Bayesian Deep Learning for Safe AI

Jiyoon Lee

April 24, 2020

Uncertainty

Uncertainty

Bayesian approach

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

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

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

Non-Bayesian approach

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell
DeepMind
{balajiln,apritzel,cblundell}@google.com

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

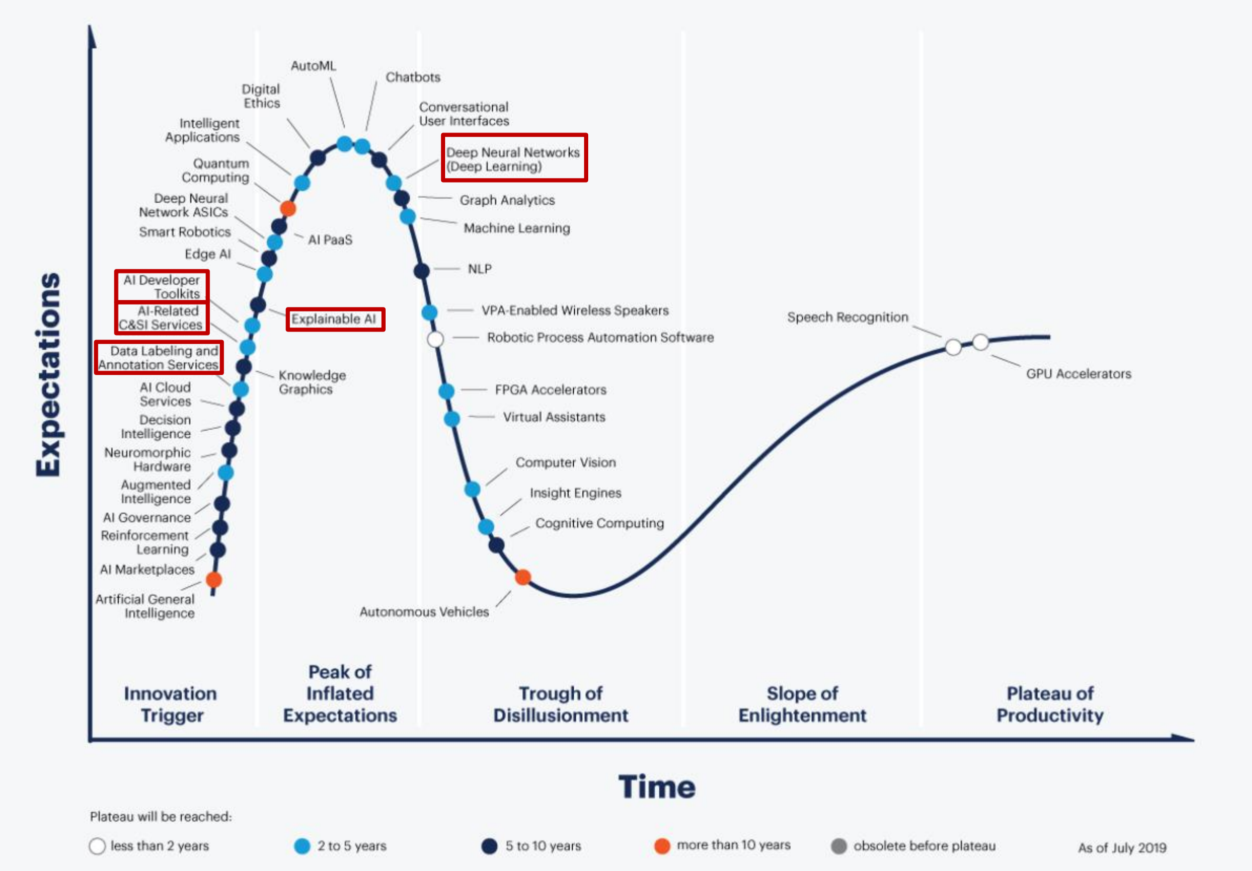
Contents

1. Introduction
   - Background
   - Importance of Uncertainty
   - Intuition of Uncertainty
   - Types of Uncertainties
2. Bayesian Neural Networks
   - Frequentist way & Bayesian way
   - Dropout as Bayesian Approximation
   - Bayesian Neural Networks for Computer Vision
3. Non-Bayesian Approaches
   - Simple and Scalable Deep Ensembles
4. Applications
   - Uncertainty-Aware Hierarchical Feedforward Attention Network
5. Conclusions
6. Appendix
   - Posterior Approximation using variational inference

Introduction

Importance of Uncertainty

❖ Gartner Hype Cycle for Emerging Technologies, 2019



Deep Neural Nets

AI Developer Toolkits

AI-Related C&SI Services

Data Labeling and Annotation

Explainable AI

Introduction

Background

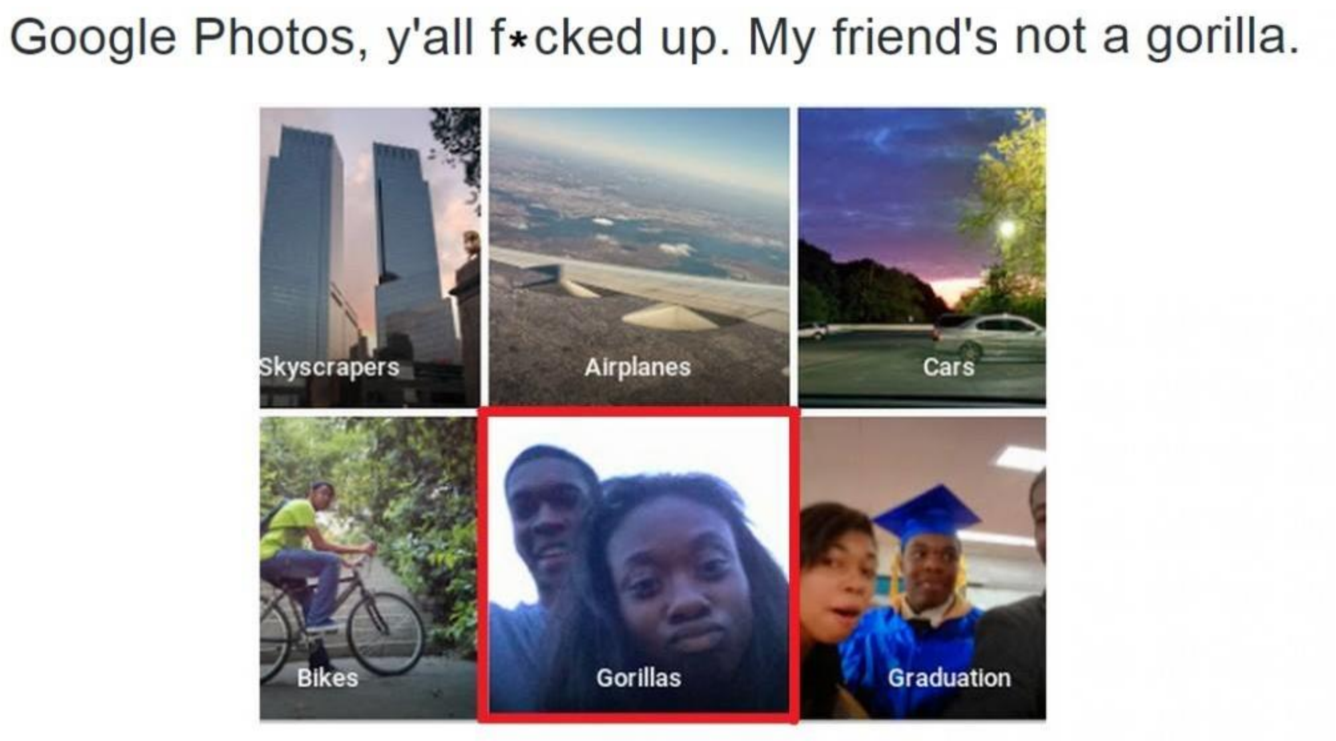
"We expected"

Deep learning

Introduction

Background

“기계학습 맹점, 흑인 사진 ‘고릴라’로 인식… 구글, 인종차별 사태 수습에 진땀”



https://news.joins.com/article/18159469

Introduction

Background

"테슬라 사고는 역광 때문… 눈 , 비 등 '악천후', 자율주행 난관으로 떠올라"

https://news.joins.com/article/22520461

예측 결과 믿을 수 있나요?

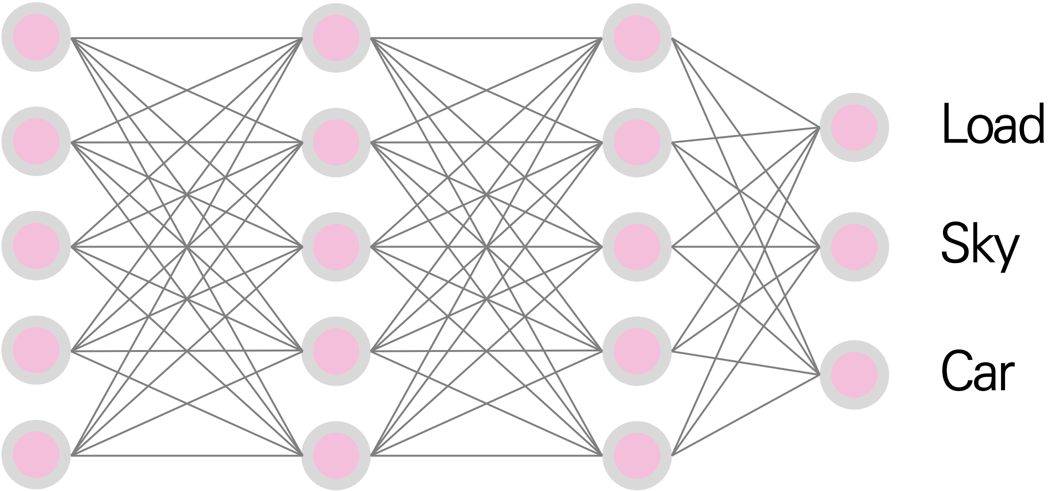
예측 결과에 대해 얼마나 확신하는가에 대한 정보가 필요

“ Uncertainty ”

Introduction

Intuition of Uncertainties

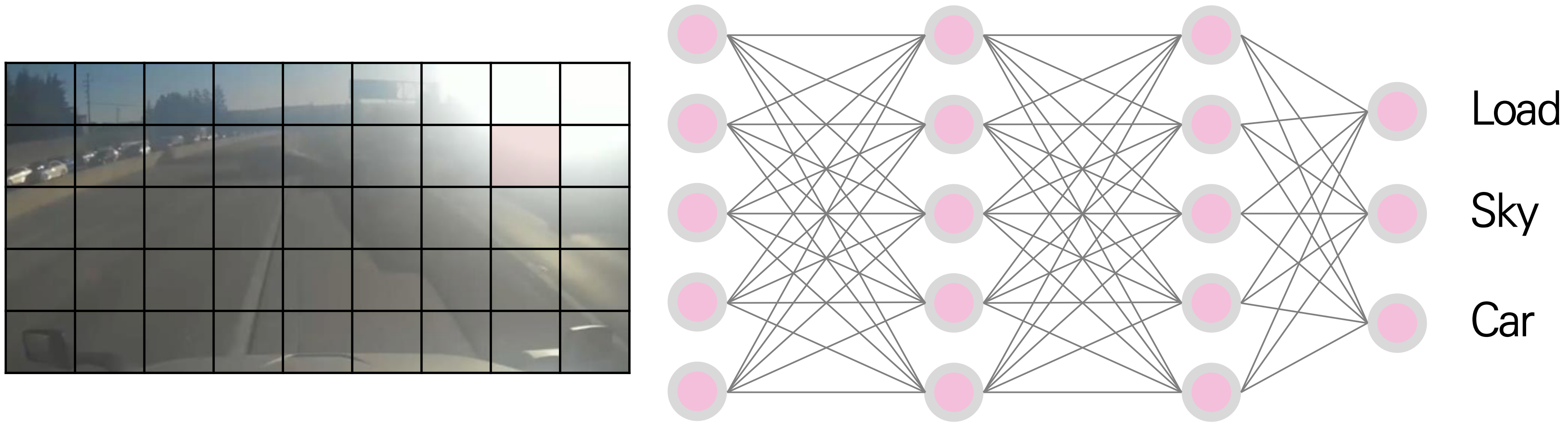
❖ Classification



Introduction

Intuition of Uncertainties

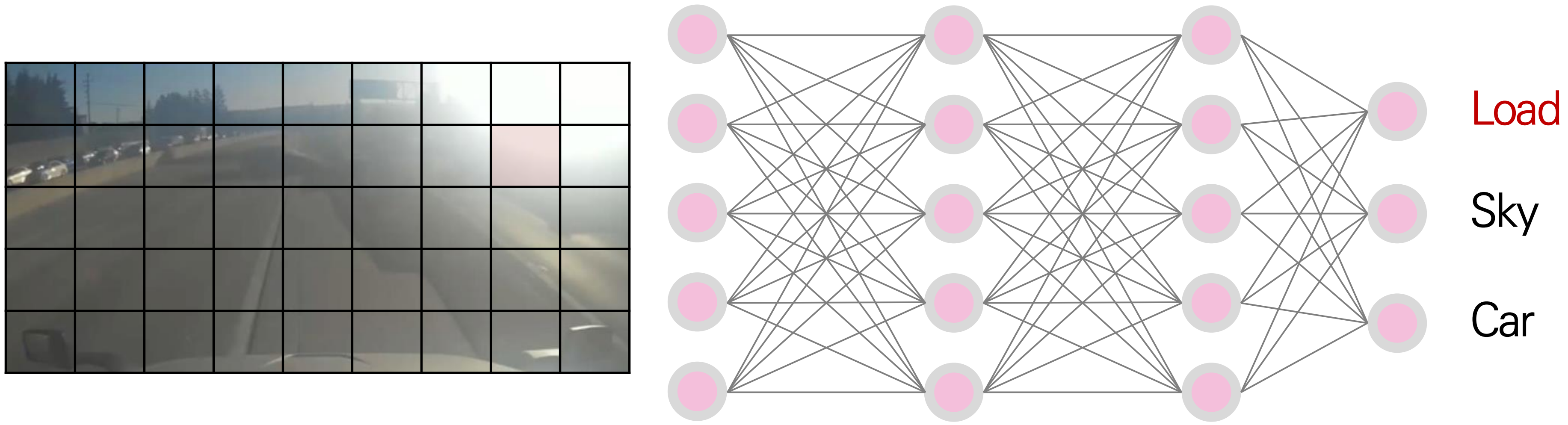
❖ Classification



Introduction

Intuition of Uncertainties

❖ Classification



Introduction

Intuition of Uncertainties

❖ Classification + Uncertainty estimation

➢ 오토파일럿 모드에서 운전자에게 운전 권한을 전환하기 위한 의사결정 기준으로 활용

Uncertainty 추정에

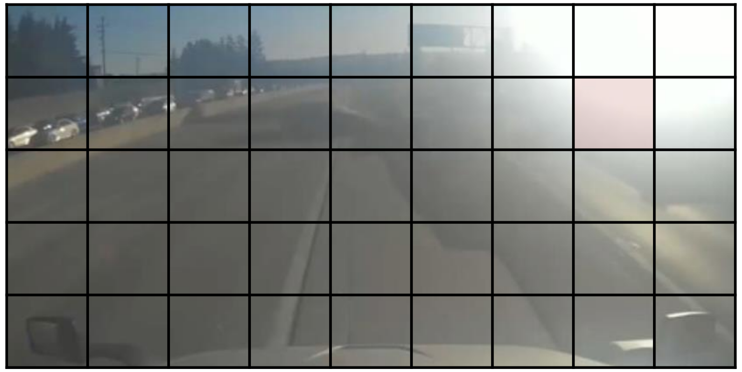
예측 확률을 적용해보자

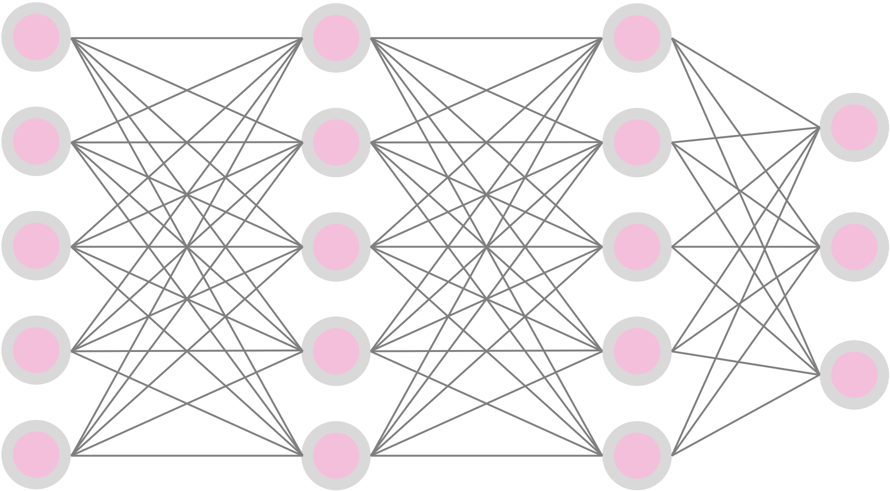
Introduction

Intuition of Uncertainties

❖ Standard Deep Neural Networks

➢ Softmax는 logit값을 확률 값으로 변환함으로써 예측확률이 도출





Load $P(y = load) = 0.5$

Sky $P(y = sky) = 0.2$

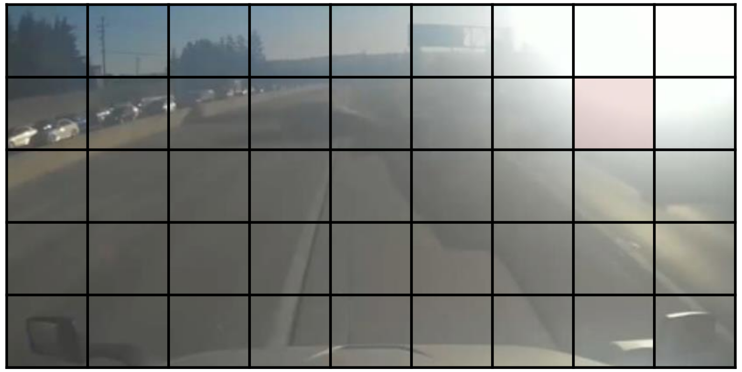
Car $P(y = car) = 0.3$

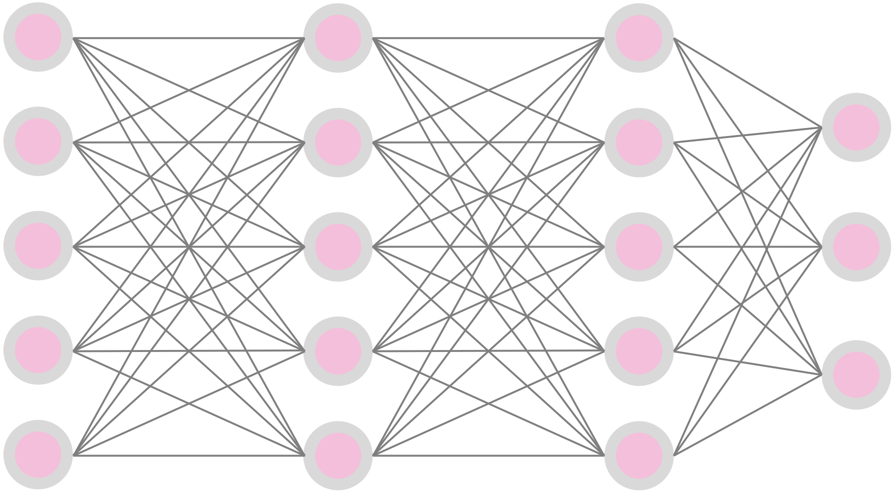
Introduction

Intuition of Uncertainties

❖ Standard Deep Neural Networks

- ➢ Softmax는 logit값을 확률 값으로 변환함으로써 예측확률이 도출
- ➢ 예측확률을 uncertainty에 대한 지표로 활용해보자





Load $P(y = load) = 0.5$

Sky $P(y = sky) = 0.2$

Car $P(y = car) = 0.3$

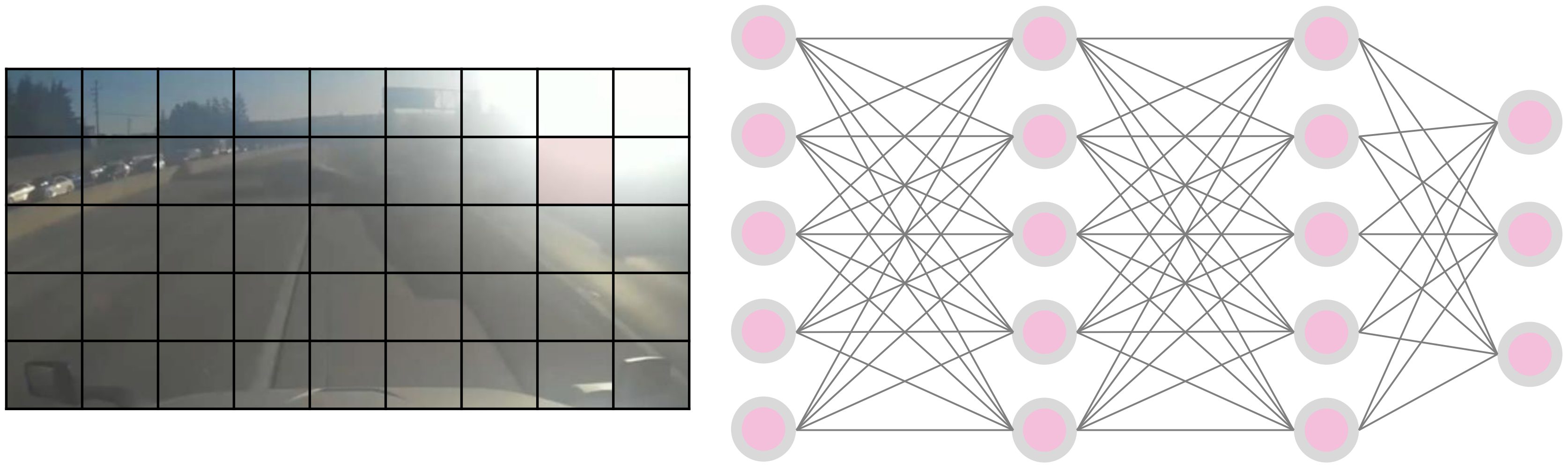
High Uncertainty, $if\ argmax\ \ P(\hat{y}) \leq 0.6$

Introduction

Intuition of Uncertainties

❖ Standard Deep Neural Networks

- ➢ Softmax로부터 도출된 예측확률의 경우, 불확실하더라도 높은 경향을 보이는 경우 존재 "overconfidence"
- ➢ 예측확률을 uncertainty로 규정하는 것은 부적절



Load $P(y = load) = 0.8$

Sky $P(y = sky) = 0.1$

Car $P(y = car) = 0.1$

High Uncertainty, $if\ argmax\ \ P(\hat{y}) \leq 0.6$

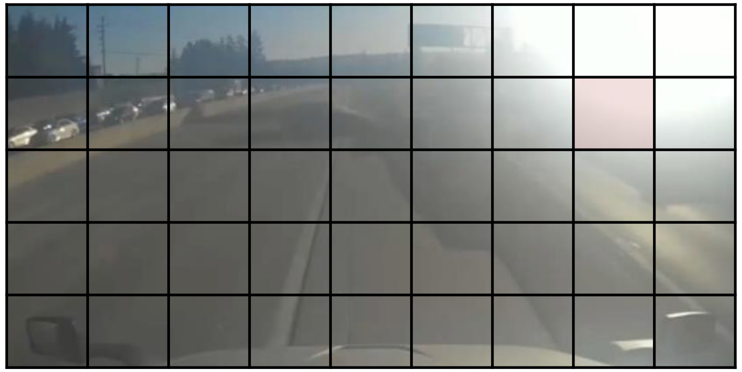
Reject

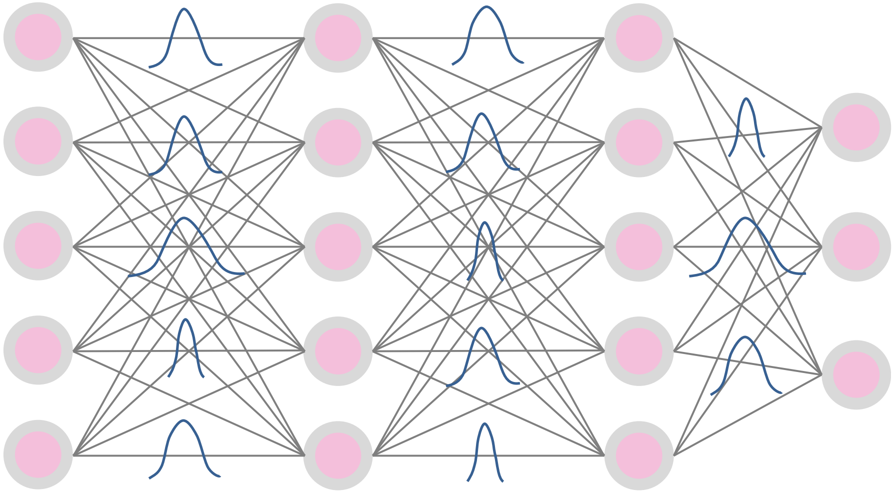
Introduction

Intuition of Uncertainties

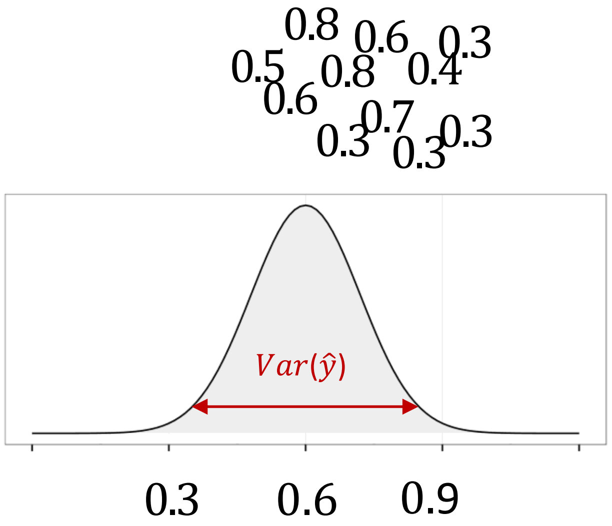
❖ Bayesian Neural Networks

- 예측에 대한 uncertainty를 추정하는 것이 목표
- 예측 값을 분포로 추정할 수 있음





Load $P(y = load) = 0.8$



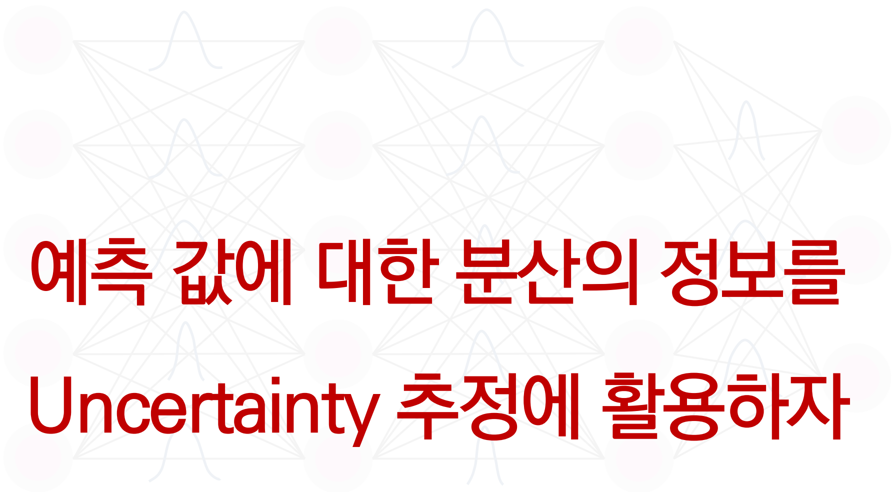
Introduction

Intuition of Uncertainties

❖ Bayesian Neural Networks

➢ 예측에 대한 uncertainty를 추정하는 것이 목표

➢ 예측 값을 분포로 추정할 수 있음



예측 값에 대한 분산의 정보를

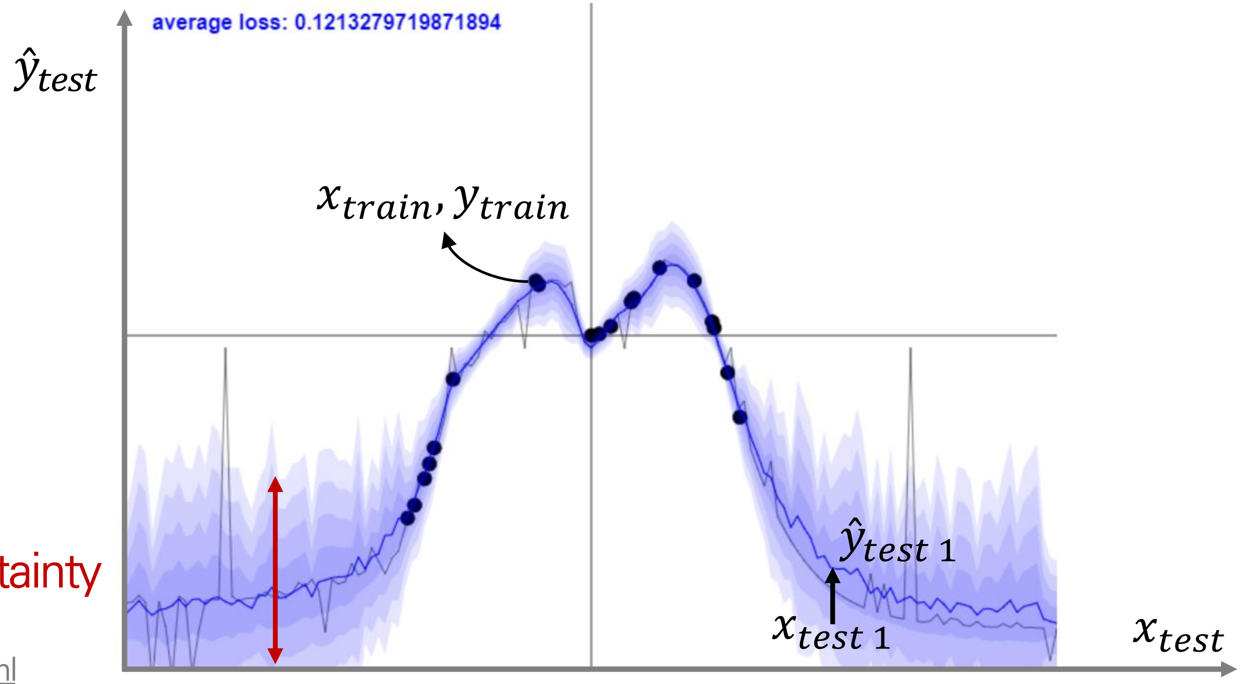
Uncertainty 추정에 활용하자

Introduction

Intuition of Uncertainties

❖ Bayesian Neural Networks Results

- Point: train data points
- Black line: $\hat{y}_{test}$
- Blue line: $E(\hat{y}_{test})$, Blue shade: $Var(\hat{y}_{test})$



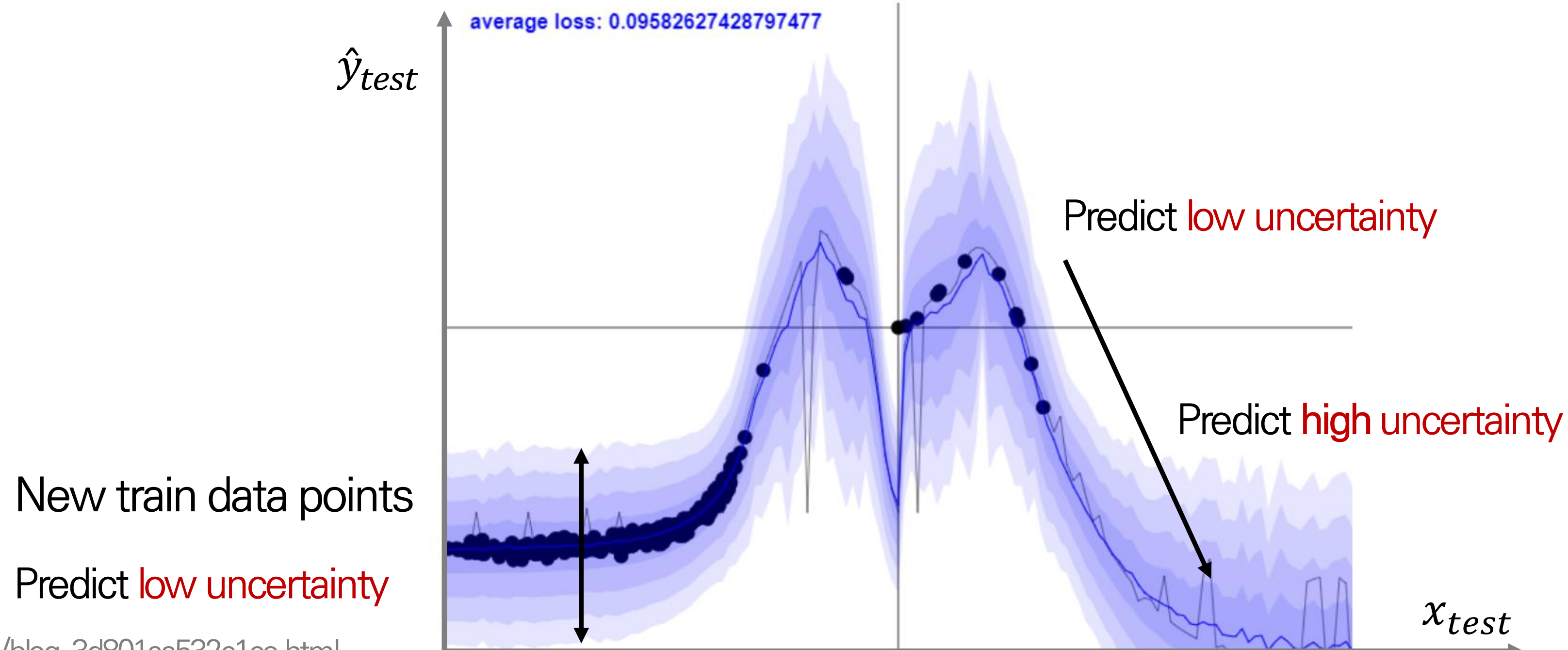
http://mlg.eng.cam.ac.uk/yarin/blog_3d801aa532c1ce.html

Introduction

Intuition of Uncertainties

❖ Bayesian Neural Networks Results

- ➢ Point: train data points
- ➢ Black line: $\hat{y}_{test}$
- ➢ Blue line: $E(\hat{y}_{test})$, Blue shade: $Var(\hat{y}_{test})$



http://mlg.eng.cam.ac.uk/yarin/blog_3d801aa532c1ce.html

Bayesian Neural Net 목표

Uncertainty 추정하고,

더 나은 예측 성능 도출하자

Uncertainty: 예측에 대한 불확실성

모델이 불확실한 경우

데이터가 불확실한 경우

Introduction

Types of Uncertainties

❖ Epistemic uncertainty (model uncertainty)

- ➢ 모델이 데이터에 대해 얼마나 적합하게 구축되었는지에 대해 모르는 정도
- ➢ 데이터의 어떤 특징을 학습하는지에 대해 모르는 정도
- ➢ 더 많은 데이터가 학습된다면 줄일 수 있음, reducible uncertainty

❖ Aleatoric uncertainty (data uncertainty)

- ➢ 데이터에 내재된 노이즈로 인해 이해하지 못하는 정도
  (e.g. measurement noise, randomness inherent in the coin flipping)
- ➢ 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- ➢ 측정 정밀도를 높이면 줄일 수 있음

구조차량이 촬영한 지난해 9월 테슬라 자동주행(오토파일럿 모드) 차량의 중앙분리대 충돌사고 현장. 지난달 사망 사고처럼 오전 역광이 내리쬐는 상황에서 발생했다. 2016년 발생한 트레일러 충돌사고도 역광이 원인이었다.[미 ABC 방송 캡처]

Tesla 자율주행 사고

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).

Introduction

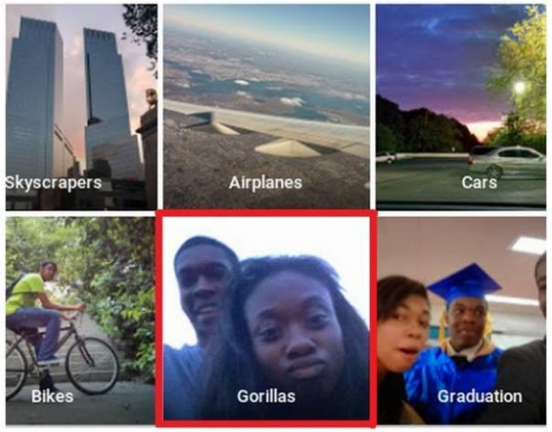
Types of Uncertainties

❖ Epistemic uncertainty (model uncertainty)

➢ 모델이 데이터에 대해 얼마나 적합하게 구축되었는지에 대해 모르는 정도

➢ 데이터의 어떤 특징을 학습하는지에 대해 모르는 정도

➢ 더 많은 데이터가 학습된다면 줄일 수 있음, reducible uncertainty



Google photo 오분류

❖ Why Epistemic uncertainty?

➢ Epistemic uncertainty는 학습데이터가 부족하여 학습되지 않은 상태를 식별할 수 있기 때문에 중요

➢ 높은 불확실성은 모델은 추가적인 학습이 필요할 가능성이 높다는 의미로, 안전이 중요한 문제상황에서 높게 발생하는 경우 모델을 신뢰할 수 없음

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).

Introduction

Types of Uncertainties

❖ Aleatoric uncertainty (data uncertainty)

- ➢ 데이터에 내재된 노이즈로 인해 이해하지 못하는 정도
  (e.g. measurement noise, randomness inherent in the coin flipping)
- ➢ 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- ➢ 측정 정밀도를 높이면 줄일 수 있음

구조차량이 촬영한 지난해 9월 테슬라 자동주행(오토파일럿 모드) 차량의 중앙분리대 충돌사고 현장. 지난달 사망 사고처럼 오전 역광이 내리쬐는 상황에서 발생했다. 2016년 발생한 트레일러 충돌사고도 역광이 원인이었다.[미 ABC 방송 캡처]

Tesla 자율주행 사고

❖ Why Aleatoric uncertainty?

- ➢ Aleatoric uncertainty는 실제 상황에서와 같이 일부 데이터의 노이즈가 높게 존재하는 경우 중요
- ➢ 노이즈가 큰 데이터에 대해 학습과정에서 제약을 부여할 수 있으므로, 예측 성능 안정화 과정에 기여

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).

Introduction

Importance of Uncertainties

- ❖ Explainable AI
  - ➢ 모델과 데이터에 대한 불확실한 정도를 표현함으로써 설명력 향상
- ❖ Medical imaging
  - ➢ 헬스케어 도메인과 같이
- ❖ Autonomous vehicles
  - ➢ 운전 프로세스의 의사결
- ❖ Active learning
  - ➢ 어떠한 데이터에 레이블을 부여할지에 대한 문제에 적합
- ❖ Out-of-distribution detection
  - ➢ Unknown class를 구분해내는 문제에 적용 가능

Nearly *all applications* !

Uncertainty

Bayesian approach

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

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

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

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

Non-Bayesian approach

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan Alexander Pritzel Charles Blundell
DeepMind
{balajiln,apritzel,cblundell}@google.com

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

Bayesian Neural Networks

References

❖ ICML 2016

❖ 1747회 인용건수

Yarin Gal

Associate Professor, University of Oxford
Verified email at cs.ox.ac.uk - Homepage

Machine Learning Artificial Intelligence Probability Theory Statistics

FOLLOW

TITLE	CITED BY	YEAR
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)	1747	2015
A theoretically grounded application of dropout in recurrent neural networks Y Gal, Z Ghahramani Advances in Neural Information Processing Systems, 1019-1027	888	2016
What uncertainties do we need in Bayesian deep learning for computer vision? A Kendall, Y Gal Advances in neural information processing systems, 5574-5584	806	2017
Uncertainty in Deep Learning Y Gal University of Cambridge	523	2016
Multi-task learning using uncertainty to weigh losses for scene geometry and semantics A Kendall, Y Gal, R Cipolla Proceedings of the IEEE Conference on Computer Vision and Pattern ...	436	2018
Deep Bayesian Active Learning with Image Data Y Gal, R Islam, Z Ghahramani International Conference on Machine Learning (ICML), 1183-1192	295	2017
Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference Y Gal, Z Ghahramani 4th International Conference on Learning Representations (ICLR) workshop track	289	2015
Concrete dropout Y Gal, J Hron, A Kendall Advances in Neural Information Processing Systems, 3581-3590	162	2017
Distributed variational inference in sparse Gaussian process regression and latent variable models Y Gal, M van der Wilk, C Rasmussen Advances in Neural Information Processing Systems, 3257-3265	133	2014

Bayesian Neural Networks

Frequentist way & Bayesian way

Bayesian Neural Networks

❖ Frequentist (빈도주의자)

- 모델의 파라미터는 고정적

  Parameter is deterministic

- 확률은 사건의 빈도(데이터)를 기반으로 도출

  Probabilities are fundamentally related to frequencies of events

- Linear regression

  $y = X\beta + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2)$

❖ Bayesian (베이지안)

- 모델의 파라미터에 분포를 가정

  Parameter is stochastic

- 확률은 우리가 갖고 있는 사전 지식과 데이터를 활용하여 추정

  Probabilities are fundamentally related to frequencies of events

- Bayesian linear regression

  $y = X\beta + \varepsilon \qquad \beta \sim N(0, \alpha^{-1}\mathrm{I_p})$

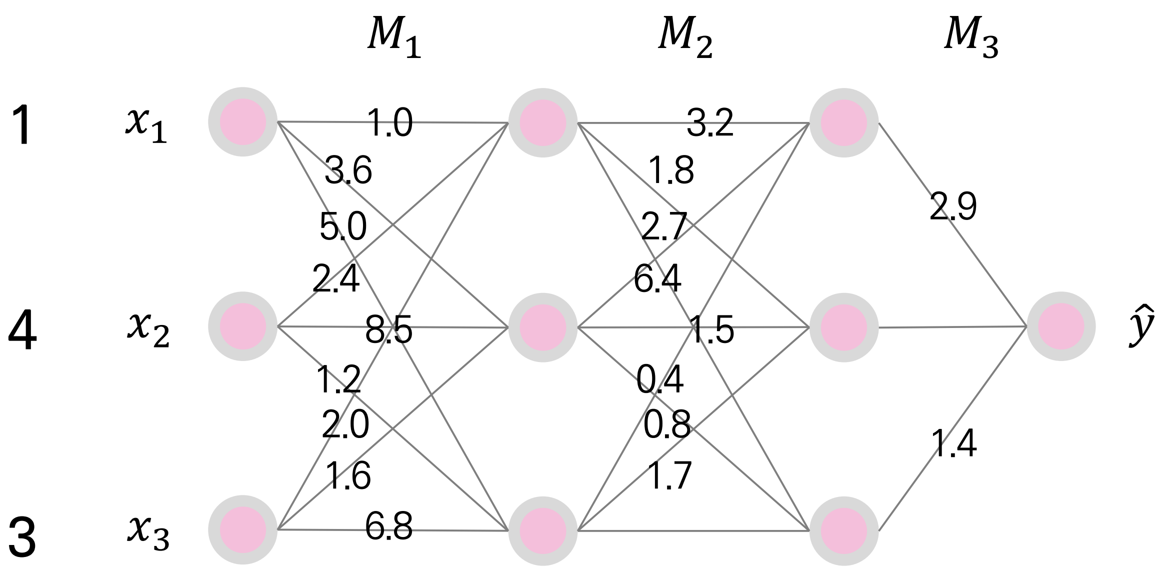
  $\varepsilon \sim N(0, \sigma^2)$

Bayesian Neural Networks

Frequentist way & Bayesian way

Bayesian Neural Networks

❖ Frequentist : Standard Deep Learning / Deterministic Deep Learning



3895	3895	3895	⋯	3895
T=1	T=2	T=3	⋯	T=T

동일한 입력 값에 대해서는 같은 예측 값이 도출

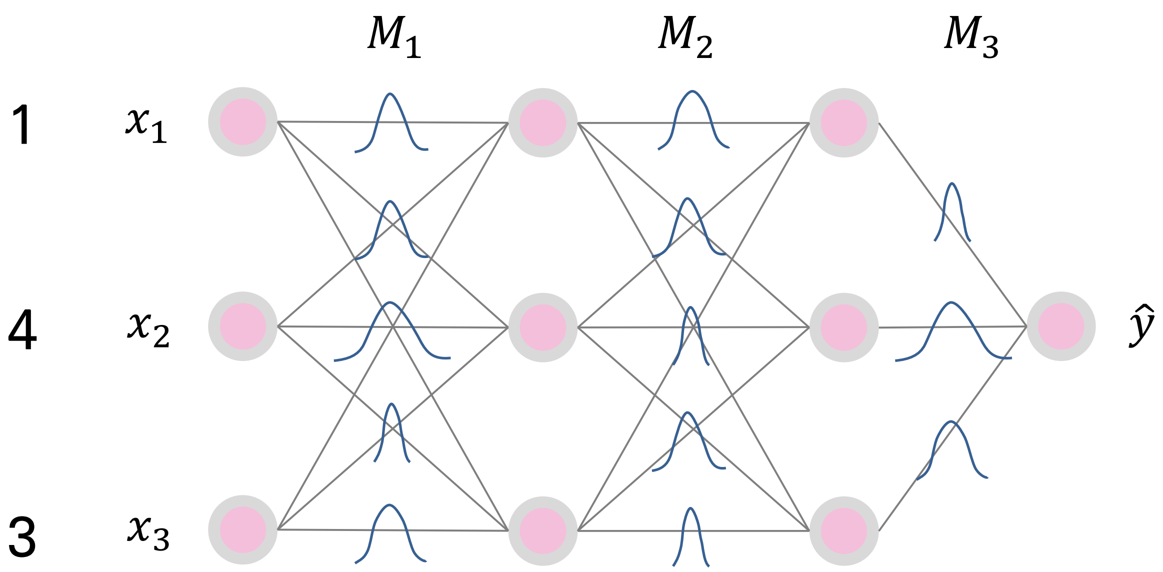
$\hat{y}$ = 3895

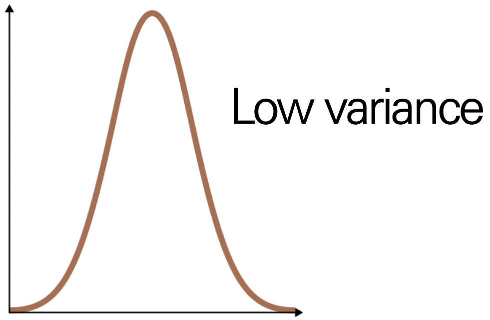
Bayesian Neural Networks

Frequentist way & Bayesian way

Bayesian Neural Networks

❖ Bayesian : Bayesian Deep learning / Stochastic Deep Learning





3895	3871	3767	⋯	3541
T=1	T=2	T=3	⋯	T=T

동일한 입력 값에 대해서도 **다른 예측 값이** 도출

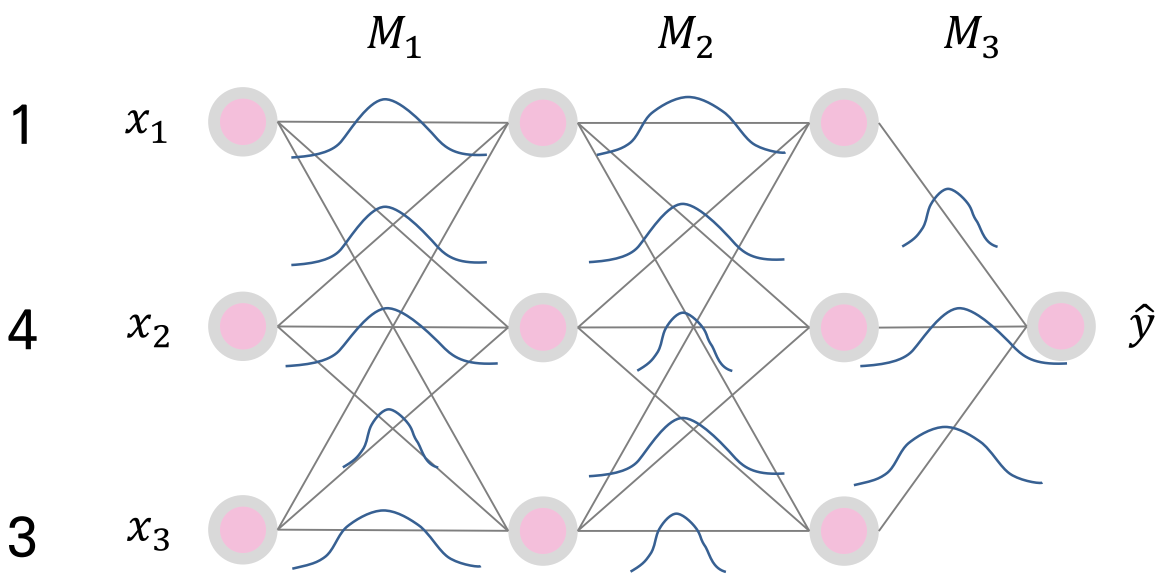
$\hat{y} \sim N(3895, 10^2)$

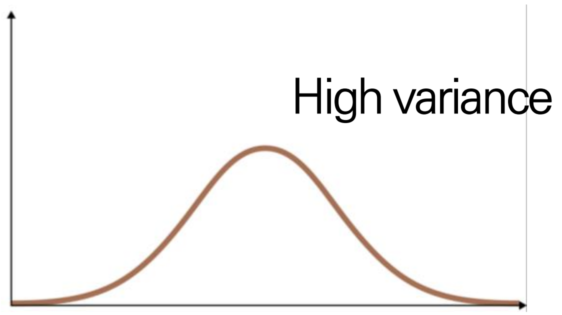
Bayesian Neural Networks

Frequentist way & Bayesian way

Bayesian Neural Networks

❖ Bayesian : Bayesian Deep learning / Stochastic Deep Learning





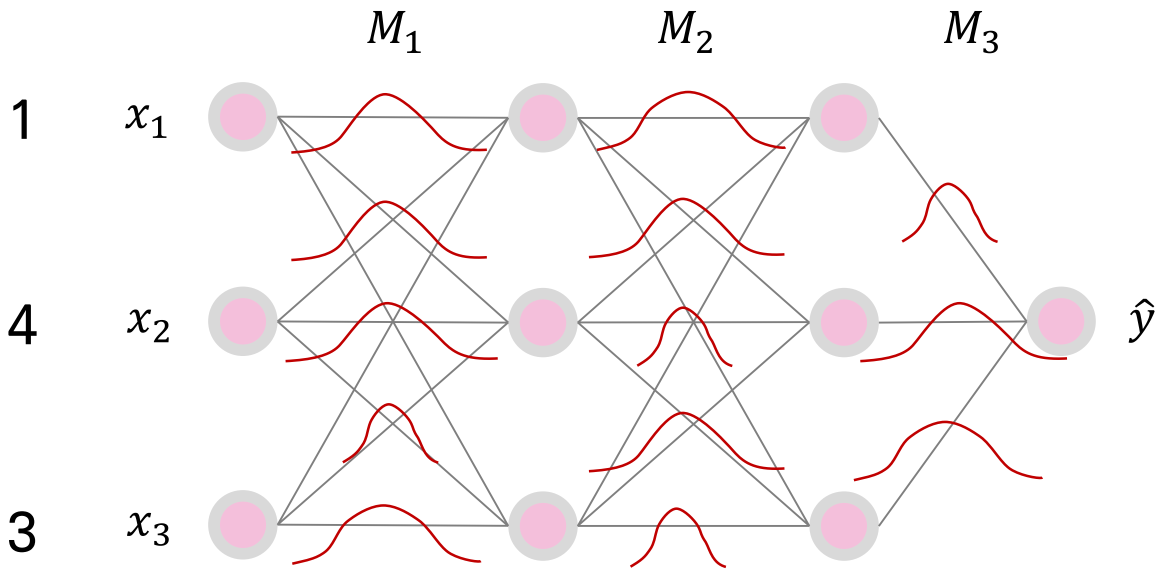
3895	5948	1767	⋯	6750
T=1	T=2	T=3	⋯	T=T

동일한 입력 값에 대해서도 **다른 예측 값이 도출**

$\hat{y} \sim N(3895, 3000^2)$

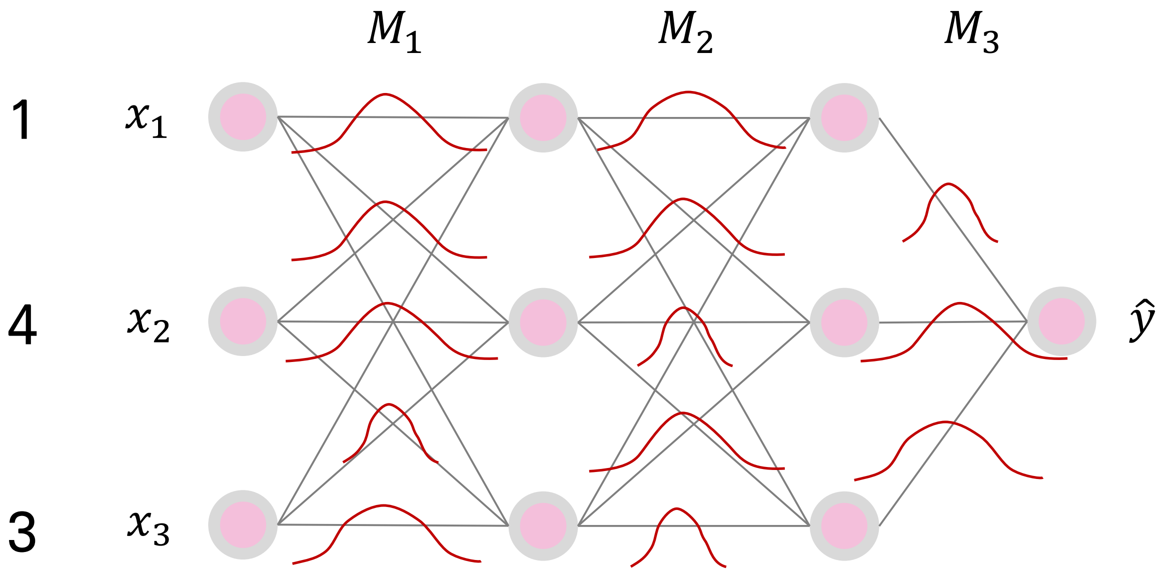
Bayesian Neural Networks

어떻게 parameter의 분포를 추정할까?



Bayesian Neural Networks

어떻게 parameter의 분포를 추정할까?



Posterior, Likelihood, Prior, Evidence

$$p(W|X,Y) = \frac{p(Y|X,W)p(w)}{p(Y|X)}$$

Evidence

$$p(Y|X) = \int p(Y|X,W)p(W)dw$$

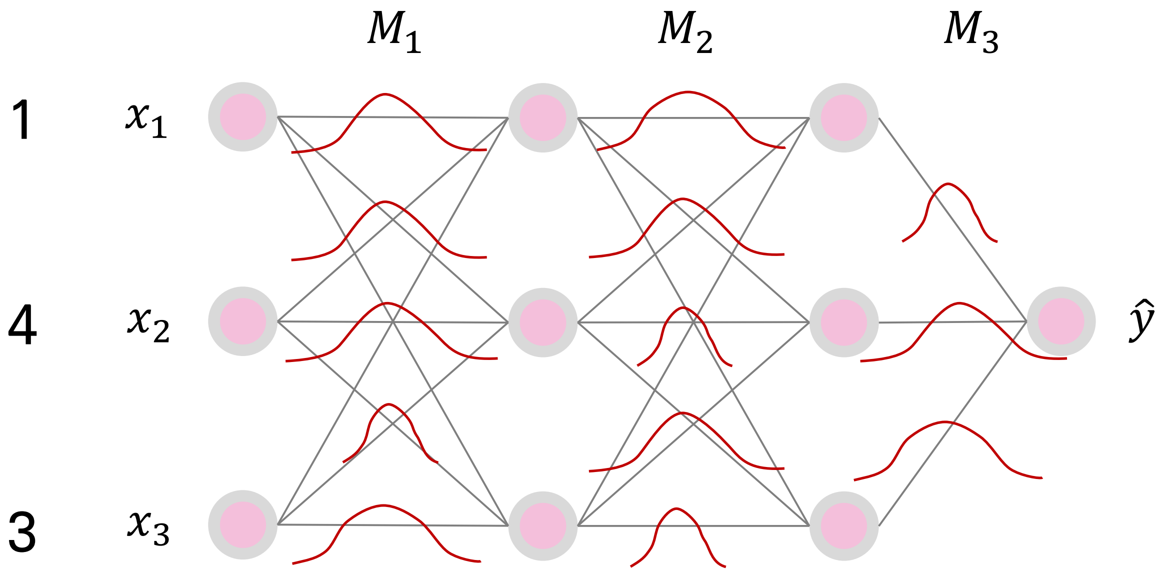
This integration is not computable in general

Bayesian Neural Networks

Frequentist way & Bayesian way

Bayesian Neural Networks

어떻게 parameter의 분포를 추정할까?



Posterior, Likelihood, Prior, Evidence

$$p(W|X,Y) = \frac{p(Y|X,W)\,p(w)}{p(Y|X)}$$

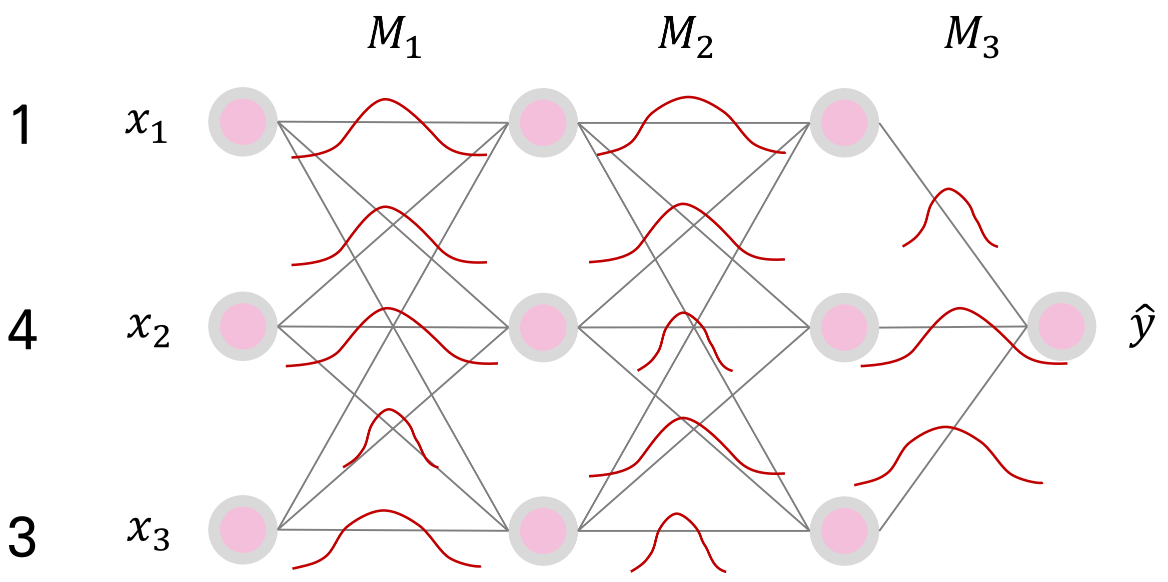
임의로 분포를 가정하고,
이를 posterior와 비슷하게 근사

$q_\theta(W)$

Variational distribution

Bayesian Neural Networks

어떻게 parameter의 분포를 추정할까?



Variational inference

Kullback-Leibler Divergence
(두 확률분포의 차이를 계산)

$$q_\theta(W)^* = \underset{q \in Q}{argmin}\ KL(q_\theta(W)||p(W|X,Y))$$

Posterior: $p(W|X,Y)$

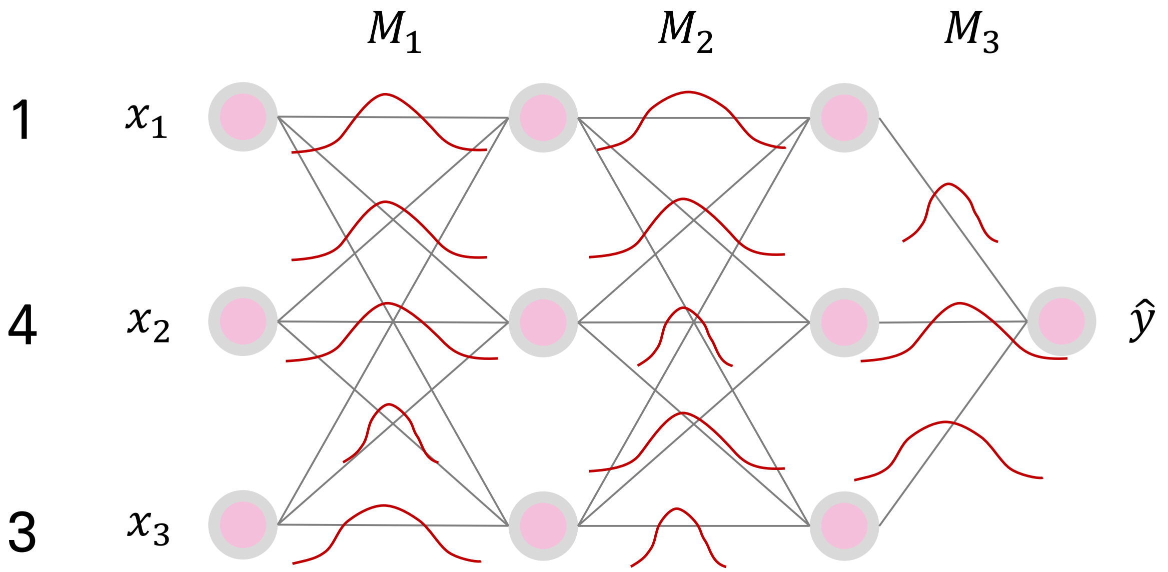
Variational distribution: $q_\theta(W)$

Bayesian Neural Networks

Frequentist way & Bayesian way

Bayesian Neural Networks

어떻게 parameter의 분포를 추정할까?



MC Dropout

with L2 regularization

Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ Loss function 정의 (Appendix 참고)

$$\textit{Minimize}\ \boldsymbol{KL(q_\theta(W)||p(W|X,Y))}$$

$$= \quad \textit{Maximize ELBO}$$

$$= \quad \textit{Minimize} -\sum_{i=1}^{N}\int q_\theta(W) ln\left(p(y_i|f^w(x_i))\right)dw + KL(q_\theta(W)||p(W))$$

$$= \quad \textit{Minimize} -\frac{N}{M}\sum_{i\in S} ln(p(y_i|f^{g(\theta,\epsilon)}(x_i))) + KL(q_\theta(W)||p(W))$$

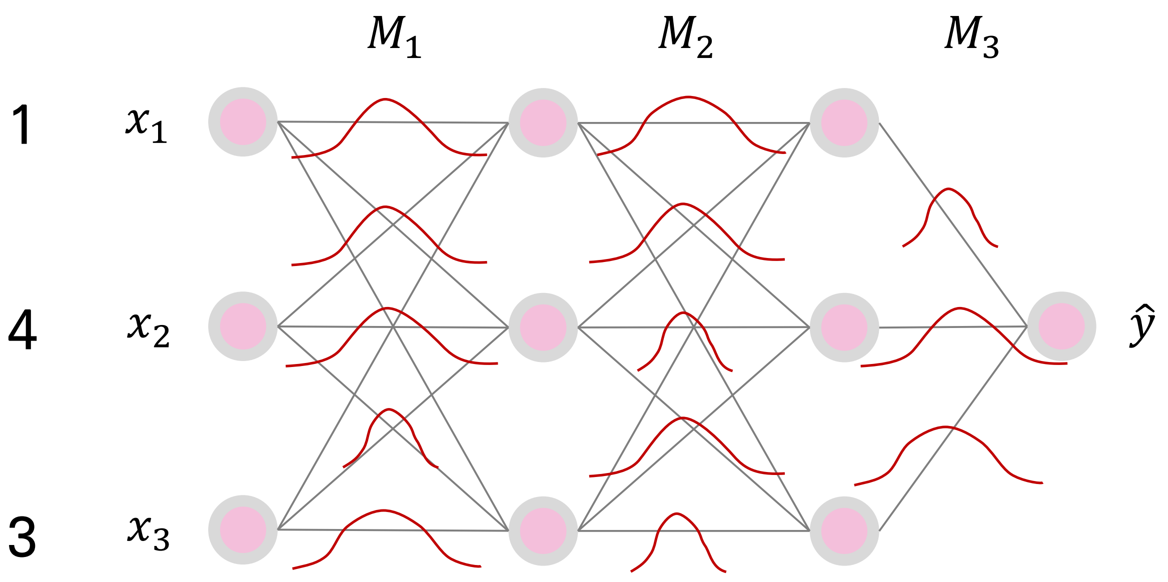
$$= \quad \textit{Minimize} -\boldsymbol{\frac{1}{M}\sum_{i\in S} ln(p(y_i|f^{g(\theta,\hat{\epsilon})}(x_i)) + \lambda_1||M_1||^2 + \lambda_2||M_2||^2 + \lambda_3||b||^2}$$

$$\boldsymbol{g(\theta,\hat{\epsilon}) = w_{l,i}}$$

Gal, Y. (2016). Uncertainty in deep learning. University of Cambridge, 1, 3.

Bayesian Neural Networks

어떻게 parameter의 분포를 추정할까?



MC Dropout

with L2 regularization

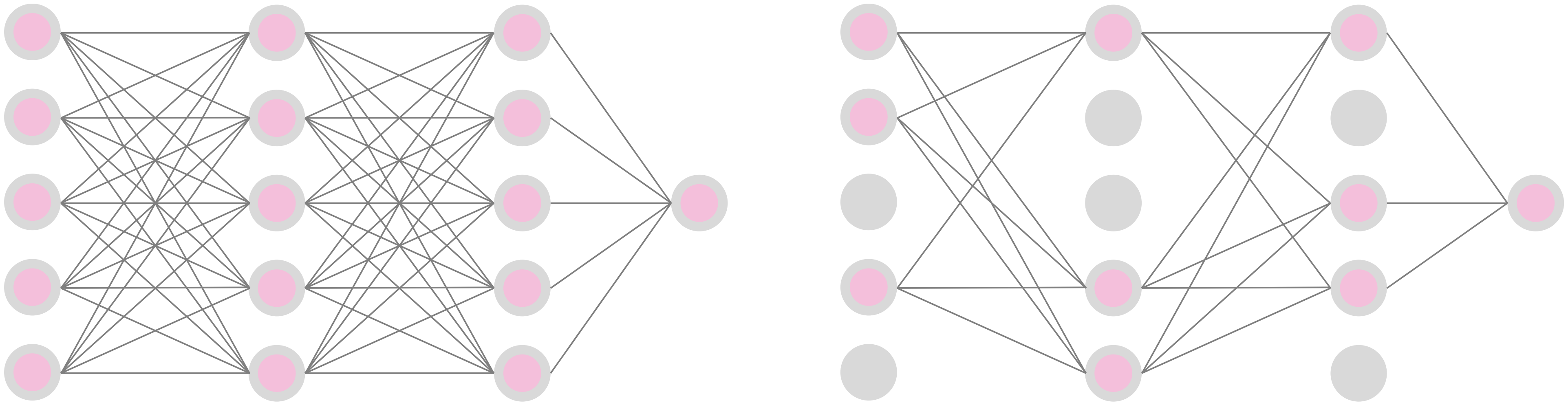
Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ What is Dropout?

- 모델 정규화(regularization) 방법으로, 미니배치마다 무작위로 노드 연결 끊음
- p : keep probability, $1-p$: dropout probability

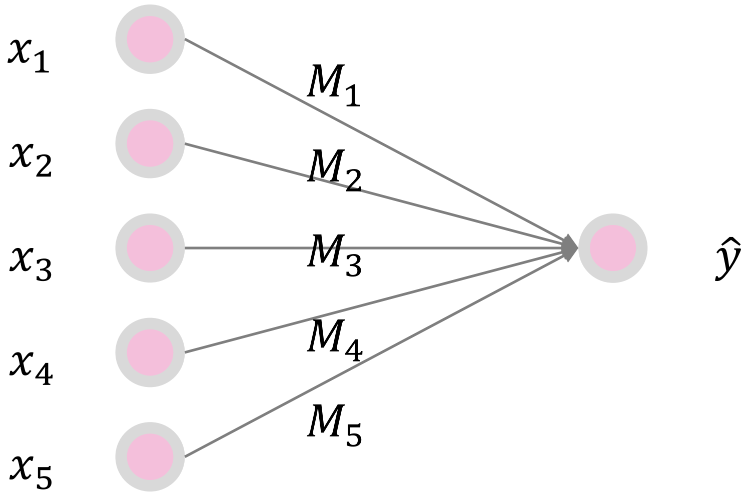
Standard Neural Net | After applying dropout



❖ Neural Net without Dropout

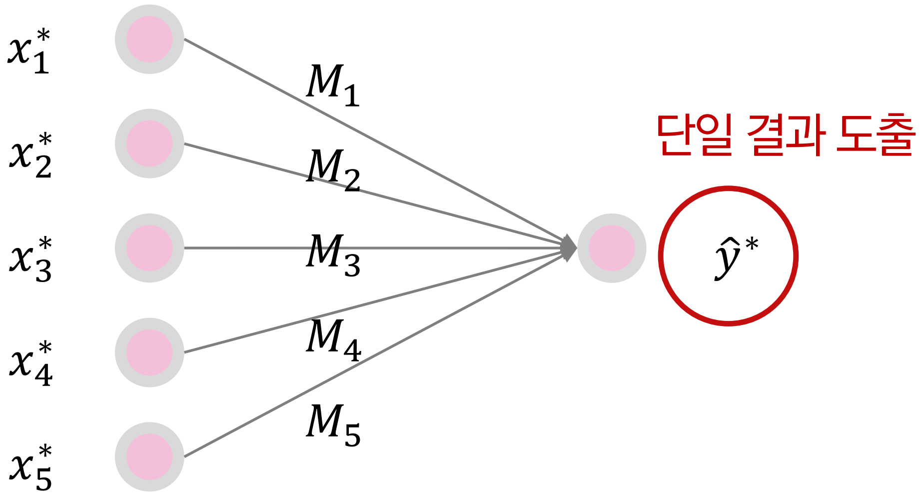
➢ 드롭아웃을 사용하지 않은 딥러닝 알고리즘의 경우 학습 이후 **추론 단계에는 파라미터가 고정적**

Training phase



$$\hat{y} = M_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5$$

Testing phase



$$\hat{y}^* = M_1 x_1^* + w_2 x_2^* + w_3 x_3^* + w_4 x_4^* + w_5 x_5^*$$

Bayesian Neural Networks

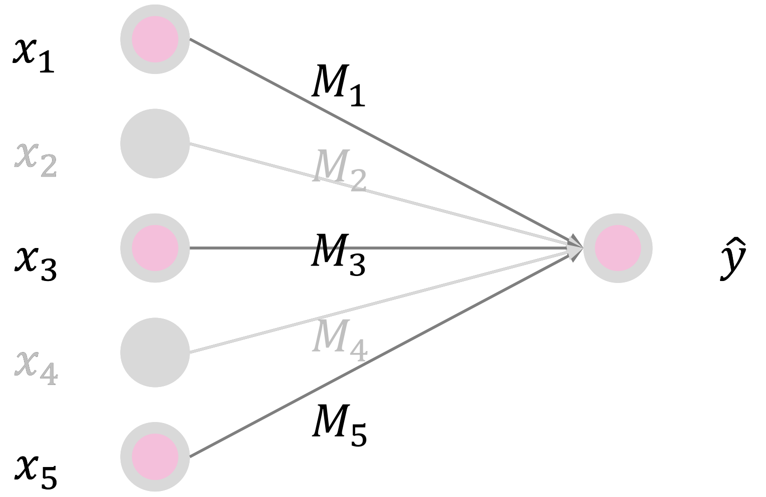
Dropout as Bayesian Approximation

❖ Neural Net with Dropout

➢ 드롭아웃을 사용하는 딥러닝 알고리즘의 경우 학습 이후 추론 단계에서는 **고정적인 파라미터에 가중치 p**를 곱함

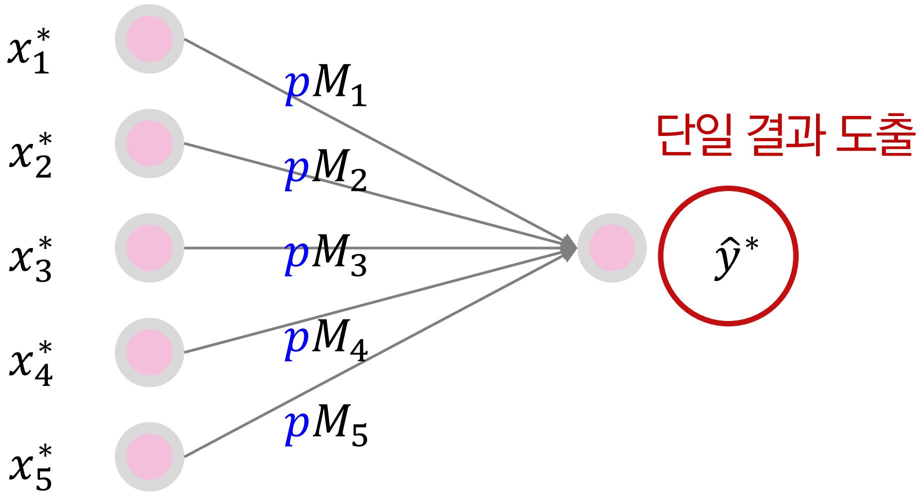
➢ p : keep probability, $1-p$: dropout probability

Training phase



$$\hat{y} = M_1x_1 + M_2x_2 + M_3x_3 + M_4x_4 + M_5x_5$$

Testing phase



$$\hat{y}^* = pM_1x_1 + pM_2x_2 + pM_3x_3 + pM_4x_4 + pM_5x_5$$

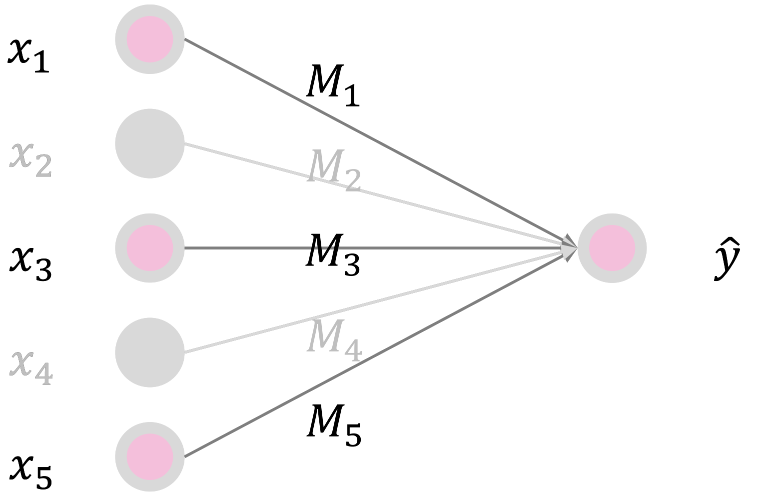
Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ Neural Net with MC Dropout

- 모델의 학습과정 뿐만 아니라 **추론 단계에서도 dropout 적용**
- T : the number of stochastic forward passes

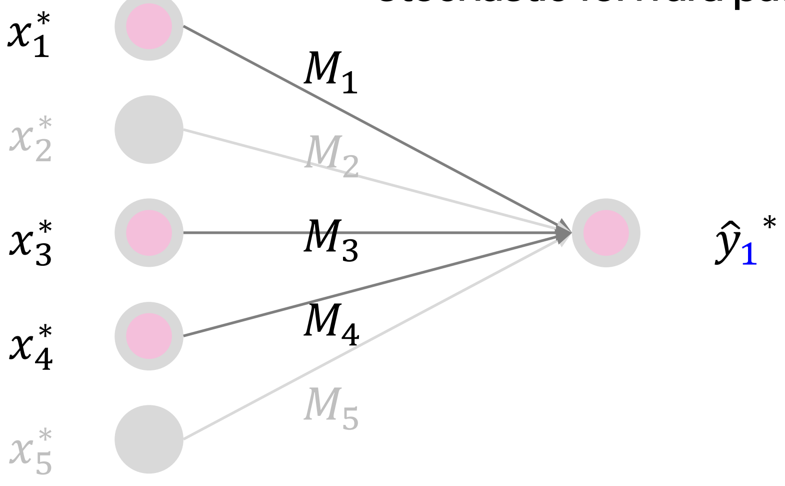
Training phase



$$\hat{y} = M_1 x_1 + M_2 x_2 + M_3 x_3 + M_4 x_4 + M_5 x_5$$

Testing phase

T stochastic forward passes



$$\hat{y}_1{}^* = M_1 x_1^* + M_2 x_2^* + M_3 x_3^* + M_4 x_4^* + M_5 x_5^*$$

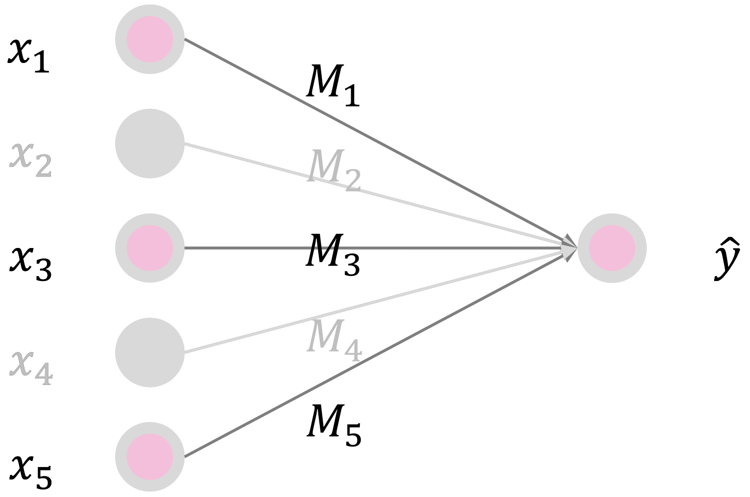
Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ Neural Net with MC Dropout

- 모델의 학습과정 뿐만 아니라 추론 단계에서도 dropout 적용
- ***T*** : the number of stochastic forward passes

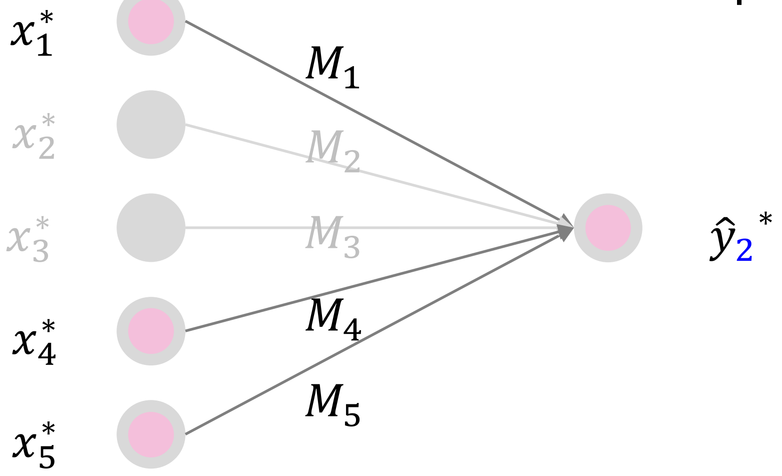
Training phase



$$\hat{y} = M_1x_1 + M_2x_2 + M_3x_3 + M_4x_4 + M_5x_5$$

Testing phase

T stochastic forward passes



$$\hat{y}_2{}^* = M_1x_1^* + M_2x_2^* + M_3x_3^* + M_4x_4^* + M_5x_5^*$$

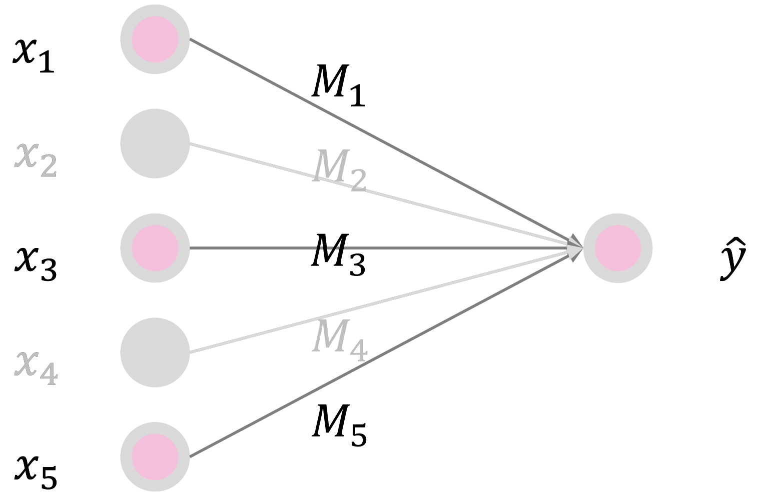
Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ Neural Net with MC Dropout

- 모델의 학습과정 뿐만 아니라 추론 단계에서도 dropout 적용
- ***T*** : the number of stochastic forward passes

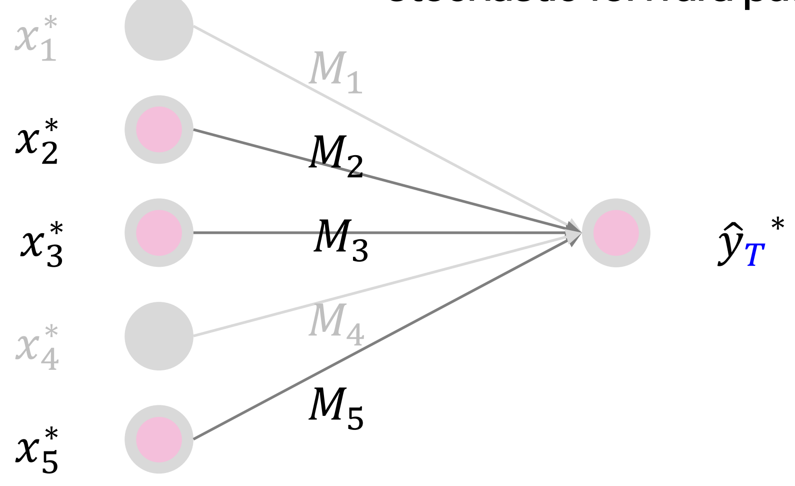
Training phase



$$\hat{y} = M_1x_1 + M_2x_2 + M_3x_3 + M_4x_4 + M_5x_5$$

Testing phase

T stochastic forward passes



$$\hat{y}_T{}^* = M_1x_1^* + M_2x_2^* + M_3x_3^* + M_4x_4^* + M_5x_5^*$$

Bayesian Neural Networks

Dropout as Bayesian Approximation

$\epsilon_j^{(l)} \sim Bernoulli(p)$

$\tilde{o}^{(l)} = \epsilon^{(l)} * o^{(l)}$

$z_i^{(l+1)} = M_i^{(l+1)} \tilde{o}^{(l)} + b_i^{(l+1)}$

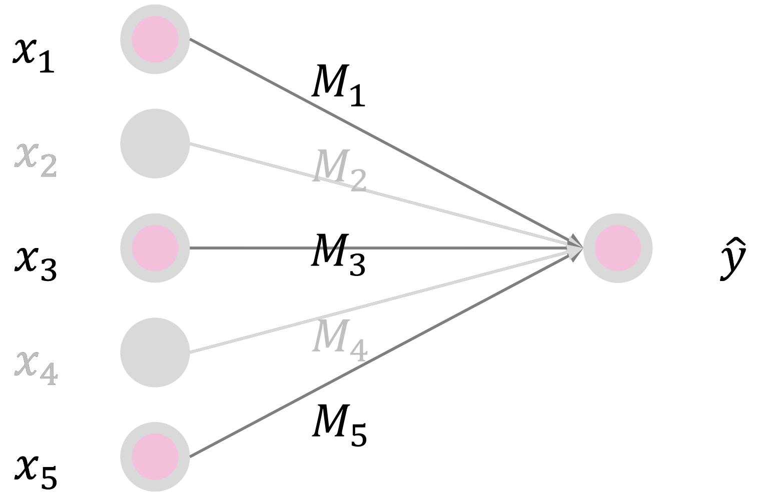
$w_i^{(l+1)}$

$y_i^{(l+1)} = f(z_i^{(l+1)})$

❖ Neural Net with MC Dropout

- 모델의 학습과정 뿐만 아니라 추론 단계에서도 dropout 적용
- ***T*** : the number of stochastic forward passes

Training phase



$$\hat{y} = M_1x_1 + M_2x_2 + M_3x_3 + M_4x_4 + M_5x_5$$

Testing phase

$$\hat{y}_1^* = M_1x_1^* + M_2x_2^* + M_3x_3^* + M_4x_4^* + M_5x_5^*$$

$$\hat{y}_2^* = M_1x_1^* + M_2x_2^* + M_3x_3^* + M_4x_4^* + M_5x_5^*$$

$$\vdots$$

$$\hat{y}_T^* = M_1x_1^* + M_2x_2^* + M_3x_3^* + M_4x_4^* + M_5x_5^*$$

다중 결과 도출

Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ Neural Net with MC Dropout

- 최종 예측 값은 ***T*** 번 도출한 예측 값의 평균을 사용
- ***T*** 번 도출한 예측 값의 분산을 epistemic uncertainty로 해석

Testing phase

$$\hat{y}_1^{\ *} = M_1 x_1^* + M_2 x_2^* + M_3 x_3^* + M_4 x_4^* + M_5 x_5^*$$

$$\hat{y}_2^{\ *} = M_1 x_1^* + M_2 x_2^* + M_3 x_3^* + M_4 x_4^* + M_5 x_5^*$$

$$\vdots$$

$$\hat{y}_T^{\ *} = M_1 x_1^* + M_2 x_2^* + M_3 x_3^* + M_4 x_4^* + M_5 x_5^*$$

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

최종 예측 값

$$Var(y^*) \approx \tau^{-1} \mathrm{I_D} + \frac{1}{T} \sum_{t=1}^{T} \widehat{y_t^*}^{\mathrm{T}} \widehat{y_t^*} - E(y^*)^T E(y^*)$$

$$\tau = \frac{pl^2}{2N\lambda}$$ p:probability of units not being dropped

Epistemic uncertainty

Bayesian Neural Networks

Dropout as Bayesian Approximation

❖ Loss function 정의

$$Minimize\ -\frac{1}{M}\sum_{i\in S} ln(p(y_i|f^{g(\theta,\hat{\epsilon})}(x_i))) + \lambda_1\|M_1\|_2^2 + \lambda_2\|M_2\|_2^2 + \lambda_3\|b\|_2^2$$

Regression: MSE

Classification: Softmax cross entropy

L2 regularization weighted with MC dropout

$\epsilon_j{}^{(l)} \sim Bernoulli(p)$

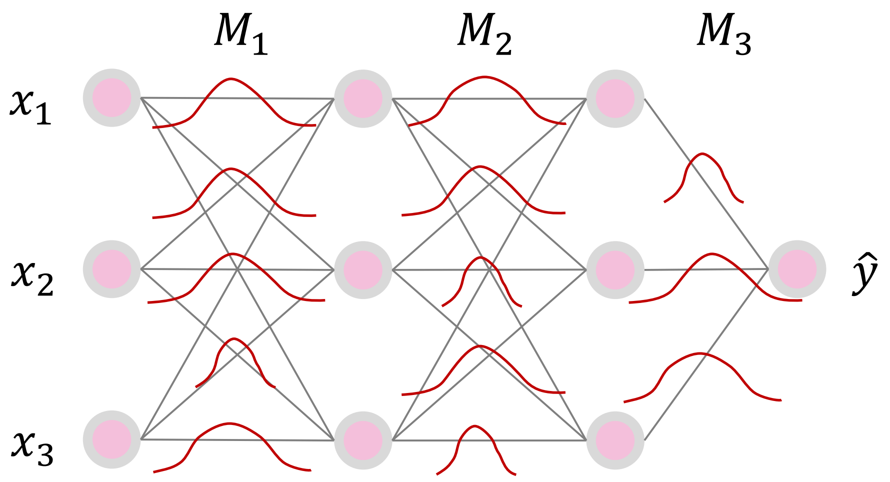
$\tilde{o}^{(l)} = \epsilon^{(l)} * o^{(l)}$

$z_i{}^{(l+1)} = M_i^{(l+1)}\tilde{o}^{(l)} + b_i^{(l+1)}$

$w_i^{(l+1)}$

$y_i{}^{(l+1)} = f\left(z_i{}^{(l+1)}\right)$

$g(\theta,\hat{\epsilon}) = w_{l,i}$

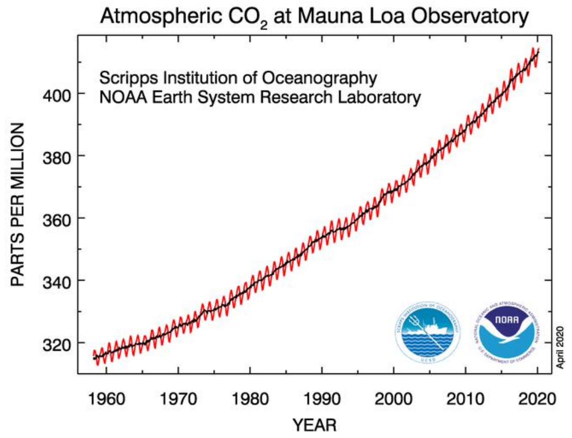


Bayesian Neural Networks

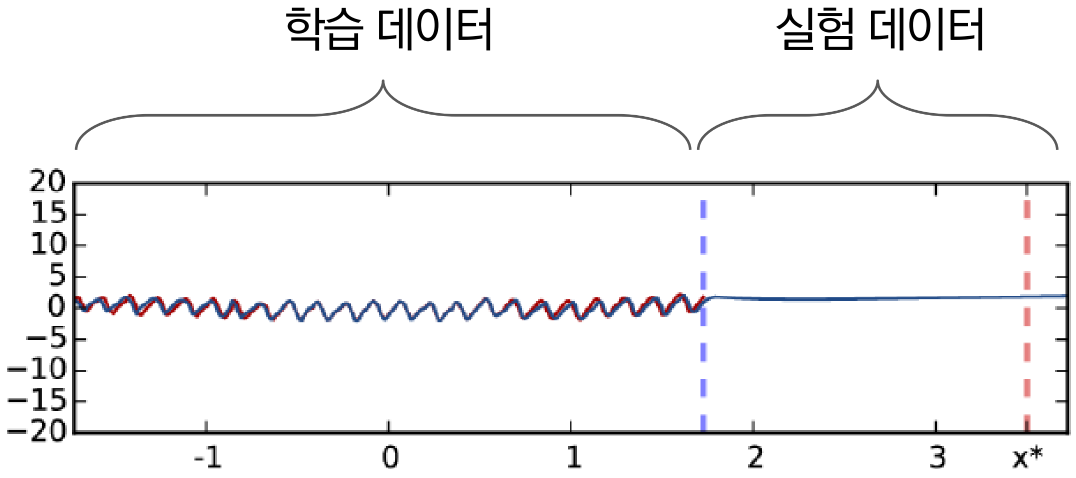
Dropout as Bayesian Approximation Results

❖ Predictive mean and uncertainties on the Mauna Loa CO_2 dataset

- Red: 실제 값
- Blue: 예측 값
- Red line: 학습 데이터와 충분히 다른 실험 데이터



Mauna Loa CO_2 dataset before pre-processing



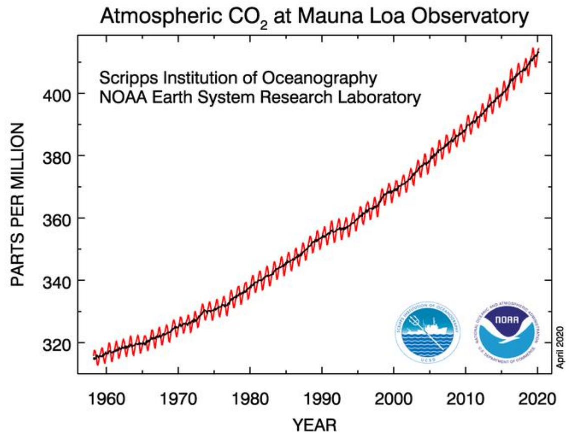
Standard dropout

Bayesian Neural Networks

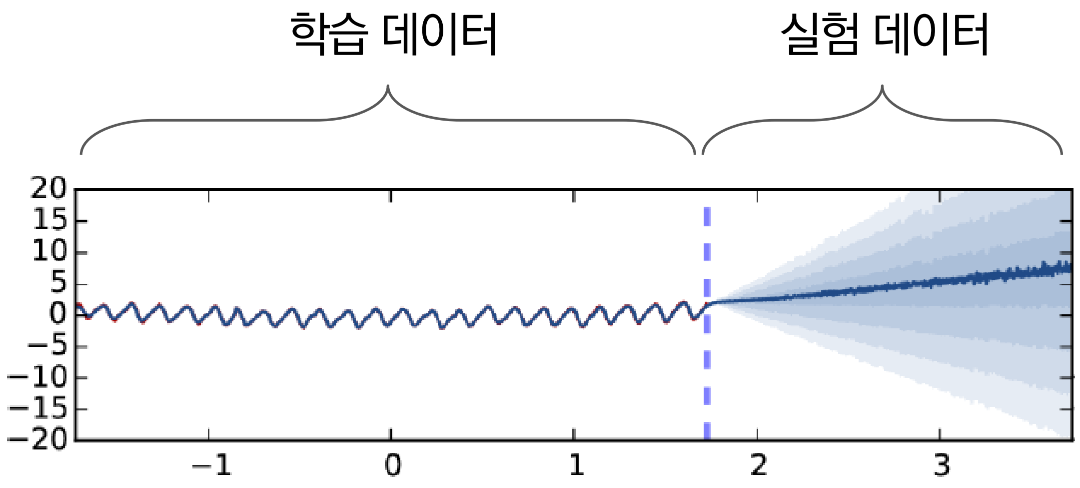
Dropout as Bayesian Approximation Results

❖ Predictive mean and uncertainties on the Mauna Loa CO_2 dataset

- ➢ Red: 실제 값
- ➢ Blue: 예측 값 / **Blue shade: 예측 값에 대한 2σ 신뢰 구간 (uncertainty 정보)**
- ➢ Red line: 학습 데이터와 충분히 다른 실험 데이터



Mauna Loa CO_2 dataset before pre-processing



MC dropout with ReLU non-linearities

Bayesian Neural Networks

Dropout as Bayesian Approximation Results

❖ Average test performance in RMSE and predictive log likelihood

- RMSE (root mean squared error) = $\sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}$
- Log likelihood = $-\frac{1}{N}\sum_{i=1}^{N}\frac{\|y_i-\hat{y}_i\|^2}{2\sigma^2} - \frac{1}{2}log\sigma^2$

	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**

Bayesian Neural Networks

Dropout as Bayesian Approximation Critic

❖ Uncertainty

- Dropout을 적용하여 Bayesian neural net을 구현함으로써 범용화에 기틀을 마련
- 모델의 불확실성인 epistemic uncertainty를 모델링

❖ Model performance

- Dropout과 L2 regularization term을 적용하여 overfitting을 방지, 성능 개선
- 모델의 불확실성인 epistemic uncertainty를 모델링하는 과정에서 도출되는 T 개의 예측 값을 평균하여 최종 예측 값으로 사용하기 때문에, outlier에 대한 보정이 가능

❖ Disadvantages

- Dropout rate에 의존적인 결과 도출
- 모델 수렴이 어려울 수 있고, standard neural net구조보다 학습 시간 오래 걸림

Bayesian Neural Networks

References

- ❖ NeurIPS 2017
- ❖ 803회 인용건수

What uncertainties do we need in **bayesian deep learning** for **computer vision**?

A Kendall, Y Gal - Advances in neural information processing ..., 2017 - papers.nips.cc

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 ...

☆ 99 803회 인용 관련 학술자료 전체 11개의 버전 ≫

- ❖ 추정하고자 하는 uncertainty가 더욱 세분화 됨
  - ➢ **Aleatoric uncertainty** as well as Epistemic uncertainty
- ❖ Computer vision tasks에 적용
  - ➢ CNN architecture에 적용 가능

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.

1 Introduction

Understanding what a model does not know is a critical part of many machine learning systems. Today, deep learning algorithms are able to learn powerful representations which can map high dimensional data to an array of outputs. However these mappings are often taken blindly and assumed to be accurate, which is not always the case. In two recent examples this has had disastrous consequences. In May 2016 there was the first fatality from an assisted driving system, caused by the perception system confusing the white side of a trailer for bright sky [1]. In a second recent example, an image classification system erroneously identified two African Americans as gorillas [2], raising concerns of racial discrimination. 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.

Quantifying uncertainty in computer vision applications can be largely divided into regression settings such as depth regression, and classification settings such as semantic segmentation. Existing approaches to model uncertainty in such settings in computer vision include particle filtering and conditional random fields [3, 4]. However many modern applications mandate the use of *deep learning* to achieve state-of-the-art performance [5], with most deep learning models not able to represent uncertainty. Deep learning does not allow for uncertainty representation in regression settings for example, and deep learning classification models often give normalised score vectors, which do not necessarily capture model uncertainty. For both settings uncertainty can be captured with *Bayesian deep learning* approaches – which offer a practical framework for understanding uncertainty with deep learning models [6].

In Bayesian modeling, there are two main types of uncertainty one can model [7]. *Aleatoric* uncertainty captures noise inherent in the observations. This could be for example sensor noise or motion noise, resulting in uncertainty which cannot be reduced even if more data were to be collected. On the other hand, *epistemic* uncertainty accounts for uncertainty in the model parameters – uncertainty

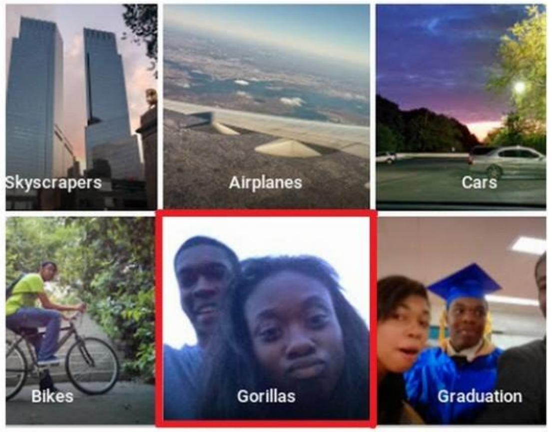
31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision

❖ Epistemic uncertainty (model uncertainty)

- ➢ 모델이 데이터에 대해 얼마나 적합하게 구축되었는지에 대해 모르는 정도
- ➢ 데이터의 어떤 특징을 학습하는지에 대해 모르는 정도
- ➢ 더 많은 데이터가 학습된다면 줄일 수 있음, reducible uncertainty



Google photo 오분류

❖ Aleatoric uncertainty (data uncertainty)

- ➢ 데이터에 내재된 노이즈로 인해 이해하지 못하는 정도
  (e.g. measurement noise, randomness inherent in the coin flipping)
- ➢ 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- ➢ 측정 정밀도를 높이면 줄일 수 있음

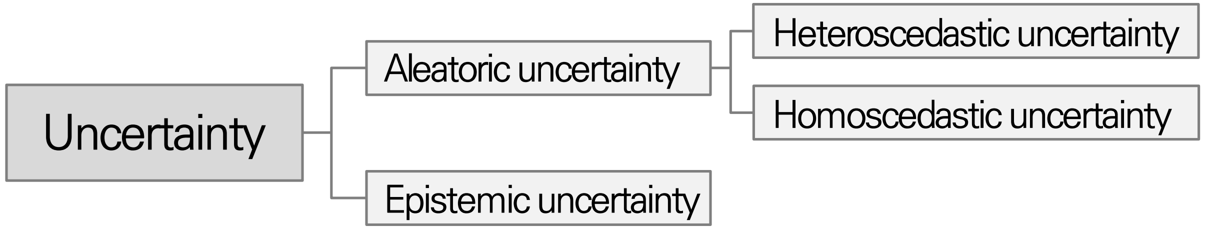
구조차량이 촬영한 지난해 9월 테슬라 자동주행(오토파일럿 모드) 차량의 중앙분리대 충돌사고 현장. 지난달 사망 사고처럼 오전 역광이 내리쬐는 상황에서 발생했다. 2016년 발생한 트레일러 충돌사고도 역광이 원인이었다.[미 ABC 방송 캡처]

Tesla 자율주행 사고

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).

Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision



❖ Aleatoric uncertainty (data uncertainty)

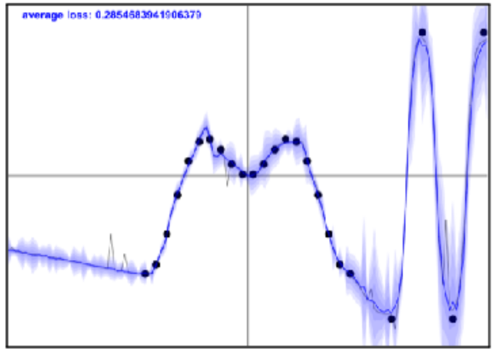
- 데이터에 내재된 노이즈로 인해 이해하지 못하는 정도
- 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- 측정 정밀도를 높이면 줄일 수 있음

❖ Homoscedastic uncertainty

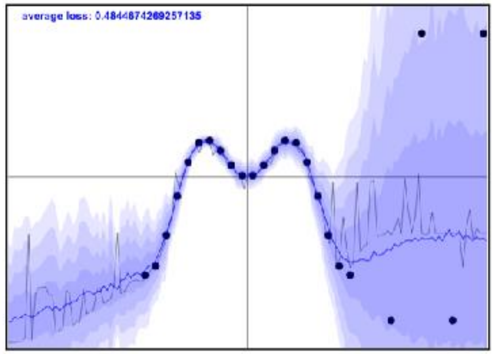
- 서로 다른 입력 값에 대해서도 동일한 constant값을 지님

❖ Heteroscedastic uncertainty

- 서로 다른 입력 값에 대해서 다른 값을 지님, input-dependent uncertainty
- 잘 모델링 된다면, 노이즈가 큰 데이터에 대해 학습과정에서 보정해줄 수 있음



Homoscedastic uncertainty

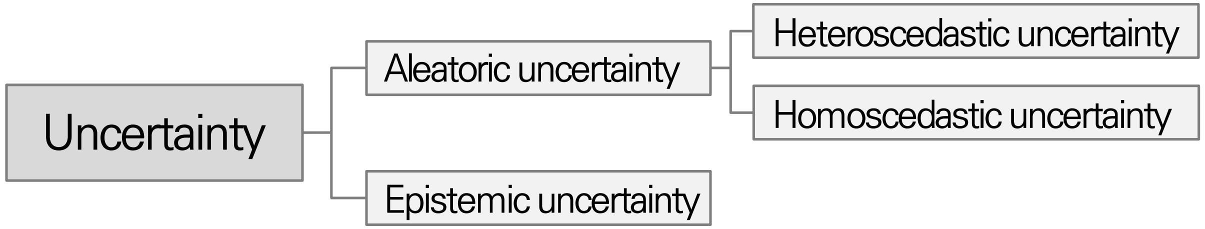


Heteroscedastic uncertainty

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).

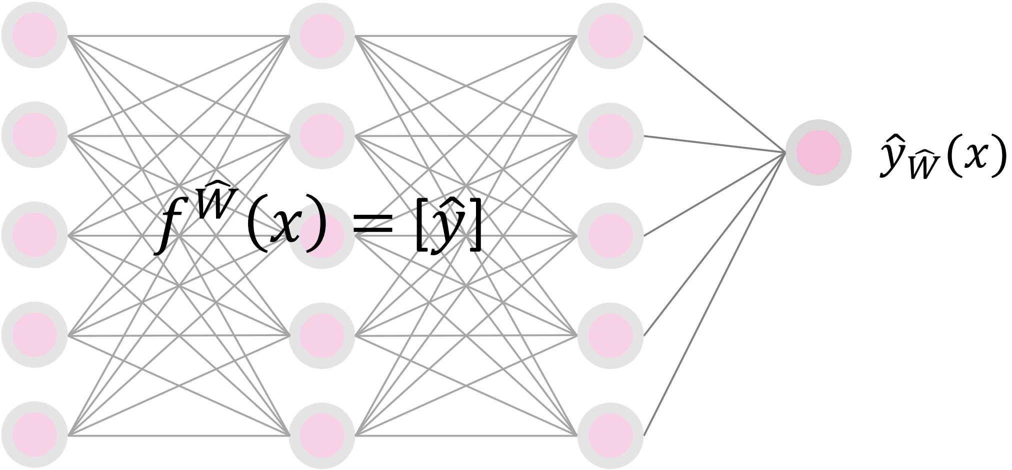
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision



❖ Standard Deep Neural Networks

- Regression task



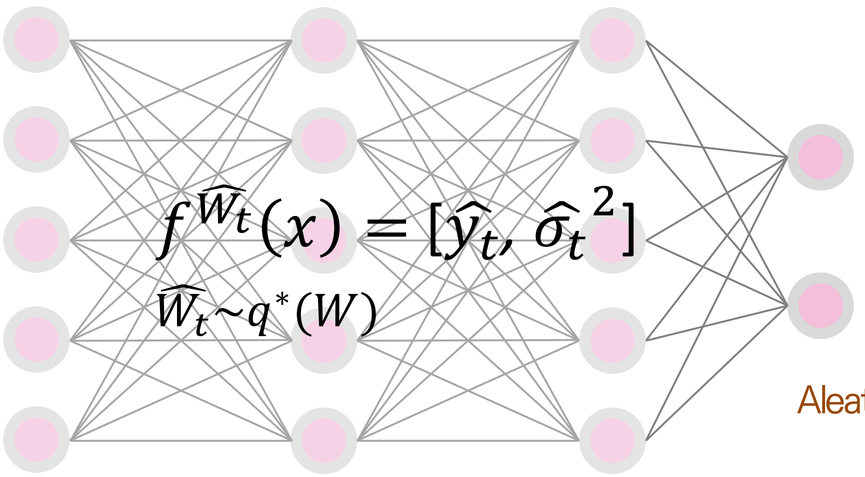
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision

- Uncertainty
  - Aleatoric uncertainty
    - Heteroscedastic uncertainty
    - Homoscedastic uncertainty
  - Epistemic uncertainty

❖ Density network architecture

➢ Regression task



MC Dropout
with L2 regularization

After T stochastic forward passes

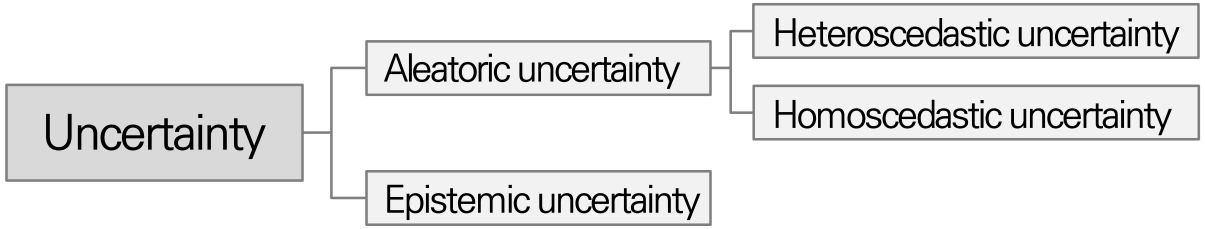
$$E(y^*) \approx \frac{1}{T}\sum_{t=1}^{T}\hat{y}_t$$ 최종 예측 값

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

Total uncertainty | Epistemic uncertainty | Aleatoric uncertainty

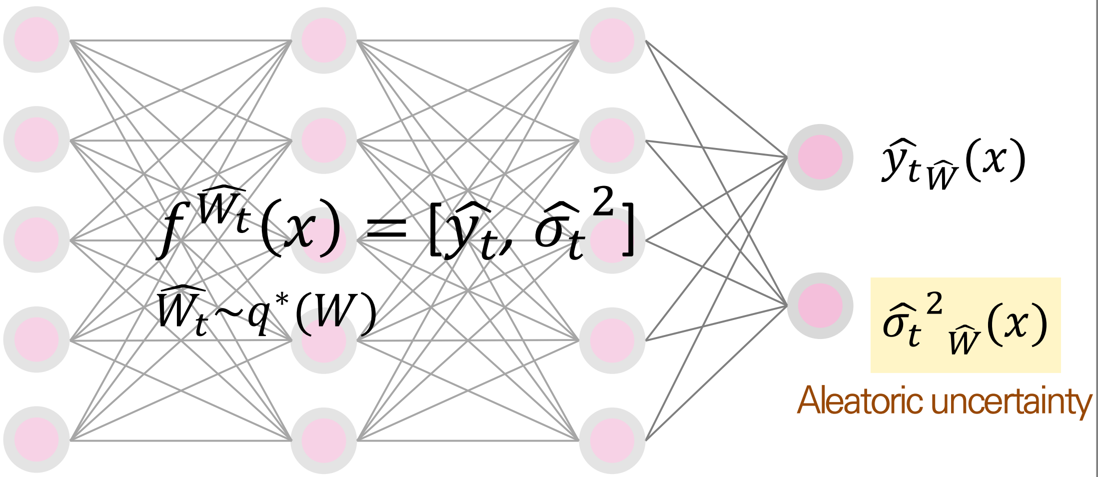
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision



❖ Density network architecture

➢ Loss function에 heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축



MC Dropout
with L2 regularization

Heteroscedastic uncertainty as learned loss attenuation

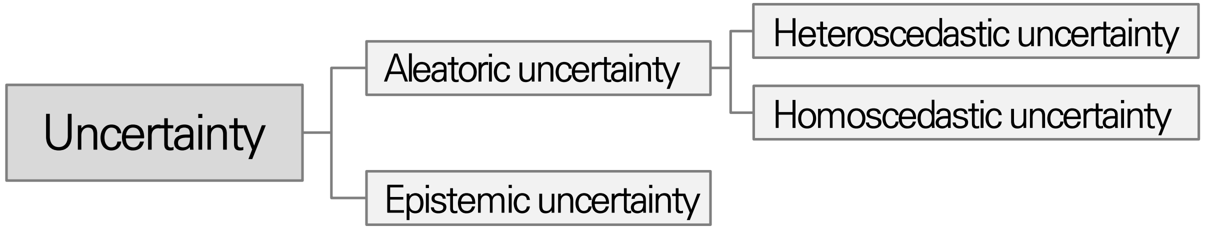
$$L_{BNN}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\sigma(x_i)^2} \|y_i - f(x_i)\|^2 + \frac{1}{2} log\sigma(x_i)^2$$

Residual's weight — Uncertainty regularization

- Residual's weight : Aleatoric heteroscedastic uncertainty가 큰 예측 값에 대해서는 residual을 적게 반영
- Uncertainty regularization : Aleatoric uncertainty가 모든 데이터에 대해 무한히 커지는 상황을 제약

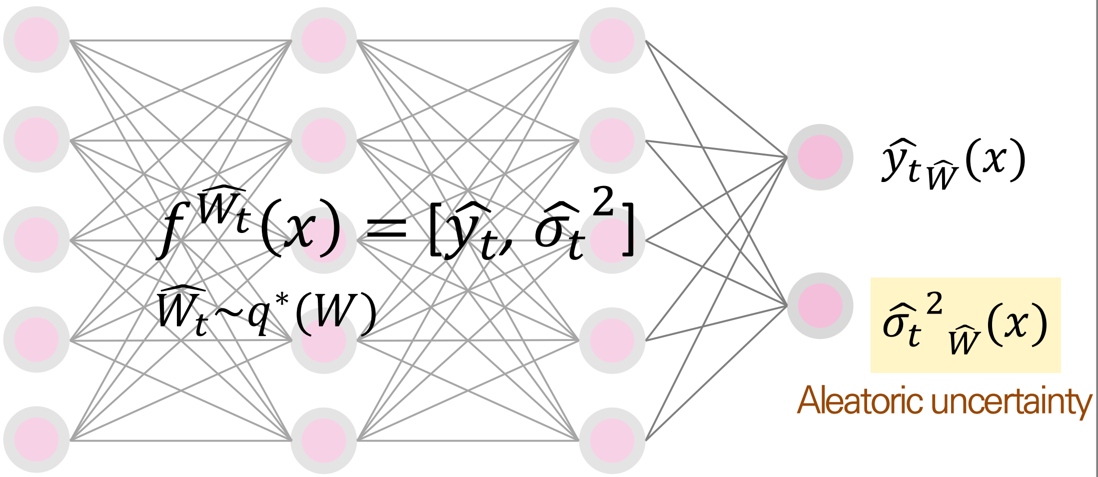
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision



❖ Density network architecture

➢ Loss function에 heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축



MC Dropout
with L2 regularization

Heteroscedastic uncertainty as learned loss attenuation

$$L_{BNN}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\sigma(x_i)^2}\|y_i - f(x_i)\|^2 + \frac{1}{2}\log\sigma(x_i)^2$$

Residual's weight — Uncertainty regularization

- 노이즈가 큰 데이터(높은 heteroscedastic uncertainty가 예측된 값)에 대해서는 loss에 적게 반영

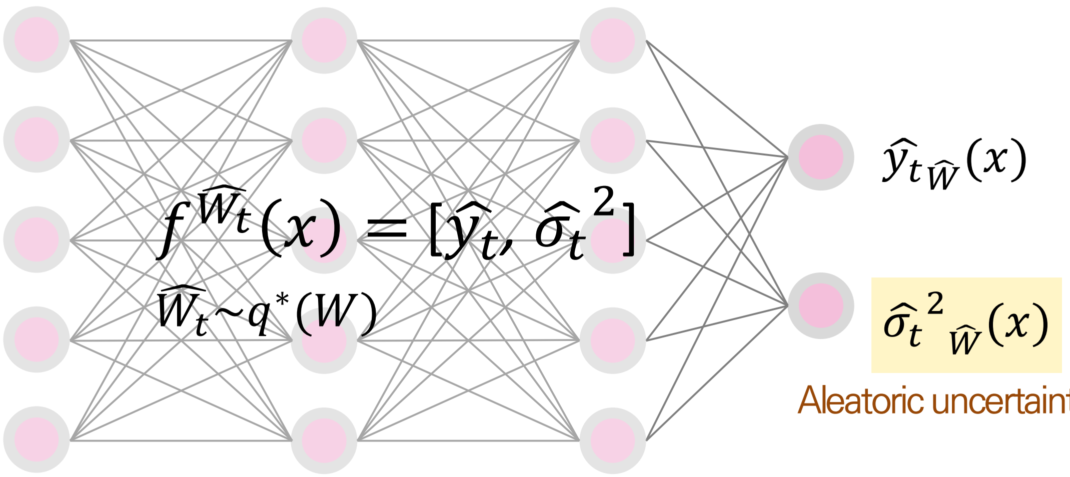
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision

- Uncertainty
  - Aleatoric uncertainty
    - Heteroscedastic uncertainty
    - Homoscedastic uncertainty
  - Epistemic uncertainty

❖ Density network architecture

➢ Loss function에 heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축



Aleatoric uncertainty

MC Dropout
with L2 regularization

Heteroscedastic uncertainty as learned loss attenuation

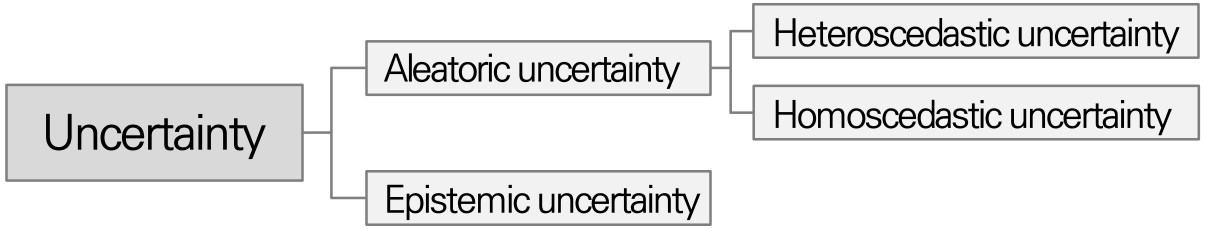
$$L_{BNN}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\sigma(x_i)^2} \|y_i - f(x_i)\|^2 + \frac{1}{2} log\sigma(x_i)^2$$

Residual's weight   Uncertainty regularization

- 노이즈가 적은 데이터(낮은 heteroscedastic uncertainty가 예측된 값)에 대해서는 loss에 크게 반영

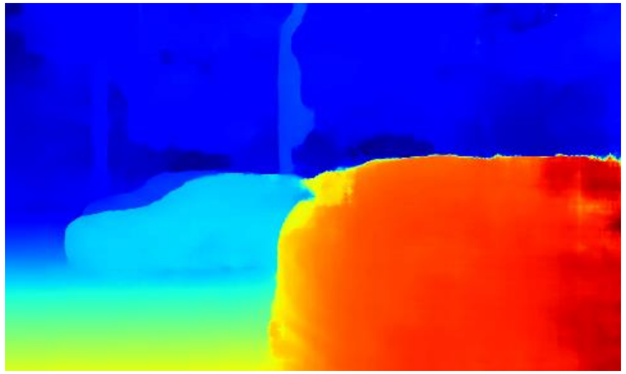
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results

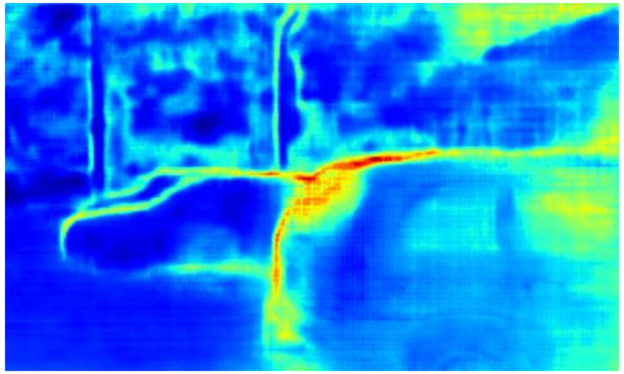


- ❖ Computer vision tasks
  - ➢ Depth regression (regression task)
  - ➢ Semantic segmentation (classification task)

Original image



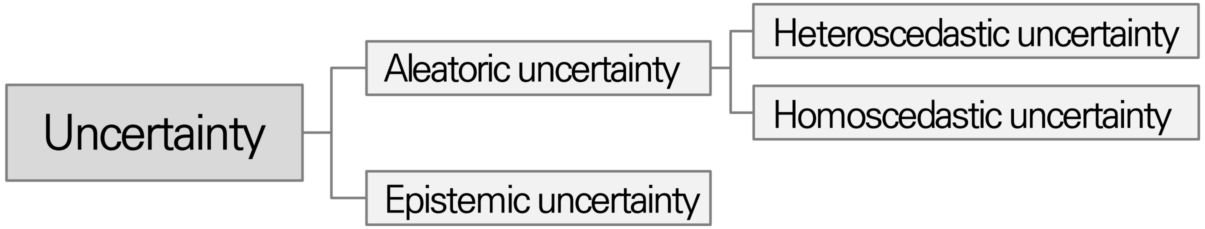
Depth regression



Semantic segmentation

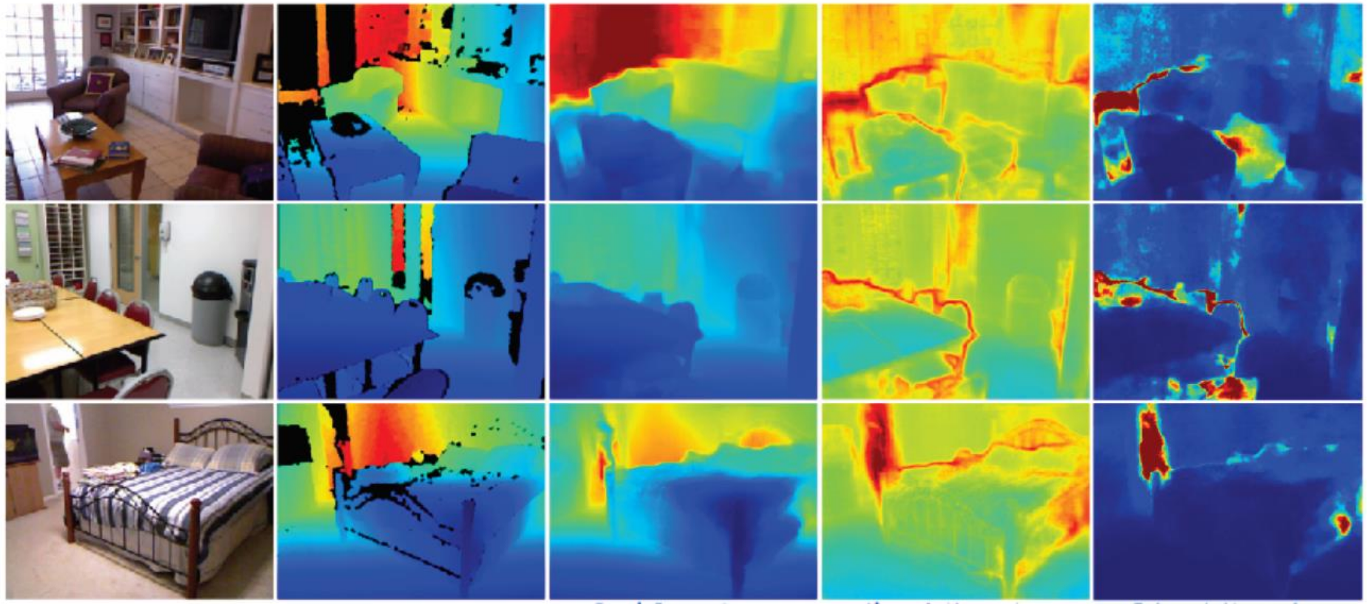
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Depth regression

- 테두리에 대한 예측에 high aleatoric uncertainty
- 예측이 틀린 부분에 high epistemic uncertainty



Input Image | Ground Truth | Depth Regression | Aleatoric Uncertainty | Epistemic Uncertainty

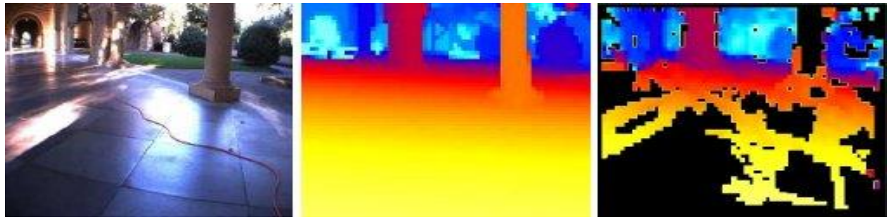
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results

- Uncertainty
  - Aleatoric uncertainty
    - Heteroscedastic uncertainty
    - Homoscedastic uncertainty
  - Epistemic uncertainty

❖ Depth regression performance with *Make3D* dataset

- ➢ *Make*3D dataset 은 **실외, 실내**에 대한 이미지 데이터셋
- ➢ 534건(345×460)차원 이미지 데이터
- ➢ 데이터는 (R, G, B, D) 로 구성



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].

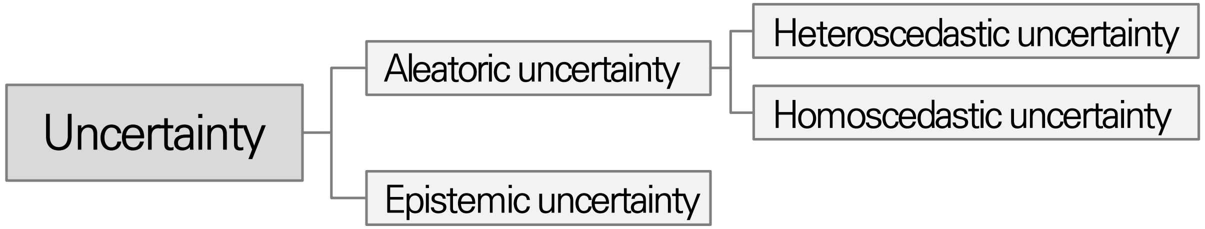
NYU v2 Depth	rel	rms	$\log_{10}$	δ_1	δ_2	δ_3
Karsch et al. [33]	0.374	1.12	0.134	-	-	-
Ladicky et al. [36]	-	-	-	54.2%	82.9%	91.4%
Liu et al. [34]	0.335	1.06	0.127	-	-	-
Li et al. [35]	0.232	0.821	0.094	62.1%	88.6%	96.8%
Eigen et al. [27]	0.215	0.907	-	61.1%	88.7%	97.1%
Eigen and Fergus [32]	0.158	0.641	-	76.9%	95.0%	98.8%
Laina et al. [26]	0.127	0.573	0.055	81.1%	95.3%	98.8%
This work:						
DenseNet Baseline	0.117	0.517	0.051	80.2%	95.1%	98.8%
+ Aleatoric Uncertainty	0.112	0.508	0.046	81.6%	95.8%	98.8%
+ Epistemic Uncertainty	0.114	0.512	0.049	81.1%	95.4%	98.8%
+ Aleatoric & Epistemic	**0.110**	**0.506**	**0.045**	**81.7%**	**95.9%**	**98.9%**

(b) NYUv2 depth dataset [23].

http://make3d.cs.cornell.edu/data.html

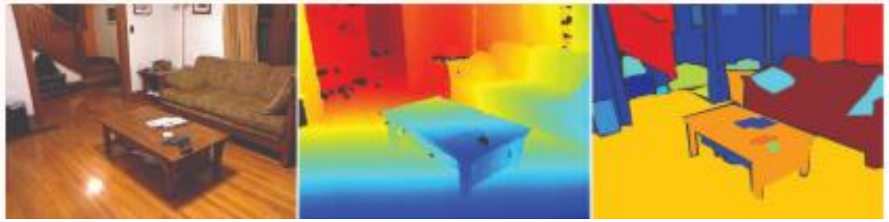
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Depth regression performance with $NYU\ v_2$ dataset

- ➢ $NYU\ v_2$ dataset 은 실내에 대한 이미지 데이터셋
- ➢ 1449건의 (640×480)차원 이미지 데이터
- ➢ 데이터는 (R, G, B, D) + structure classes로 구성
- ➢ D = [0, 250] , #structure classis = 1000개 이상



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].

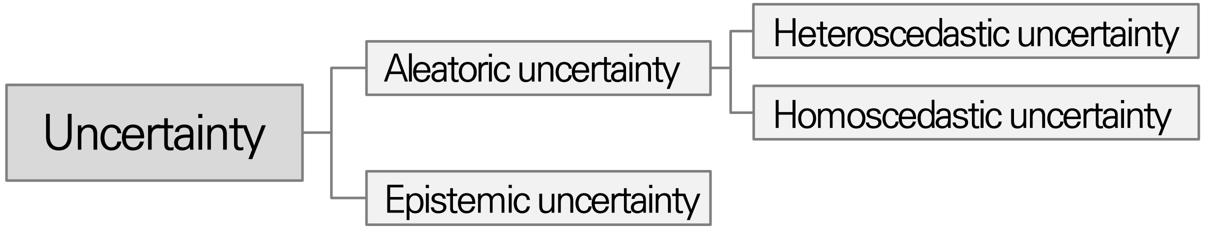
NYU v2 Depth	rel	rms	$\log_{10}$	δ_1	δ_2	δ_3
Karsch et al. [33]	0.374	1.12	0.134	-	-	-
Ladicky et al. [36]	-	-	-	54.2%	82.9%	91.4%
Liu et al. [34]	0.335	1.06	0.127	-	-	-
Li et al. [35]	0.232	0.821	0.094	62.1%	88.6%	96.8%
Eigen et al. [27]	0.215	0.907	-	61.1%	88.7%	97.1%
Eigen and Fergus [32]	0.158	0.641	-	76.9%	95.0%	98.8%
Laina et al. [26]	0.127	0.573	0.055	81.1%	95.3%	98.8%
This work:						
DenseNet Baseline	0.117	0.517	0.051	80.2%	95.1%	98.8%
+ Aleatoric Uncertainty	0.112	0.508	0.046	81.6%	95.8%	98.8%
+ Epistemic Uncertainty	0.114	0.512	0.049	81.1%	95.4%	98.8%
+ Aleatoric & Epistemic	**0.110**	**0.506**	**0.045**	**81.7%**	**95.9%**	**98.9%**

(b) NYUv2 depth dataset [23].

https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Depth regression performance metric

- REL (average relative error) = $\frac{1}{N}\sum\frac{|D-\hat{D}|}{D}$
- RMS (root mean squared error) = $\sqrt{\frac{1}{N}\sum\left(D-\hat{D}\right)^2}$
- log_{10} (average log_{10} error) = $\frac{1}{N}\sum\left|log_{10}D - log_{10}\hat{D}\right|$

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

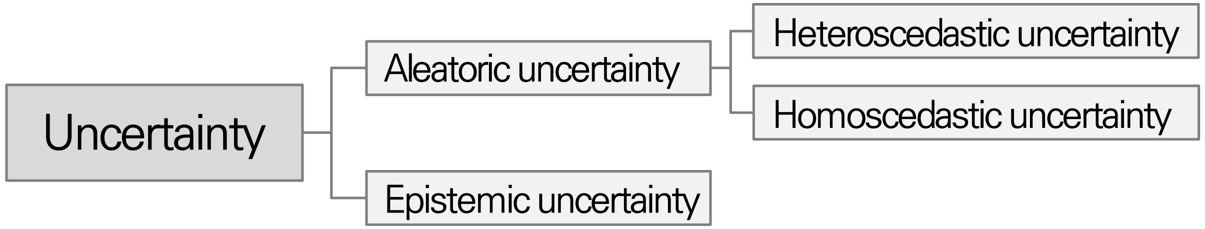
(a) Make3D depth dataset [25].

NYU v2 Depth	rel	rms	$\log_{10}$	δ_1	δ_2	δ_3
Karsch et al. [33]	0.374	1.12	0.134	-	-	-
Ladicky et al. [36]	-	-	-	54.2%	82.9%	91.4%
Liu et al. [34]	0.335	1.06	0.127	-	-	-
Li et al. [35]	0.232	0.821	0.094	62.1%	88.6%	96.8%
Eigen et al. [27]	0.215	0.907	-	61.1%	88.7%	97.1%
Eigen and Fergus [32]	0.158	0.641	-	76.9%	95.0%	98.8%
Laina et al. [26]	0.127	0.573	0.055	81.1%	95.3%	98.8%
This work:						
DenseNet Baseline	0.117	0.517	0.051	80.2%	95.1%	98.8%
+ Aleatoric Uncertainty	0.112	0.508	0.046	81.6%	95.8%	98.8%
+ Epistemic Uncertainty	0.114	0.512	0.049	81.1%	95.4%	98.8%
+ Aleatoric & Epistemic	**0.110**	**0.506**	**0.045**	**81.7%**	**95.9%**	**98.9%**

(b) NYUv2 depth dataset [23].

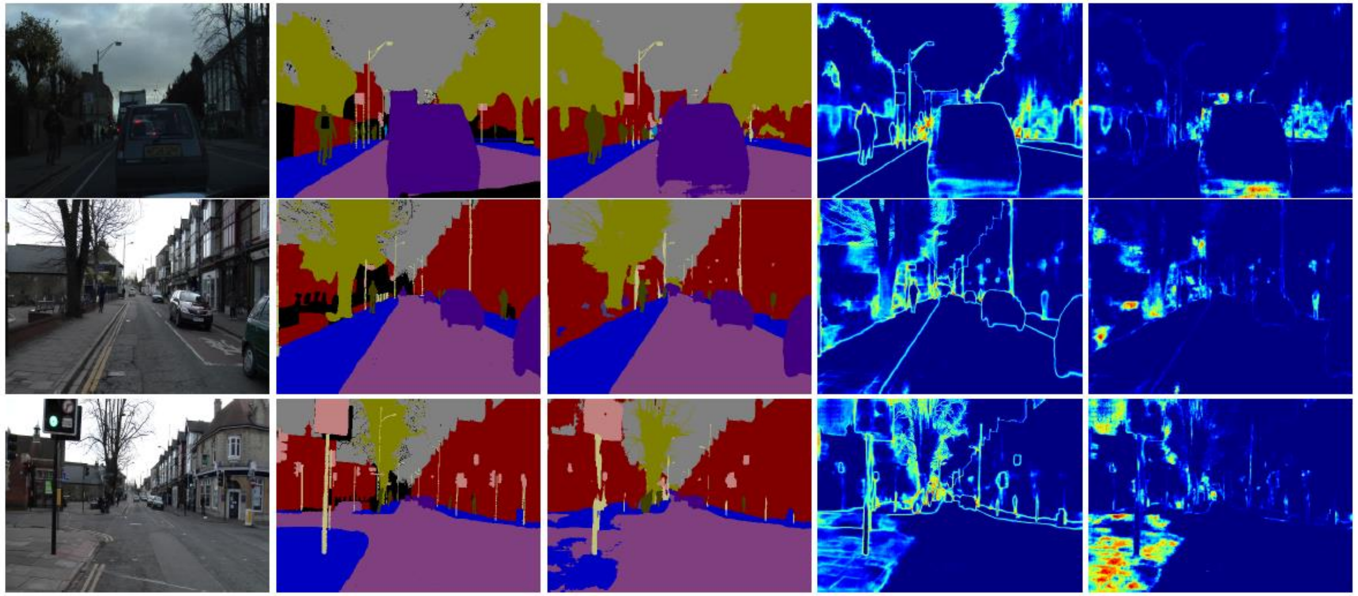
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Semantic segmentation

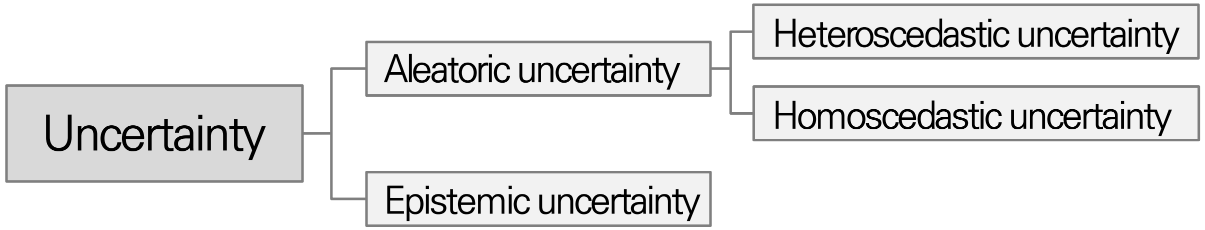
- ➢ 테두리에 대한 예측에 high aleatoric uncertainty
- ➢ 예측이 틀린 부분에 high epistemic uncertainty



Input Image | Ground Truth | Semantic Segmentation | Aleatoric Uncertainty | Epistemic Uncertainty

Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Semantic segmentation performance with *CamVid* dataset

- ➢ *CamVid* dataset 은 도로에 대한 이미지 데이터셋
- ➢ 600건의 (360 ×480)차원 이미지 데이터
- ➢ 데이터는 (R, G, B) + structure classes로 구성
- ➢ #structure classis = 11 (32)

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.

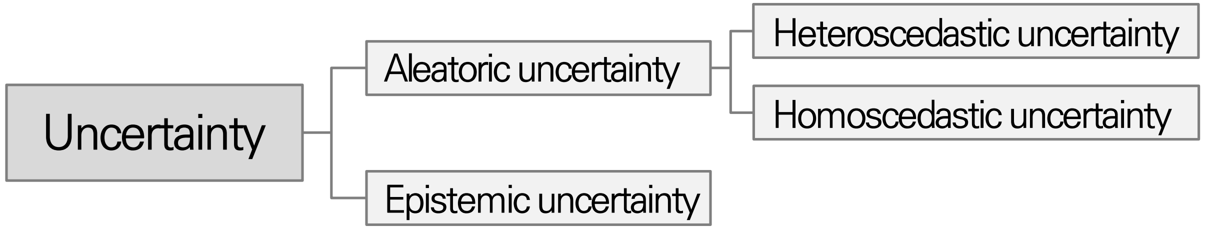
NYUv2 40-class	Accuracy	IoU
SegNet [28]	66.1	23.6
FCN-8 [29]	61.8	31.6
Bayesian SegNet [22]	68.0	32.4
Eigen and Fergus [32]	65.6	34.1
This work:		
DeepLabLargeFOV	70.1	36.5
+ Aleatoric Uncertainty	70.4	37.1
+ Epistemic Uncertainty	70.2	36.7
+ Aleatoric & Epistemic	**70.6**	**37.3**

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

http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/

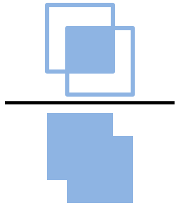
Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Semantic segmentation performance metric

➢ IoU = area of overlap/ area of union



area of overlap

area of union

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**

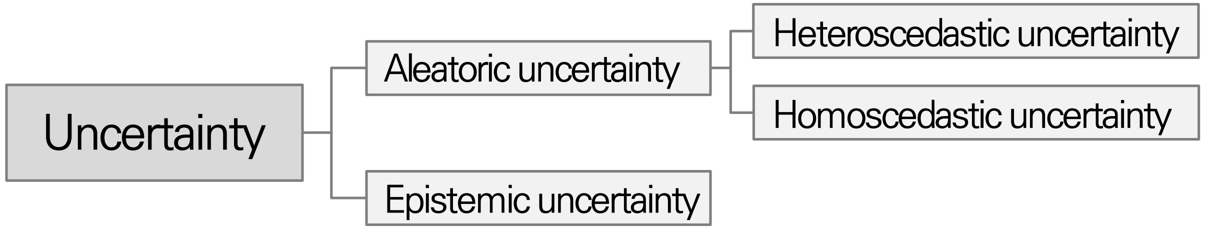
(a) CamVid dataset for road scene segmentation.

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.

Bayesian Neural Networks

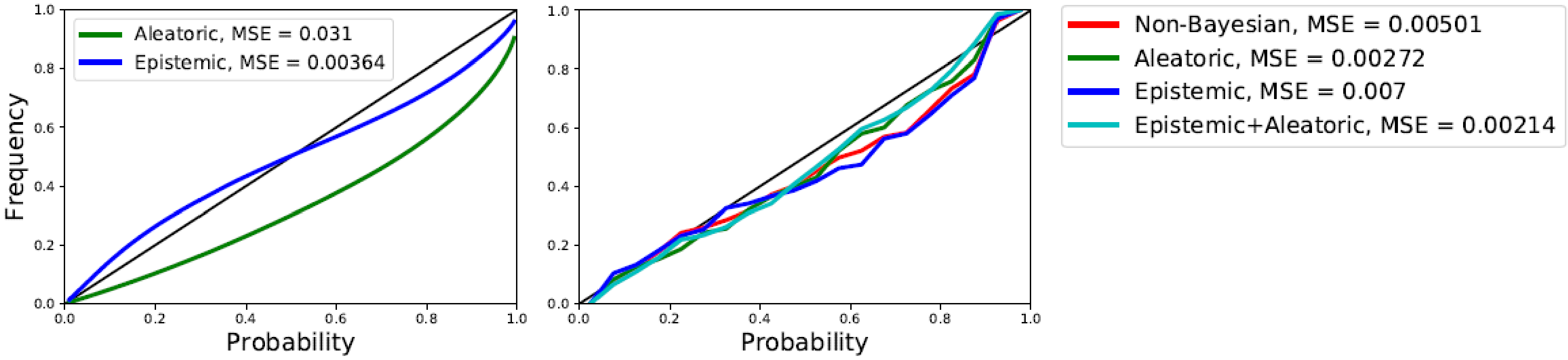
Bayesian Neural Networks for Computer Vision Results



❖ Uncertainty calibration plots (x-axis: predicted probability, y-axis: True probability)

➢ 해당 plot을 통해 추정한 uncertainty의 타당성을 확인할 수 있음

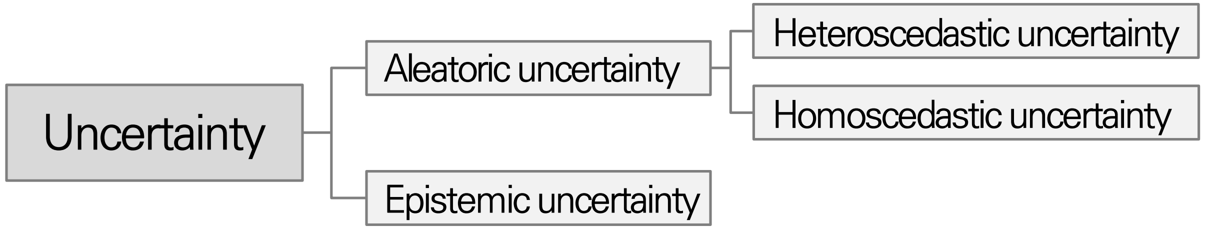
➢ Epistemic + Aleatoric 을 모두 사용하여 모델링한 경우, 최종 uncertainty가 가장 타당한 것을 확인



(a) Regression (Make3D) (b) Classification (CamVid)

Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Results



❖ Aleatoric and epistemic uncertainties for a range of different dataset combinations

- Aleatoric uncertainty는 학습 데이터가 증가해도 줄일 수 없고, epistemic uncertainty는 줄일 수 있다는 가설 증명
- 학습 데이터셋 크기를 (¼, ½, 1)로 조정하며 실험
- $Make3D$ dataset: 실내, 실외 데이터/ $NYU\ v_2$ dataset: 실내 데이터 / $CamVid$ dataset: 도로 주행 데이터

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

Bayesian Neural Networks

Bayesian Neural Networks for Computer Vision Critic

❖ Uncertainty

➢ 모델의 불확실성인 epistemic uncertainty를 모델링 뿐만 아니라, 데이터의 불확실성인 aleatoric uncertainty를 모델링

❖ Model performance

➢ Dropout과 L2 regularization term을 적용하여 overfitting을 방지

➢ 모델의 불확실성인 epistemic uncertainty를 모델링하는 과정에서 도출되는 T 개의 예측 값을 평균하여 최종 예측 값으로 사용하기 때문에, outlier에 대한 보정이 가능

➢ Heteroscedastic aleatoric uncertainty를 loss function에 반영함으로써 더욱 강건한 모델 구축 가능

❖ Disadvantages

➢ Dropout rate에 의존적인 결과 도출

➢ 모델 수렴이 어려울 수 있고, standard neural net구조보다 학습 시간 오래 걸림

➢ 정해진 architecture 구조 내에서만 BNN 구현 가능

Uncertainty

Uncertainty

Bayesian approach


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

1. Introduction


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

1 Introduction

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

Non-Bayesian approach


Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan Alexander Pritzel Charles Blundell
DeepMind
{balajiln,apritzel,cblundell}@google.com

Abstract

1 Introduction

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

Uncertainty

Bayesian approach

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

Yarin Gal, Zoubin Ghahramani — University of Cambridge

Epistemic

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

Alex Kendall, Yarin Gal — University of Cambridge

Epistemic & Aleatoric

Non-Bayesian approach

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell — DeepMind

Aleatoric

Non-Bayesian Approaches

Simple and Scalable Deep Ensembles

- ❖ ICML 2017
- ❖ 578회 인용건수

Simple and **scalable predictive uncertainty** estimation using deep ensembles
B Lakshminarayanan, A Pritzel… - Advances in neural …, 2017 - papers.nips.cc
Deep neural networks (NNs) are powerful black box predictors that have recently achieved impressive performance on a wide spectrum of tasks. Quantifying predictive uncertainty in NNs is a challenging and yet unsolved problem. Bayesian NNs, which learn a distribution …
☆ 99 578회 인용 관련 학술자료 전체 14개의 버전 »

- ❖ 기존의 BNN의 경우, 모델 구조가 한정적, 계산량多
- ❖ Ensemble을 이용하여 간단하게 uncertainty 모델링
  - ➢ Simple: 구조의 제한이 비교적 없음
  - ➢ Scalable: 병렬연산이 가능하기 때문에 계산 효율 증가

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan **Alexander Pritzel** **Charles Blundell**
DeepMind
{balajiln,apritzel,cblundell}@google.com

Abstract

Deep neural networks (NNs) are powerful black box predictors that have recently achieved impressive performance on a wide spectrum of tasks. Quantifying predictive uncertainty in NNs is a challenging and yet unsolved problem. Bayesian NNs, which learn a distribution over weights, are currently the state-of-the-art for estimating predictive uncertainty; however these require significant modifications to the training procedure and are computationally expensive compared to standard (non-Bayesian) NNs. We propose an alternative to Bayesian NNs that is simple to implement, readily parallelizable, requires very little hyperparameter tuning, and yields high quality predictive uncertainty estimates. Through a series of experiments on classification and regression benchmarks, we demonstrate that our method produces well-calibrated uncertainty estimates which are as good or better than approximate Bayesian NNs. To assess robustness to dataset shift, we evaluate the predictive uncertainty on test examples from known and unknown distributions, and show that our method is able to express higher uncertainty on out-of-distribution examples. We demonstrate the scalability of our method by evaluating predictive uncertainty estimates on ImageNet.

1 Introduction

Deep neural networks (NNs) have achieved state-of-the-art performance on a wide variety of machine learning tasks [35] and are becoming increasingly popular in domains such as computer vision [32], speech recognition [25], natural language processing [42], and bioinformatics [2, 61]. Despite impressive accuracies in supervised learning benchmarks, NNs are poor at quantifying predictive uncertainty, and tend to produce overconfident predictions. Overconfident incorrect predictions can be harmful or offensive [3], hence proper uncertainty quantification is crucial for practical applications.

Evaluating the quality of predictive uncertainties is challenging as the 'ground truth' uncertainty estimates are usually not available. In this work, we shall focus upon two evaluation measures that are motivated by practical applications of NNs. Firstly, we shall examine *calibration* [12, 13], a frequentist notion of uncertainty which measures the discrepancy between subjective forecasts and (empirical) long-run frequencies. The quality of calibration can be measured by *proper scoring rules* [17] such as log predictive probabilities and the Brier score [9]. Note that calibration is an orthogonal concern to accuracy: a network's predictions may be accurate and yet miscalibrated, and vice versa. The second notion of quality of predictive uncertainty we consider concerns generalization of the predictive uncertainty to domain shift (also referred to as *out-of-distribution* examples [23]), that is, measuring if the network *knows what it knows*. For example, if a network trained on one dataset is evaluated on a completely different dataset, then the network should output high predictive uncertainty as inputs from a different dataset would be far away from the training data. Well-calibrated predictions that are robust to model misspecification and dataset shift have a number of important practical uses (e.g., weather forecasting, medical diagnosis).

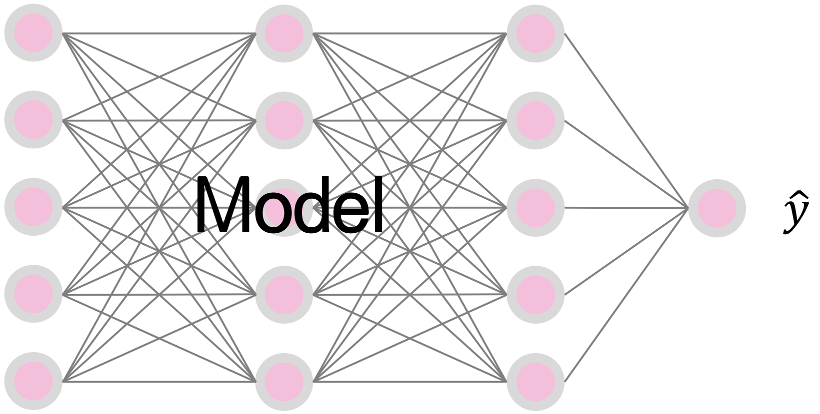
31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

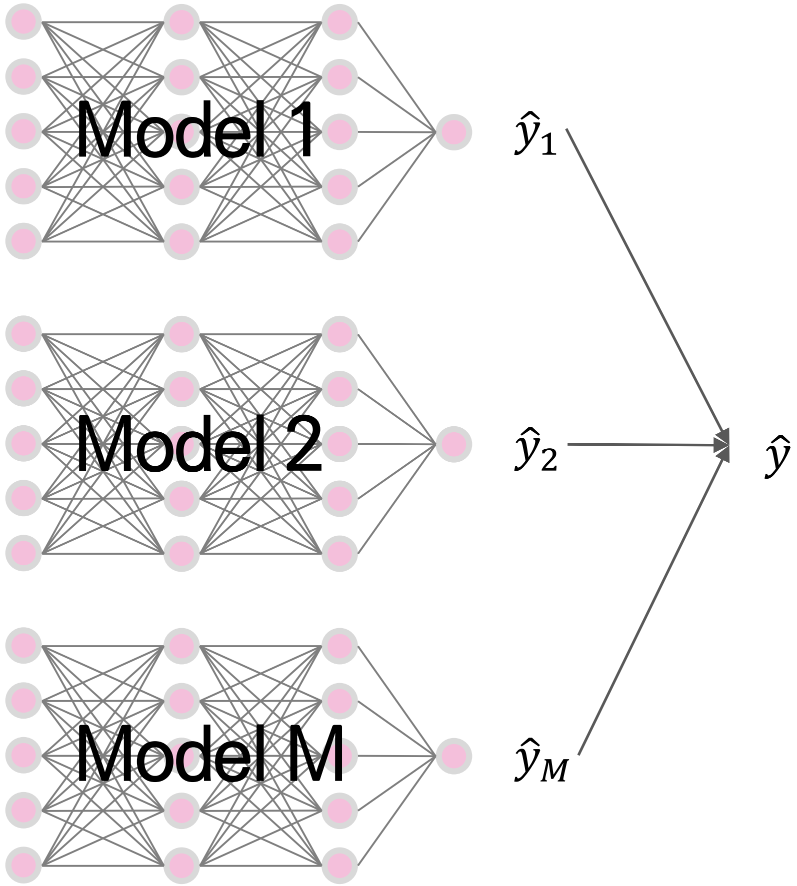
Non-Bayesian Approaches

Simple and Scalable Deep Ensembles

❖ Ensemble learning method

➢ 도출된 다수의 결과를 종합하여 최종 예측 수행

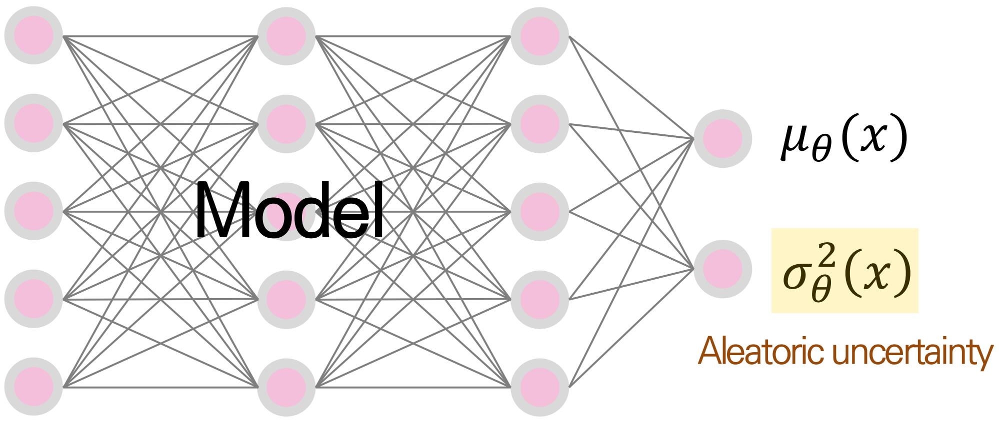




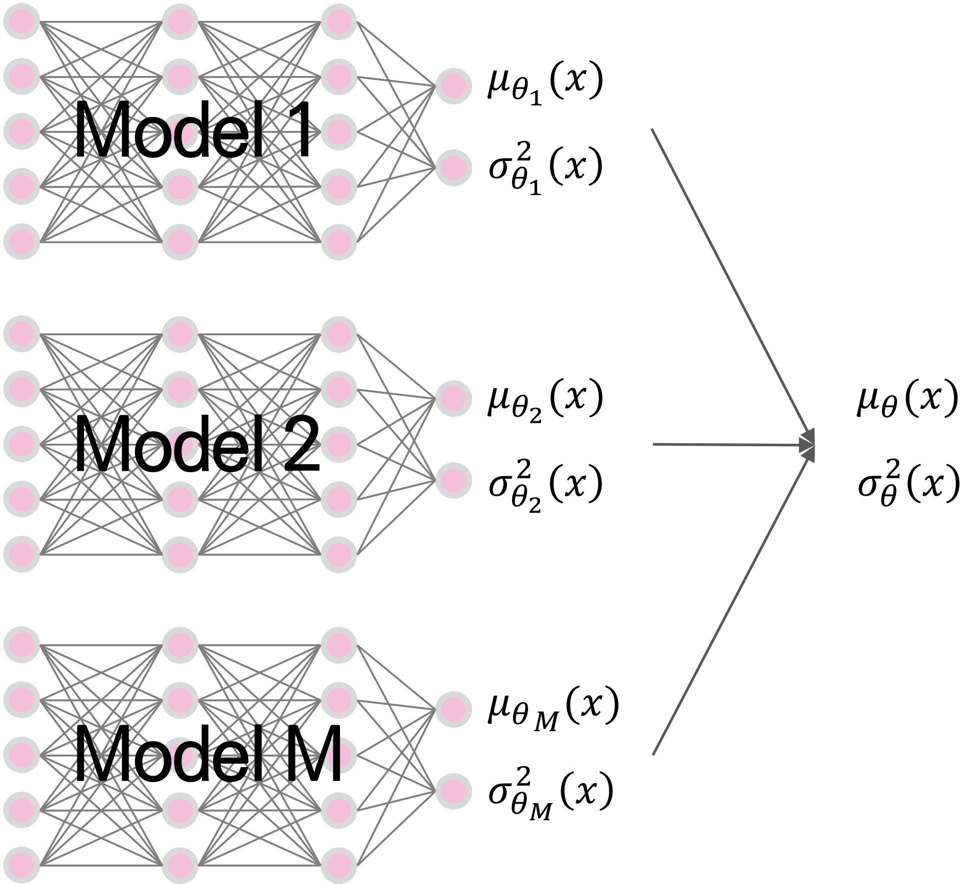
Non−Bayesian Approaches

Simple and Scalable Deep Ensembles

❖ Ensemble learning method + uncertainty (aleatoric uncertainty) = Deep ensembles



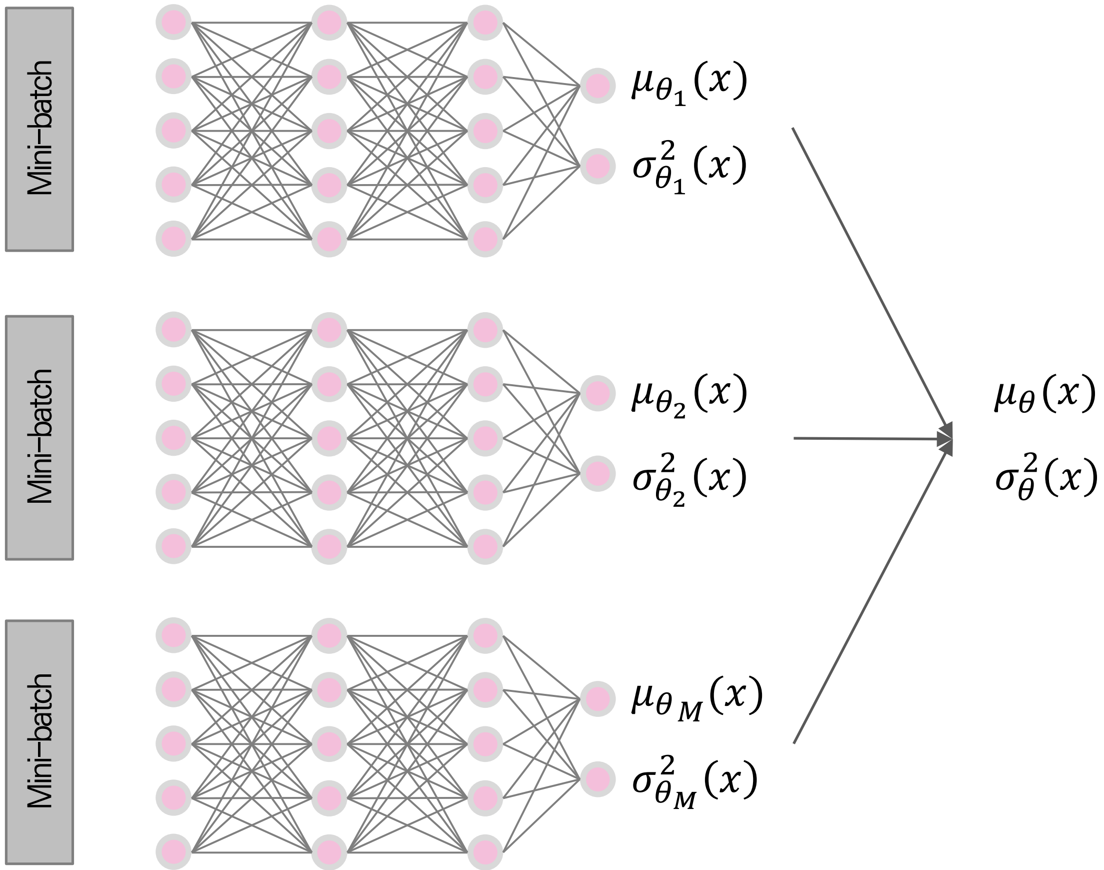
$$f_\theta(x) = [\, \mu_\theta(x), \sigma_\theta(x) \,]$$



Non-Bayesian Approaches

Simple and Scalable Deep Ensembles

❖ Deep Ensembles **architecture**



M 개의 mini-batch 구성하여 모델링 수행 후,

$$\mu_\theta(x) = \frac{1}{M}\sum_{m=1}^{M} \mu_{\theta_m}(x)$$

최종 예측 값

$$\sigma^2_\theta(x) = \frac{1}{M}\sum_{m=1}^{M} \left(\sigma^2_{\theta_m}(x) + \mu^2_{\theta_m}(x)\right) - \mu^2_\theta(x)$$

Uncertainty

Non-Bayesian Approaches

Simple and Scalable Deep Ensembles

❖ Deep Ensembles with scoring rule

➢ Loss function을 구성하는 과정에서 scoring rule 제안

➢ Scoring rule: 예측 분산이 클 때, loss에 반영할 수 있는 방법 → 사실상 일반적인 loss function이 해당 조건을 만족함

❖ Regression

➢ $\sigma_\theta^2(x)$ 을 반영하여 MSE 보정

➢ Negative Log-likelihood(NLL)

Uncertainty regularization

$$L_{Ensemble}(\theta) = \frac{(y - \mu_\theta(x))^2}{2\sigma_\theta^2(x)} + \frac{log\sigma_\theta^2(x)}{2} + constant$$

Residual's weight

❖ Classification

➢ 실제 label의 one-hot 벡터와 예측확률 사이의 MSE (mean squared error)

$$L_{Ensemble}(\theta) = \frac{1}{C}\sum_{c=1}^{C}\left(\delta_{c=y} - p_\theta(y = c|x)\right)^2$$

$\delta_{c=y}$: 실제 label의 one-hot encoding 벡터 $[1, 0, 0, 0]$

$p_\theta(y = c|x)$: 예측 확률 $[0.8, 0.05, 0.05, 0.1]$

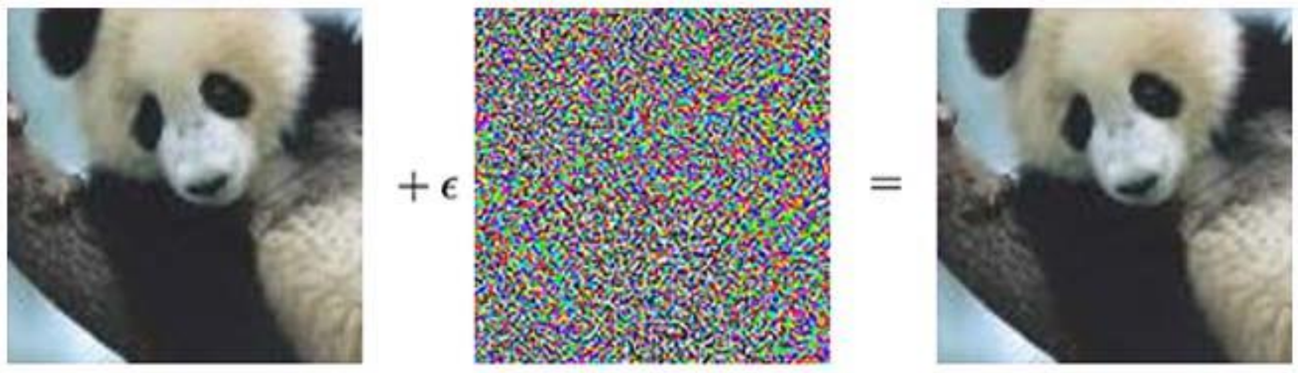
Non-Bayesian Approaches

Simple and Scalable Deep Ensembles

❖ Deep Ensembles with adversarial training

- Adversarial training은 일종의 data augmentation 방법
- 사람의 눈에는 동일해 보이지만, 모델은 헷갈려 하는 데이터를 perturbation을 더함으로써 생성함
- 이러한 데이터를 학습 시 추가적으로 사용하면 noise에 강건한 모델 구축 가능

x $\quad$ $perturbation$ $\quad$ $x_{autmented}$



판다 57.7% $\quad$ 긴팔원숭이 99.9%

$$x_{autmented} = x + \epsilon sign(\nabla_x l(\theta, x, y))$$

Non-Bayesian Approaches

Simple and Scalable Deep Ensembles Results

❖ Deep Ensembles training procedure

Algorithm 1 Pseudocode of the training procedure for our method

1: ▷ *Let each neural network parametrize a distribution over the outputs, i.e.* $p_\theta(y|\mathbf{x})$. *Use a proper scoring rule as the training criterion* $\ell(\theta, \mathbf{x}, y)$. *Recommended default values are* $M = 5$ *and* $\epsilon = 1\%$ *of the input range of the corresponding dimension (e.g 2.55 if input range is [0,255]).*
2: Initialize $\theta_1, \theta_2, \ldots, \theta_M$ randomly
3: **for** $m = 1 : M$ **do** ▷ *train networks independently in parallel*
4: Sample data point n_m randomly for each net ▷ *single* n_m *for clarity, minibatch in practice*
5: Generate adversarial example using $\mathbf{x}'_{n_m} = \mathbf{x}_{n_m} + \epsilon \,\text{sign}\big(\nabla_{\mathbf{x}_{n_m}} \ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m})\big)$
6: Minimize $\ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m}) + \ell(\theta_m, \mathbf{x}'_{n_m}, y_{n_m})$ w.r.t. θ_m ▷ *adversarial training (optional)*

1. Loss function, 네트워크 개수 M, adversarial training ratio ϵ 정의
2. 각 네트워크의 파라미터 초기화
3. M개의 네트워크에 대해 반복 수행 (독립적으로 병렬처리 가능)
   4. 전체 데이터 셋에서 각 네트워크를 학습시키기 위한 mini-batch 데이터셋 구축
   5. 해당 mini-batch에 대한 adversarial example 생성하여 데이터 증폭 (optional)
   6. Score rule인 loss를 최소화 하도록 네트워크 파라미터 학습

- Deep ensembles
- Adversarial training
- Score rule

Non-Bayesian Approaches

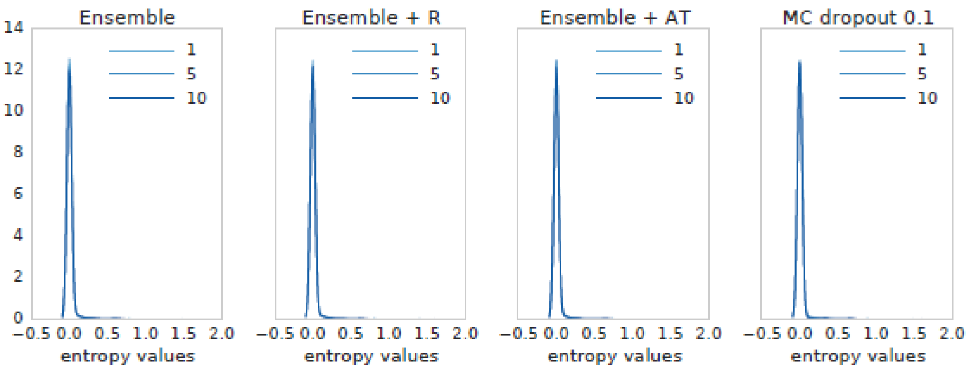
Simple and Scalable Deep Ensembles Results

MNIST

Not-MNIST

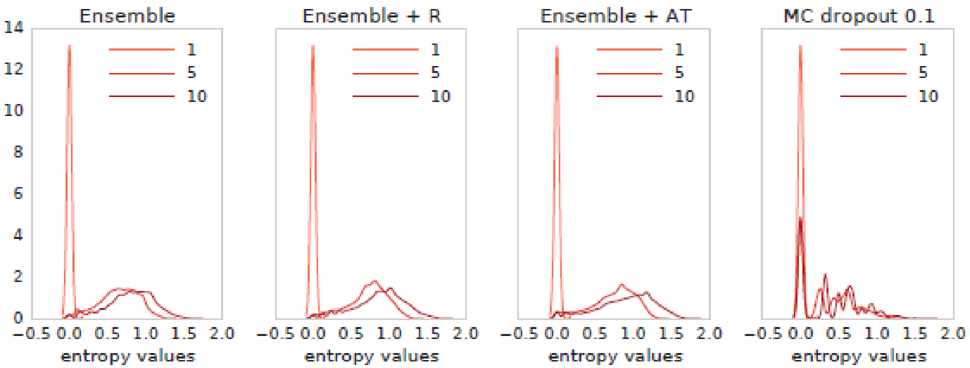
- ❖ Histogram of the predictive entropy on test examples
- ❖ R: random noise / AT: adversarial training
- ❖ MC dropout (첫번째 논문과 비교)

Train: MNIST
Test: MNIST



- ➢ Entropy가 0, uncertainty가 없음

Train: MNIST
Test: Not-MNIST



- ➢ M이 커질수록 entropy가 커짐, uncertainty을 잘 모델링
- ➢ Mc dropout 방법론보다 성능이 더 우수

Non-Bayesian Approaches

Simple and Scalable Deep Ensembles Results Critic

❖ Uncertainty

- ➢ BNN에서 파라미터의 분포를 가정하는 것 자체가 한계점이라고 주장
- ➢ NN구조에 대한 제약 없이 ensemble 구조로 간단하게 uncertainty 모델링

❖ Model performance

- ➢ BNN의 epistemic uncertainty만 추정한 경우 보다 좋은 성능을 보임

❖ Disadvantages

- ➢ Uncertainty에 대한 정의가 불명확하고, 실험적으로 증명하고자 함
- ➢ Scalability를 기여점으로 주장하고 있으나, 실제 구현과정에서는 dropout기법보다 더 많은 시간이 소요

Conclusions

Uncertainty

References

- Bayesian Neural Nets
  - http://dmqa.korea.ac.kr/activity/seminar/252
  - https://www.slideshare.net/rsilveira79/uncertainty-in-deep-learning
  - https://alexgkendall.com/computer_vision/bayesian_deep_learning_for_safe_ai/
  - https://getpocket.com/redirect?url=http%3A%2F%2Fmlg.eng.cam.ac.uk%2Fyarin%2Fblog_3d801aa532c1ce.html
  - https://towardsdatascience.com/building-a-bayesian-deep-learning-classifier-ece1845bc09
- Variational Inference
  - http://dmqa.korea.ac.kr/activity/seminar/253
- Non-Bayesian approach
  - https://www.slideshare.net/DonghyeonKim7/2018-133403439

Thank you

Appendix

Appendix

Posterior Approximation using variational inference

❖ Bayes' Rule

What we know: Likelihood(Model), Prior(Assumption)

What we do not know: Posterior, Evidence

What we want know: Posterior

$$\underbrace{p(W|X,Y)}_{\text{Posterior}} = \frac{\overbrace{p(Y|X,W)}^{\text{Likelihood}}\overbrace{p(w)}^{\text{Prior}}}{\underbrace{p(Y|X)}_{\text{Evidence}}}$$

This integration is not computable in general

$$\underbrace{p(Y|X)}_{\text{Evidence}} = \int p(Y|X,W)p(W)dw$$

Our goal

$$p(y^*|x^*,X,Y) = \int p(y^*|f^*)p(f^*|x^*,W)\underbrace{p(W|X,Y)}_{\text{Posterior}}df * dw$$

$$= \int \underbrace{p(y^*|x^*,W)}_{\text{NN output}}\underbrace{p(W|X,Y)}_{\text{Posterior}}dw$$

Bayesian networks are easy to formulate, but difficult to perform inference in

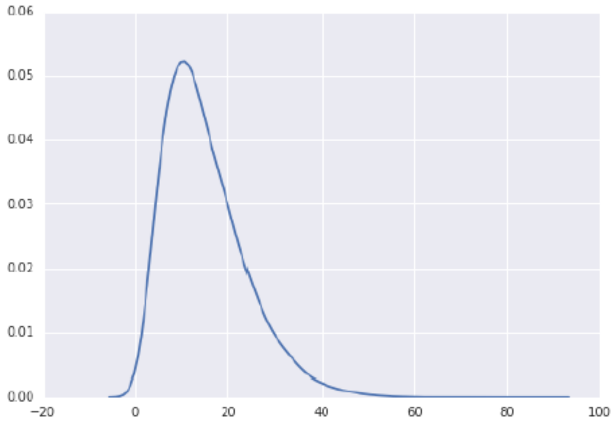
Appendix

Posterior Approximation using variational inference

❖ Bayes' Rule

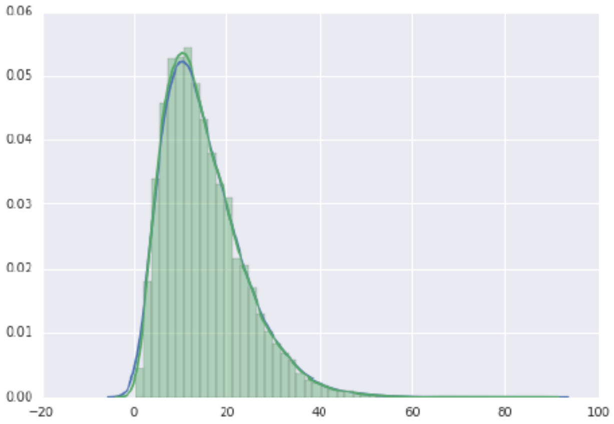
➢ Methods for Intractable Posterior

True Posterior



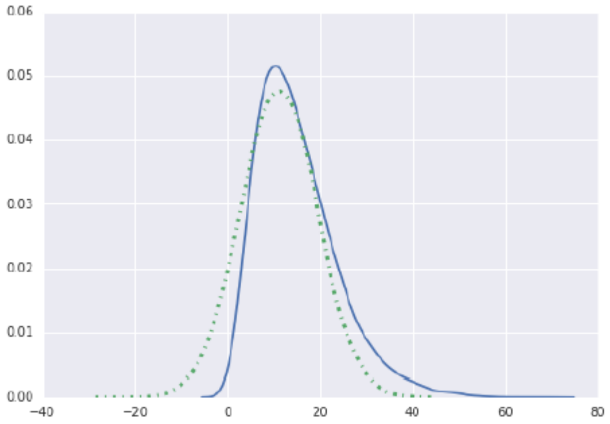
$$p(W|X,Y) = \frac{p(Y|X,W)p(w)}{p(Y|X)}$$

Sampling-based



$$W_1, W_1, W_1, \cdots, W_1 \sim p(W|X,Y)$$

Approximate Inference



$$q_\theta(W) \approx p(W|X,Y)$$

http://dmqa.korea.ac.kr/activity/seminar/253

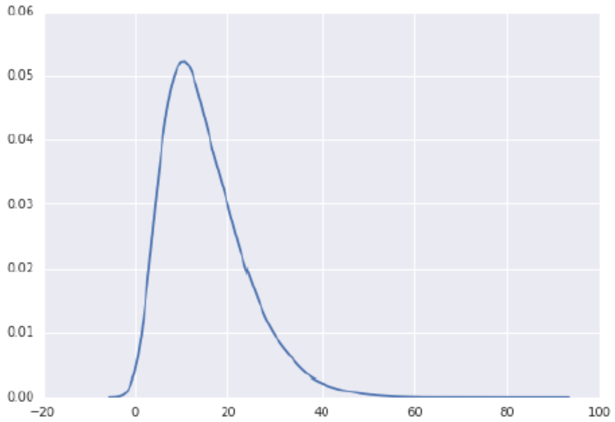
Appendix

Posterior Approximation using variational inference

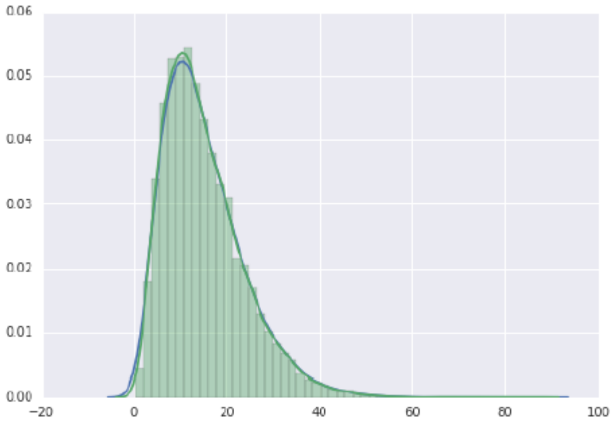
❖ Bayes' Rule

➢ Methods for Intractable Posterior

True Posterior

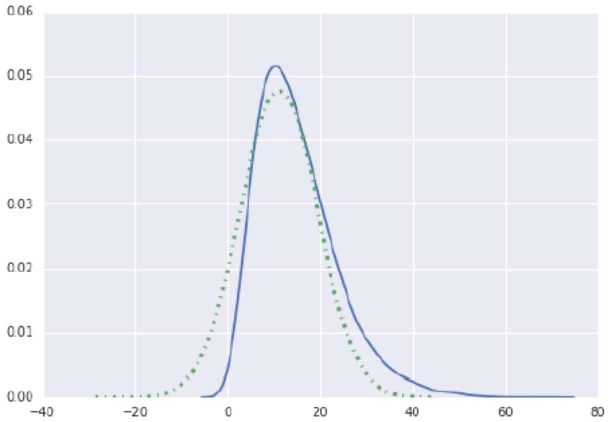


Sampling−based



Naïve Monte Carlo	Metropolis−Hastings
Rejection Sampling	Gibbs Sampling
Importance Sampling	Reversible−Jump MCMC

Approximate Inference



Laplace Approximation

Expectation Propagation

Variational Inference

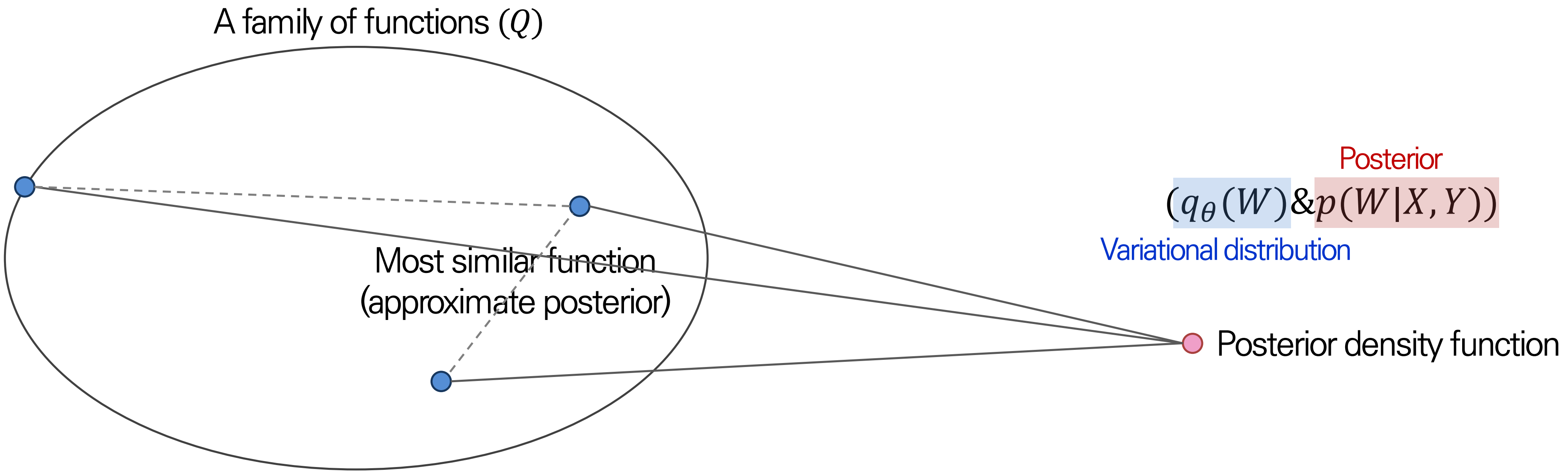
Appendix

Posterior Approximation using variational inference

❖ Variational Inference

- ➢ Approximation by using an "easier" distribution $q_\theta(W)$

Variational distribution, where θ are the variational parameters



Appendix

Posterior Approximation using variational inference

❖ Variational Inference

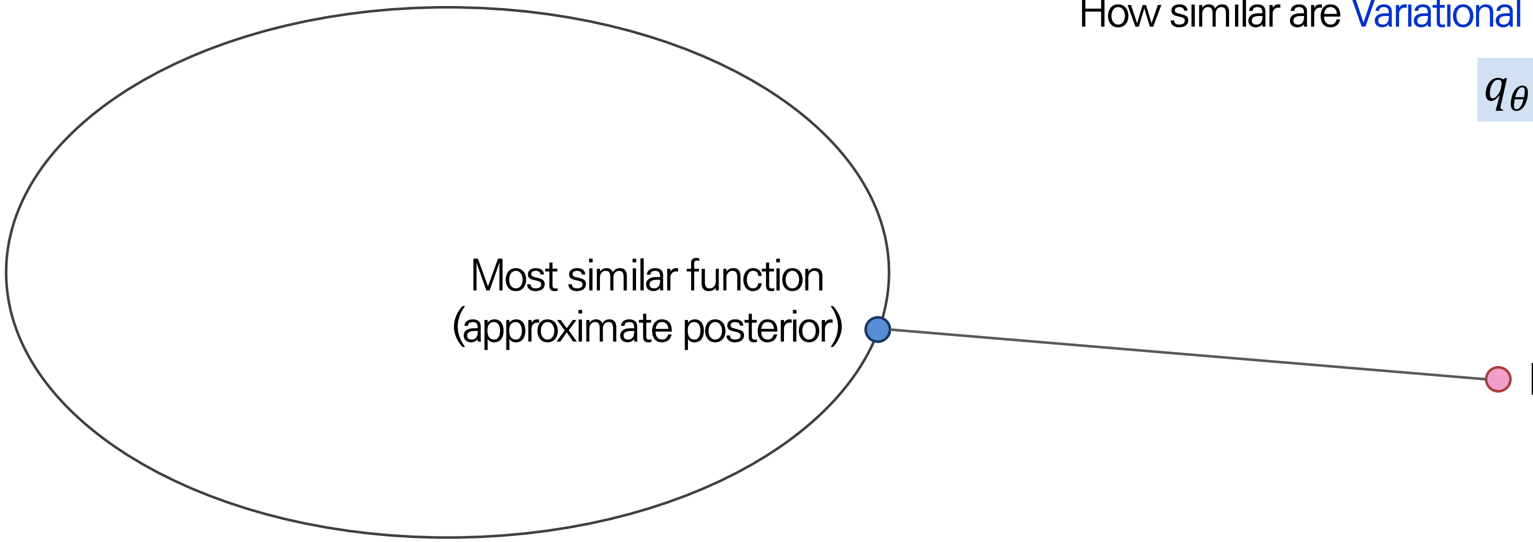
- ➢ Approximation by using an “easier” distribution $q_\theta(W)$

Variational distribution, where θ are the variational parameters

A family of functions (Q)

How similar are Variational distribution and Posterior ?

$q_\theta(W)$ $p(W|X,Y)$



Appendix

Posterior Approximation using variational inference

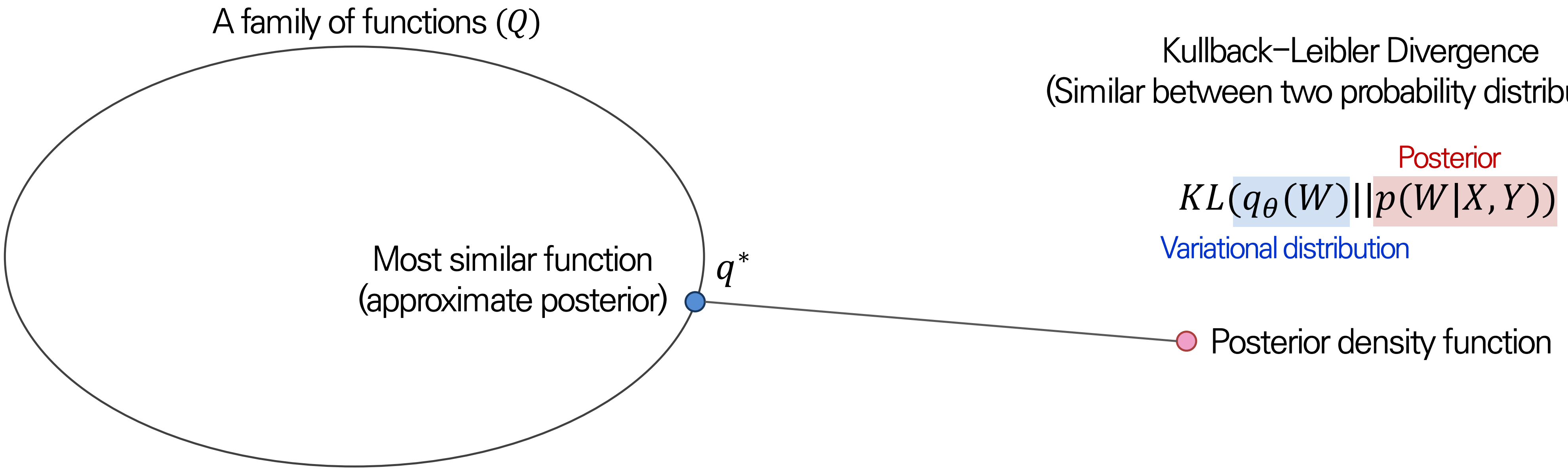
Why Kullback−Leibler Divergence?

- Because it allows us to derive a cost that is tractable to optimization
- Not without paying a price though

❖ Variational Inference

➢ Approximation by using an “easier” distribution $q_\theta(W)$

Variational distribution, where θ are the variational parameters



Kullback−Leibler Divergence
(Similar between two probability distributions)

Posterior

$$KL(q_\theta(W)||p(W|X,Y))$$

Variational distribution

Appendix

Posterior Approximation using variational inference

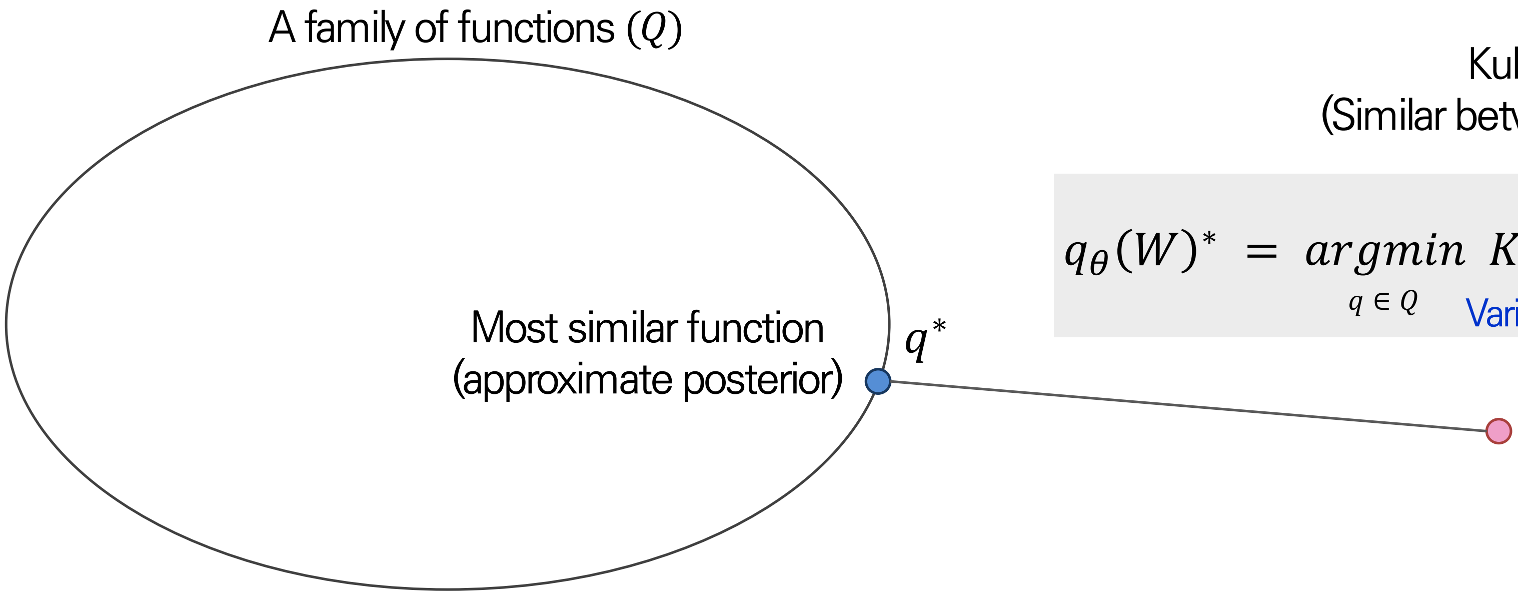
Why Kullback–Leibler Divergence?

- Because it allows us to derive a cost that is tractable to optimization
- Not without paying a price though

❖ Variational Inference

➢ Approximation by using an "easier" distribution $q_\theta(W)$

Variational distribution, where θ are the variational parameters



Kullback–Leibler Divergence
(Similar between two probability distributions)

$$q_\theta(W)^* = \underset{q \in Q}{argmin}\ KL(q_\theta(W) || p(W|X,Y))$$

Variational distribution: $q_\theta(W)$

Posterior: $p(W|X,Y)$

Appendix

Posterior Approximation using variational inference

$$q_\theta(W)^* = \underset{q \in Q}{argmin}\ KL(q_\theta(W)||p(W|X,Y))$$

❖ Variational Inference

➢ Approximation by using an "easier" distribution $q_\theta(W)$

$$KL(q_\theta(W)||p(W|X,Y)) = \int q_\theta(W) ln \frac{q_\theta(W)}{p(W|X,Y)} dw$$

$$p(W|X,Y) = \frac{p(X,Y|W)p(W)}{p(X,Y)}$$

$$= \int q_\theta(W) ln \frac{q_\theta(W)p(X,Y)}{p(X,Y|W)p(W)} dw$$

$$= \int q_\theta(W) ln \frac{q_\theta(W)}{p(W)} dw + \int q_\theta(W) ln(p(X,Y))dw - \int q_\theta(W) ln(p(X,Y|\mathrm{W}))dw$$

Appendix

Posterior Approximation using variational inference

$$q_\theta(W)^* = \underset{q \in Q}{argmin}\ KL(q_\theta(W)||p(W|X,Y))$$

❖ Variational Inference

➢ Approximation by using an “easier” distribution $q_\theta(W)$

$$KL(q_\theta(W)||p(W|X,Y)) = \int q_\theta(W) ln \frac{q_\theta(W)}{p(W)} dw + \int q_\theta(W) ln(p(X,Y))dw - \int q_\theta(W) ln\,(p(X,Y|\mathrm{W}))dw$$

$$\ln\big(p(X,Y)\big) = KL(q_\theta(W)||p(W|X,Y)) - KL(q_\theta(W)||p(W)) + \int q_\theta(W) ln\,(p(X,Y|\mathrm{W}))dw$$

$$\ln\big(p(X,Y)\big) \geq -KL(q_\theta(W)||p(W)) + \int q_\theta(W) ln\,(p(X,Y|\mathrm{W}))dw \qquad (KL(q_\theta(W)||p(W|X,Y)) \geq 0)$$

$$\ln\big(p(Y|X)\big) \geq -KL(q_\theta(W)||p(W)) + \int q_\theta(W) ln\,(p(Y|X,\mathrm{W}))dw$$

Evidence

Evidence Lower Bound (ELBO)

Appendix

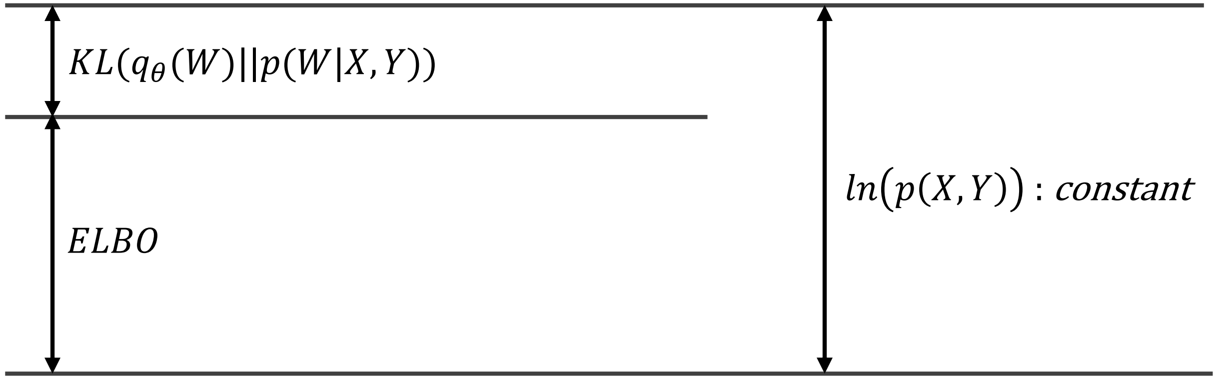
Posterior Approximation using variational inference

❖ Variational Inference

➢ Approximation by using an “easier” distribution $q_\theta(W)$

$$\ln\big(p(X,Y)\big) = KL(q_\theta(W)||p(W|X,Y)) - KL(q_\theta(W)||p(W)) + \int q_\theta(W) ln\,(p(X,Y|\mathrm{W}))dw$$

Evidence (Constant) — KL Divergence (Nonnegative) — Evidence Lower Bound (ELBO)



Minimizing KL Divergence = Maximizing ELBO

Appendix

Posterior Approximation using variational inference

❖ Variational Inference

➢ Approximation by using an "easier" distribution $q_\theta(W)$

$$\ln\big(p(X,Y)\big) = KL\big(q_\theta(W)\big|\big|p(W|X,Y)\big) - KL\big(q_\theta(W)\big|\big|p(W)\big) + \int q_\theta(W)\, ln\,(p(X,Y|\mathrm{W}))dw$$

Evidence (Constant) | KL Divergence (Nonnegative) | Evidence Lower Bound (ELBO)

Minimizing KL Divergence = Maximizing ELBO

Appendix

Training Bayesian Neural Networks

❖ The objective of Bayesian Neural Networks

$$KL(q_\theta(W)||p(W|X,Y)) = \int q_\theta(W) ln \frac{q_\theta(W)}{p(W)} dw + \underbrace{\int q_\theta(W) ln(p(X,Y)) dw}_{\text{Constant}} - \int q_\theta(W) ln\,(p(X,Y|\mathrm{W})) dw$$

$$\propto \int q_\theta(W) ln \frac{q_\theta(W)}{p(W)} dw - \int q_\theta(W) ln\,(p(X,Y|\mathrm{W})) dw$$

$$= -\int q_\theta(W) ln\,(p(X,Y|\mathrm{W})) dw + \int q_\theta(W) ln \frac{q_\theta(W)}{p(W)} dw$$

$$= -\sum_{i=1}^{N} \int q_\theta(W) ln\,(p(y_i|f^w(x_i))) dw + KL(q_\theta(W)||p(W))$$

This objective requires us to perform computations over the entire dataset, which can be too costly for large N

❖ The objective of Bayesian Neural Networks

$$Minimize \; -\sum_{i=1}^{N}\int q_\theta(W)ln\left(p(y_i|f^{w}(x_i))\right)dw + KL(q_\theta(W)||p(W))$$

$$= -\frac{N}{M}\sum_{i\in S}\int q_\theta(W)ln\left(p(y_i|f^{w}(x_i))\right)dw + KL(q_\theta(W)||p(W))$$

Mini–batch optimization

$$= -\frac{N}{M}\sum_{i\in S}\int p(\epsilon)ln(p(y_i|f^{g(\theta,\epsilon)}(x_i)))d\epsilon + KL(q_\theta(W)||p(W))$$

Reparameterization trick

$$= -\frac{N}{M}\sum_{i\in S} ln(p(y_i|f^{g(\theta,\epsilon)}(x_i))) + KL(q_\theta(W)||p(W))$$

Monte Carlo integration

❖ The objective of Bayesian Neural Networks

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

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 $\widehat{\boldsymbol{\epsilon}}_i \sim p(\boldsymbol{\epsilon})$, S a random subset of $\{1, .., N\}$ of size M.
6: Calculate stochastic derivative estimator w.r.t. θ:

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

7: Update θ:

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

8: **until** θ has converged.

Appendix

Dropout as Bayesian Approximation

❖ The objective of Bayesian Neural Networks

$$\hat{L}_{MC}(\theta) = -\frac{N}{M}\sum_{i\in S} ln(p(y_i|f^{g(\theta,\epsilon)}(x_i))) + KL(q_\theta(W)||p(W))$$

Negative log likelihood

$$\hat{L}_{dropout}(\theta) = -\frac{1}{M}\sum_{i\in S} \ln(p(y_i|f^{g(\theta,\hat{\epsilon})}(x_i)) + \lambda_1||M_1||^2 + \lambda_2||M_2||^2 + \lambda_3||b||^2$$

Negative log likelihood

$$\frac{\partial}{\partial\theta}\lambda_1||M_1||^2 + \lambda_2||M_2||^2 + \lambda_3||b||^2 = \frac{\partial}{\partial\theta}KL(q_\theta(W)||p(W))$$

$$\frac{\partial}{\partial\theta}\hat{L}_{dropout}(\theta) = \frac{1}{N}\frac{\partial}{\partial\theta}\hat{L}_{MC}(\theta)$$

We often use L_2 regularization weighted by some weight decay λ,
Resulting in a minimization objective with dropout,
we sample binary variables for every input point and for every network unit in each layer

❖ The objective of Bayesian Neural Networks

Algorithm 2 Optimisation of a neural network with dropout

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 derivative w.r.t. θ:

$$\widehat{\Delta\theta} \leftarrow -\frac{1}{M\tau}\sum_{i \in S}\frac{\partial}{\partial\theta}\log p(\mathbf{y}_i|\mathbf{f}^{g(\theta,\widehat{\epsilon}_i)}(\mathbf{x})) + \frac{\partial}{\partial\theta}\left(\lambda_1||\mathbf{W}_1||^2 + \lambda_2||\mathbf{W}_2||^2 + \lambda_3||\mathbf{b}||^2\right).$$

7: Update θ:

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

8: **until** θ has converged.