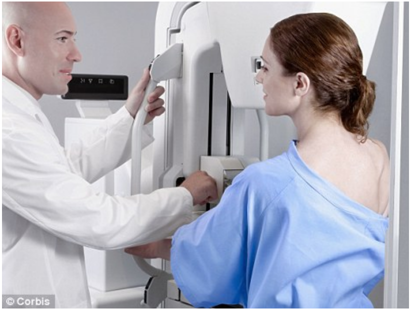
- We train a model to recognise dog breeds
- And are given a cat to classify

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

?

Why should I care about uncertainty?

- 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.[8]



- For the practitioner: pass inputs with low confidence to

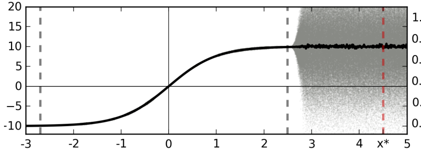
Why should I care about uncertainty?

- 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.[8]
- For the practitioner: pass inputs with low confidence to specialised models
- But I already have uncertainty in classification! well... no
- We need to be able to tell **what our model knows** and what it doesn't.[9]

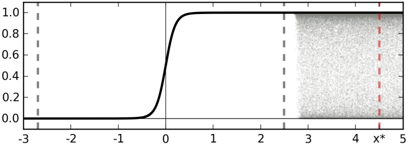
[8] Complete references at end of slides
[9] Friendly introduction given in `yarin.co/blog`

UNIVERSITY OF CAMBRIDGE

- 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.[8]
- For the practitioner: pass inputs with low confidence to specialised models
- But I already have uncertainty in classification! well... no



(a) Softmax *input* as a function of data $\mathbf{x}$: $f(\mathbf{x})$



(b) Softmax *output* as a function of data $\mathbf{x}$: $\sigma(f(\mathbf{x}))$

- 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.[8]
- For the practitioner: pass inputs with low confidence to specialised models
- But I already have uncertainty in classification! well... no
- We need to be able to tell **what our model knows** and what it doesn't.[9]

---

[8] Complete references at end of slides

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

- We fit a **distribution**; Already used its first moment:

$$\mathbb{E}(\mathbf{y}^*) = \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t)$$

with $\widehat{\omega}_t \sim q_\theta(\omega)$.

- For uncertainty (in regression) look at the **second moment**:

$$\boxed{\mathrm{Var}(\mathbf{y}^*)} = \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t)^T \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t) - \mathbb{E}(\mathbf{y}^*)^T \mathbb{E}(\mathbf{y}^*)$$

- As simple as looking at the **sample variance** of stochastic forward passes through the network (plus obs. noise).[10]

[10] See `yarin.co/dropout` for more details

- We fit a **distribution**; Already used its first moment:

$$\mathbb{E}(\mathbf{y}^*) = \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t)$$

with $\widehat{\omega}_t \sim q_\theta(\omega)$.

- For uncertainty (in regression) look at the **second moment**:

$$\boxed{\mathrm{Var}(\mathbf{y}^*)} = \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t)^T \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t) - \mathbb{E}(\mathbf{y}^*)^T \mathbb{E}(\mathbf{y}^*)$$

- As simple as looking at the **sample variance** of stochastic forward passes through the network (plus obs. noise).[10]

[10] See `yarin.co/dropout` for more details

- We fit a **distribution**; Already used its first moment:

$$\mathbb{E}(\mathbf{y}^*) = \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t)$$

with $\widehat{\omega}_t \sim q_\theta(\omega)$.

- For uncertainty (in regression) look at the **second moment**:

$$\boxed{\text{Var}(\mathbf{y}^*)} = \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t)^T \widehat{\mathbf{y}}(\mathbf{x}^*, \widehat{\omega}_t) - \mathbb{E}(\mathbf{y}^*)^T \mathbb{E}(\mathbf{y}^*)$$

- As simple as looking at the **sample variance** of stochastic forward passes through the network (plus obs. noise).[10]

[10] See `yarin.co/dropout` for more details

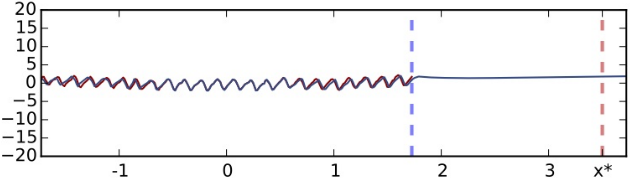
Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
- What does my model know?
  - Why should I care about uncertainty?
  - How can I get uncertainty in deep learning?
  - What does this uncertainty look like?
  - Real-world implications
- Why does my model predict this and not that, and other open problems
- Conclusions

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

- Normal dropout (weight averaging, 5 layers, ReLU units):

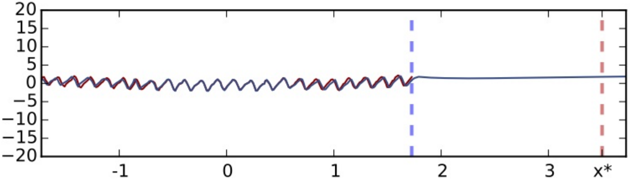


- Same network, Bayesian perspective:

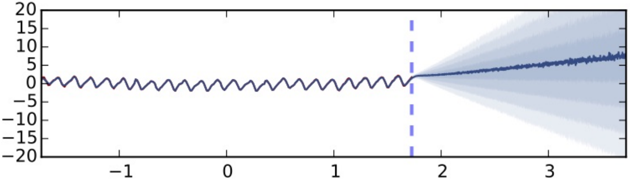
What does this uncertainty look like?

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

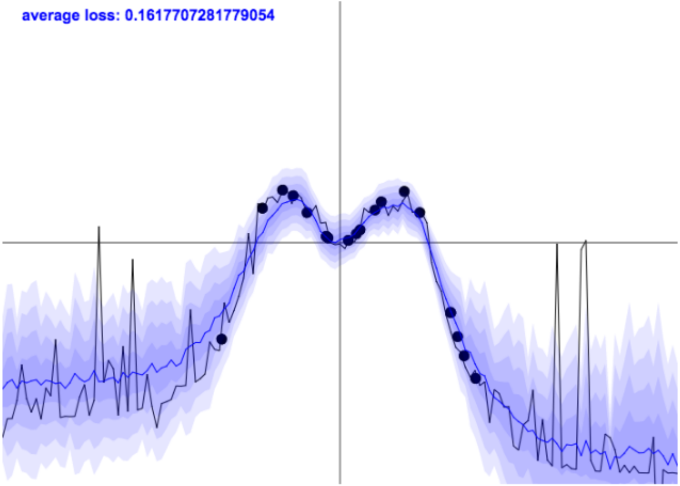
- Normal dropout (weight averaging, 5 layers, ReLU units):



- Same network, Bayesian perspective:



What does this uncertainty look like?



[Online demo] [11]

[11] yarin.co/blog

UNIVERSITY OF CAMBRIDGE

- How good is our uncertainty estimate?

Dataset	Avg. Test RMSE and Std. Errors VI	PBP	Dropout	Avg. Test LL and Std. Errors VI	PBP	Dropout
Boston Housing	4.32 ±0.29	3.01 ±0.18	**2.97 ±0.85**	-2.90 ±0.07	-2.57 ±0.09	**-2.46 ±0.25**
Concrete Strength	7.19 ±0.12	5.67 ±0.09	**5.23 ±0.53**	-3.39 ±0.02	-3.16 ±0.02	**-3.04 ±0.09**
Energy Efficiency	2.65 ±0.08	1.80 ±0.05	**1.66 ±0.19**	-2.39 ±0.03	-2.04 ±0.02	**-1.99 ±0.09**
Kin8nm	**0.10 ±0.00**	**0.10 ±0.00**	**0.10 ±0.00**	0.90 ±0.01	0.90 ±0.01	**0.95 ±0.03**
Naval Propulsion	**0.01 ±0.00**	**0.01 ±0.00**	**0.01 ±0.00**	3.73 ±0.12	3.73 ±0.01	**3.80 ±0.05**
Power Plant	4.33 ±0.04	4.12 ±0.03	**4.02 ±0.18**	-2.89 ±0.01	-2.84 ±0.01	**-2.80 ±0.05**
Protein Structure	4.84 ±0.03	4.73 ±0.01	**4.36 ±0.04**	-2.99 ±0.01	-2.97 ±0.00	**-2.89 ±0.01**
Wine Quality Red	0.65 ±0.01	0.64 ±0.01	**0.62 ±0.04**	-0.98 ±0.01	-0.97 ±0.01	**-0.93 ±0.06**
Yacht Hydrodynamics	6.89 ±0.67	**1.02 ±0.05**	1.11 ±0.38	-3.43 ±0.16	-1.63 ±0.02	**-1.55 ±0.12**
Year Prediction MSD	9.034 ±NA	8.879 ±NA	**8.849 ±NA**	-3.622 ±NA	-3.603 ±NA	**-3.588 ±NA**

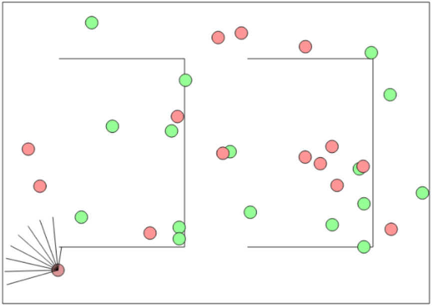
Table 1: **Average test performance in RMSE and predictive log likelihood** for a popular variational inference method (VI, Graves [20]), Probabilistic back-propagation (PBP, Hernández-Lobato and Adams [27]), and dropout uncertainty (Dropout).

Outline

UNIVERSITY OF CAMBRIDGE

- Many unanswered questions
- Why does my model work?
- What does my model know?
  - Why should I care about uncertainty?
  - How can I get uncertainty in deep learning?
  - What does this uncertainty look like?
  - Real-world implications
- Why does my model predict this and not that, and other open problems
- Conclusions

- We have a "Roomba"[12]
- Penalised -5 for walking into a wall, $+10$ reward for collecting dirt
- Our environment is stochastic and ever changing
- We want a net to learn what actions to do in different situations



[12] Code based on Karpathy and authors. `github.com/karpathy/convnetjs`

Behavioural policies:

- **Epsilon-greedy** – take random actions with probability ϵ and optimal actions otherwise
- Using uncertainty we can learn faster
- **Thompson sampling** – draw realisation from current belief over world, choose action with highest value
- In practice: simulate a stochastic forward pass through the dropout network and choose action with highest value

Behavioural policies:

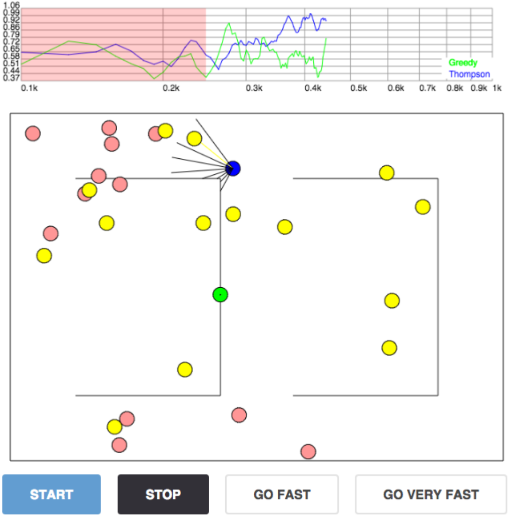
- **Epsilon-greedy** – take random actions with probability ϵ and optimal actions otherwise
- Using uncertainty we can learn faster
- **Thompson sampling** – draw realisation from current belief over world, choose action with highest value
- In practice: simulate a stochastic forward pass through the dropout network and choose action with highest value

Behavioural policies:

- **Epsilon-greedy** – take random actions with probability ϵ and optimal actions otherwise
- Using uncertainty we can learn faster
- **Thompson sampling** – draw realisation from current belief over world, choose action with highest value
- In practice: simulate a stochastic forward pass through the dropout network and choose action with highest value

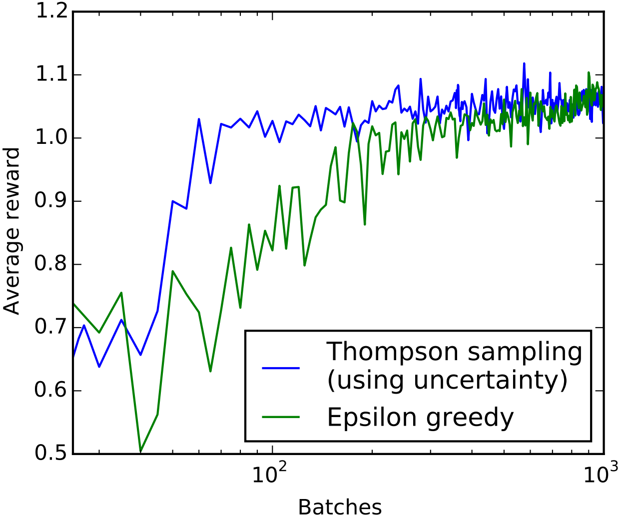
Behavioural policies:

- **Epsilon-greedy** – take random actions with probability ϵ and optimal actions otherwise
- Using uncertainty we can learn faster
- **Thompson sampling** – draw realisation from current belief over world, choose action with highest value
- In practice: simulate a stochastic forward pass through the dropout network and choose action with highest value



[Online demo] [13]

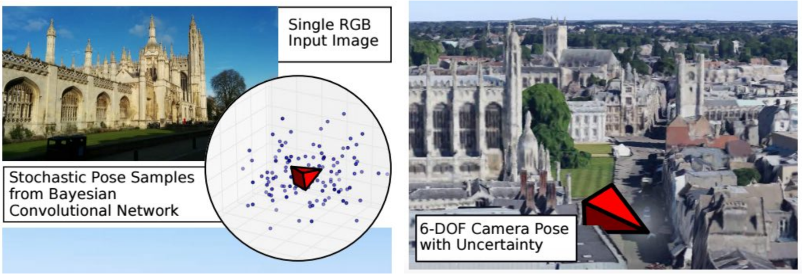
[13] yarin.co/blog



Average reward over time (log scale)

RW: Camera local. (with uncertainty!)

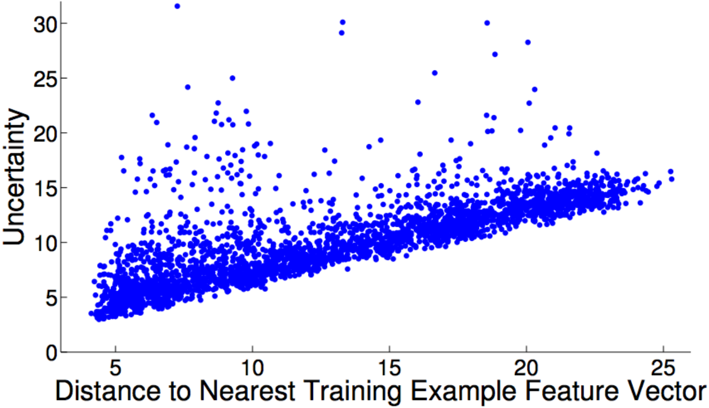
- Where was a picture taken? (Kendall and Cipolla, 2015)[14]



- Uncertainty increases as a test photo diverges from training distribution
- Test photos with high uncertainty (strong occlusion from vehicles, pedestrians or other objects)
- Localisation error correlates with uncertainty

[14] Figures used with author permission

- Where was a picture taken? (Kendall and Cipolla, 2015)[14]
- Uncertainty increases as a test photo diverges from training distribution



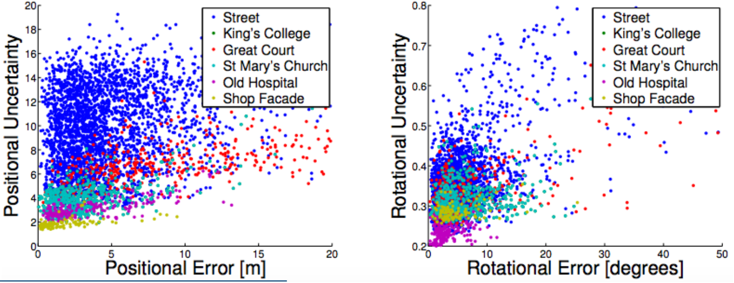
- Test photos with high uncertainty (strong occlusion from

RW: Camera local. (with uncertainty!)

- Where was a picture taken? (Kendall and Cipolla, 2015)[14]
- Uncertainty increases as a test photo diverges from training distribution
- Test photos with high uncertainty (strong occlusion from vehicles, pedestrians or other objects)

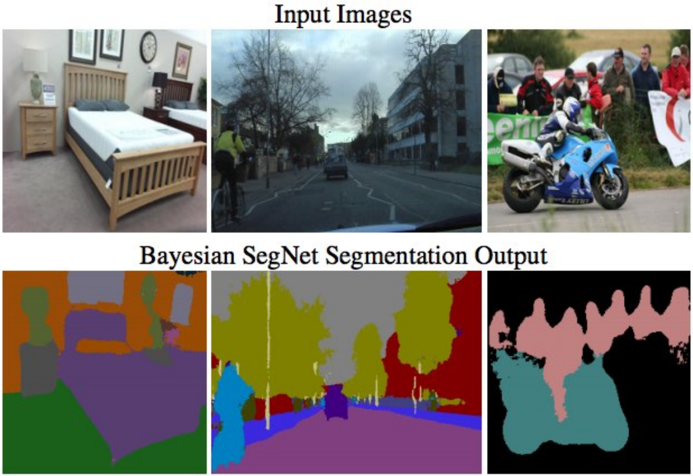
UNIVERSITY OF CAMBRIDGE

- Where was a picture taken? (Kendall and Cipolla, 2015)[14]
- Uncertainty increases as a test photo diverges from training distribution
- Test photos with high uncertainty (strong occlusion from vehicles, pedestrians or other objects)
- Localisation error correlates with uncertainty



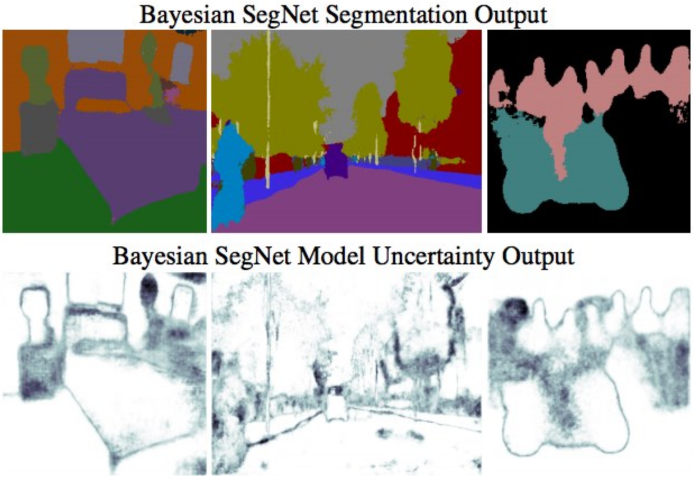
[14] Figures used with author permission

- Scene understanding: what's in a photo and where? (Kendall, Badrinarayanan, and Cipolla, 2015)[15]



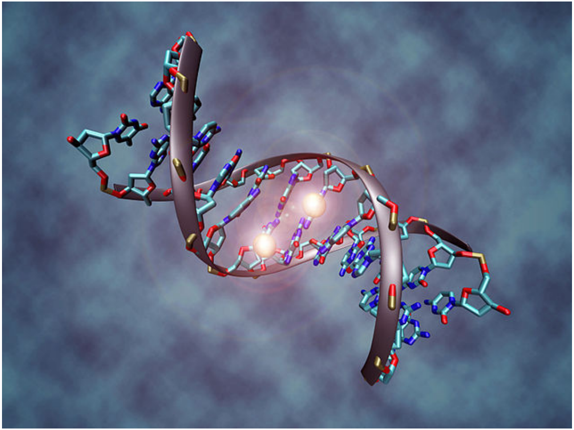
[15] Figures used with author permission

- Scene understanding: what's in a photo and where? (Kendall, Badrinarayanan, and Cipolla, 2015)[15]



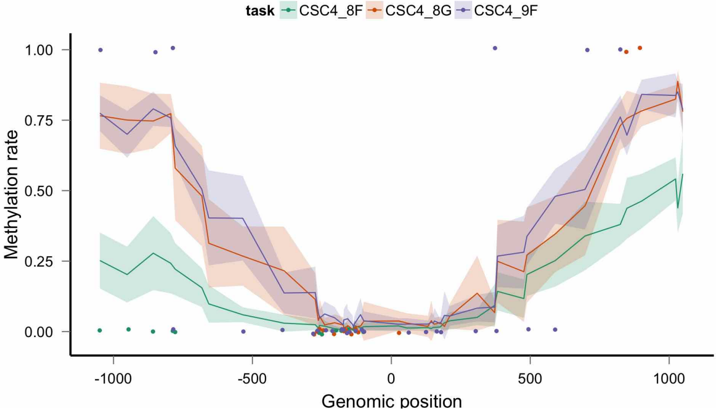
[15] Figures used with author permission

- Angermueller and Stegle (2015) fit a network to predict **DNA methylation** – used for gene regulation



- Look at methylation rate of different embryonic stem cells. **Uncertainty increases** in genomic contexts that are hard to predict (e.g. LMR or H3K27me3)

- Angermueller and Stegle (2015) fit a network to predict **DNA methylation** – used for gene regulation
- Look at methylation rate of different embryonic stem cells. **Uncertainty increases** in genomic contexts that are hard to predict (e.g. LMR or H3K27me3)



Outline

- Many unanswered questions
- Why does my model work?
- What does my model know?
- Why does my model predict this and not that, and other open problems
- Conclusions

Use the theory to answer many questions: **How can we...**

- ... build interpretable models?
- ... combine Bayesian techniques & deep models?
- ... practically use deep learning uncertainty in existing models?
- ... extend deep learning in a principled way?

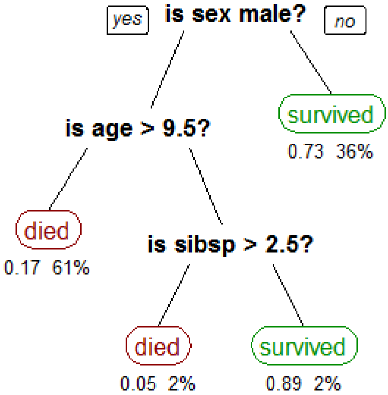
Many unanswered questions left

- Interpretable models?
  - Will *you* trust a decision made by a black-box?

  - Rich literature in interpretable Bayesian models (e.g. Sun (2006), Letham (2014))
  - Combine Bayesian and deep models in a principled way?
- Combine Bayesian techniques & deep models?
  - Unsupervised learning – Bayesian data analysis?
  - Bayesian models with complex data? (sequence data, image data)

- Interpretable models?
  - Will *you* trust a decision made by a black-box?

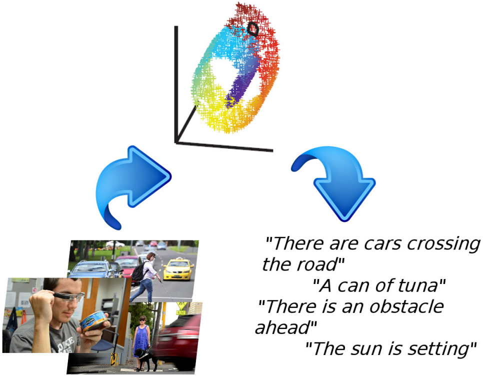
  - Rich literature in interpretable Bayesian models (e.g. Sun (2006), Letham (2014))



Many unanswered questions left

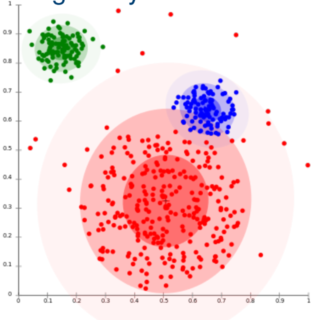
UNIVERSITY OF CAMBRIDGE

- Interpretable models?
  - Will *you* trust a decision made by a black-box?
  - Rich literature in interpretable Bayesian models (e.g. Sun (2006), Letham (2014))
  - Combine Bayesian and deep models in a principled way?



Many unanswered questions left

- Interpretable models?
  - Will *you* trust a decision made by a black-box?
  - Rich literature in interpretable Bayesian models (e.g. Sun (2006), Letham (2014))
  - Combine Bayesian and deep models in a principled way?
- Combine Bayesian techniques & deep models?
  - Unsupervised learning – Bayesian data analysis?



Many unanswered questions left

- Interpretable models?
  - Will *you* trust a decision made by a black-box?
  - Rich literature in interpretable Bayesian models (e.g. Sun (2006), Letham (2014))
  - Combine Bayesian and deep models in a principled way?
- Combine Bayesian techniques & deep models?
  - Unsupervised learning – Bayesian data analysis?
  - Bayesian models with complex data? (sequence data, image data)

Many unanswered questions left

- Practical deep learning uncertainty?
  - Capture language ambiguity?

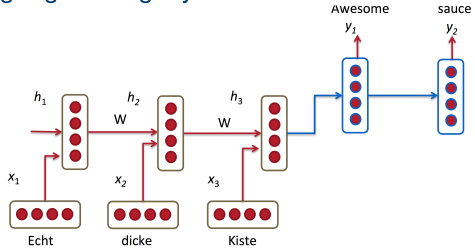
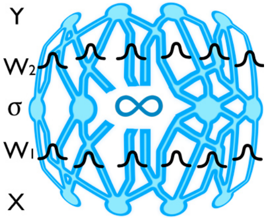


Image Source: `cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf`

  - Weight uncertainty for model debugging?
- Principled extensions of deep learning?
  - Dropout in recurrent networks?
  - New appr. distributions = new stochastic reg. techniques?
  - Model compression: $W_i \sim$ discrete distribution w. continuous base measure?

Many unanswered questions left

- Practical deep learning uncertainty?
  - Capture language ambiguity?
  - Weight uncertainty for model debugging?



- Principled extensions of deep learning?
  - Dropout in recurrent networks?
  - New appr. distributions = new stochastic reg. techniques?
  - Model compression: $W_i \sim$ discrete distribution w. continuous base measure?

Many unanswered questions left

- Practical deep learning uncertainty?
  - Capture language ambiguity?
  - Weight uncertainty for model debugging?
- Principled extensions of deep learning?
  - Dropout in recurrent networks?

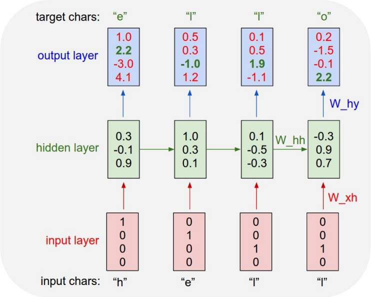


Image Source: `karpathy.github.io/2015/05/21/rnn-effectiveness`

  - New appr. distributions = new stochastic reg. techniques?

Many unanswered questions left

- Practical deep learning uncertainty?
  - Capture language ambiguity?
  - Weight uncertainty for model debugging?
- Principled extensions of deep learning?
  - Dropout in recurrent networks?
  - New appr. distributions = new stochastic reg. techniques?

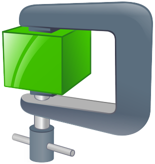
$$q_\theta(\omega) = ?$$

  - Model compression: $\mathbf{W}_i \sim$ discrete distribution w. continuous base measure?

Many unanswered questions left

UNIVERSITY OF CAMBRIDGE

- Practical deep learning uncertainty?
  - Capture language ambiguity?
  - Weight uncertainty for model debugging?
- Principled extensions of deep learning?
  - Dropout in recurrent networks?
  - New appr. distributions = new stochastic reg. techniques?
  - Model compression: $\mathbf{W}_i \sim$ discrete distribution w. continuous base measure?



Many unanswered questions left

- Practical deep learning uncertainty?
  - Capture language ambiguity?
  - Weight uncertainty for model debugging?
- Principled extensions of deep learning?
  - Dropout in recurrent networks?
  - New appr. distributions = new stochastic reg. techniques?
  - Model compression: $\mathbf{W}_i \sim$ discrete distribution w. continuous base measure?

Work in progress!

Outline

- Many unanswered questions
- Why does my model work?
- What does my model know?
- Why does my model predict this and not that, and other open problems
- Conclusions

Conclusions

UNIVERSITY OF CAMBRIDGE

The theory above means that modern deep learning:

- captures stochastic processes underlying observed data
- can use vast Bayesian statistics literature
- can be explained by mathematically rigorous theory
- can be extended in a principled way
- can be combined with Bayesian models / techniques in a practical way (we saw this!)
- has uncertainty estimates built-in (we saw this as well!)

Conclusions

UNIVERSITY OF CAMBRIDGE

The theory above means that modern deep learning:

- captures stochastic processes underlying observed data
- can use vast Bayesian statistics literature
- can be explained by mathematically rigorous theory
- can be extended in a principled way
- can be combined with Bayesian models / techniques in a practical way (we saw this!)
- has uncertainty estimates built-in (we saw this as well!)

Conclusions

The theory above means that modern deep learning:

- captures stochastic processes underlying observed data
- can use vast Bayesian statistics literature
- can be explained by mathematically rigorous theory
- can be extended in a principled way
- can be combined with Bayesian models / techniques in a practical way (we saw this!)
- has uncertainty estimates built-in (we saw this as well!)

Conclusions

UNIVERSITY OF CAMBRIDGE

The theory above means that modern deep learning:

- captures stochastic processes underlying observed data
- can use vast Bayesian statistics literature
- can be explained by mathematically rigorous theory
- can be extended in a principled way
- can be combined with Bayesian models / techniques in a practical way (we saw this!)
- has uncertainty estimates built-in (we saw this as well!)

Conclusions

UNIVERSITY OF CAMBRIDGE

The theory above means that modern deep learning:

- captures stochastic processes underlying observed data
- can use vast Bayesian statistics literature
- can be explained by mathematically rigorous theory
- can be extended in a principled way
- can be combined with Bayesian models / techniques in a practical way (we saw this!)
- has uncertainty estimates built-in (we saw this as well!)

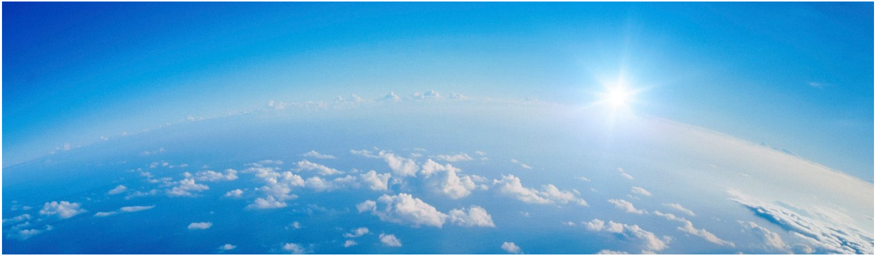
Conclusions

The theory above means that modern deep learning:

- captures stochastic processes underlying observed data
- can use vast Bayesian statistics literature
- can be explained by mathematically rigorous theory
- can be extended in a principled way
- can be combined with Bayesian models / techniques in a practical way (we saw this!)
- has uncertainty estimates built-in (we saw this as well!)

But...

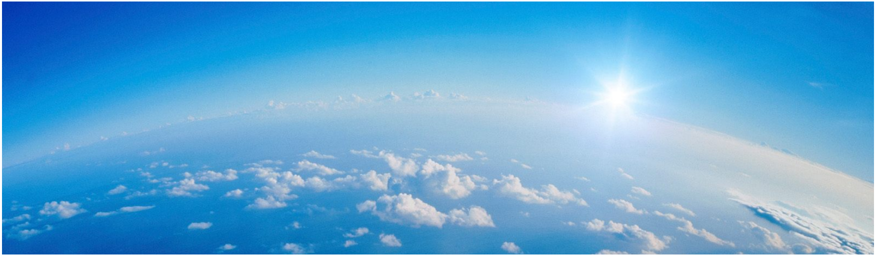
New horizons



Most exciting is work to come:

- **Practical uncertainty** in deep learning
- **Principled extensions** to deep learning
- **Hybrid** deep learning – Bayesian models

and much, much, more.



Most exciting is work to come:

- **Practical uncertainty** in deep learning
- **Principled extensions** to deep learning
- **Hybrid** deep learning – Bayesian models

and much, much, more.

Thank you for listening.

References

- Y Gal, Z Ghahramani, **"Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning"**, arXiv preprint, arXiv:1506.02142 (2015).
- Y Gal, Z Ghahramani, **"Dropout as a Bayesian Approximation: Appendix"**, arXiv preprint, arXiv:1506.02157 (2015).
- Y Gal, Z Ghahramani, **"Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference"**, arXiv preprint, arXiv:1506.02158 (2015).
- A Kendall, R Cipolla, **"Modelling Uncertainty in Deep Learning for Camera Relocalization"**, arXiv preprint, arXiv:1509.05909 (2015)
- C Angermueller and O Stegle, **"Multi-task deep neural network to predict CpG methylation profiles from low-coverage sequencing data"**, NIPS MLCB workshop (2015).
- JM Hernndez-Lobato, RP Adams, **"Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks"**, ICML (2015).
- DP Kingma, T Salimans, M Welling, **"Variational Dropout and the Local Reparameterization Trick"**, NIPS (2015).
- DJ Rezende, S Mohamed, D Wierstra, **"Stochastic Backpropagation and Approximate Inference in Deep Generative Models"**, ICML (2014).

Refs – Bayesian neural networks

UNIVERSITY OF CAMBRIDGE

- Denker, Schwartz, Wittner, Solla, Howard, Jackel, and Hopfield, **"Large Automatic Learning, Rule Extraction, and Generalization"**, Complex Systems (1987).
- Tishby, Levin, and Solla, **"A statistical approach to learning and generalization in layered neural networks"**, COLT (1989).
- Denker and LeCun, **"Transforming neural-net output levels to probability distributions"**, NIPS (1991).
- D MacKay, **"A practical Bayesian framework for backpropagation networks"**, Neural Computation (1992).
- GE Hinton and D van Camp, **"Keeping the neural networks simple by minimizing the description length of the weights"**, Computational learning theory (1993).
- R Neal, **"Bayesian Learning for Neural Networks"**, PhD dissertation (1995).
- D Barber and CM Bishop, **"Ensemble learning in Bayesian neural networks"**, Computer and Systems Sciences, (1998).
- A Graves, **"Practical variational inference for neural networks"**, NIPS (2011).
- C Blundell, J Cornebise, K Kavukcuoglu, and D Wierstra, **"Weight uncertainty in neural networks"**, ICML (2015).

Refs – importance of being uncertain

- Krzywinski and Altman, **“Points of significance: Importance of being uncertain”**, Nature Methods (2013).
- Herzog and Ostwald, **“Experimental biology: Sometimes Bayesian statistics are better”**, Nature (2013).
- Nuzzo, **“Scientific method: Statistical errors”**, Nature (2014).
- Woolston, **“Psychology journal bans P values”**, Nature (2015).
- Ghahramani, **“Probabilistic machine learning and artificial intelligence”**, Nature (2015).