20190228 DMQA Seminar

Understanding Uncertainty and Bayesian Convolutional Neural Networks

이 민정

Contents

Yarin Gal

팔로우 중

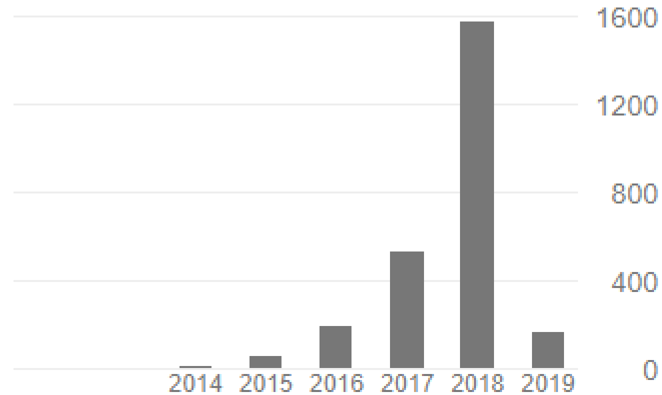
Associate Professor, University of Oxford
cs.ox.ac.uk의 이메일 확인됨 - 홈페이지

Machine Learning   Artificial Intelligence   Probability Theory   Statistics

제목	인용
Dropout as a Bayesian approximation: Representing model uncertainty in deep learning Y Gal, Z Ghahramani Proceedings of the 33rd International Conference on Machine Learning (ICML-16)	751
A theoretically grounded application of dropout in recurrent neural networks Y Gal, Z Ghahramani Advances in Neural Information Processing Systems, 1019-1027	490
What uncertainties do we need in bayesian deep learning for computer vision? A Kendall, Y Gal Advances in neural information processing systems, 5574-5584	254
Uncertainty in Deep Learning Y Gal University of Cambridge	195

인용

	전체	2014년 이후
서지정보	2588	2587
h-index	17	17
i10-index	20	20



What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?

Alex Kendall
University of Cambridge
agk34@cam.ac.uk

Yarin Gal
University of Cambridge
yg279@cam.ac.uk

Abstract

There are two major types of uncertainty one can model. *Aleatoric* uncertainty captures noise inherent in the observations. On the other hand, *epistemic* uncertainty accounts for uncertainty in the model – uncertainty which can be explained away given enough data. Traditionally it has been difficult to model epistemic uncertainty in computer vision, but with new Bayesian deep learning tools this is now possible. We study the benefits of modeling epistemic vs. aleatoric uncertainty in Bayesian deep learning models for vision tasks. For this we present a Bayesian deep learning framework combining input-dependent aleatoric uncertainty together with epistemic uncertainty. We study models under the framework with per-pixel semantic segmentation and depth regression tasks. Further, our explicit uncertainty formulation leads to new loss functions for these tasks, which can be interpreted as learned attenuation. This makes the loss more robust to noisy data, also giving new state-of-the-art results on segmentation and depth regression benchmarks.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. In *Advances in neural information processing systems* (pp. 5574-5584).

Under review as a conference paper at ICLR 2016

BAYESIAN CONVOLUTIONAL NEURAL NETWORKS WITH BERNOULLI APPROXIMATE VARIATIONAL INFERENCE

Yarin Gal & Zoubin Ghahramani
University of Cambridge
{yg279,zg201}@cam.ac.uk

ABSTRACT

Convolutional neural networks (CNNs) work well on large datasets. But labelled data is hard to collect, and in some applications larger amounts of data are not available. The problem then is how to use CNNs with small data – as CNNs overfit quickly. We present an efficient Bayesian CNN, offering better robustness to over-fitting on small data than traditional approaches. This is by placing a probability distribution over the CNN's *kernels*. We approximate our model's intractable posterior with Bernoulli variational distributions, requiring no additional model parameters.

On the theoretical side, we cast dropout network training as approximate inference in Bayesian neural networks. This allows us to implement our model using existing tools in deep learning with no increase in time complexity, while highlighting a negative result in the field. We show a considerable improvement in classification accuracy compared to standard techniques and improve on published state-of-the-art results for CIFAR-10.

Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

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

Yarin Gal YG279@CAM.AC.UK
Zoubin Ghahramani ZG201@CAM.AC.UK
University of Cambridge

Abstract

Deep learning tools have gained tremendous attention in applied machine learning. However such tools for regression and classification do not capture model uncertainty. In comparison, Bayesian models offer a mathematically grounded framework to reason about model uncertainty, but usually come with a prohibitive computational cost. In this paper we develop a new theoretical framework casting dropout training in deep neural networks (NNs) as approximate Bayesian inference in deep Gaussian processes. A direct result of this theory gives us tools to model uncertainty with dropout NNs – extracting information from existing models that has been thrown away so far. This mitigates the problem of representing uncertainty in deep

mow & Marks, 2015; Nuzzo, 2014), new needs arise from deep learning tools.

Standard deep learning tools for regression and classification do not capture model uncertainty. In classification, predictive probabilities obtained at the end of the pipeline (the softmax output) are often erroneously interpreted as model confidence. A model can be uncertain in its predictions even with a high softmax output (fig. 1). Passing a point estimate of a function (solid line 1a) through a softmax (solid line 1b) results in extrapolations with unjustified high confidence for points far from the training data. x^* for example would be classified as class 1 with probability 1. However, passing the distribution (shaded area 1a) through a softmax (shaded area 1b) better reflects classification uncertainty far from the training data.

Model uncertainty is indispensable for the deep learning

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

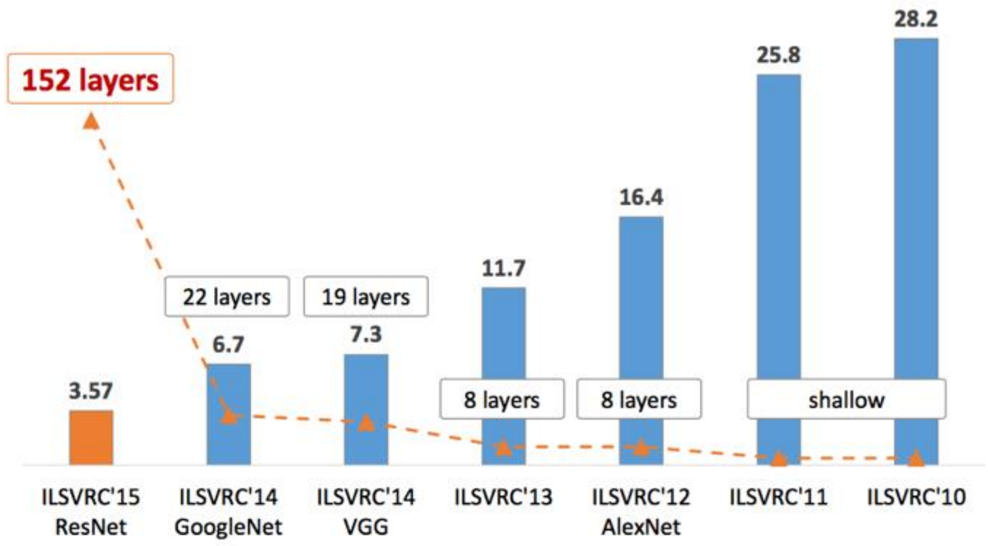
Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050-1059).

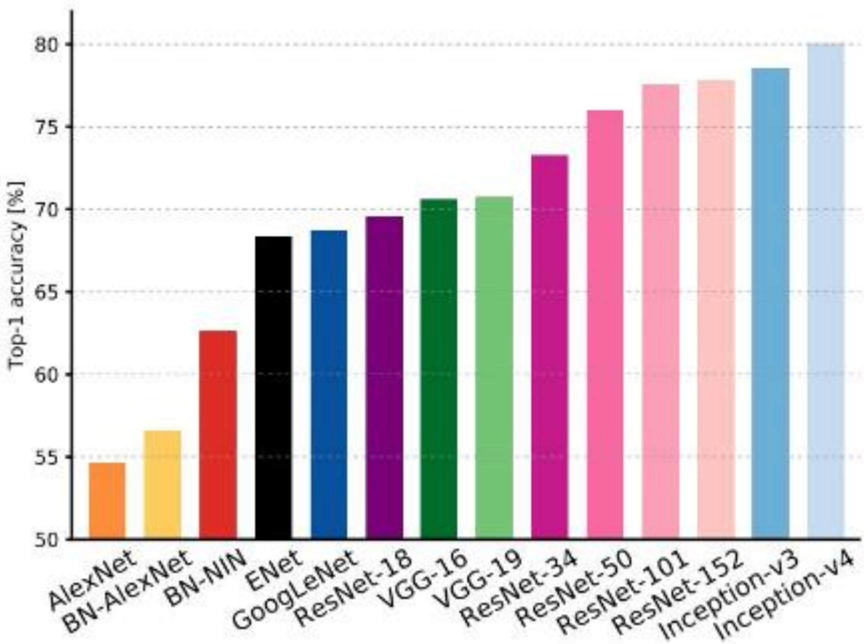
Introduction

Introduction

Convolutional Neural Networks

- Convolutional neural networks (CNNs) work well on large datasets.
- But labeled data is hard to collect, and in some applications larger amounts of data are not available.





https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

Introduction

Weakness of CNNs

- <u>CNNs overfit on small data.</u>
- CNNs can not measure uncertainty.

In May 2016, the first fatality from an assisted driving system, caused by the perception system confusing the white side of a trailer for bring sky.

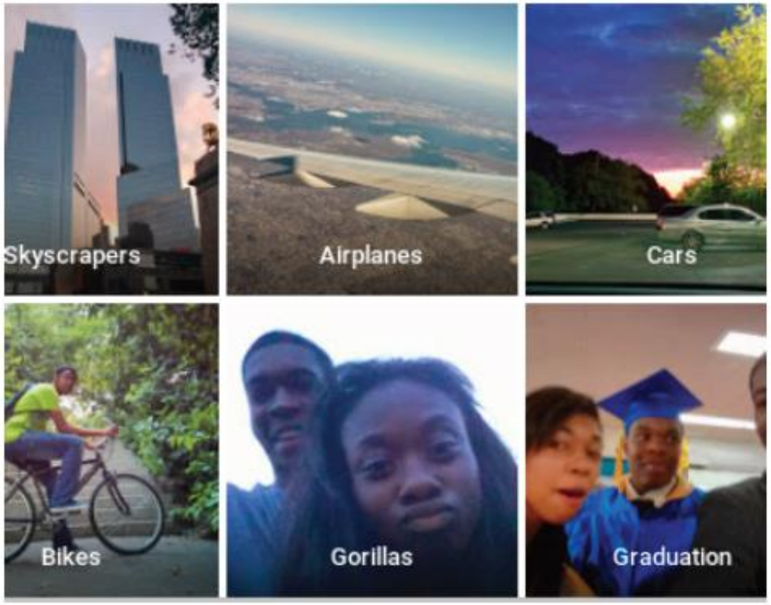
NHTSA. PE 16-007. Technical report, U.S. Department of Transportation, National Highway Traffic Safety Administration, Jan 2017. Tesla Crash Preliminary Evaluation Report

Introduction

Weakness of CNNs

- <u>CNNs overfit on small data.</u>
- CNNs can not measure uncertainty.

An image classification system erroneously identified two African Americans as gorillas.



Jessica Guynn. Google photos labeled black people 'gorillas'. USA Today, 2015.

Introduction

Weakness of CNNs

- CNNs overfit on small data.
- CNNs can not measure uncertainty.

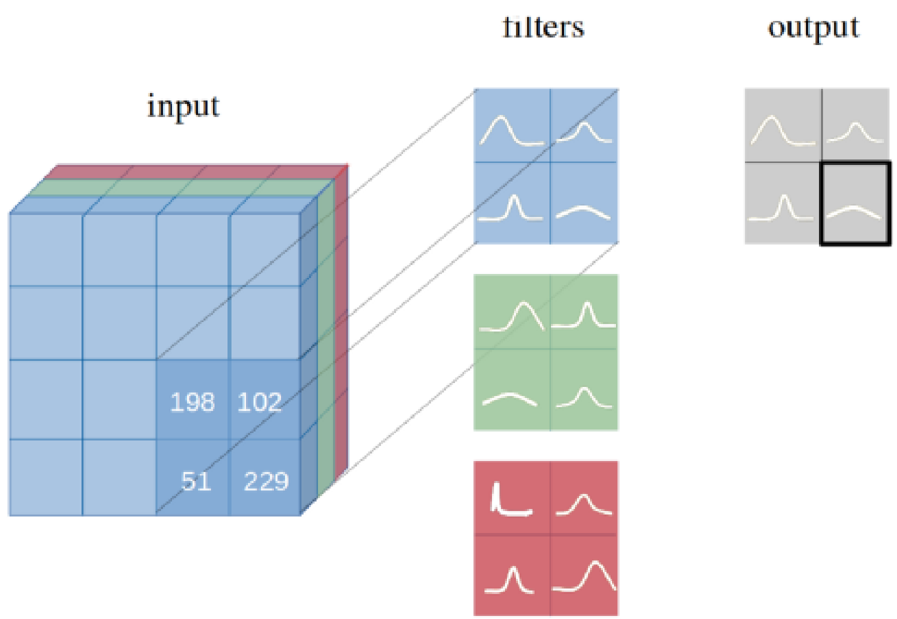
~~I don't know.~~

If both these algorithms were able to assign a high level of uncertainty to their erroneous predictions, then the system may have been able to make better decisions and likely avoid disaster.

Introduction

Bayesian Neural Networks

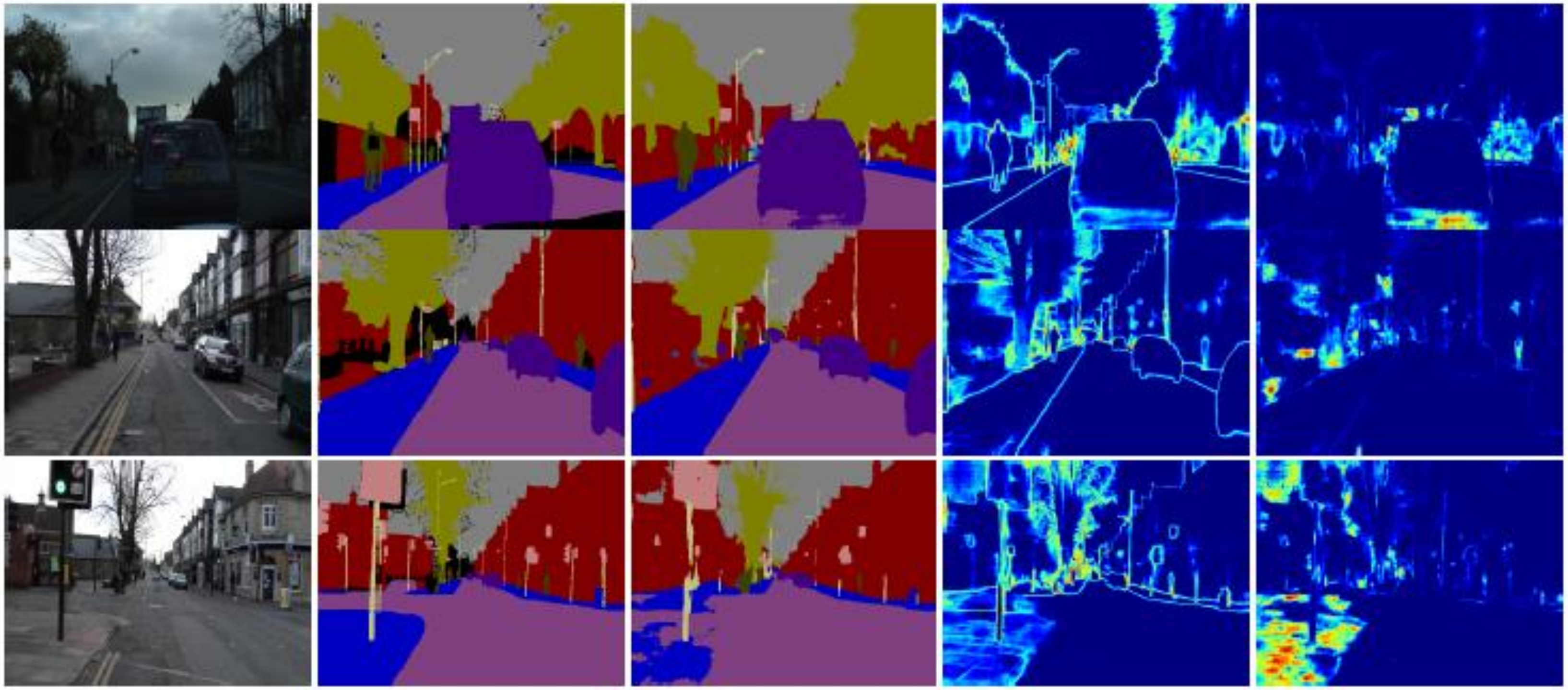
- Bayesian NNs are robust to overfitting.
- Bayesian NNs offer uncertainty estimates, and easily learn from small datasets.



Shridhar, K., Laumann, F., & Liwicki, M. (2019). A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference. *arXiv preprint arXiv:1901.02731*.

Introduction

Bayesian Neural Networks + Computer Vision



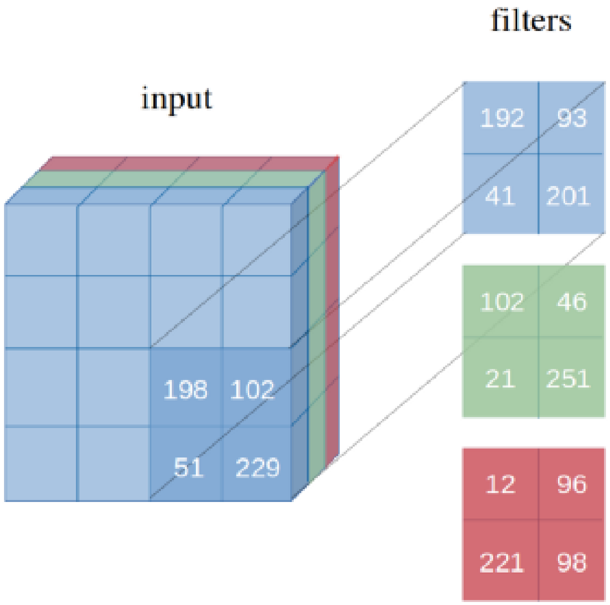
(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. In *Advances in neural information processing systems* (pp. 5574-5584).

Frequentist VS Bayesian Parameter Learning

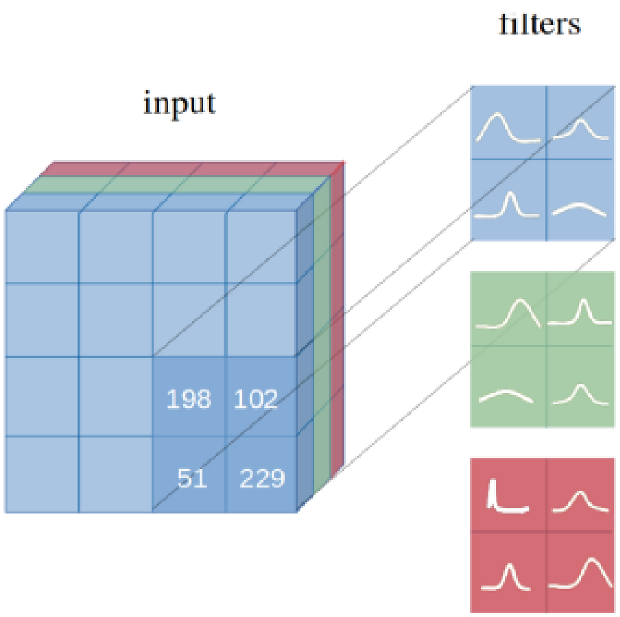
Frequentist VS Bayesian Parameter Learning

Frequentist
Network with
point-estimates as weights



Bayesian
Network with
probability distribution as weights

P(parameter|Data)



Shridhar, K., Laumann, F., & Liwicki, M. (2019). A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference. *arXiv preprint arXiv:1901.02731*.

Frequentist VS Bayesian Parameter Learning

How can Bayesian parameter learning prevents overfitting?

Frequentist VS Bayesian Parameter Learning

Flipping a coin : parameter p

- A single random variable $x \in \{0, 1\}$
- x might describe the outcome of flipping a coin with $x = 1$ representing 'heads', and $x = 0$ representing 'tails'.
- $P(x = 1|p) = p, P(x = 0|p) = 1 - p$
- The probability distribution over x can be written in the form

$$\mathrm{Bern(x|p)} = p^x(1-p)^{1-x}$$

- Now suppose we have a data set $\mathrm{D} = \{x_1, \ldots \ldots, x_N\}$ of observed values of x.

Head

Tail

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Frequentist VS Bayesian Parameter Learning

Flipping a coin : parameter p

\# 1

Head

\#2

Head

\#3

Head

……..

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Frequentist VS Bayesian Parameter Learning

Flipping a coin (Maximum Likelihood Estimator, MLE)

- Now suppose we have a data set $\mathrm{D} = \{x_1, \dots\dots, x_N\}$ of observed values of x.

- $L = P(D|p) = \prod_{n=1}^{N} P(x_n|p) = \prod_{n=1}^{N} \mathrm{Bern}(x_n|\mathrm{p}) = \prod_{n=1}^{N} p^{x_n}(1-p)^{1-x_n}$
- $\ln L = \ln P(D|p) = \sum_{n=1}^{N} \ln P(x_n|p) = \sum_{n=1}^{N} \{x_n \ln p + (1-x_n)\ln(1-p)\}$
- $\frac{\partial lnL}{\partial p} = 0$
- $p = \frac{1}{N}\sum_{n=1}^{N} x_n$

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Frequentist VS Bayesian Parameter Learning

Flipping a coin (Bayesian approach)

- $P(p|D) = \frac{P(D|p)P(p)}{P(D)} = \frac{Likelihood \times Prior}{Evidence} = \frac{P(D|p)P(p)}{\int P(D|p)P(p)}$
- We need to introduce a prior distribution $P(p)$ over the parameter p.
- The likelihood function takes the form of the product of factors of the form $p^{x_n}(1-p)^{1-x_n}$. If we choose a prior to be proportional to powers of p and (1-p), then the posterior distribution will have the same functional form as the prior.
- This property is called **conjugacy**.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Frequentist VS Bayesian Parameter Learning

Flipping a coin (Bayesian approach)

- $P(p|D) = \frac{P(D|p)P(p)}{P(D)} = \frac{Likelihood \times Prior}{Evidence} = \frac{P(D|p)P(p)}{\int P(D|p)P(p)}$
- We need to introduce a prior distribution $P(p)$ over the parameter p.

When likelihood function is a discrete distribution [edit]

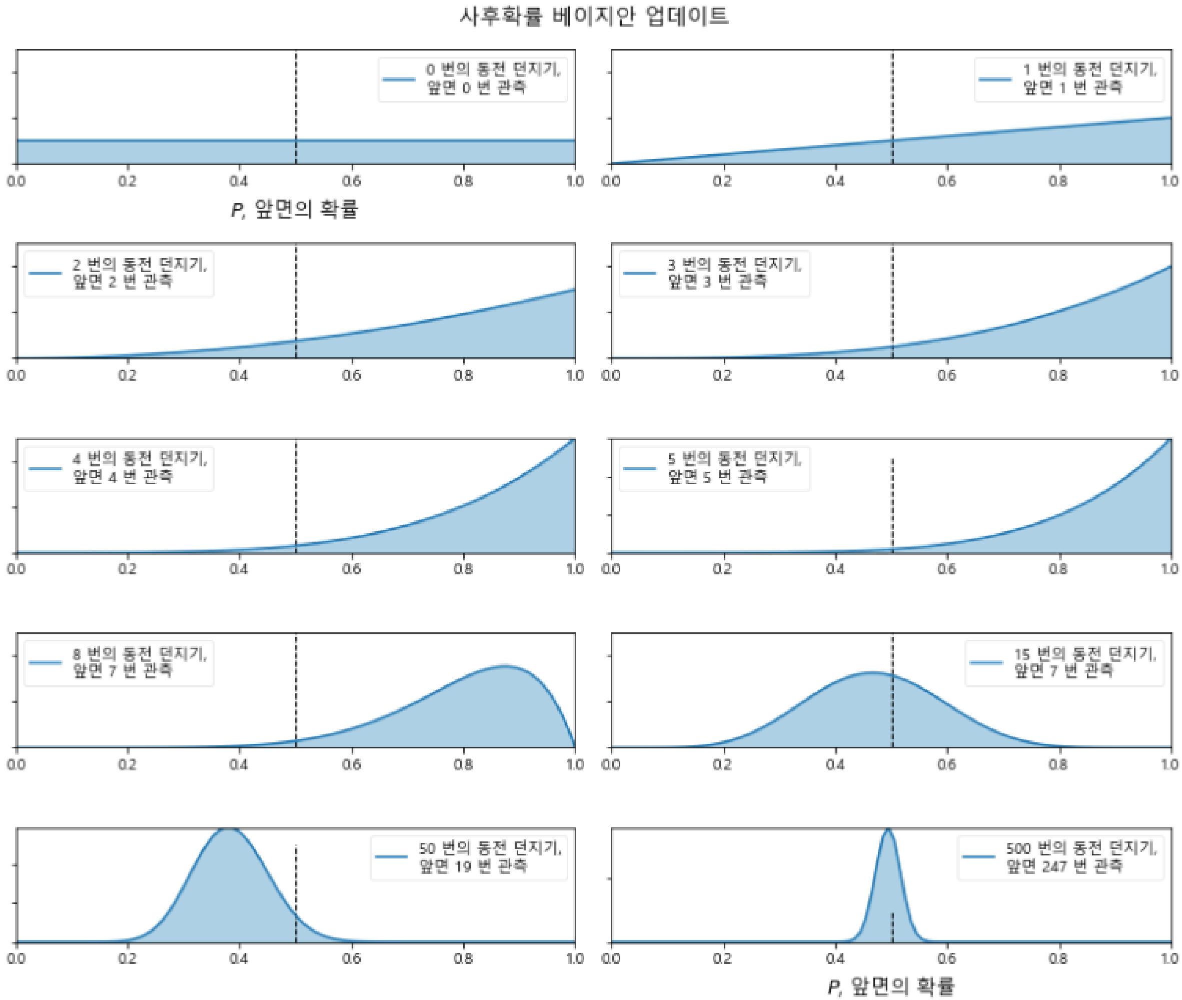
pdf $p(p;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$

Likelihood	Model parameters	Conjugate prior distribution	Prior hyperparameters	Posterior hyperparameters
Bernoulli	p (probability)	Beta	α, β	$\alpha + \sum_{i=1}^{n} x_i,\ \beta + n - \sum_{i=1}^{n} x_i$
Binomial	p (probability)	Beta	α, β	$\alpha + \sum_{i=1}^{n} x_i,\ \beta + \sum_{i=1}^{n} N_i - \sum_{i=1}^{n} x_i$
Negative binomial with known failure	p (probability)	Beta	α, β	$\alpha + \sum_{i=1}^{n} x_i,\ \beta + rn$

https://en.wikipedia.org/wiki/Conjugate_prior

Frequentist VS Bayesian Parameter Learning

Flipping a coin (Bayesian approach)



동전던지기_예

프로그래머를 위한 베이지안 with 파이썬, 캐머런 데이비슨 필론 지음, 곽승주 옮김

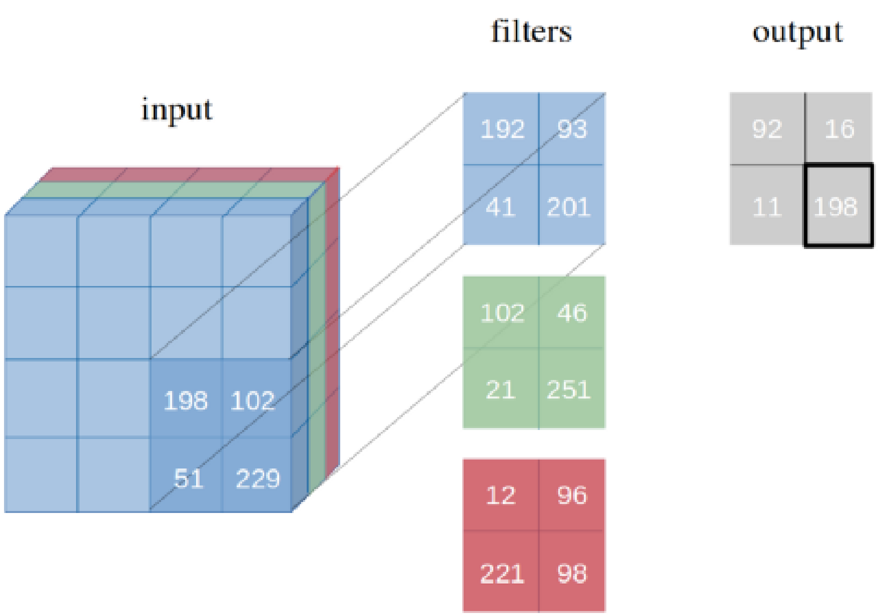
Frequentist VS Bayesian Parameter Learning

How can Bayesian parameter learning measures uncertainty?

Frequentist VS Bayesian Parameter Learning

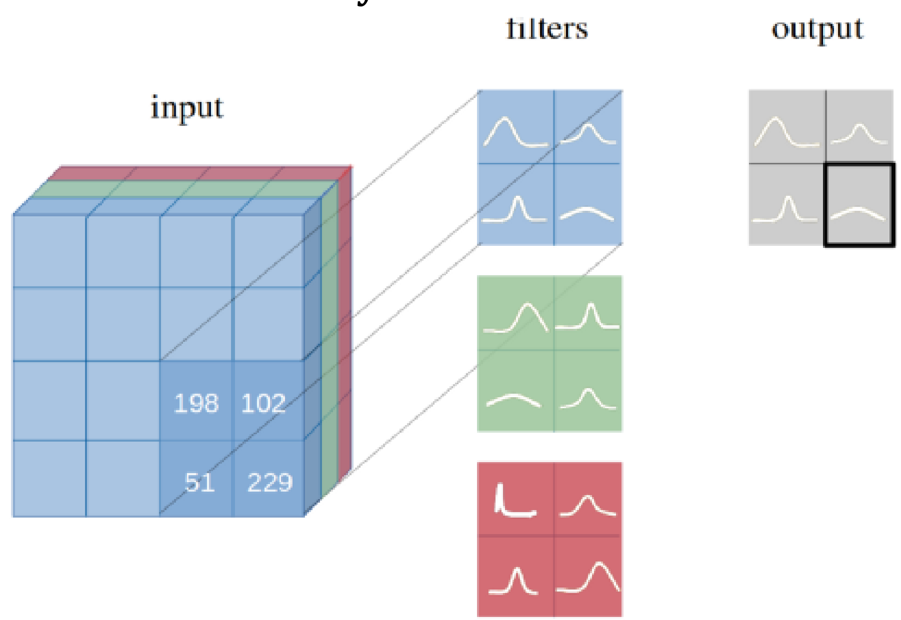
Frequentist
Network with
point-estimates as weights

$$y^* = Wx^*$$



Bayesian
Network with
probability distribution as weights
P(w|Data)

$$P(y^*|x^*,X,Y) = \int p(y^*|\,x^*,w)\boxed{P(w|X,Y)}dw$$

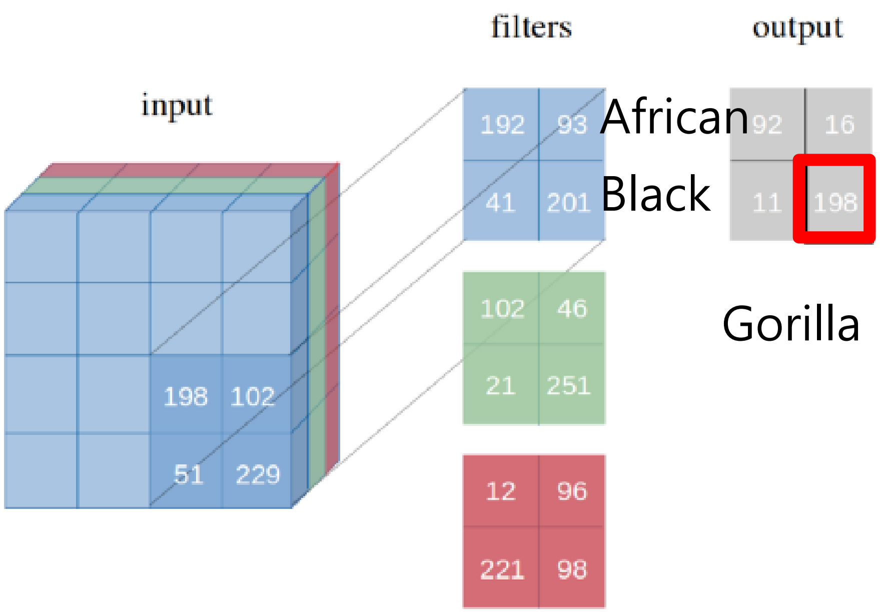


Shridhar, K., Laumann, F., & Liwicki, M. (2019). A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference. *arXiv preprint arXiv:1901.02731*.

Frequentist VS Bayesian Parameter Learning

Frequentist
Network with
point-estimates as weights

$$y^* = Wx^*$$



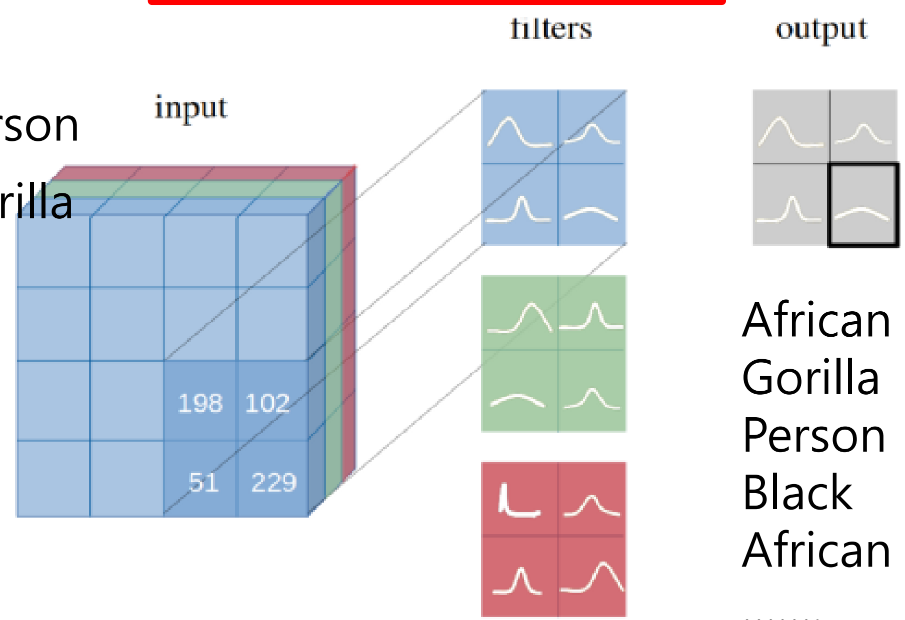
African
Black
Person
Gorilla

Gorilla

Bayesian
Network with
probability distribution as weights
P(w|Data)

$$P(y^*|x^*,X,Y) = \int p(y^*|\,x^*,w)P(w|X,Y)dw$$

$$\boxed{y^* = Wx^*, \qquad W \sim P(w|X,Y)}$$



African
Gorilla
Person
Black
African
....

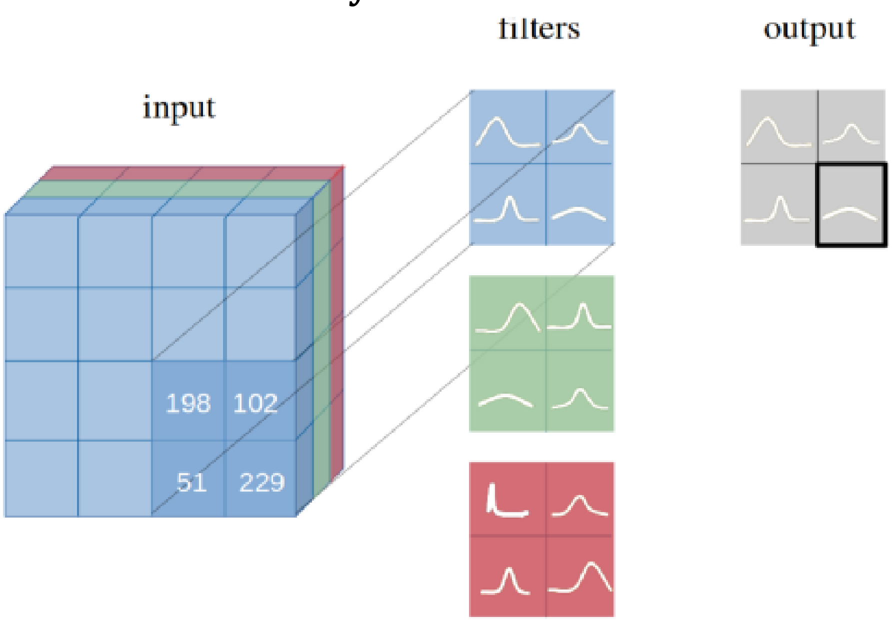
Shridhar, K., Laumann, F., & Liwicki, M. (2019). A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference. *arXiv preprint arXiv:1901.02731*.

Approximate Inference for Bayesian Neural Networks

Approximate Inference for Bayesian NNs

Bayesian
Network with
probability distribution as weights
P(w|Data)

$$P(y^*|x^*,X,Y) = \int p(y^*|\, x^*, w)\boxed{P(w|X,Y)}dw$$



$$P(w|D)$$

$$= \frac{P(D|w)P(w)}{P(D)}$$

$$= \frac{Likelihood \times Prior}{Evidence}$$

$$= \frac{P(D|w)P(w)}{\int P(D|w)P(w)\, dw}$$

Shridhar, K., Laumann, F., & Liwicki, M. (2019). A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference. *arXiv preprint arXiv:1901.02731*.

Approximate Inference for Bayesian NNs

- The distribution $P(w|D)$ is intractable.
- We need to approximate it with **a variational distribution $\boldsymbol{q_\theta(w)}$**

$$P(w|D)$$

$$= \frac{P(D|w)P(w)}{P(D)}$$

$$= \frac{\boxed{Likelihood} \times Prior}{\boxed{Evidence}}$$

$$= \frac{P(D|w)P(w)}{\int P(D|w)P(w)dw}$$

Posterior Distribution $P(w|D)$ **Inference**

Variational Inference

Optimization

Minimize the Kullback-Leibler divergence

$\mathrm{KL}(q_\theta(w)||P(w|D))$

$q_\theta(w)=P(w|D) \longrightarrow \mathrm{KL}(q_\theta(w)||P(w|D)) = 0$

Ex) $N(\mu,\sigma^2) \longrightarrow \theta = \mu,\sigma^2$

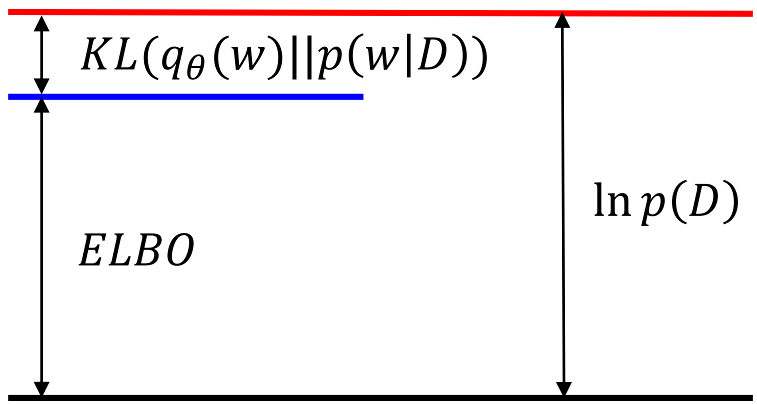
Approximate Inference for Bayesian NNs

❖ Indirectly reducing the gap between the variational distribution and the posterior distribution by <u>maximizing ELBO</u>.

$$\mathrm{KL}(q_\theta(w)||\boxed{p(w|D)}) = \int q_\theta(w) ln \frac{q_\theta(w)}{\boxed{p(w|D)}} dw$$

Log marginal likelihood $\ln p(D) = ELBO(variational\ free\ energy) + KL(q_\theta(w)||p(w|D))$

$$= \int q_\theta(w) \ln(D|w)\, dw - \int q_\theta(w) \ln \frac{q_\theta(w)}{p(w)} dw + KL(q_\theta(w)||p(w|D))$$



Appendix 1. Variational inference
Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

ELBO = evidence lower bound

Approximate Inference for Bayesian NNs

❖ Indirectly reducing the gap between the variational distribution and the posterior distribution by maximizing ELBO.

Approximate posterior distribution
by variational inference

$$\underset{\theta}{\text{Minimize}}\ \mathrm{KL}(q_{\theta}(w)||\mathrm{p}(w|D))$$

$$= \underset{\theta}{\text{Maximize}}\ \mathrm{ELBO}$$

Approximate Inference for Bayesian NNs

The objective of Bayesian NNs

Minimize KL($q_\theta(w)||\mathrm{p}(w|D)$)
= Maximize ELBO

$$= Minimize -\sum_{i=1}^{N}\int q_\theta(w) logp(y_i|f^{w}(x_i))dw + KL\big(q_\theta(w)||\mathrm{p}(w)\big)$$

$$= -\frac{\mathrm{N}}{\mathrm{M}}\sum_{i\in S}\int q_\theta(w) logp(y_i|f^{w}(x_i))dw + KL\big(q_\theta(w)||\mathrm{p}(w)\big)$$ Mini-batch optimization

$$= -\frac{\mathrm{N}}{\mathrm{M}}\sum_{i\in S}\int p(\epsilon)\ logp\big(y_i\big|f^{g(\theta,\epsilon)}(x_i)\big)d\epsilon + KL\big(q_\theta(w)||\mathrm{p}(w)\big)$$ Reparameterization trick

$$= -\frac{\mathrm{N}}{\mathrm{M}}\sum_{i\in S}\log p\big(y_i\big|f^{g(\theta,\epsilon)}(x_i)\big) + KL\big(q_\theta(w)||\mathrm{p}(w)\big)$$ Monte Carlo integration

Appendix 2. Reparameterization trick
Gal, Y. (2016). *Uncertainty in deep learning* (Doctoral dissertation, PhD thesis, University of Cambridge).

Approximate Inference for Bayesian NNs

The objective of Bayesian NNs

❖ By using stochastic gradient descent, we can update θ

Algorithm 1 Minimise divergence between $q_\theta(\omega)$ and $p(\omega|X, Y)$

1: Given dataset $\mathbf{X}, \mathbf{Y}$,
2: Define learning rate schedule η,
3: Initialise parameters θ randomly.
4: **repeat**
5: Sample M random variables $\widehat{\epsilon}_i \sim p(\epsilon)$, S a random subset of $\{1, .., N\}$ of size M.
6: Calculate stochastic derivative estimator w.r.t. θ:

$$\widehat{\Delta\theta} \leftarrow -\frac{N}{M}\sum_{i\in S}\frac{\partial}{\partial\theta}\log p(\mathbf{y}_i|\mathbf{f}^{g(\theta,\widehat{\epsilon}_i)}(\mathbf{x}_i)) + \frac{\partial}{\partial\theta}\mathrm{KL}(q_\theta(\omega)||p(\omega)).$$

7: Update θ:

$$\theta \leftarrow \theta + \eta\widehat{\Delta\theta}.$$

8: **until** θ has converged.

Approximate Inference for Bayesian NNs

The objective of Bayesian NNs

Posterior Distribution $P(w|D)$ **Inference**

Variational Inference

Optimization θ

Dropout as a Bayesian Approximation

Dropout as a Bayesian Approximation

MC Dropout to Bayesian CNNs

CNNs + L2norm + MC dropout
= Bayesian CNNs

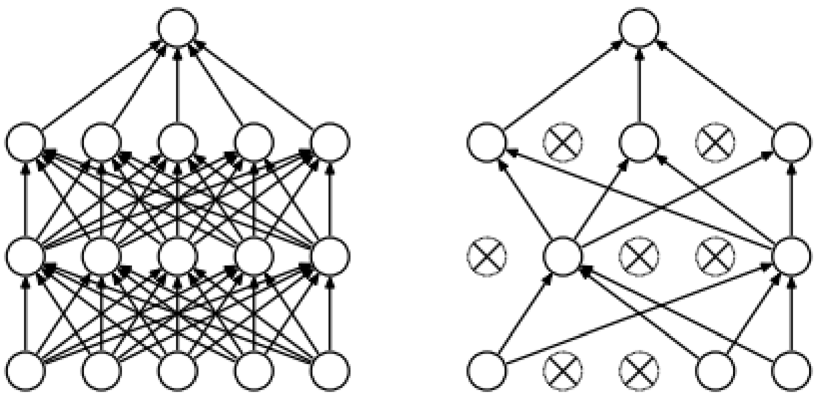
We can implement Bayesian CNNs using existing tools in deep learning

Dropout as a Bayesian Approximation

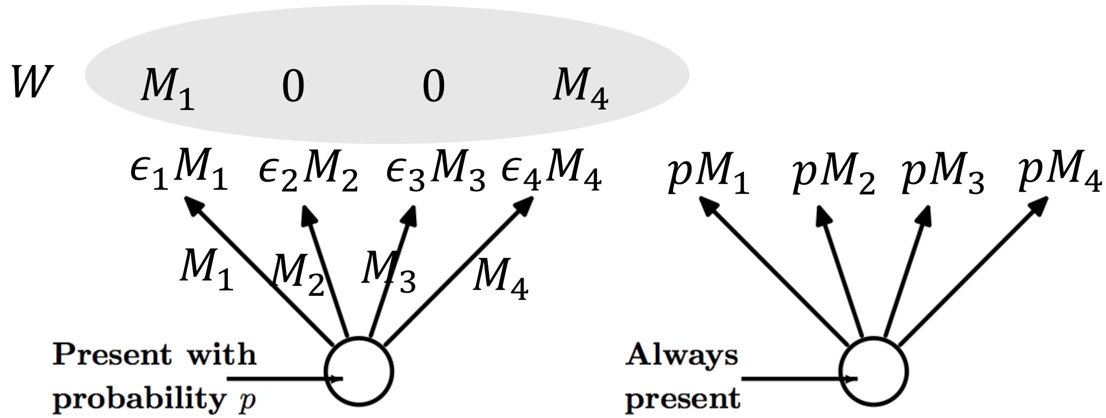
Dropout

Drop Prop : 1-p, Keep Prop : p

- Standard dropout is a technique used to avoid over-fitting in neural networks.
- Setting 1-p proportion of the elements (nodes) of the layer to zero.



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



(a) At training time (b) At test time

$$\epsilon_i \sim Beroulli(p)$$

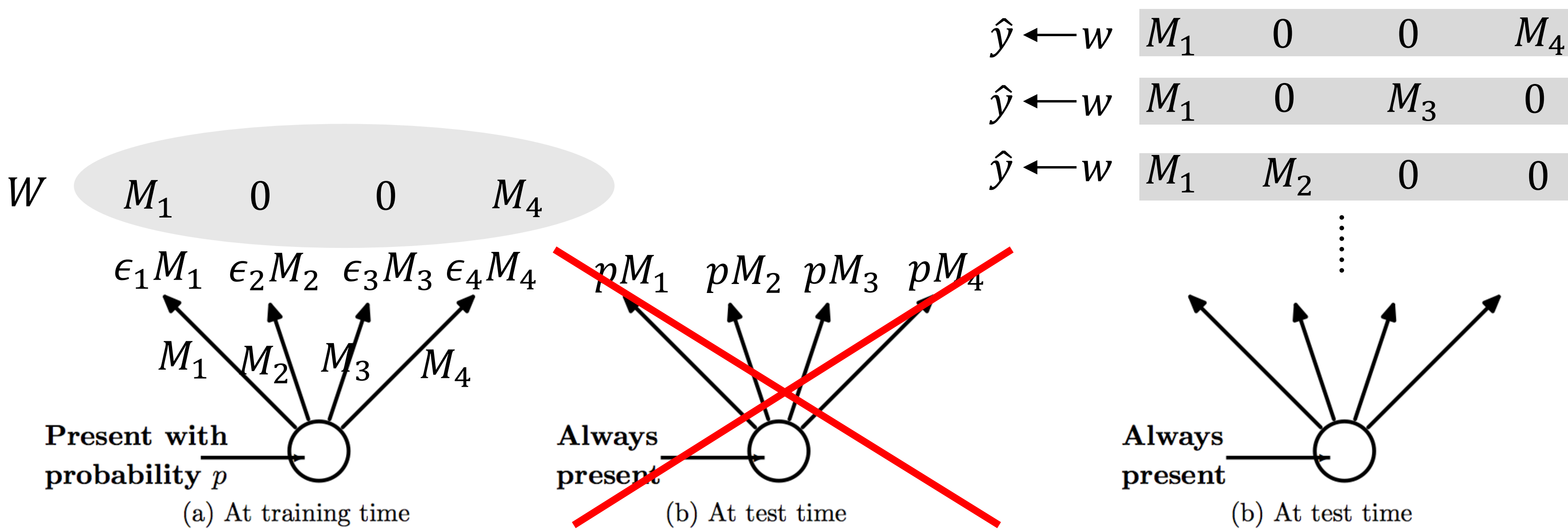
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

Dropout as a Bayesian Approximation

MC Dropout to Bayesian CNNs

Drop Prop : 1-p, Keep Prop : p

- The standard dropout test time approximation does not perform well when dropout is applied after convolutions.
- Averaging stochastic forward passes through the model at test time (using MC dropout).



$\epsilon_i \sim Beroulli(p)$

MC : Monte Carlo

Dropout as a Bayesian Approximation

MC Dropout to Bayesian CNNs

Drop Prop : 1-p, Keep Prop : p

- $W = \epsilon M, \epsilon \sim Bernoulli(p)$

$$\text{Minimize KL}(q_\theta(w)||\text{p}(w|D))$$

$$= \text{Maximize ELBO}$$

$$= \text{Minimize } -\frac{\text{N}}{\text{M}}\sum_{i \in S} \log p\left(y_i | f^{g(\theta,\epsilon)}(x_i)\right) + KL\left(q_\theta(w)||\text{p}(w)\right)$$

Regression : MSE
Classification : Softmax cross entropy

L2 norm ($||M||_2^2$) + MC dropout

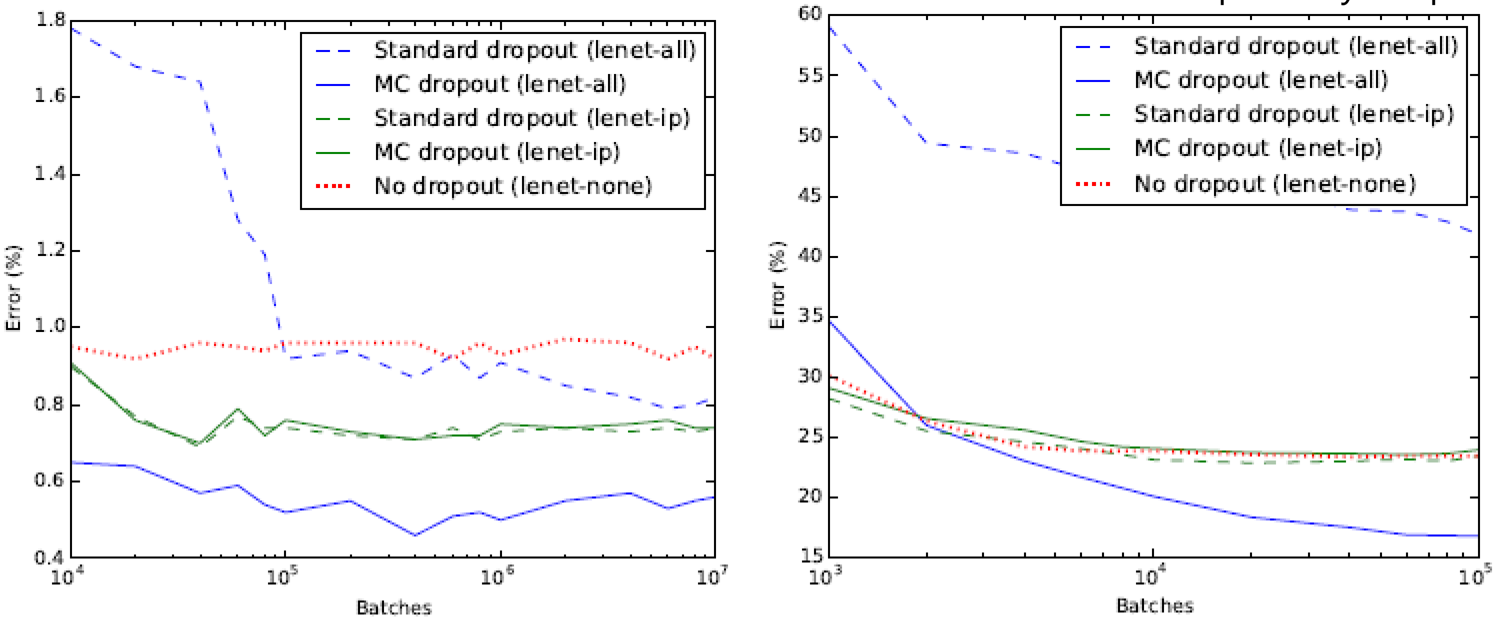
Gal, Y. (2016). *Uncertainty in deep learning* (Doctoral dissertation, PhD thesis, University of Cambridge).

Dropout as a Bayesian Approximation

Results

- Although Standard dropout lenet-all performs very badly on both datasets (dashed blue line), when evaluating the same network with MC dropout (solid blue line) the model outperforms all others.

lenet-ip : only dropout after FC layer



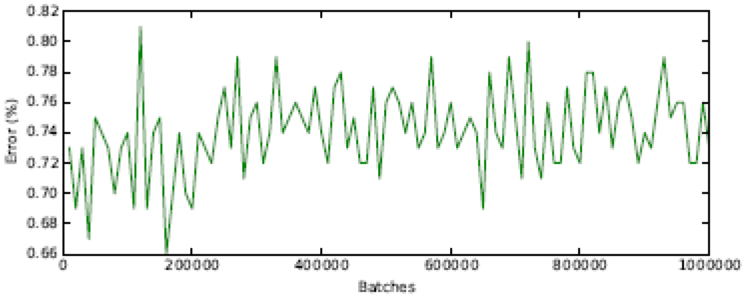
(a) MNIST (b) CIFAR-10

Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

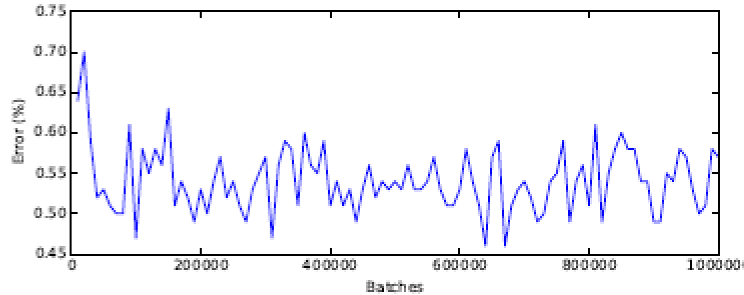
Dropout as a Bayesian Approximation

Results

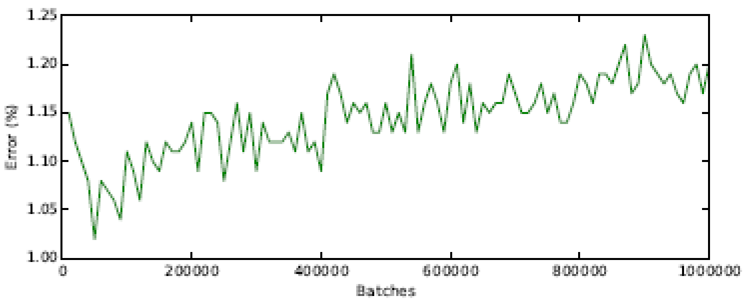
- Evaluate our model's tendency to over-fit on training sets decreasing in size.
- Randomly split the MNIST dataset into smaller training sets of sizes 1/4.
- Test error of LeNet trained on random subsets of MNIST decreasing in size.



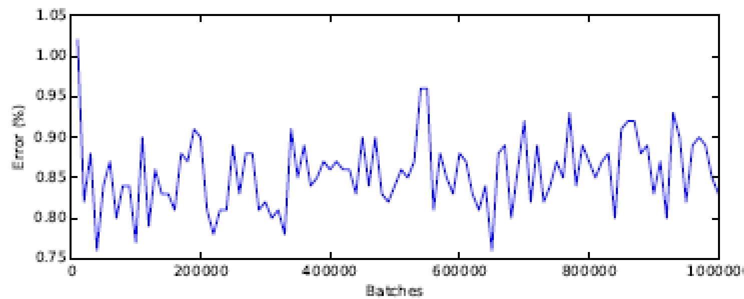
(a) Entire MNIST, Standard dropout + lenet-ip



(b) Entire MNIST, MC dropout + lenet-all



(c) 1/4 of MNIST, Standard dropout + lenet-ip



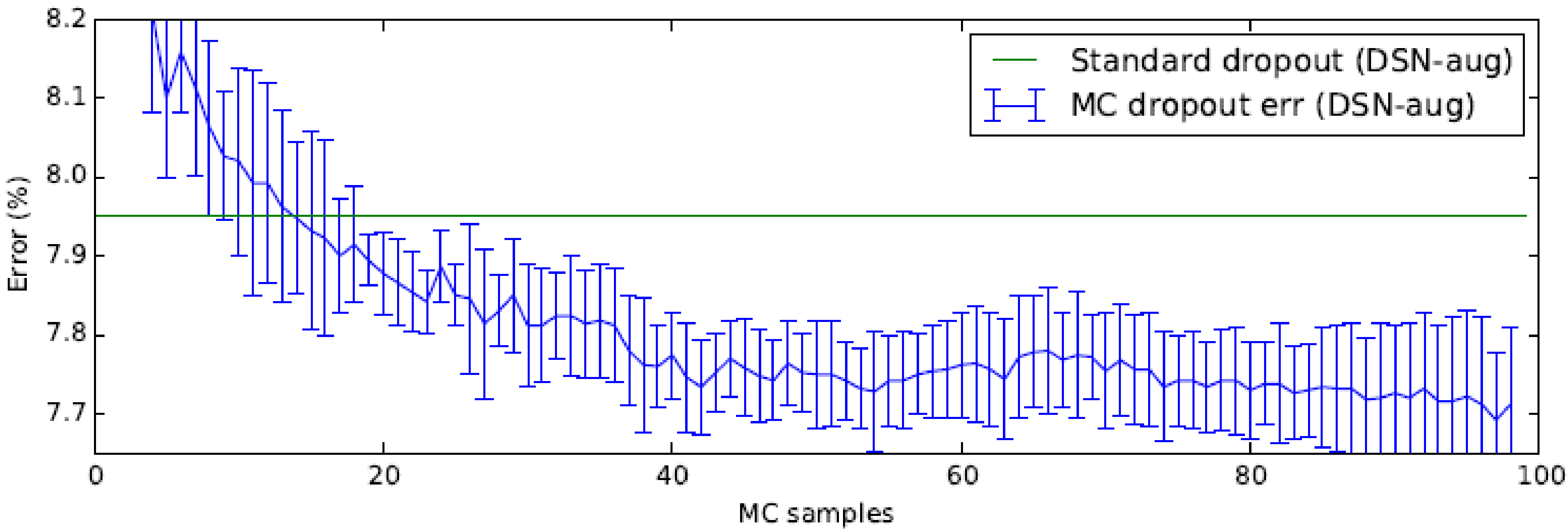
(d) 1/4 of MNIST, MC dropout + lenet-all

Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

Dropout as a Bayesian Approximation

Results

- In green is test error with Standard dropout. MC dropout achieves a significant improvement (more than 1 standard deviation) after 20 samples.



Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

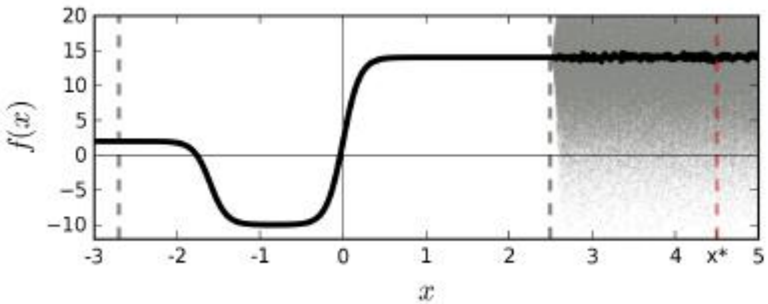
Uncertainty in Bayesian Neural Networks

Uncertainty in Bayesian NNs

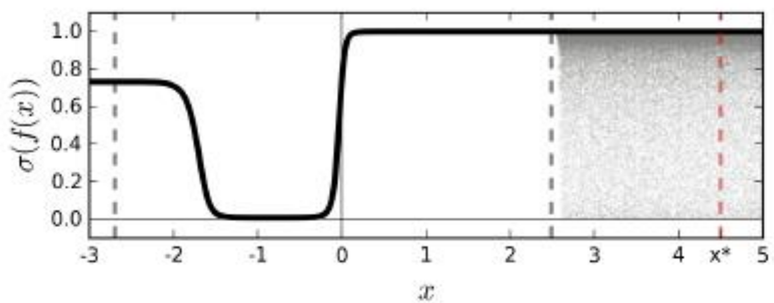
Uncertainty

- Standard deep learning tools for regression and classification do not capture model uncertainty.
- In classification, predictive probabilities obtained at the end of the pipeline (the softmax output) are often erroneously interpreted as model confidence.
- Extrapolations with unjustified high confidence for points far from the training data

counter example



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



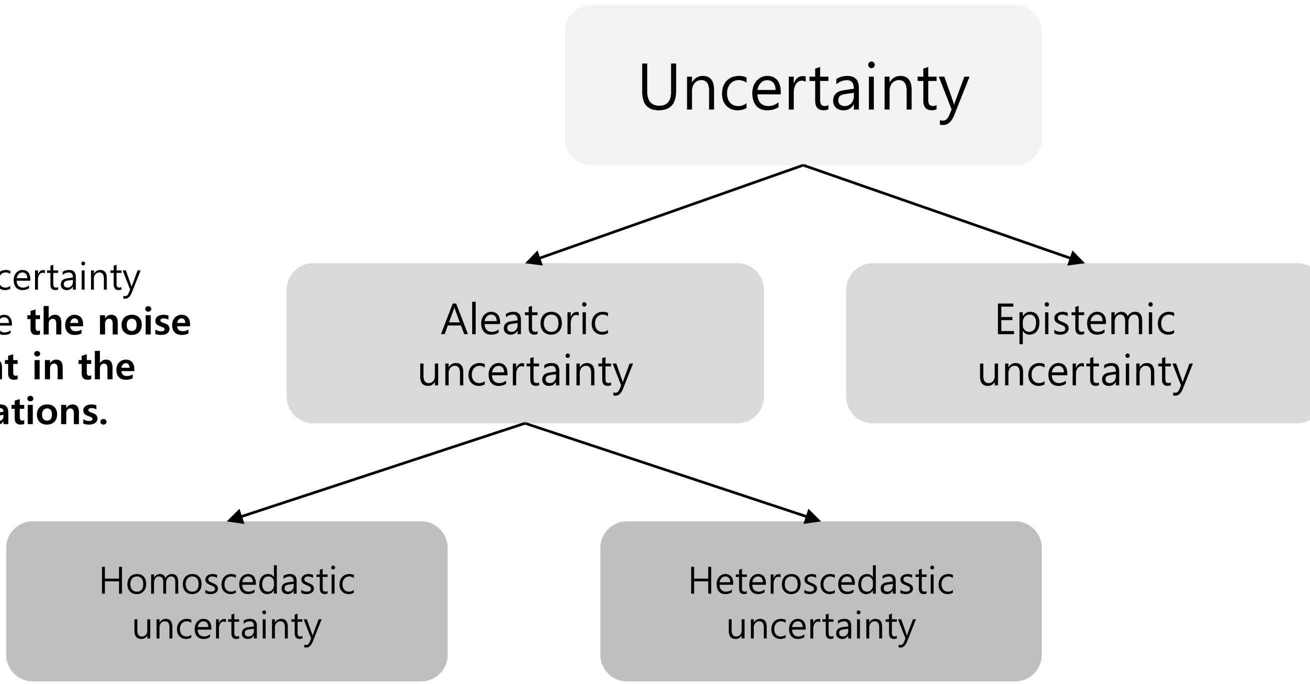
(b) $\sigma(f(\mathbf{x}))$ as a function of data $\mathbf{x}$ (softmax *output*)

Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050-1059).

Uncertainty in Bayesian NNs

Uncertainty

- In Bayesian modeling, there are two main types of uncertainty.

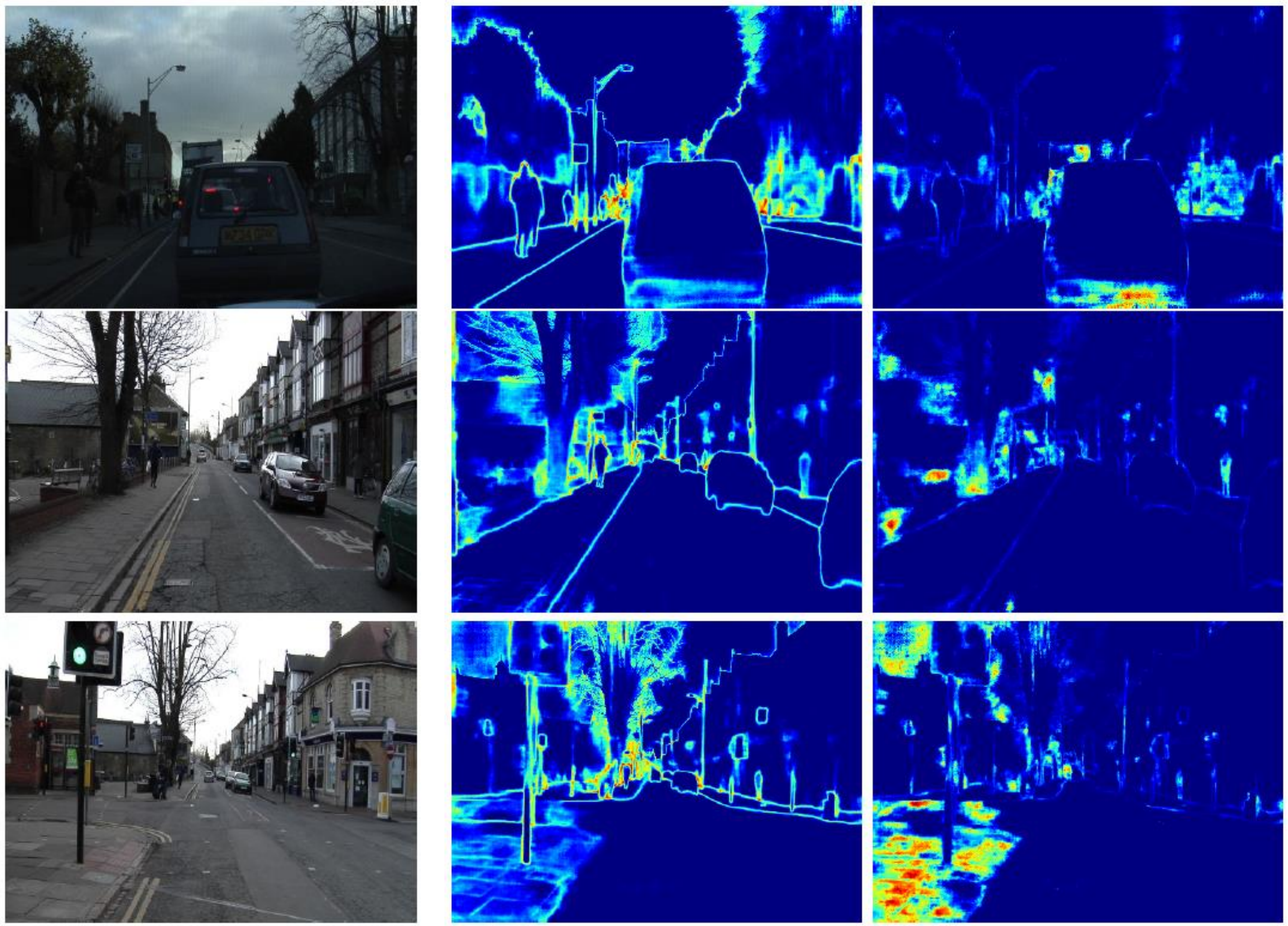


- This uncertainty measure **the noise inherent in the observations.**

- The uncertainty **caused by the model itself.**
- This uncertainty can be **reduced given more data**
- This uncertainty is often **referred to as model uncertainty.**

- Aleatoric uncertainty can further be categorized into **homoscedastic** uncertainty, the uncertainty which stays constant for different inputs,
and **heteroscedastic** uncertainty which depends on the inputs to the model.

Uncertainty in Bayesian NNs



Input Image　　Aleatoric　　Epistemic

Uncertainty in Bayesian NNs

How to make a model
which can measure these uncertainties?

Uncertainty in Bayesian NNs

Epistemic Uncertainty modeling

- Epistemic uncertainty is modeled by placing a prior distribution over a models weights, and then trying to capture how much these weights vary given some data.

L2 norm +MC dropout

$$Minimize - \frac{1}{N}\sum_{i=1}^{N} \log p\left(y_i \middle| f^{\widehat{W}_i}(x_i)\right) + \boxed{KL\left(q_\theta(w) \| \mathrm{p}(w)\right)}$$

Full batch, N data points, $\widehat{W}_i \sim q_\theta^*(W)$,
θ the set of the simple distribution's parameters to be optimized.

Uncertainty in Bayesian NNs

Heteroscedastic uncertainty modeling

- There are homoscedastic and heteroscedastic uncertainty in aleatoric.
- Heteroscedastic models are useful in cases where parts of the observation space might have higher noise levels than others.

$$Minimize -\frac{1}{N}\sum_{i=1}^{N} \log p\left(y_i \middle| f^{\widehat{W}_i}(x_i)\right)$$

$$y = f + \epsilon, \epsilon \sim N(0, \sigma^2)$$
$$y \sim N(f, \sigma^2)$$

The negative log likelihood can be further simplified as

$$-\log p\left(y_i \middle| f^{\widehat{W}_i}(x_i)\right) \propto \frac{1}{2\sigma^2}\left|\left|y_i - f^{\widehat{W}_i}(x_i)\right|\right|^2 + \frac{1}{2}\log \sigma^2$$

for a Gaussian likelihood, with σ the model's observation noise parameter-capturing how much noise we have in the outputs.

Uncertainty in Bayesian NNs

Heteroscedastic uncertainty modeling

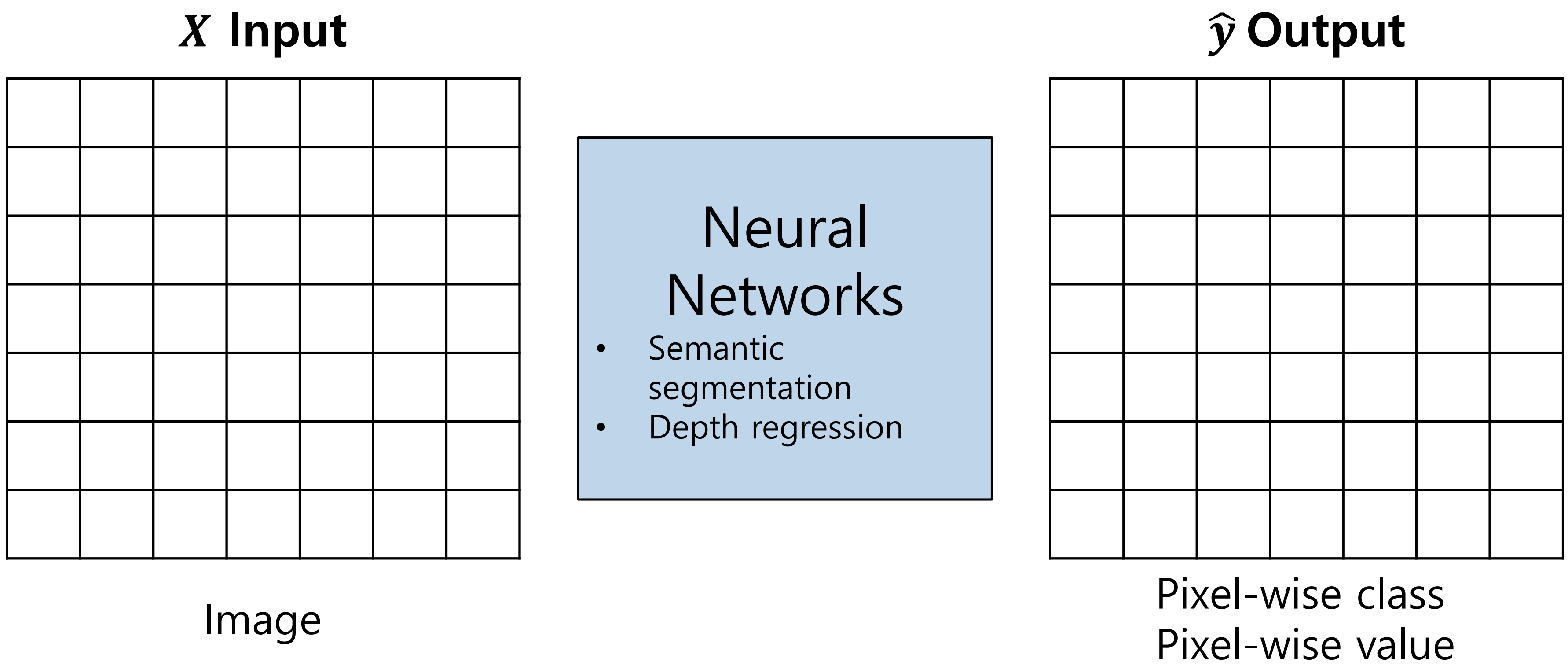
- There are homoscedastic and heteroscedastic uncertainty in aleatoric.
- Heteroscedastic models are useful in cases where parts of the observation space might have higher noise levels than others.

$$Minimize \; -\frac{1}{N}\sum_{i=1}^{N}\log p\left(y_i \middle| f^{\widehat{W}_i}(x_i)\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(x_i)^2}\left|\left|y_i - f^{\widehat{W}_i}(x_i)\right|\right|^2 + \frac{1}{2}\log\sigma(x_i)^2$$

Uncertainty in Bayesian NNs

Epistemic + Heteroscedastic + **Computer vision model**



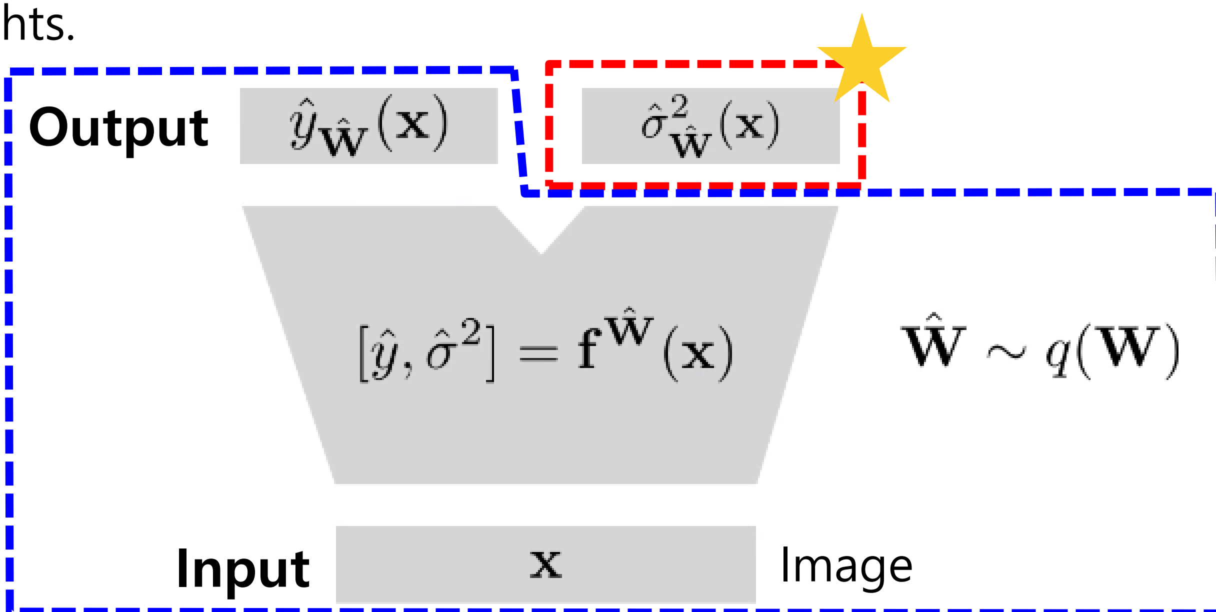
D : the number of output pixels y_i

Uncertainty in Bayesian NNs

Epistemic + **Heteroscedastic** + **Computer vision model**

Head split to predict both $\hat{y}$ and $\hat{\sigma}^2$

- Turn the heteroscedastic NN into a Bayesian NN by placing a distribution over its weights.



$$\frac{1}{D}\sum_{i=1}^{N}\frac{1}{2\widehat{\sigma}_i^{\,2}}||y_i-\widehat{y}_i||^2+\frac{1}{2}\log\widehat{\sigma}_i^{\,2}=\frac{1}{D}\sum_{i=1}^{N}\frac{1}{2}\exp(-s_i)||y_i-\widehat{y}_i||^2+\frac{1}{2}s_i$$

+ L2 norm ($||M||_2^2$) + MC dropout

Uncertainty in Bayesian NNs

Loss attenuation

- The predictive uncertainty acts as a robust regression function by allowing the network to learn to attenuate the effect from erroneous labels.

Discouraged from predicting high uncertainty for all points

$$\frac{1}{D}\sum_{i=1}^{N}\frac{1}{2\widehat{\sigma}_i^{\,2}}\left|\left|y_i-\widehat{y}_i\right|\right|^2+\frac{1}{2}\log\widehat{\sigma}_i^{\,2}=\frac{1}{D}\sum_{i=1}^{N}\frac{1}{2}\exp(-s_i)\left|\left|y_i-\widehat{y}_i\right|\right|^2+\frac{1}{2}s_i$$

The residual's weighting

Uncertainty in Bayesian NNs

Measuring Uncertainty

- The predictive uncertainty for pixel y in this combined model can be approximated using:

$$[\hat{y}, \hat{\sigma}^2] = \mathbf{f}^{\hat{\mathbf{W}}}(\mathbf{x})$$

$$Var(y) \approx \boxed{\frac{1}{T}\sum_{t=1}^{T} \hat{y}_t^2 - \left(\frac{1}{T}\sum_{t=1}^{T} \hat{y}_t\right)^2} + \boxed{\frac{1}{T}\sum_{t=1}^{T} \hat{\sigma}_t^2}$$

Epistemic uncertainty

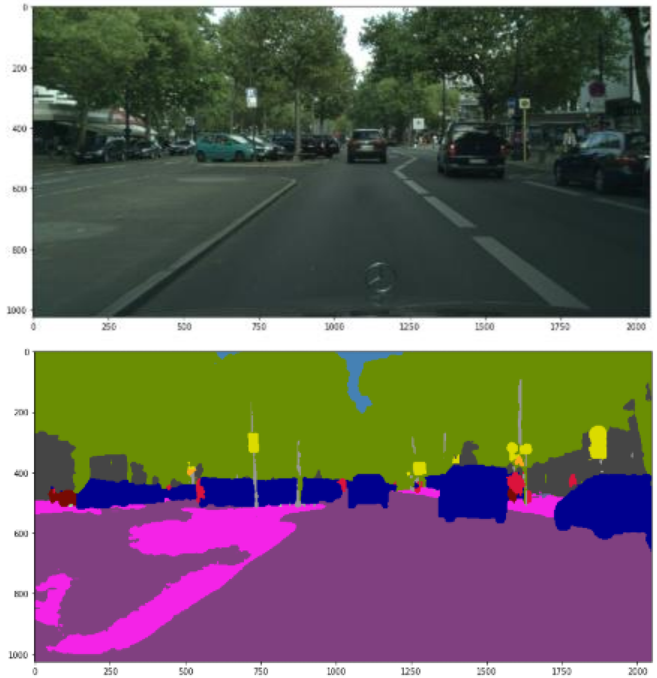
Heteroscedastic uncertainty

Uncertainty in Bayesian NNs

Experiments

❖ Evaluating methods with pixel-wise depth regression & semantic segmentation.

semantic segmentation



pixel-wise depth regression

Uncertainty in Bayesian NNs

Experiments

- Evaluating methods with pixel-wise depth regression & semantic segmentation.

Tasks	Dataset	
Semantic Segmentation	CamVid	A road scene 367 training images and 233 test images 11 classes Resize images to 360×480 pixels
	NYU v2 40-class	A indoor segmentation dataset 1449 images with resolution 640 ×480 from 464 different indoor scenes.
Depth Regression	Make3D	400 training and 134 testing images Gathered using a 3D laser scanner Resizing images to 345×460 pixels
	NYU v2 Depth	The same dataset used for segmentation above

Uncertainty in Bayesian NNs

Results

- Semantic segmentation performance : Modeling both aleatoric and epistemic uncertainty gives a notable improvement in segmentation accuracy over state of the art baselines.

CamVid	IoU
SegNet [28]	46.4
FCN-8 [29]	57.0
DeepLab-LFOV [24]	61.6
Bayesian SegNet [22]	63.1
Dilation8 [30]	65.3
Dilation8 + FSO [31]	66.1
DenseNet [20]	66.9
This work:	
DenseNet (Our Implementation)	67.1
+ Aleatoric Uncertainty	67.4
+ Epistemic Uncertainty	67.2
+ Aleatoric & Epistemic	**67.5**

(a) CamVid dataset for road scene segmentation.

NYUv2 40-class	Accuracy	IoU
SegNet [28]	66.1	23.6
FCN-8 [29]	61.8	31.6
Bayesian SegNet [22]	68.0	32.4
Eigen and Fergus [32]	65.6	34.1
This work:		
DeepLabLargeFOV	70.1	36.5
+ Aleatoric Uncertainty	70.4	37.1
+ Epistemic Uncertainty	70.2	36.7
+ Aleatoric & Epistemic	**70.6**	**37.3**

(b) NYUv2 40-class dataset for indoor scenes.

Appendix 3. IoU

Uncertainty in Bayesian NNs

Results

- Depth regression performance : comparison to previous approaches on depth regression data NYUv2 Depth. Modeling the combination of uncertainties improves accuracy.

Make3D	rel	rms	$\log_{10}$
Karsch et al. [33]	0.355	9.20	0.127
Liu et al. [34]	0.335	9.49	0.137
Li et al. [35]	0.278	7.19	0.092
Laina et al. [26]	0.176	4.46	0.072
This work:			
DenseNet Baseline	0.167	3.92	0.064
+ Aleatoric Uncertainty	**0.149**	3.93	**0.061**
+ Epistemic Uncertainty	0.162	**3.87**	0.064
+ Aleatoric & Epistemic	**0.149**	4.08	0.063

(a) Make3D depth dataset [25].

NYU v2 Depth	rel	rms	$\log_{10}$
Karsch et al. [33]	0.374	1.12	0.134
Ladicky et al. [36]	-	-	-
Liu et al. [34]	0.335	1.06	0.127
Li et al. [35]	0.232	0.821	0.094
Eigen et al. [27]	0.215	0.907	-
Eigen and Fergus [32]	0.158	0.641	-
Laina et al. [26]	0.127	0.573	0.055
This work:			
DenseNet Baseline	0.117	0.517	0.051
+ Aleatoric Uncertainty	0.112	0.508	0.046
+ Epistemic Uncertainty	0.114	0.512	0.049
+ Aleatoric & Epistemic	**0.110**	**0.506**	**0.045**

(b) NYUv2 depth dataset [23].

Uncertainty in Bayesian NNs

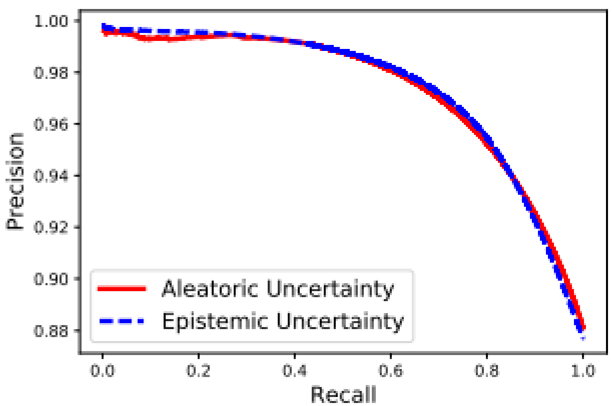
Results

Precision = TP/(TP+FP)

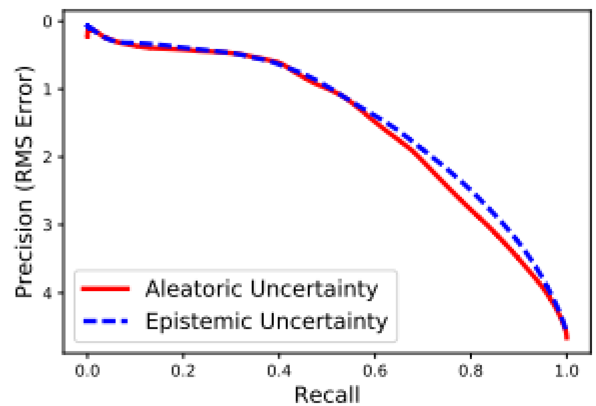
Recall = TP/(TP+FN)

		Predicted Condition	
		Positive 1	Negative 0
True condition	Positive 1	True positive	False negative
	Negative 0	False positive	True negative

by removing pixels with uncertainty larger than various percentile thresholds.



(a) Classification (CamVid)



(b) Regression (Make3D)

Increasing uncertainty,

Decreasing precision
= Increasing FP
= Increasing recall
= Decreasing FN

Figure 2: Precision Recall plots demonstrating both measures of uncertainty can effectively capture accuracy, as precision decreases with increasing uncertainty.

Uncertainty in Bayesian NNs

Results

- This shows that aleatoric uncertainty remains approximately constant, while epistemic uncertainty decreases the closer the test data is to the training distribution, demonstrating that epistemic uncertainty can be explained away with sufficient training data (but not for out-of-distribution data).

Train dataset	Test dataset	RMS	Aleatoric variance	Epistemic variance
Make3D / 4	Make3D	5.76	0.506	7.73
Make3D / 2	Make3D	4.62	0.521	4.38
Make3D	Make3D	3.87	0.485	2.78
Make3D / 4	NYUv2	-	0.388	15.0
Make3D	NYUv2	-	0.461	4.87

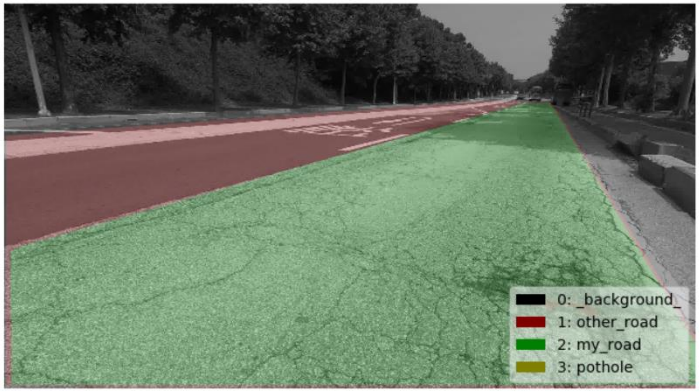
(a) Regression

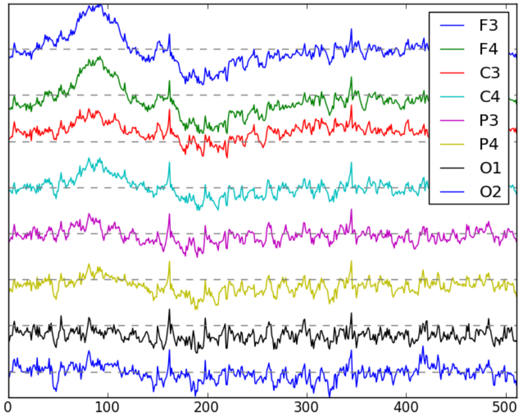
Train dataset	Test dataset	IoU	Aleatoric entropy	Epistemic logit variance ($\times 10^{-3}$)
CamVid / 4	CamVid	57.2	0.106	1.96
CamVid / 2	CamVid	62.9	0.156	1.66
CamVid	CamVid	67.5	0.111	1.36
CamVid / 4	NYUv2	-	0.247	10.9
CamVid	NYUv2	-	0.264	11.8

(b) Classification

Conclusions

- Bayesian parameter learning prevents overfitting
- L2 norm + MC dropout : practical Bayesian NNs
- Bayesian NNs in computer vision can measure uncertainty
- Measuring uncertainty helps make decision





Thank You!

Appendix 1. Variational Inference

Log marginal likelihood $\ln p(D)$

$$= \int \ln p(D)\, q_\theta(w) dw$$

$$= \int q_\theta(w) ln \frac{p(D)p(w|D)}{p(w|D)} dw$$

$$= \int q_\theta(w) ln \frac{p(w,D)}{p(w|D)} dw$$

$$= \int q_\theta(w) ln \frac{p(D|w)p(w)}{p(w|D)} dw$$

$$= \int q_\theta(w) ln \frac{q_\theta(w)p(D|w)p(w)}{q_\theta(w)p(w|D)} dw$$

$$= \int q_\theta(w) ln \frac{p(D|w)p(w)}{q_\theta(w)} dw + \int q_\theta(w) ln \frac{q_\theta(w)}{p(w|D)} dw$$

$$= ELBO(variational\ free\ energy) + KL(q_\theta(w)||p(w|D))$$

Appendix 2. Reparameterization trick

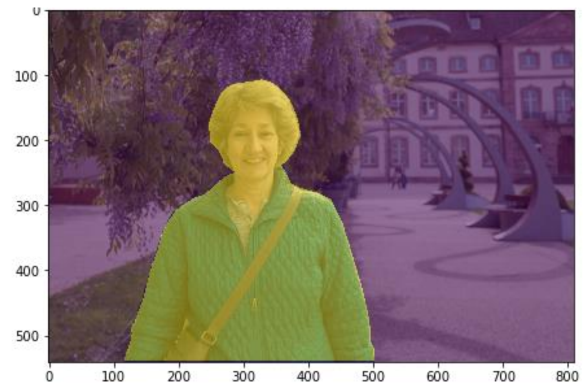
Target	$p(z;\theta)$	Base $p(\epsilon)$	One-liner $g(\epsilon;\theta)$
Exponential	$\exp(-x); x > 0$	$\epsilon \sim [0;1]$	$\ln(1/\epsilon)$
Cauchy	$\frac{1}{\pi(1+x^2)}$	$\epsilon \sim [0;1]$	$\tan(\pi\epsilon)$
Laplace	$\mathcal{L}(0;1) = \exp(-\lvert x\rvert)$	$\epsilon \sim [0;1]$	$\ln(\frac{\epsilon_1}{\epsilon_2})$
Laplace	$\mathcal{L}(\mu;b)$	$\epsilon \sim [0;1]$	$\mu - bsgn(\epsilon)\ln(1-2\lvert\epsilon\rvert)$
Std Gaussian	$\mathcal{N}(0;1)$	$\epsilon \sim [0;1]$	$\sqrt{\ln(\frac{1}{\epsilon_1})}\cos(2\pi\epsilon_2)$
Gaussian	$\mathcal{N}(\mu;RR^\top)$	$\epsilon \sim \mathcal{N}(0;1)$	$\mu + R\epsilon$
Rademacher	$Rad(\frac{1}{2})$	$\epsilon \sim Bern(\frac{1}{2})$	$2\epsilon - 1$
Log-Normal	$\ln\mathcal{N}(\mu;\sigma)$	$\epsilon \sim \mathcal{N}(\mu;\sigma^2)$	$\exp(\epsilon)$
Inv Gamma	$i\mathcal{G}(k;\theta)$	$\epsilon \sim \mathcal{G}(k;\theta^{-1})$	$\frac{1}{\epsilon}$

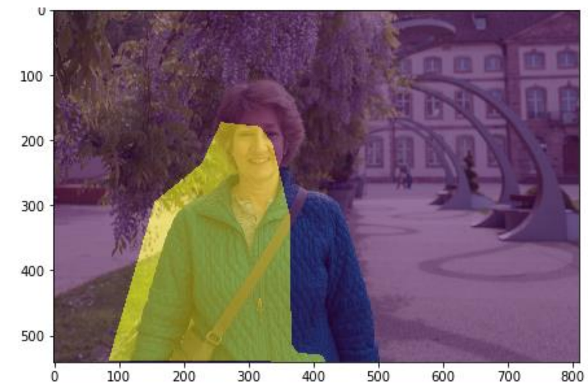
http://blog.shakirm.com/2015/10/machine-learning-trick-of-the-day-4-reparameterisation-tricks/

Appendix 3. IoU

$$IoU = \frac{target \cap prediction}{target \cup prediction}$$

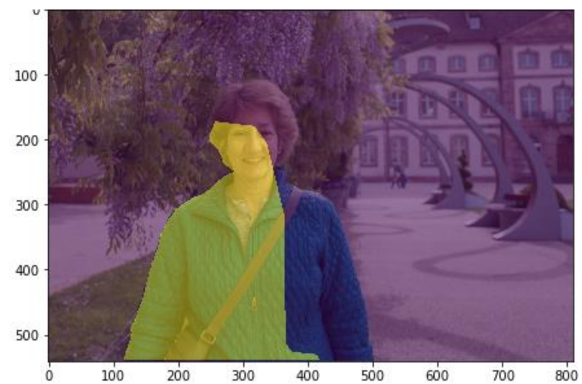
Ground Truth

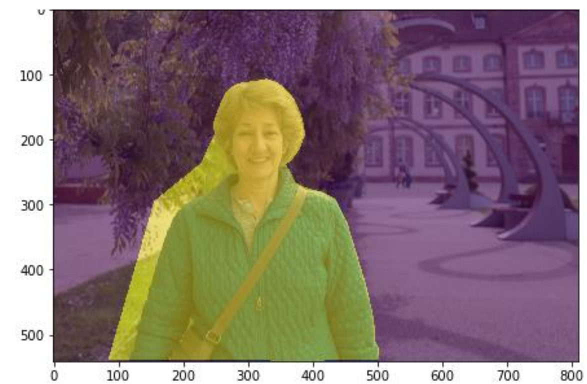




Prediction

Intersection
A∩B





Union
A∪B

https://www.jeremyjordan.me/evaluating-image-segmentation-models/