Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

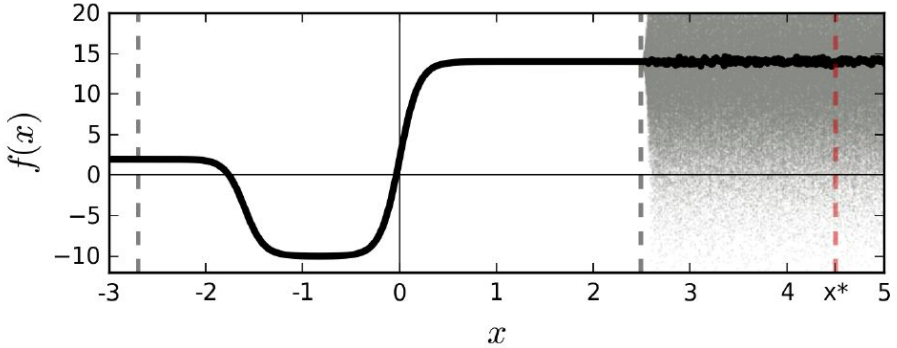
Yarin Gal
Zoubin Ghahramani

JMLR-2016

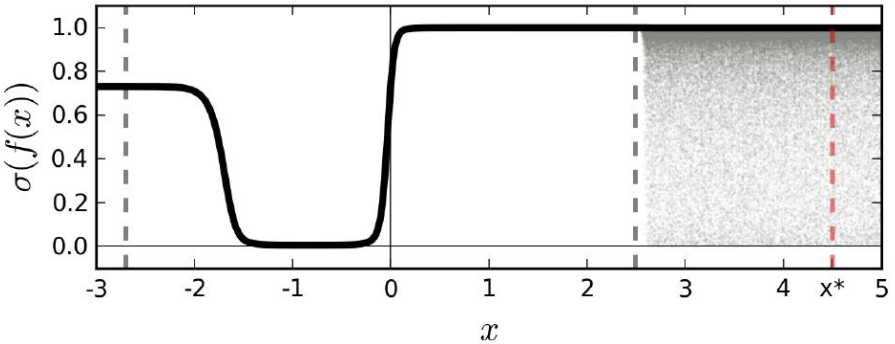
- Preliminary concepts
- Dropout as Approximation
- Experiments

Model Uncertainty in Deep Learning

- ➔ Model is uncertain about a test point beyond the range of training data
- ➔ Still give a point estimate with high confidence



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



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

Figure 1. **A sketch of softmax input and output for an idealised binary classification problem.** Training data is given between the dashed grey lines. Function point estimate is shown with a solid line. Function uncertainty is shown with a shaded area. Marked with a dashed red line is a point x^* far from the training data. Ignoring function uncertainty, point x^* is classified as class 1 with probability 1.

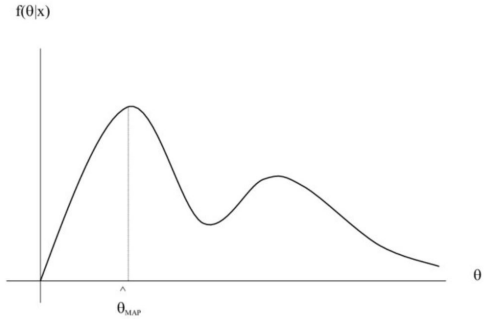
Necessity of Evaluating Uncertainty

- ➔ Treat uncertain inputs and special cases explicitly
  - ◆ In case of uncertain classification, involve human to check
    - In post office, check the characters of the zipcode to sort
  - ◆ In reinforcement learning (RL)
    - With uncertainty information an agent can decide when to exploit and when to explore its environment.

Bayesian Inference

$$posterior = \frac{likelihood * prior}{evidence}, \; or \; p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{p(\mathcal{D})}$$

$$evidence = p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)dw$$



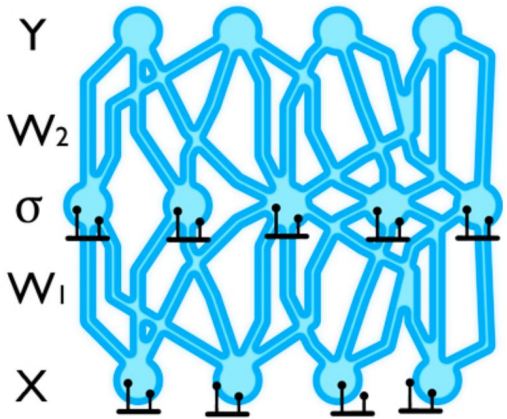
<u>Maximum A Posteriori (MAP) Estimation :</u>

- ➔ Train NN and get optimum $\hat{w}$
- ➔ Compute MAP

<u>Full Predictive Distribution :</u>

$$p(y|\mathcal{D}, x) = \int p(y|w, x)p(w|\mathcal{D})dw$$

Dropout in NN as a regularizer



for each layer i = 1,...,L,

- ➔ *K_i* units
- ➔ *W_i* weight matrices of dimensions $K_i \times K_{i-1}$

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

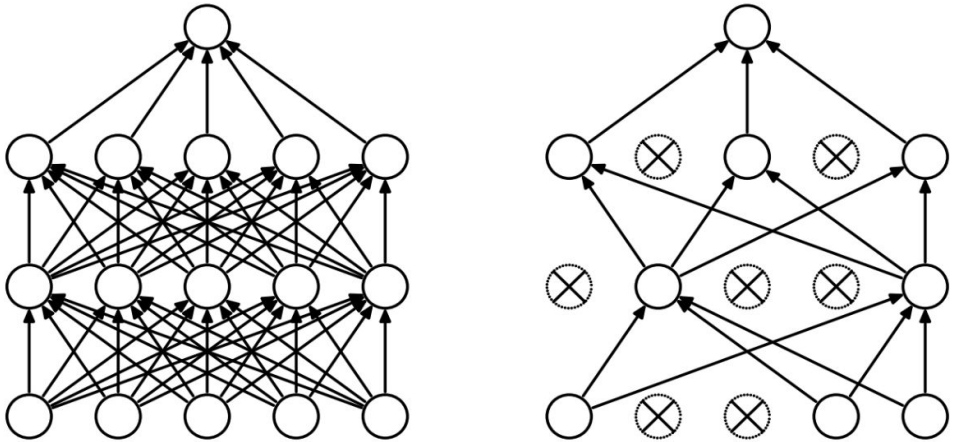
$$\mathbf{z}_{i,j} \sim \text{Bernoulli}(p_i) \text{ for } i = 1, ..., L,\ j = 1, ..., K_{i-1}$$

$\mathbf{z}_{i,j} = 0$ unit *j* in layer *i -1* being dropped out as an input to layer *i*

Dropout in NN as a regularizer

Minimize objective function :

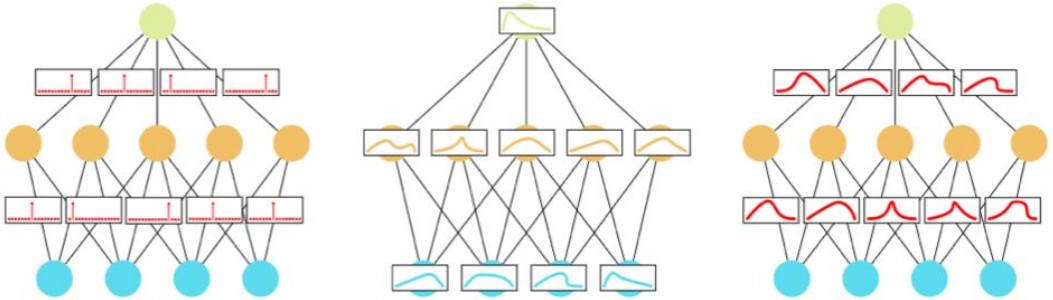
$$\mathcal{L}_{\text{dropout}} := \frac{1}{N}\sum_{i=1}^{N} E(\mathbf{y}_i, \widehat{\mathbf{y}}_i) + \lambda \sum_{i=1}^{L} \left(||\mathbf{W}_i||_2^2 + ||\mathbf{b}_i||_2^2 \right)$$



(a) Standard Neural Net (b) After applying dropout.

In the original dropout mechanism, some neurons are randomly shut down during training. Srivastava et al. (2014), Figure 1.

Bayesian Neural Network (apply probability distribution)



(a) (b) (c)

Fig. 3: (a) Point estimate neural network, (b) stochastic neural network with a probability distribution for the activations, and (c) stochastic neural network with a probability distribution over the weights.

Probability distribution over NN

function $\mathbf{y} = \mathbf{f}(\mathbf{x})$

Prior : $p(\mathbf{f})$

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{f})p(\mathbf{f})$$

Gaussian Processes

- ➔ A process to model distributions over functions
- ➔ large neural networks ≡ Gaussian processes
- ➔ Given a training dataset of *N* -
  - Inputs **X =** $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
  - Corresponding outputs **: Y =** $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$
- ➔ Goal : estimate function $y = f(x)$
- ➔ Following the Bayesian approach

posterior distribution over the space of functions, given our dataset (**X,Y**)

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \mathbf{f})p(\mathbf{f})$$

$p(\mathbf{f})$ - *prior distribution* over the space of functions

Gaussian Processes

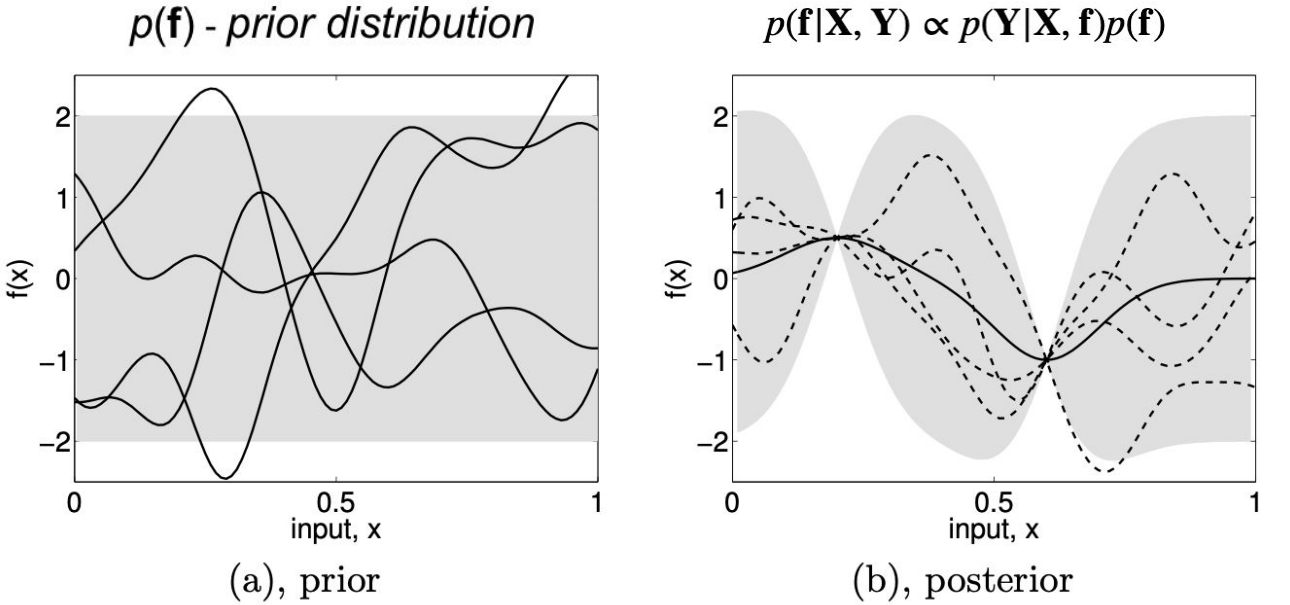


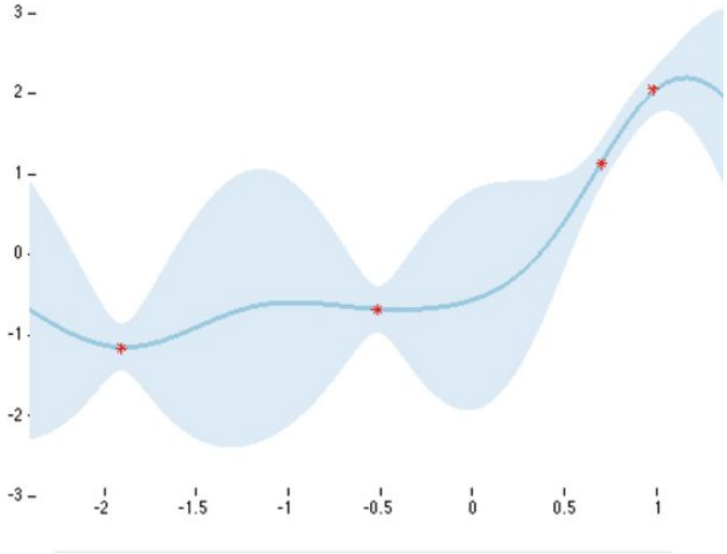
Figure 1.1: Panel (a) shows four samples drawn from the prior distribution. Panel (b) shows the situation after two datapoints have been observed. The mean prediction is shown as the solid line and four samples from the posterior are shown as dashed lines. In both plots the shaded region denotes twice the standard deviation at each input value x.

Gaussian Processes (Example)

Posterior with 4 data points

- ➔ Covariance function - squared exponential
- ➔ Predictive mean - bold blue line
- ➔ Predictive uncertainty - light blue shape

- ➔ Model uncertainty
  - ◆ Small near the data
  - ◆ Increases as we move away from data-points



Posterior in Gaussian Processes :

➔ Posterior distribution is a joint Gaussian distribution over all function values

$$\mathbf{F} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\mathbf{X}, \mathbf{X}))$$
$$\mathbf{Y} \mid \mathbf{F} \sim \mathcal{N}(\mathbf{F}, \tau^{-1}\mathbf{I}_N)$$

➔ **K**(**X**,**X**) is covariance function

defines the similarity between every pair of input points $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \int p(\mathbf{w})p(b)\sigma(\mathbf{w}^T\mathbf{x} + b)\sigma(\mathbf{w}^T\mathbf{y} + b)\mathrm{d}\mathbf{w}\mathrm{d}b$$

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

Then predictive probability (marginalized over weights)

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})\mathrm{d}\boldsymbol{\omega}$$

Predictive Probability in Gaussian Processes :

➔ predictive probability

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) \mathrm{d}\boldsymbol{\omega}$$

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}\left(\mathbf{y}; \widehat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\omega}), \tau^{-1}\mathbf{I}_D\right)$$

$p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y})$ —> Intractable

need to approximate

- Preliminary concepts
- Dropout as Approximation
- Experiments

Approximating Posterior

- ➔ variational distribution $q(\omega)$ approximates posterior $p(\omega|\mathbf{X}, \mathbf{Y})$
- ➔ By minimising the KL divergence

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

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

According to Gaussian Process properties

$$\mathrm{KL}(q(\omega)||p(\omega)) = \sum_{i=1}^{L} \left(\frac{p_i l^2}{2\tau N} ||\mathbf{M}_i||_2^2 + \frac{l^2}{2\tau N} ||\mathbf{m}_i||_2^2 \right)$$

Approximating Posterior

$$-\int q(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} = -\sum_{n=1}^{N} \int q(\boldsymbol{\omega}) \log p(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}$$

approximate each term in the sum by Monte Carlo integration sampling from $q(\boldsymbol{\omega})$

$$\text{sample } \widehat{\boldsymbol{\omega}}_n \sim q(\boldsymbol{\omega})$$

get unbiased estimate $-\log p(\mathbf{y}_n|\mathbf{x}_n, \widehat{\boldsymbol{\omega}}_n)$

$$\mathcal{L}_{\text{GP-MC}} \propto -\log p(\mathbf{y}_n|\mathbf{x}_n, \widehat{\boldsymbol{\omega}}_n) + \mathrm{KL}(q(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

Dropout as a Bayesian Approximation

define $q(\boldsymbol{\omega})$ as:

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

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

sample $\widehat{\boldsymbol{\omega}}_n \sim q(\boldsymbol{\omega})$ $\qquad \widehat{\boldsymbol{\omega}}_n = \{\mathbf{W}_i\}_{i=1}^{L}$

$$\mathcal{L}_{\text{GP-MC}} \propto -\log p(\mathbf{y}_n|\mathbf{x}_n, \widehat{\boldsymbol{\omega}}_n) + \text{KL}(q(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

Obtaining Model Uncertainty

Approximate predictive distribution $$q(\mathbf{y}^*|\mathbf{x}^*) = \int p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}) q(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}$$

sample T sets of vectors from the Bernoulli distribution $\{\mathbf{z}_1^t, ..., \mathbf{z}_L^t\}_{t=1}^T$

giving $\{\mathbf{W}_1^t, ..., \mathbf{W}_L^t\}_{t=1}^T$

Estimate First-Moment : $$\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) \approx \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$$

Estimate Second-Moment : $$\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}\left((\mathbf{y}^*)^T(\mathbf{y}^*)\right) \approx \tau^{-1}\mathbf{I}_D + \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)^T \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$$

Obtaining Model Uncertainty

First-Moment : $\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) \approx \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$

the model precision

$$\tau = \frac{pl^2}{2N\lambda}.$$

Second-Moment : $\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}((\mathbf{y}^*)^T(\mathbf{y}^*)) \approx \tau^{-1}\mathbf{I}_D$

$$+ \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)^T\widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$$

model's predictive variance Model Uncertainty

$$\mathrm{Var}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) \approx$$

$$\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}((\mathbf{y}^*)^T(\mathbf{y}^*)) - \mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*)^T\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*)$$

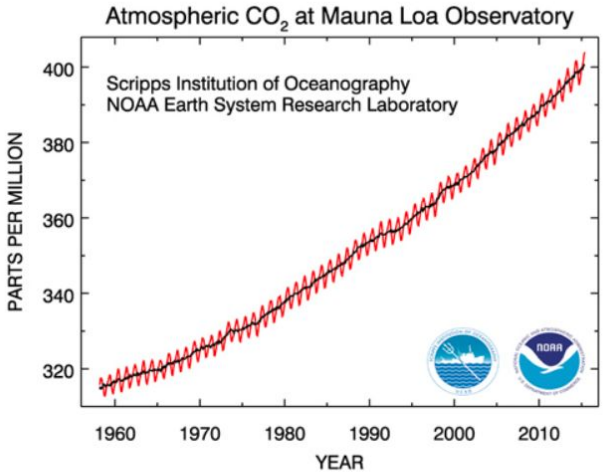
Obtaining Model Uncertainty - Discussion

- ➔ To estimate the predictive mean and predictive uncertainty
  - ◆ Dropout is done in test time - NN model itself is not changed
- ➔ Collect the results of stochastic forward passes through the model
  - ◆ this information can be used with existing NN models trained with dropout
- ➔ The forward passes can be done concurrently
  - ◆ constant running time

- Preliminary concepts
- Dropout as Approximation
- Experiments

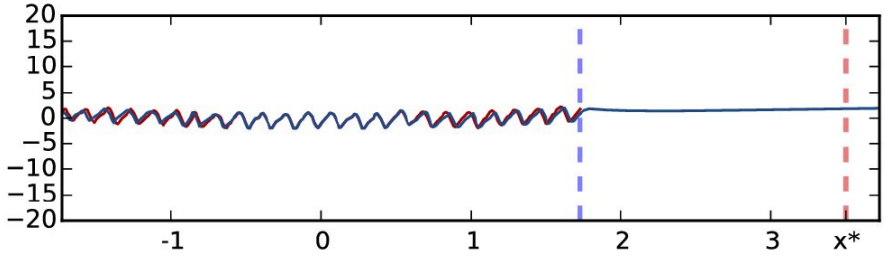
Model Uncertainty in Regression Tasks

- ➔ Dataset - the Mauna Loa CO2 concentrations dataset
- ➔ NN hidden layers - 5
- ➔ NN 1024 hidden units in each layer
- ➔ NN non-linearities - ReLU, TanH
- ➔ dropout probabilities - 0.1/0.2
- ➔ number of forward iterations - 1000

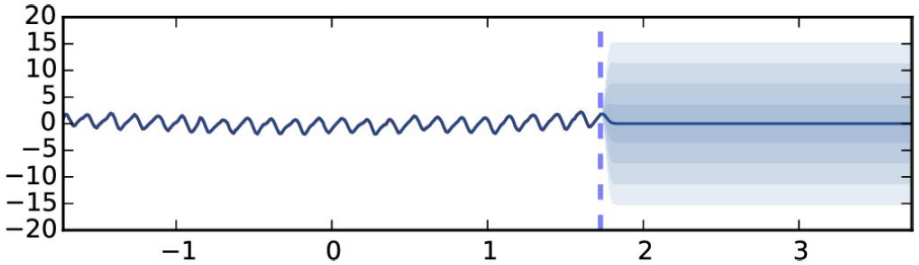


This is what the CO_2 dataset looks like before pre-processing.

Model Uncertainty in Regression Tasks



(a) Standard dropout with weight averaging

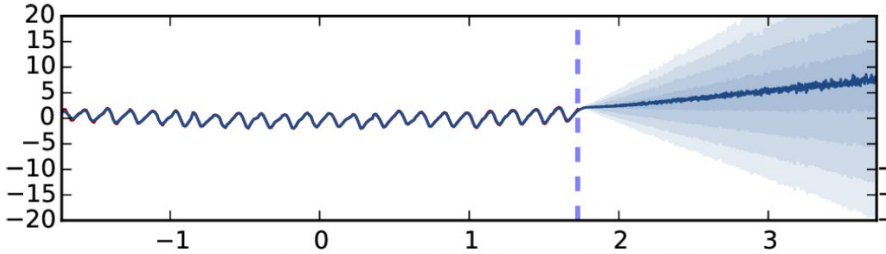


(b) Gaussian process with SE covariance function

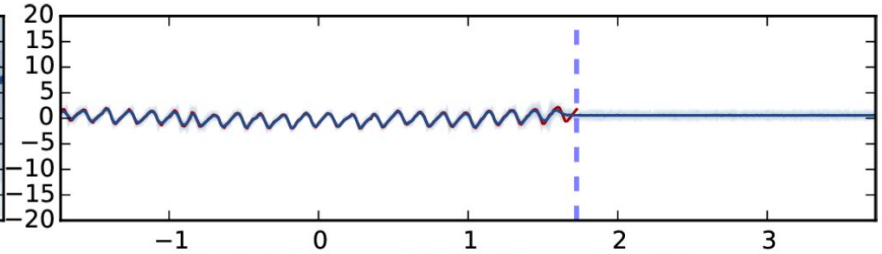
➔ standard dropout NN model predicts
- ◆ 0 with high confidence, not sensible

➔ GP model predicts
- ◆ 0, but the model in uncertain
- ◆ The shades of blue represent model uncertainty:
- ◆ each colour gradient represents half a standard deviation

Model Uncertainty in Regression Tasks

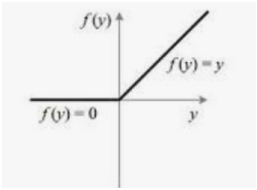


(c) MC dropout with ReLU non-linearities

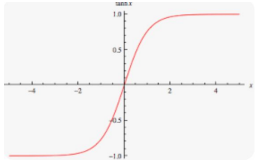


(d) MC dropout with TanH non-linearities

- ➔ MC dropout with ReLU predicts
  - ◆ ~7, but the model in uncertain
  - ◆ the uncertainty is increasing far from the data
- ➔ GP model predicts
  - ◆ 0, but the model in uncertain
  - ◆ the uncertainty stays bounded



ReLU doesn't saturate



TanH saturates

Model Uncertainty in Regression Tasks - discussion

- ➔ Models initialised with different dropout probability
  - ◆ initially exhibit smaller uncertainty
  - ◆ converged uncertainty at the end is almost indistinguishable
- ➔ Moments (mean and uncertainty) of the dropout models converge to the moments of the approximated GP model
- ➔ The number of forward iterations may be small to get a reasonable estimation to the predictive mean and uncertainty

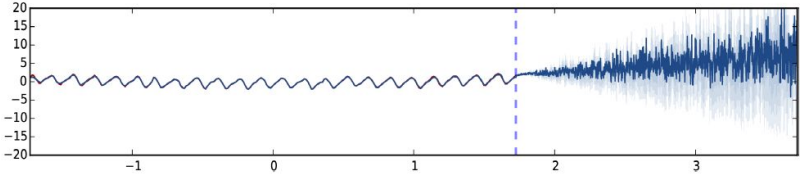
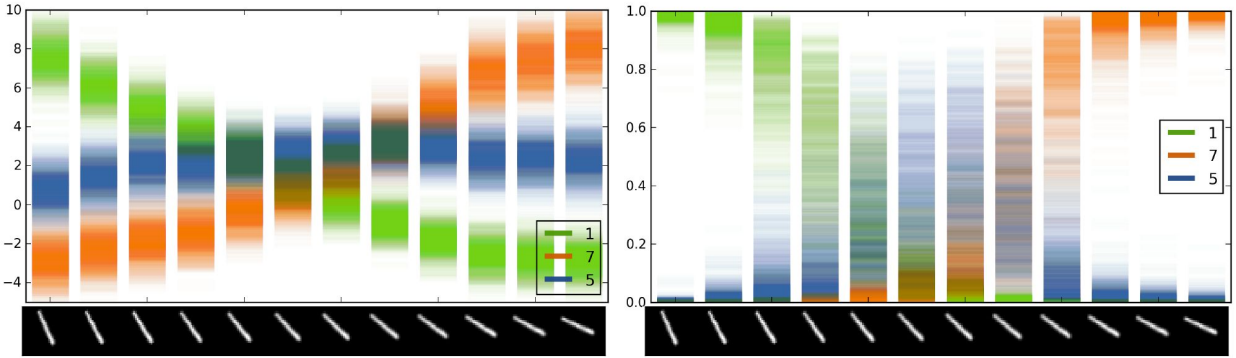


Figure 3. Predictive mean and uncertainties on the Mauna Loa CO_2 concentrations dataset for the MC dropout model with ReLU non-linearities, approximated with 10 samples.

Model Uncertainty in Classification Tasks

- ➔ Dataset - the full MNIST dataset (LeCun & Cortes, 1998)
- ➔ LeNet convolutional neural network model (LeCun et al., 1998)
- ➔ dropout applied before the last fully connected inner-product layer
- ➔ dropout probabilities - 0.5
- ➔ number of forward iterations - 100

Model Uncertainty in Classification Tasks



(a) Softmax *input* scatter (b) Softmax *output* scatter

Figure 4. **A scatter of 100 forward passes of the softmax input and output for dropout LeNet.** On the X axis is a rotated image of the digit 1. The input is classified as digit 5 for images 6-7, even though model uncertainty is extremly large (best viewed in colour).

- ➔ For the 12 images, the model predicts classes [1 1 1 1 1 5 5 7 7 7 7 7]
- ➔ If the uncertainty envelope of a class is far from that of other classes' (left-most image) then the input is classified with high confidence
- ➔ If the uncertainty envelope intersects that of other classes then the softmax output uncertainty can be as large as the entire space.
- ➔ expect the model to ask an external annotator for a label for this input images 6-7
- ➔ the model uncertainty in the softmax output can be summarised by taking the mean of the distribution

Predictive Performance

- ➔ Predictive log-likelihood captures how well a model fits the data
- ➔ larger values indicating better model fit
- ➔ Uncertainty quality can be determined from this quantity as well
- ➔ Experiment is done to compare the RMSE and predictive log-likelihood
- ➔ Experiment is done on
  - ◆ a popular variational inference method (VI, Graves (2011))
  - ◆ Probabilistic back-propagation (PBP, Hern´andez-Lobato & Adams (2015))
  - ◆ dropout uncertainty (Dropout)

Predictive Performance

	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**

Model Uncertainty in Reinforcement Learning

- ➔ An agent receives various rewards from different states
- ➔ Agent's aim is to maximise its expected reward over time
- ➔ agent tries to learn to avoid transitioning into states with low rewards,
  - ◆ Instead pick actions that lead to better states.
- ➔ with uncertainty information an agent can decide when to exploit and when to explore
- ➔ RL uses NNs to estimate agents' Q-value functions (a function that estimates the quality of different actions)

Model Uncertainty in Reinforcement Learning

Experiment set-up

- ➔ simulate an agent in a 2D world
- ➔ Agent can take one of 5 actions controlling two motors at its base
- ➔ action - turn the motors at different angles and different speeds
- ➔ environment - red circles, green circle
- ➔ positive reward
  - ◆ reaching to red circle
  - ◆ not looking at (white) walls
  - ◆ for walking in a straight line
- ➔ negative reward
  - ◆ reaching to green circle

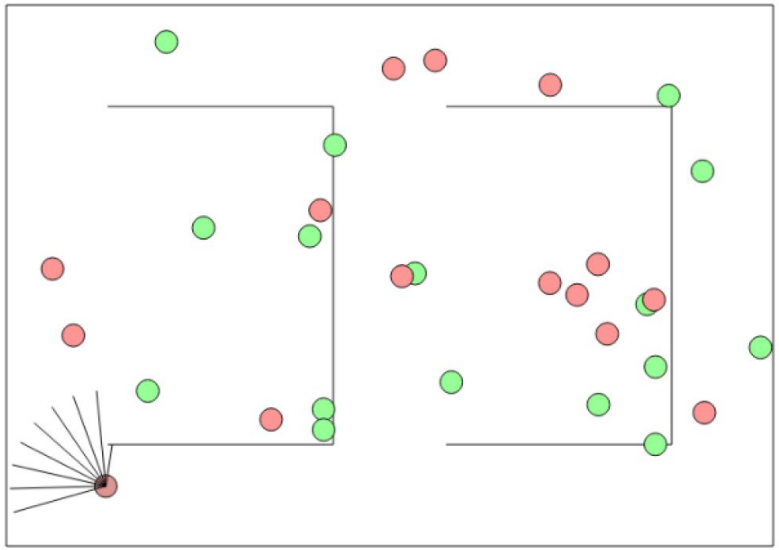


Figure 5. Depiction of the reinforcement learning problem used in the experiments. The agent is in the lower left part of the maze, facing north-west.

Model Uncertainty in Reinforcement Learning :approaches

Epsilon greedy search

- ➔ the agent selects the best action following its current Q-function estimation with some probability
- ➔ explores otherwise

Dropout approach

- ➔ Use dropout Q-network
- ➔ Use Thompson sampling

Model Uncertainty in Reinforcement Learning

gets reward larger than 1 (converge faster)

➔ 25 batches

➔ 17 batches

Dropout approach seems to stop improving after 1K batches

➔ still sampling random moves

➔ only exploits at this stage

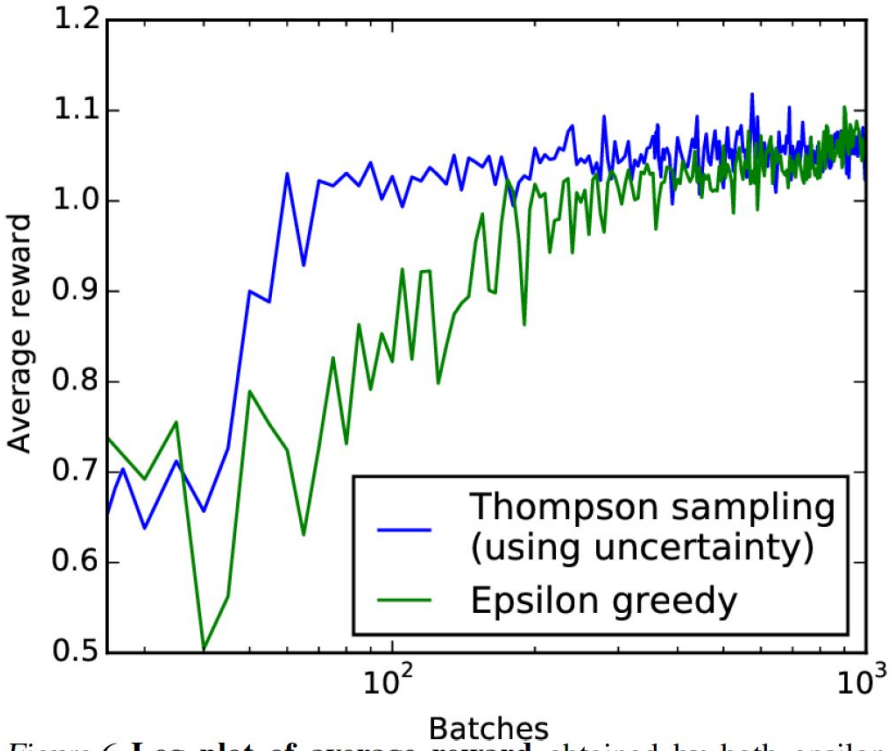


Figure 6. **Log plot of average reward** obtained by both epsilon greedy (in green) and our approach (in blue), as a function of the number of batches.

Conclusions

- ➔ Built a probabilistic interpretation of dropout
  - ◆ that make possible to obtain model uncertainty out of existing deep learning models
- ➔ Studied the properties of this uncertainty
- ➔ Bernoulli dropout is used to approximate variational distribution
  - ◆ Other variants of dropout follow this interpretation as well and correspond to alternative approximating distributions
- ➔ Future Research
  - ◆ Experiment using this approach with different non-linearity function (activation function) and different regularisation