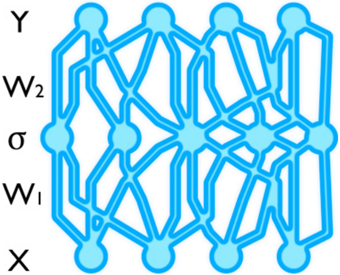
UNIVERSITY OF CAMBRIDGE

Dropout as a Bayesian Approximation

Yarin Gal • Zoubin Ghahramani

yg279@cam.ac.uk

Modern deep learning



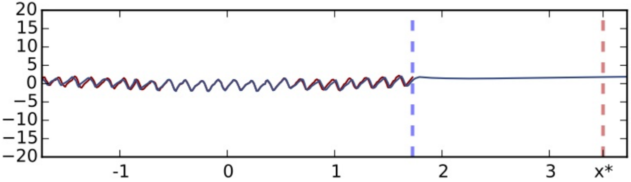
Conceptually simple models...

- 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.

- **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*?



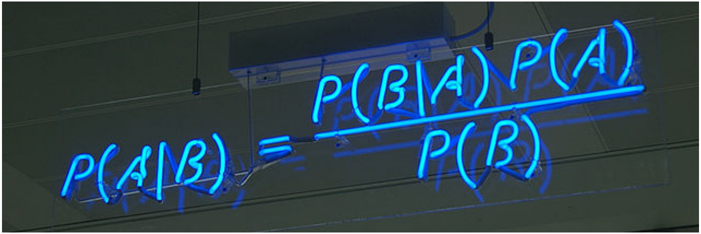
But important unanswered questions...

UNIVERSITY OF CAMBRIDGE

- **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*?



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

Bayesian modelling and inference

UNIVERSITY OF CAMBRIDGE

- 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

Bayesian modelling and inference

- 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

UNIVERSITY OF CAMBRIDGE

- 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

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

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

- Identical to minimising

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

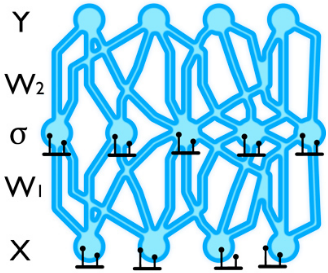
- We can approximate the **predictive distribution**

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

What to this and deep learning?

We'll look at dropout specifically:

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



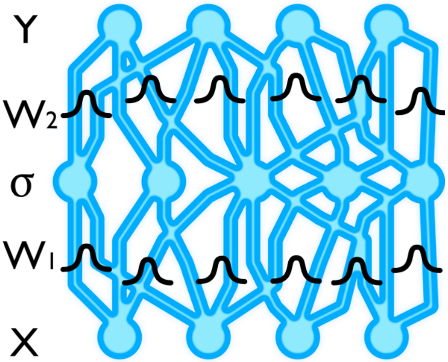
- It somehow circumvents **over-fitting**
- And improves **performance**

The link — Bayesian neural networks

- Place prior $p(\mathbf{w}_{ik})$:

$$p(\mathbf{w}_{ik}) \propto e^{-\frac{1}{2}\mathbf{w}_{ik}^T \mathbf{w}_{ik}}$$

for layer i and column k (and write $\omega := \{\mathbf{w}_{ik}\}_{i,k}$).



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

The link — Bayesian neural networks

- Place prior $p(\mathbf{w}_{ik})$:

$$p(\mathbf{w}_{ik}) \propto e^{-\frac{1}{2}\mathbf{w}_{ik}^T \mathbf{w}_{ik}}$$

  for layer i and column k (and write $\boldsymbol{\omega} := \{\mathbf{w}_{ik}\}_{i,k}$).

- 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...

- Place prior $p(\mathbf{w}_{ik})$:

$$p(\mathbf{w}_{ik}) \propto e^{-\frac{1}{2}\mathbf{w}_{ik}^T \mathbf{w}_{ik}}$$

  for layer i and column k (and write $\omega := \{\mathbf{w}_{ik}\}_{i,k}$).

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

- 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...

- Place prior $p(\mathbf{w}_{ik})$:

$$p(\mathbf{w}_{ik}) \propto e^{-\frac{1}{2}\mathbf{w}_{ik}^T \mathbf{w}_{ik}}$$

  for layer i and column k (and write $\omega := \{\mathbf{w}_{ik}\}_{i,k}$).

- 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

UNIVERSITY OF CAMBRIDGE

- 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?

UNIVERSITY OF CAMBRIDGE

- Approximate posterior $p(\omega|\mathbf{X},\mathbf{Y})$ with $q_\theta(\omega)$ (def later)
- KL divergence to minimise:

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

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

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

- Approximate posterior $p(\omega|\mathbf{X},\mathbf{Y})$ with $q_\theta(\omega)$ (def later)
- KL divergence to minimise:

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

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

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

- Approximate posterior $p(\omega|\mathbf{X},\mathbf{Y})$ with $q_\theta(\omega)$ (def later)
- KL divergence to minimise:

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

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

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

- Unbiased estimator:

$$\mathbb{E}_{\widehat{\boldsymbol{\omega}} \sim q_\theta(\boldsymbol{\omega})}\left(\widehat{\mathcal{L}}(\theta)\right) = \mathcal{L}(\theta)$$

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

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

  w.r.t. θ.

- Unbiased estimator:

$$\mathbb{E}_{\widehat{\boldsymbol{\omega}} \sim q_\theta(\boldsymbol{\omega})}\big(\widehat{\mathcal{L}}(\theta)\big) = \mathcal{L}(\theta)$$

- 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.

- Unbiased estimator:

$$\mathbb{E}_{\widehat{\omega}\sim q_\theta(\omega)}\big(\widehat{\mathcal{L}}(\theta)\big) = \mathcal{L}(\theta)$$

- 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.

Specifying q()

- Given variational parameters $\theta = \{\mathbf{m}_{ik}\}_{i,k}$:

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

$$q_\theta(\mathbf{W}_i) = \prod_k q_{\mathbf{m}_{ik}}(\mathbf{w}_{ik})$$

$$q_{\mathbf{m}_{ik}}(\mathbf{w}_{ik}) = p\delta_{\mathbf{0}}(\mathbf{w}_{ik}) + (1-p)\delta_{\mathbf{m}_{ik}}(\mathbf{w}_{ik})$$

$\rightarrow$ k'th column of the i'th layer is a mixture of two components

- Or, in a more compact way:

$$\mathbf{z}_{ik} \sim \text{Bernoulli}(p_i) \text{ for each layer } i \text{ and column } k$$

$$\mathbf{W}_i = \mathbf{M}_i \cdot \text{diag}([\mathbf{z}_{ik}]_{k=1}^{K})$$

with $\mathbf{z}_{ik}$ Bernoulli r.v.s.

UNIVERSITY OF CAMBRIDGE

In summary:

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

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

$$\widehat{\mathbf{W}}_i = \mathbf{M}_i \cdot \text{diag}([\widehat{\mathbf{z}}_{ik}]_{k=1}^K)$$

$$\widehat{\omega} = \{\widehat{\mathbf{W}}_i\}_{i=1}^L$$

  - Minimise (one step)

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

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 columns of $\mathbf{M}_i$ to zero
  - Minimise (one step)

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

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

In summary:

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

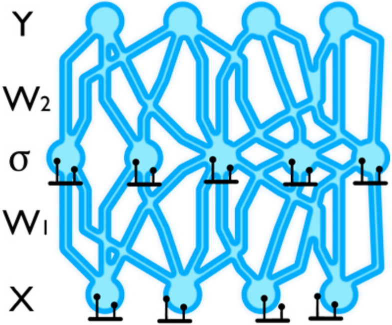
- Repeat:
  - = Randomly set units of the network to zero
  - Minimise (one step)

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

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

Deep learning as approx. inference

Sounds familiar?



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

Implementing VI with $q_\theta(\cdot)$ above = implementing dropout in deep network

Diving deeper into dropout

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time ≈ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i\mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

UNIVERSITY OF CAMBRIDGE

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

Diving deeper into dropout

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

UNIVERSITY OF CAMBRIDGE

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i\mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

Diving deeper into dropout

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

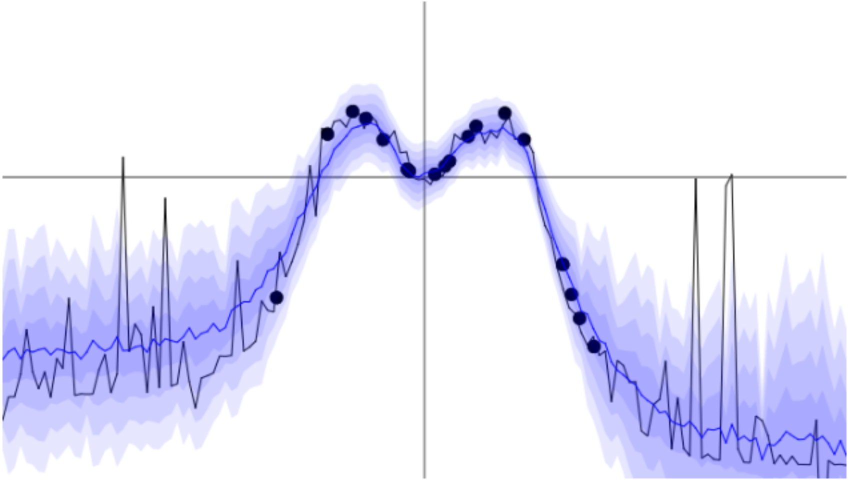
Diving deeper into dropout

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

Diving deeper into dropout

- We **fit to the distribution** that generated our observed data, not just its mean
- What can we say about $q_\theta(\cdot)$?
  - Many Bernoullis = cheap multi-modality
  - Dropout at test time $\approx$ propagate the mean $\mathbb{E}(\mathbf{W}_i) = p_i \mathbf{M}_i$
  - Strong correlations between function frequencies, indp. across output dimensions
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

We fit a **distribution**...



We fit a **distribution**...

- Use first moment for **predictions**:

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

with $\widehat{\mathbf{y}}_t \sim \text{DropoutNetwork}(\mathbf{x}^*)$.

- Use second moment for **uncertainty** (in regression):

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

with $\widehat{\mathbf{y}}_t \sim \text{DropoutNetwork}(\mathbf{x}^*)$.

UNIVERSITY OF CAMBRIDGE

We fit a **distribution**...

- Use first moment for **predictions**:

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

with $\widehat{\mathbf{y}}_t \sim \text{DropoutNetwork}(\mathbf{x}^*)$.

- Use second moment for **uncertainty** (in regression):

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

with $\widehat{\mathbf{y}}_t \sim \text{DropoutNetwork}(\mathbf{x}^*)$.

Bayesian eval. – some code

In more practical terms, given point x:[1]

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

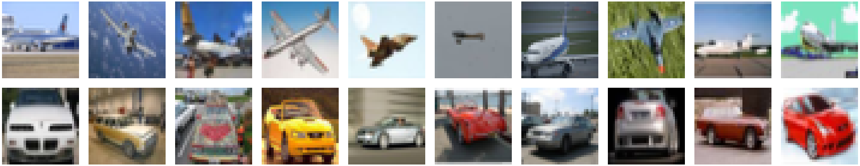
```
y = []
for _ in xrange(10):
    y.append(model.output(x, dropout=True))
y_mean = numpy.mean(y)
y_var = numpy.var(y)
```

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

Using the first moment

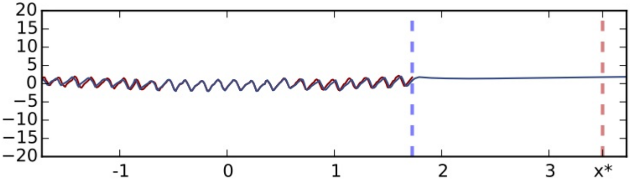
	CIFAR Test Error (and Std.)	
Model	**Standard Dropout**	**Bayesian technique**
NIN	10.43 (Lin et al., 2013)	$\mathbf{10.27 \pm 0.05}$
DSN	9.37 (Lee et al., 2014)	$\mathbf{9.32 \pm 0.02}$
Augmented-DSN	7.95 (Lee et al., 2014)	$\mathbf{7.71 \pm 0.09}$

Table : Bayesian techniques with existing state-of-the-art



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

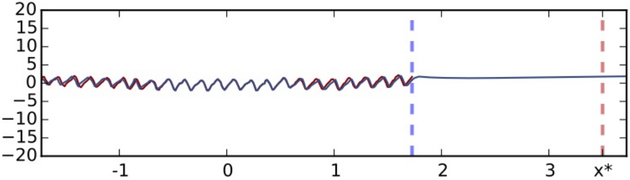
- Normal dropout (weight averaging, 5 layers, ReLU units):



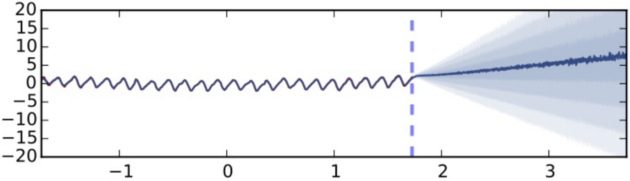
- Same network, Bayesian perspective:

What 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:



How good is our uncertainty estimate?

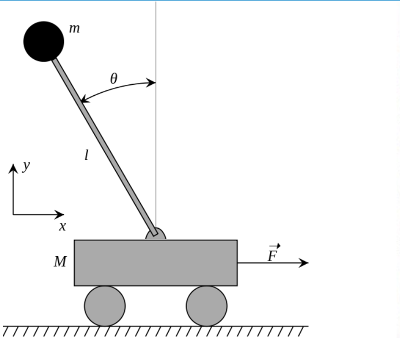
UNIVERSITY OF CAMBRIDGE

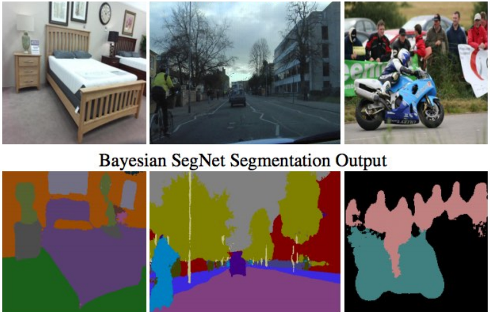
	Avg. Test RMSE and Std. Errors			Avg. Test LL and Std. Errors		
Dataset	**VI**	**PBP**	**Dropout**	**VI**	**PBP**	**Dropout**
Boston Housing	4.32 ±0.29	3.01 ±0.18	**2.97 ±0.85**	-2.90 ±0.07	-2.57 ±0.09	**-2.46 ±0.25**
Concrete Strength	7.19 ±0.12	5.67 ±0.09	**5.23 ±0.53**	-3.39 ±0.02	-3.16 ±0.02	**-3.04 ±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**

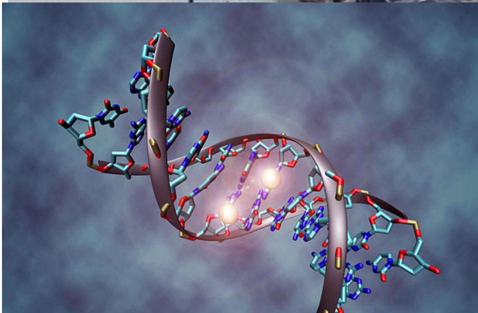
Table 1: **Average test performance in RMSE and predictive log likelihood** for a popular variational inference method (VI, Graves [20]), Probabilistic back-propagation (PBP, Hernández-Lobato and Adams [27]), and dropout uncertainty (Dropout).

Applications

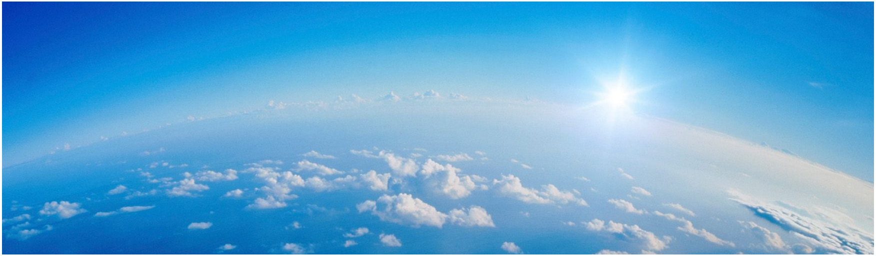
UNIVERSITY OF CAMBRIDGE







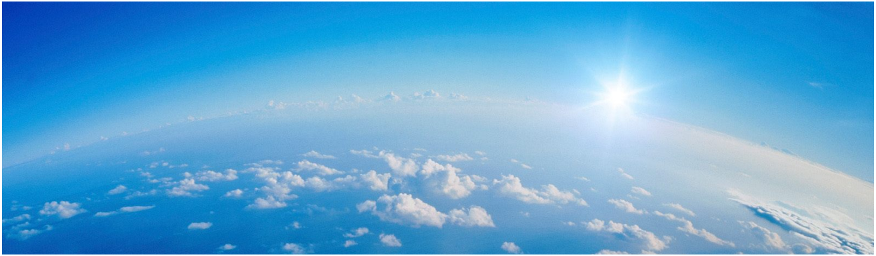
UNIVERSITY OF CAMBRIDGE



Most exciting is work to come:

- Deep learning applications using **practical uncertainty estimates**
- **Principled extensions** to deep learning tools
- **Hybrid** deep learning – Bayesian models

and much, much, more.



Most exciting is work to come:

- Deep learning applications using **practical uncertainty estimates**
- **Principled extensions** to deep learning tools
- **Hybrid** deep learning – Bayesian models

and much, much, more.

Thank you for listening.