UNIVERSITY OF CAMBRIDGE

What my deep model doesn't know...

Yarin Gal

yg279@cam.ac.uk

- We train a model to recognise dog breeds

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

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?

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.[1]

- 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.[1]
- For the practitioner: model debugging, specialised models, critical systems

Why should I care about uncertainty?

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.[1]
- For the practitioner: model debugging, specialised models, critical systems
- We need a way to tell **what our model knows** and what not.
- Luckily, if you use **dropout** you already have this information. You just **ignored** it so far.

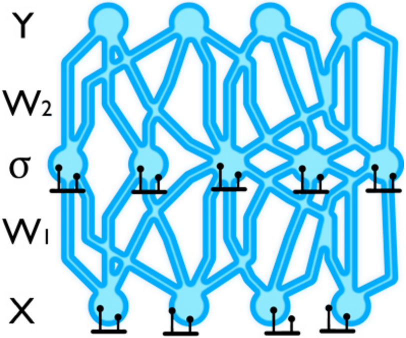
[1] Complete references at end of slides

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.[1]
- For the practitioner: model debugging, specialised models, critical systems
- We need a way to tell **what our model knows** and what not.
- Luckily, if you use **dropout** you already have this information. You just **ignored** it so far.

[1] Complete references at end of slides

- Used in **most modern deep learning models**
- It circumvents **over-fitting**
- And improves **performance**

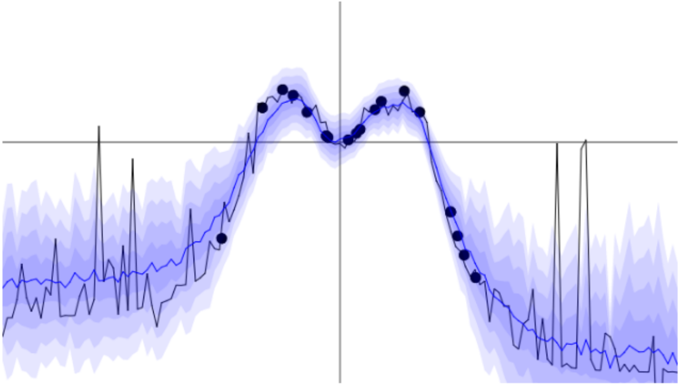


- **Training time:** drop units, **test time:** don't drop

How do we get uncertainty?

Bayesian modelling:

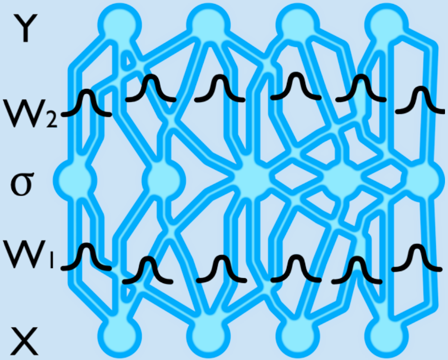
- Observed inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and outputs $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$
- Capture distribution believed to have generated outputs
- Look at the first two moments:



UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

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



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

What to this and dropout?

From Bayesian neural networks to Dropout

- Place **prior** $p(\mathbf{W})$ dist. on weights, making these r.v.s
- Given dataset $\mathbf{X}, \mathbf{Y}$, the r.v. $\mathbf{W}$ has a **posterior**: $p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big)$
- Which is difficult to evaluate...
- Can define **simple distribution** $q_\theta(\cdot)$ and approximate
$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$
- **Inference** with
$$q_\theta(\mathbf{W}) := \mathbf{M} \cdot \text{diag}(\text{Bernoulli})$$
and parameter $\mathbf{M}$

$$= \textbf{Dropout training}.$$

What to this and dropout?

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$

- **Inference** with

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

and parameter $\mathbf{M}$

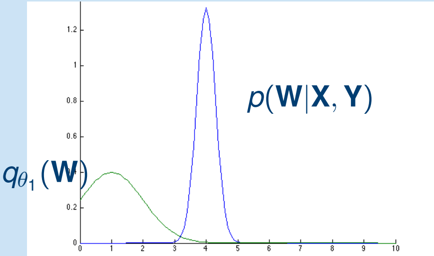
$$= \textbf{Dropout training}.$$

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$



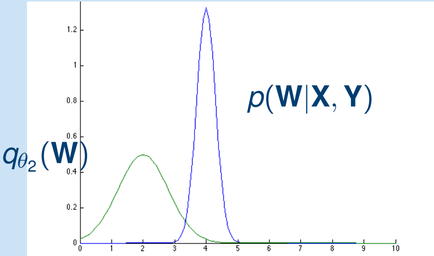
What to this and dropout?

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$



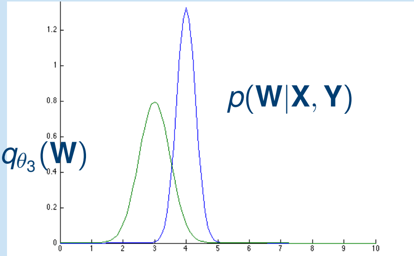
What to this and dropout?

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$

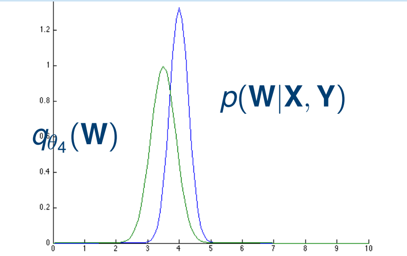


What to this and dropout?

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$

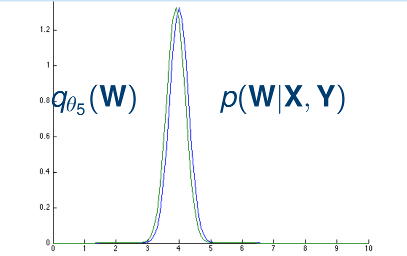


What to this and dropout?

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$



What to this and dropout?

UNIVERSITY OF CAMBRIDGE

From Bayesian neural networks to Dropout

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

$$q_\theta(\mathbf{W}) \approx p\big(\mathbf{W}|\mathbf{X}, \mathbf{Y}\big).$$

- **Inference** with

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

  and parameter $\mathbf{M}$

= **Dropout training**.

The theory above means that with dropout we:

- **capture distribution** that generated observed data
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

What does this mean

The theory above means that with dropout we:

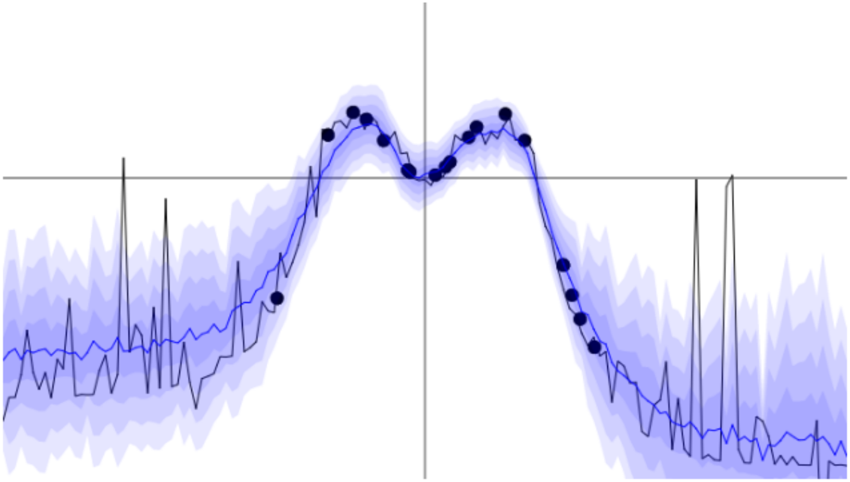
- **capture distribution** that generated observed data
- can combine model with Bayesian techniques in a **practical** way...
- have **uncertainty estimates** in the network

What does this mean

The theory above means that with dropout we:

- **capture distribution** that generated observed data
- 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 $\widehat{\mathbf{y}}_t \sim \text{DropoutNetwork}(\mathbf{x}^*)$.

Bayesian evaluation techniques

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*:[2]

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

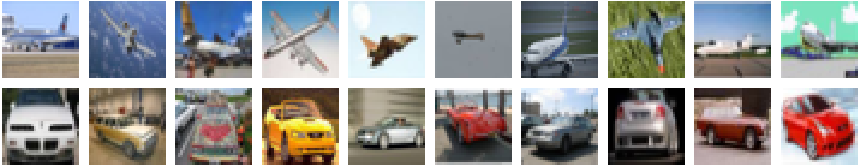
```
y = []
for _ in xrange(10):
    y.append(model.output(x, dropout=True))
y_mean = numpy.mean(y)
y_var = numpy.var(y)
```

[2] 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)	$\mathbf{7.71 \pm 0.09}$

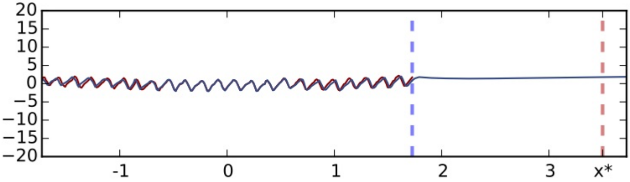
Table : Bayesian techniques with existing state-of-the-art



UNIVERSITY OF CAMBRIDGE

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

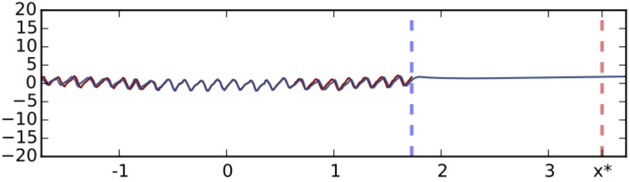
- Normal dropout (weight averaging, 5 layers, ReLU units):



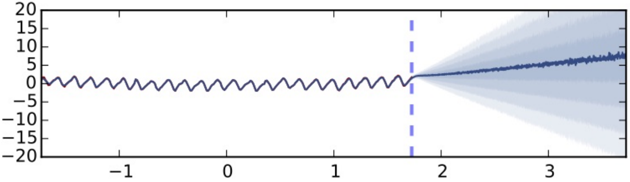
- Same network, Bayesian perspective:

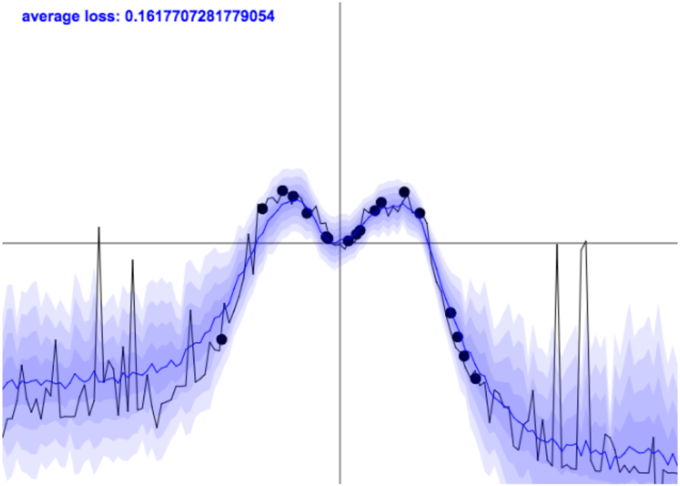
What would be the CO_2 concentration level in Mauna Loa, Hawaii, *in 20 years' time*?

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



- Same network, Bayesian perspective:





[Online demo] [3]

[3] yarin.co/blog

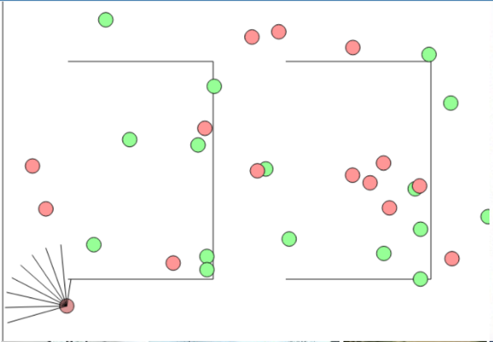
How good is our uncertainty estimate?

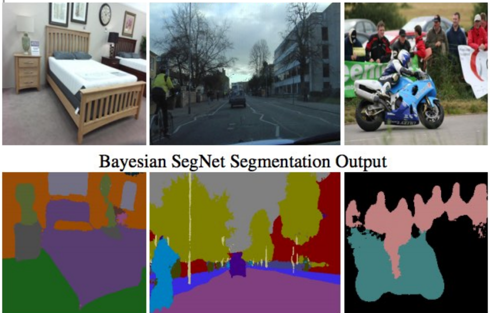
UNIVERSITY OF CAMBRIDGE

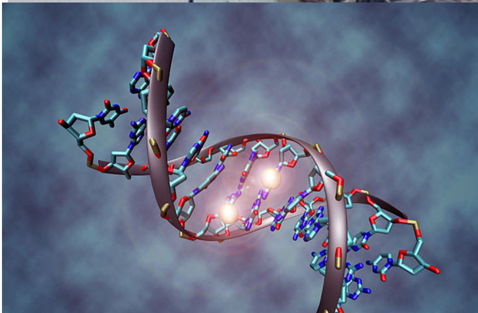
Dataset	Avg. Test RMSE and Std. Errors VI	PBP	Dropout	Avg. Test LL and Std. Errors VI	PBP	Dropout
Boston Housing	4.32 ±0.29	3.01 ±0.18	**2.97 ±0.85**	-2.90 ±0.07	-2.57 ±0.09	**-2.46 ±0.25**
Concrete Strength	7.19 ±0.12	5.67 ±0.09	**5.23 ±0.53**	-3.39 ±0.02	-3.16 ±0.02	**-3.04 ±0.09**
Energy Efficiency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

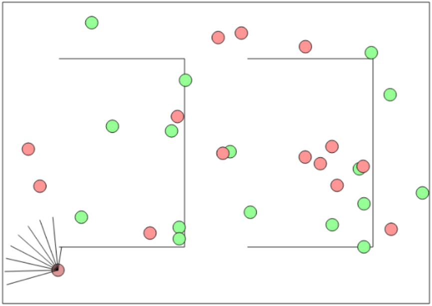
Applications







- We have a "Roomba"[4]
- 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



[4] 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:

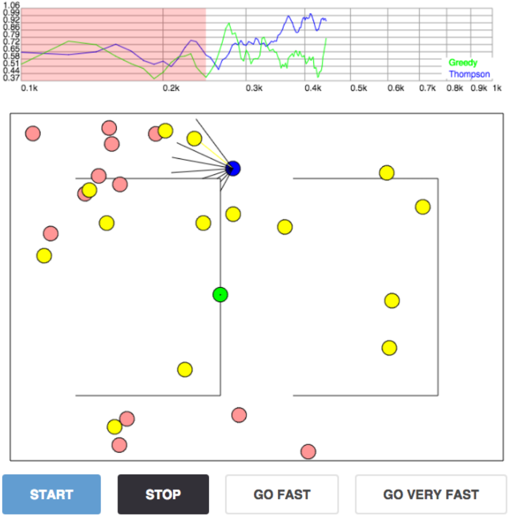
- **Epsilon-greedy** – take random actions with probability ϵ and optimal actions otherwise
- Using uncertainty we can learn faster
- **Thompson sampling** – draw realisation from current belief over world, choose action with highest value
- In practice: simulate a stochastic forward pass through the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

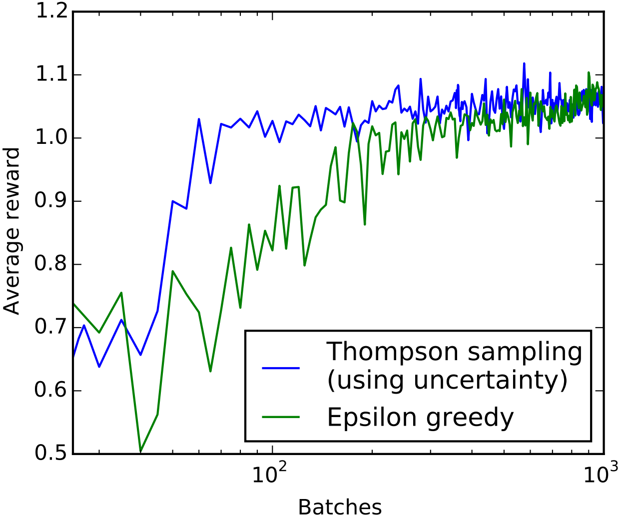
Behavioural policies:

- **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] [5]

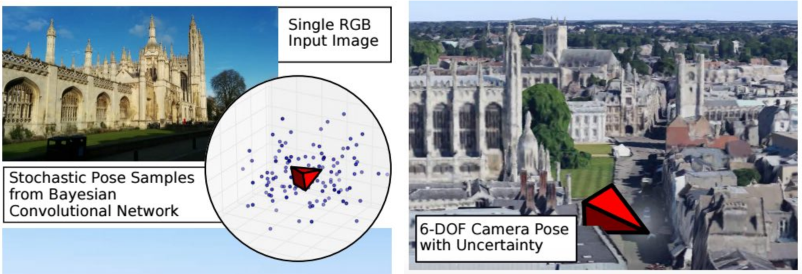
[5] yarin.co/blog



Average reward over time (log scale)

Camera pose estimation

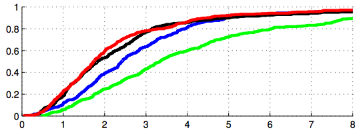
- Where was a picture taken? (Kendall and Cipolla, 2015)[6]

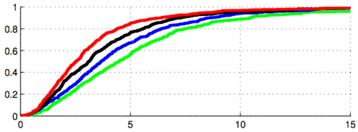


- With Bayesian techniques above: **10–15%** improvement on **state-of-the-art**
- Uncertainty increases as a test photo diverges from training distribution
- Test photos with high uncertainty (strong occlusion from vehicles, pedestrians or other objects)

Camera pose estimation

- Where was a picture taken? (Kendall and Cipolla, 2015)[6]
- With Bayesian techniques above: **10–15%** improvement on **state-of-the-art**



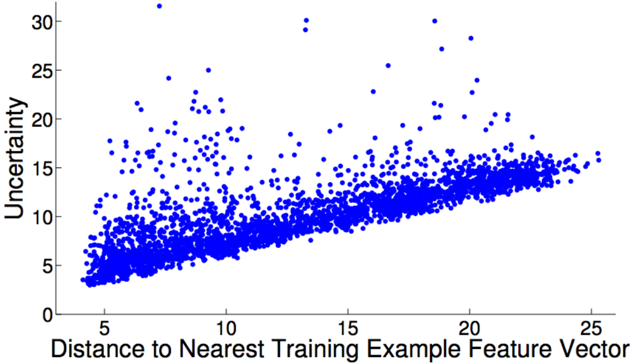


- Uncertainty increases as a test photo diverges from training distribution
- Test photos with high uncertainty (strong occlusion from vehicles, pedestrians or other objects)
- Localisation error correlates with uncertainty

[6] Figures used with author permission

Camera pose estimation

- Where was a picture taken? (Kendall and Cipolla, 2015)[6]
- With Bayesian techniques above: **10–15%** improvement on **state-of-the-art**
- Uncertainty increases as a test photo diverges from training distribution



Camera pose estimation

- Where was a picture taken? (Kendall and Cipolla, 2015)[6]
- With Bayesian techniques above: **10–15%** improvement on **state-of-the-art**
- 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

- Where was a picture taken? (Kendall and Cipolla, 2015)[6]
- With Bayesian techniques above: **10–15%** improvement on **state-of-the-art**
- 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

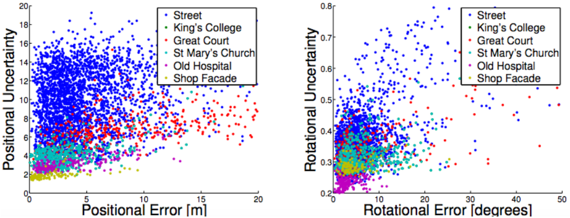
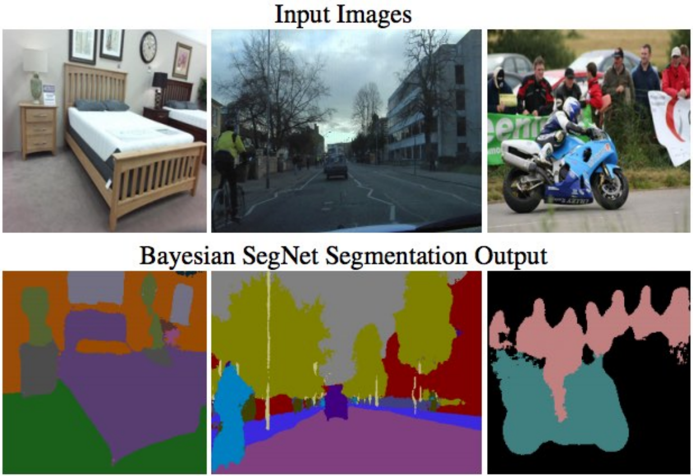


Image segmentation

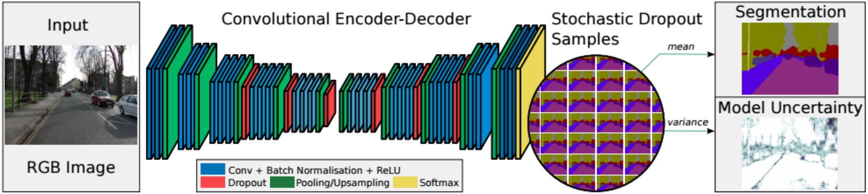
- Scene understanding: what's in a photo and where? (Kendall, Badrinarayanan, and Cipolla, 2015)[7]



[7] Figures used with author permission

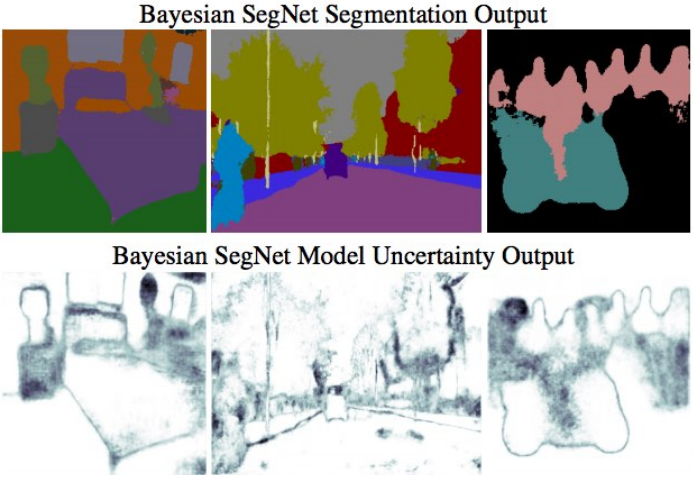
Image segmentation

- Scene understanding: what's in a photo and where? (Kendall, Badrinarayanan, and Cipolla, 2015)[7]



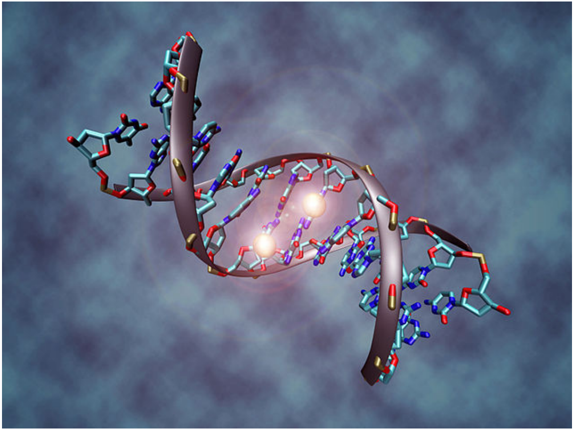
[7] Figures used with author permission

- Scene understanding: what's in a photo and where? (Kendall, Badrinarayanan, and Cipolla, 2015)[7]



[7] Figures used with author permission

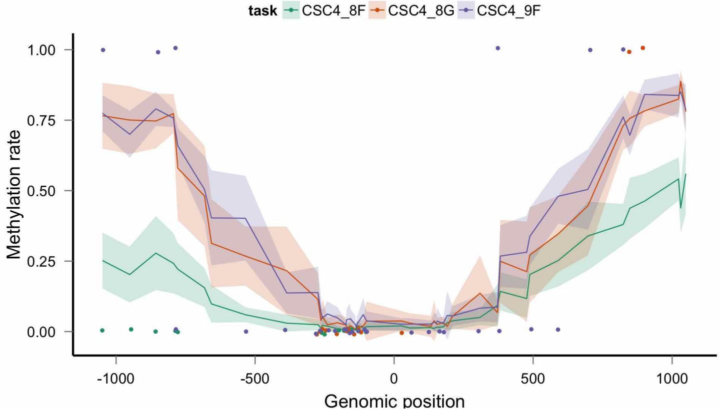
- Angermueller and Stegle (2015) fit a network to predict **DNA methylation** – controls gene regulation



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

Model confidence in bioinformatics

- Angermueller and Stegle (2015) fit a network to predict **DNA methylation** – controls 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)



UNIVERSITY OF CAMBRIDGE

If you use **dropout** you already have uncertainty information = **practical** deep learning uncertainty.

- Applications: capture language ambiguity?

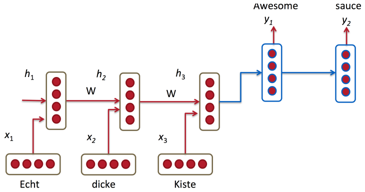
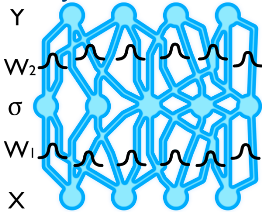


Image Source: cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf

- Tools: weight uncertainty for model debugging?



What's next

If you use dropout you already have uncertainty information = practical deep learning uncertainty.

- Applications: capture language ambiguity?

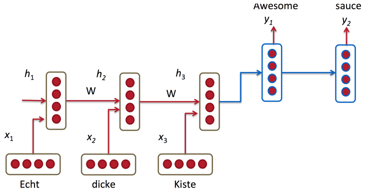
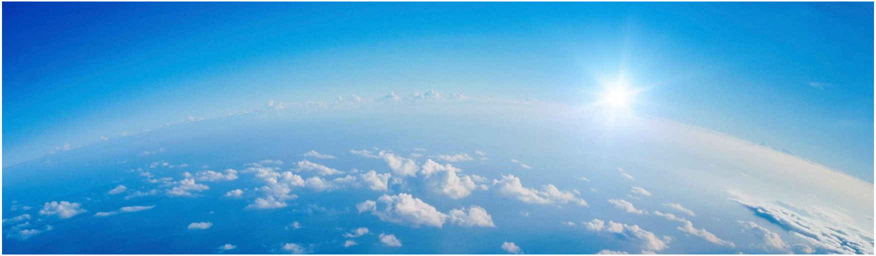


Image Source: `cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf`

- Tools: weight uncertainty for model debugging?

Work in progress!

New horizons

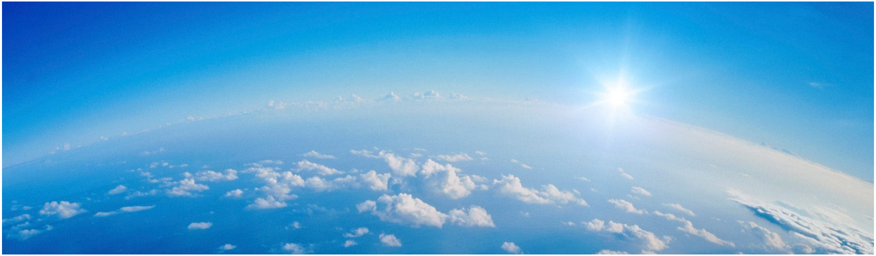


Most exciting is work to come:

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

and much, much, more.

New horizons



Most exciting is work to come:

- **Practical uncertainty** in deep learning applications
- **Principled extensions** to deep learning tools
- **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 – 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).