Uncertainty Quantification in Deep Learning

Jiyoon Lee

2022 . 01 . 28

Introduction
발표자 소개

❖ **이지윤 (Jiyoon Lee)**

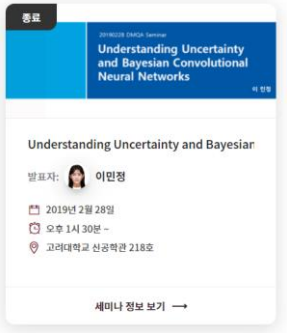
- Data Mining & Quality Analytics Lab
- Ph.D. Candidates (2018.03 ~ Present)

❖ **Research Interest**

- Explainable neural network using Attention mechanism & Bayesian neural network
- Graph-based semi-supervised learning using Label propagation

❖ **Contact**

- Tel: +82-2-3290-3769
- E-mail: jiyoonlee@korea.ac.kr



Contents

Introduction

Background

Expectation

Training dataset

Dogs

Driving

Introduction

Background

Expectation

Training dataset

Reality

Testing in reality

Dogs

Driving

Introduction

Background

뉴스투데이 김수산 리포터

[이슈톡] '자율주행'이라더니…테슬라, 전복 트럭도 못 피하고 정면 충돌

입력 2020-06-03 06:51 | 수정 2020-06-03 09:22

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

미 ABC 방송의 서부지역 네트워크인 KGO-TV는 이번 테슬라 운전자 월터 후앙의 사망 사고가 지난해 9월 발생한 테슬라의 자동주행 차량 사고와 비슷하다고 최근 보도했다. 지난달 테슬라의 사고는 오전 역광이 내리쬐는 상황에서 차량이 중앙 분리대를 들이받아 발생했는데, 6개월 전 사고도 오전 역광으로 눈부신 상황이었다는 것이다. 이에 앞서 테슬라는 지난 2016년 발생한 트레일러 충돌 사고에 대해 "자동주행 차량이 역광 탓에 흰색 트레일러를 하늘로 오인해 충돌사고를 냈다"고 사고 원인을 밝힌 바 있다.

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]

https://www.joongang.co.kr/article/22520461#home

Introduction

Background

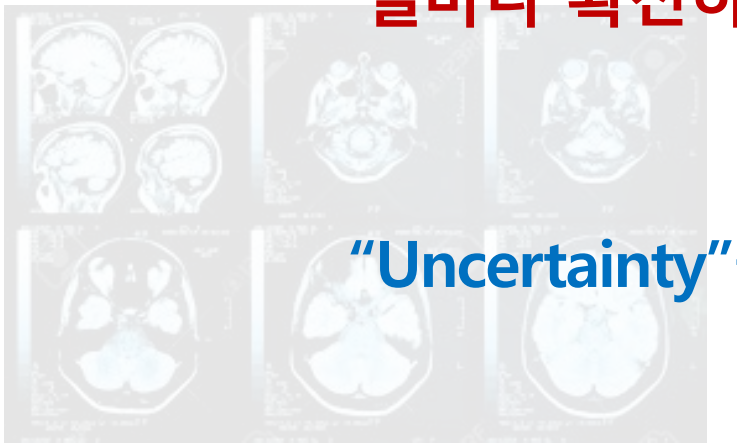
Safety critical domains

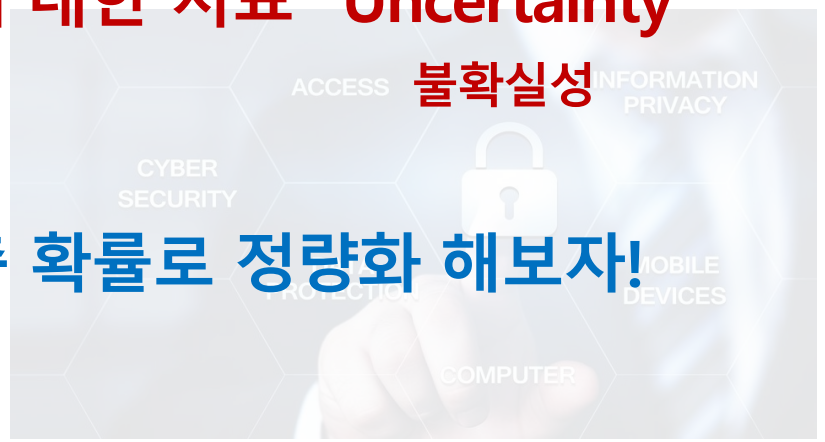
예측 “Prediction”

얼마나 확신하는지에 대한 지표 “Uncertainty”

불확실성

“Uncertainty”를 예측 확률로 정량화 해보자!





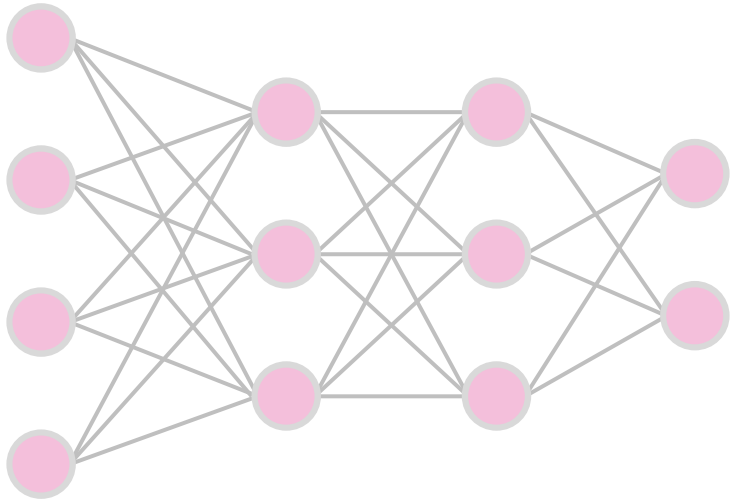
https://www.joongang.co.kr/article/22520461#home

Introduction

Uncertainty

❖ Standard Neural Networks

- SoftMax를 통해 logit값을 확률 값으로 변환함으로써 예측 확률이 도출



Certain !

$$P(y = dog|x, w) = 0.9$$

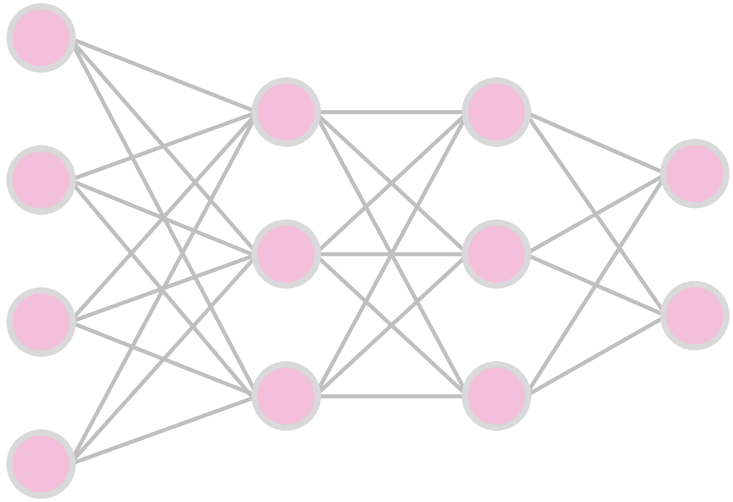
$$P(y = cat|x, w) = 0.1$$

Introduction

Uncertainty

❖ Standard Neural Networks

- SoftMax를 통해 logit값을 확률 값으로 변환함으로써 예측 확률이 도출



$$P(y = dog|x, w) = 0.1$$

$$P(y = cat|x, w) = 0.9$$

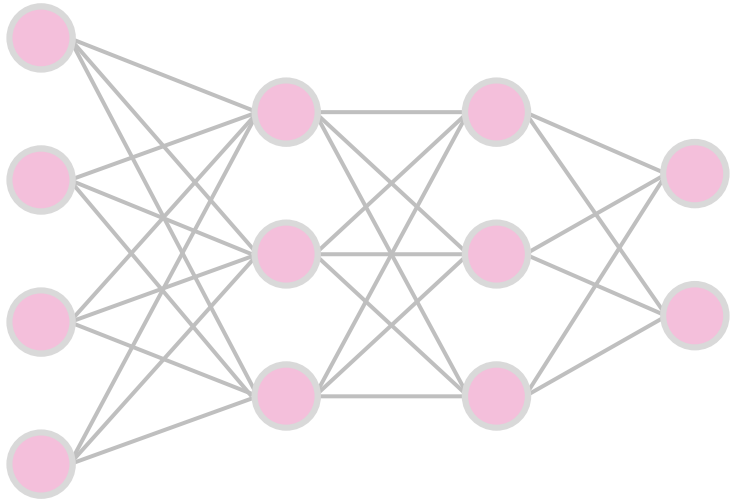
Certain !

Introduction

Uncertainty

❖ Standard Neural Networks

- SoftMax를 통해 logit값을 확률 값으로 변환함으로써 예측 확률이 도출



Uncertain ???

$$P(y = dog|x, w) = 0.5$$

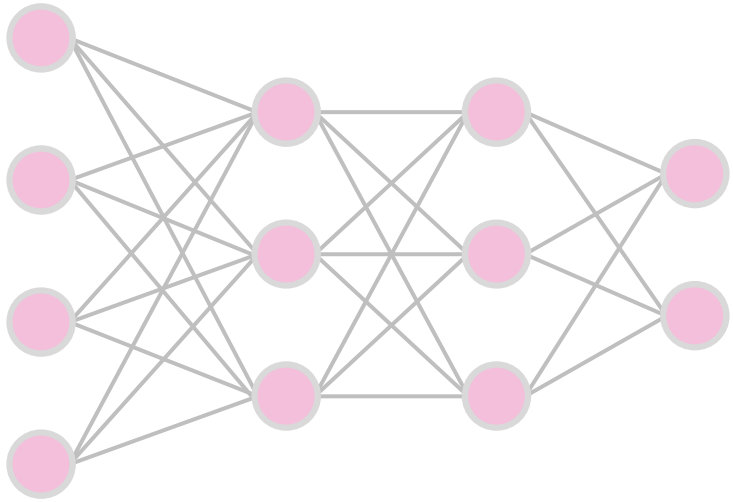
$$P(y = cat|x, w) = 0.5$$

Introduction

Uncertainty

❖ Standard Neural Networks

- SoftMax를 통해 logit값을 확률 값으로 변환함으로써 예측 확률이 도출



Certain ???

$$P(y = dog|x, w) = 0.9$$

$$P(y = cat|x, w) = 0.1$$

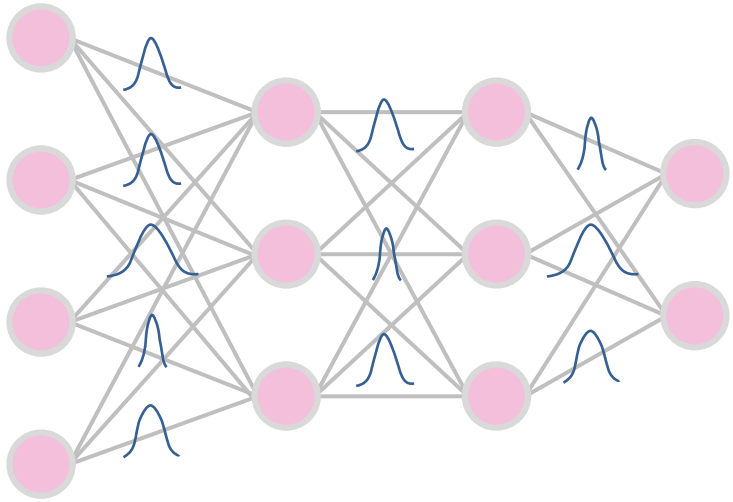
"Uncertainty"를 예측 확률로 정량화 하는 것은 한계가 있음!

Introduction

Uncertainty

❖ Bayesian Neural Networks

- Parameter w 에 분포를 가정하여, 예측 값을 분포로 추정할 수 있음
- 도출된 예측값의 분산정보를 활용하여 불확실성 정량화가 가능



$$P(y = dog|x, w) = 0.9$$

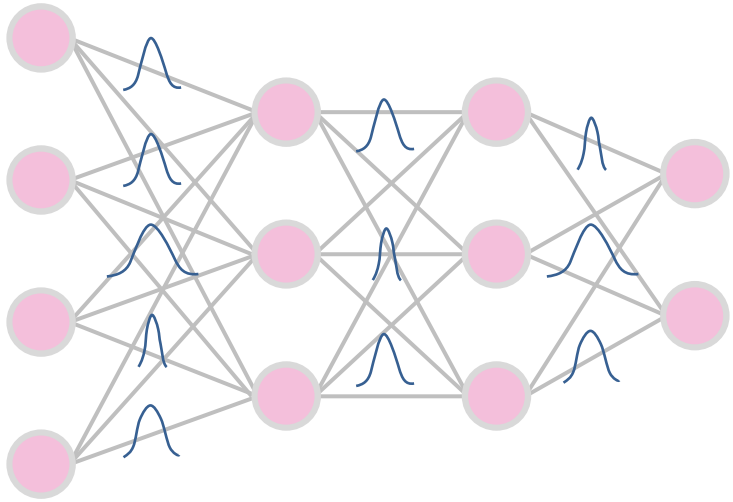
$$P(y = cat|x, w) = 0.1$$

Introduction

Uncertainty

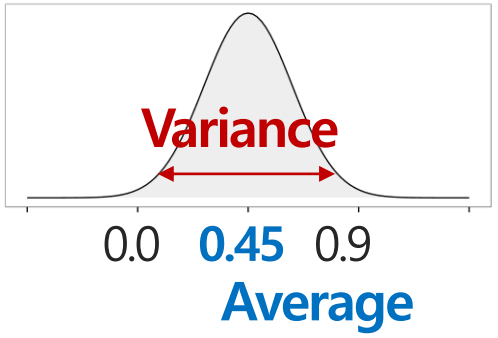
❖ Bayesian Neural Networks

- Parameter w 에 분포를 가정하여, 예측 값을 분포로 추정할 수 있음
- 도출된 예측값의 분산정보를 활용하여 불확실성 정량화가 가능



$$P(y = dog|x, w) = 0.9$$

0.2 0.4 0.4 0.9 0.3 0.8 0.4 0.5 0.4 0.2 0.4



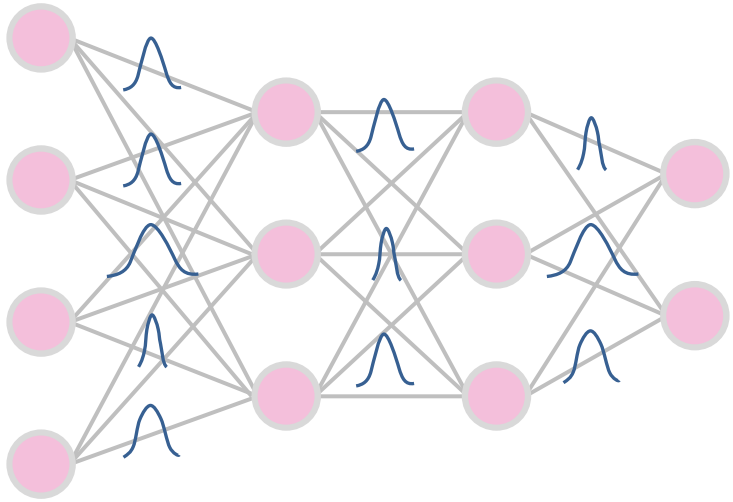
"Uncertainty"를 예측 확률의 분산으로 정량화 하고자 함

Introduction

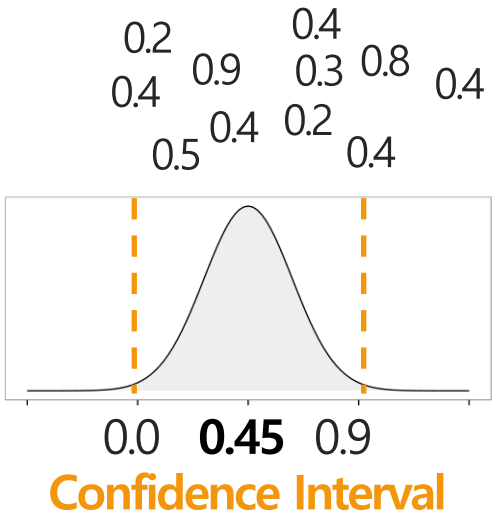
Uncertainty

❖ Bayesian Neural Networks

- Parameter w 에 분포를 가정하여, 예측 값을 분포로 추정할 수 있음
- 도출된 예측값의 분산정보를 활용하여 불확실성 정량화가 가능



$$P(y = dog|x, w) = 0.9$$



Uncertainty Quantification

Abdar, Moloud, et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges.“ *Information Fusion* (2021).

Uncertainty Quantification

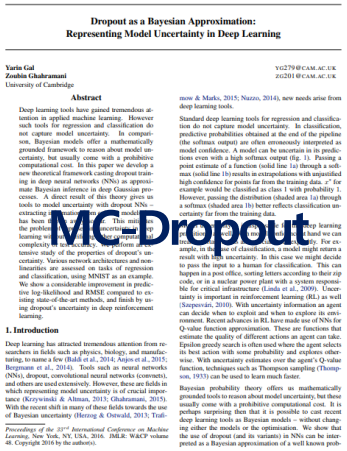
예측에 대한 불확실성

잘 정량화하자

분산정보를 어떻게 도출할지

분산정보를 어떻게 요약할지

Bayesian-based Approach

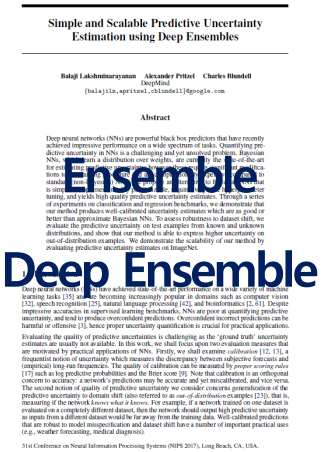


Dropout as a Bayesian Approximation:
Representing Model Uncertainty in Deep Learning

Yarin Gal
Zoubin Ghahramani
University of Cambridge

MC Dropout

Ensemble-based Approach



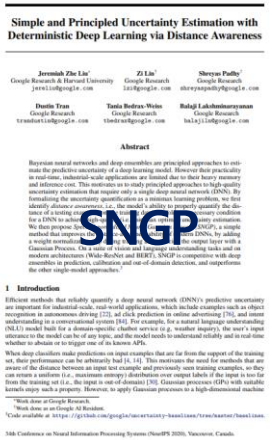
Simple and Scalable Predictive Uncertainty
Estimation using Deep Ensembles

Balaji Lakshminarayanan Alexander Pritzel Charles Blundell
DeepMind

Ensemble

Deep Ensemble

GP-based Approach



Simple and Principled Uncertainty Estimation with
Deterministic Deep Learning via Distance Awareness

SNGP

Abdar, Moloud, et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges." *Information Fusion* (2021).

Uncertainty Quantification

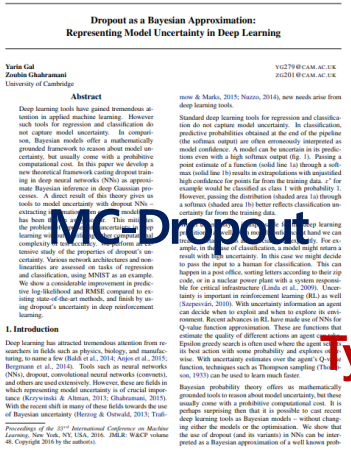
예측에 대한 불확실성

잘 정량화하자

모델이 불확실한 경우

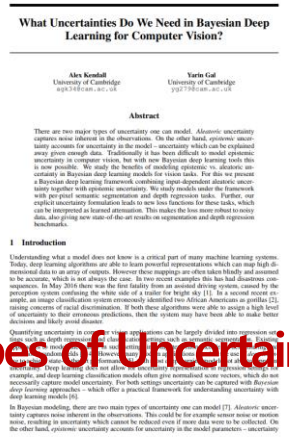
데이터가 불확실한 경우

Bayesian-based Approach


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

Yarin Gal
Zoubin Ghahramani
University of Cambridge

MC Dropout


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

Alex Kendall
University of Cambridge

Yarin Gal
University of Cambridge

Types of Uncertainty

Ensemble-based Approach


Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

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

Ensemble

Deep Ensemble

GP-based Approach


Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness

Jeremiah Zhe Liu
Google Research & Harvard University

Zi Lin
Google Research

Shreyas Padhy
Google Research

Dustin Tran
Google Research

Tania Bedrax-Weiss
Google Research

Balaji Lakshminarayanan
Google Research

SNGP

Abdar, Moloud, et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges."
Information Fusion (2021).

Introduction

Types of Uncertainty

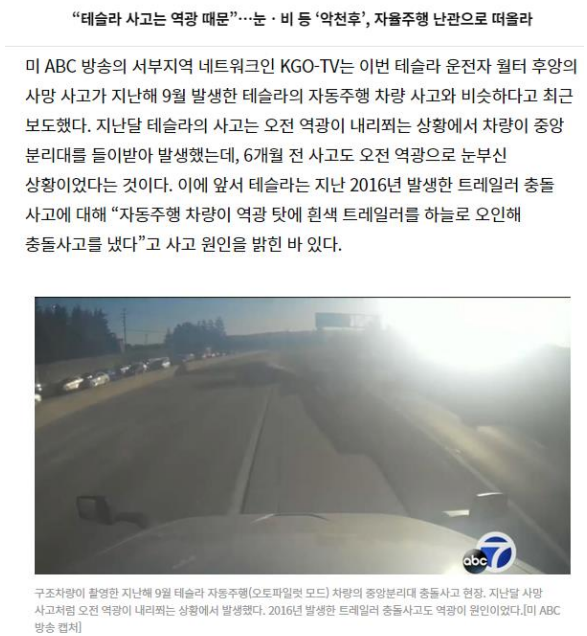
❖ Epistemic uncertainty (model uncertainty)

- 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)
- 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- 측정 정밀도를 높이면 줄일 수 있음



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

미 ABC 방송의 서부지역 네트워크인 KGO-TV는 이번 테슬라 운전자 월터 후앙의 사망 사고가 지난해 9월 발생한 테슬라의 자동주행 차량 사고와 비슷하다고 최근 보도했다. 지난달 테슬라의 사고는 오전 역광이 내리쬐는 상황에서 차량이 중앙 분리대를 들이받아 발생했는데, 6개월 전 사고도 오전 역광으로 눈부신 상황이었다는 것이다. 이에 앞서 테슬라는 지난 2016년 발생한 트레일러 충돌 사고에 대해 "자동주행 차량이 역광 탓에 흰색 트레일러를 하늘로 오인해 충돌사고를 냈다"고 사고 원인을 밝힌 바 있다.

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]

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?. *Advances in Neural Information Processing Systems*, *30*, 5574-5584.

Introduction

Types of Uncertainty

❖ 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

❖ 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?. *Advances in Neural Information Processing Systems*, *30*, 5574-5584.

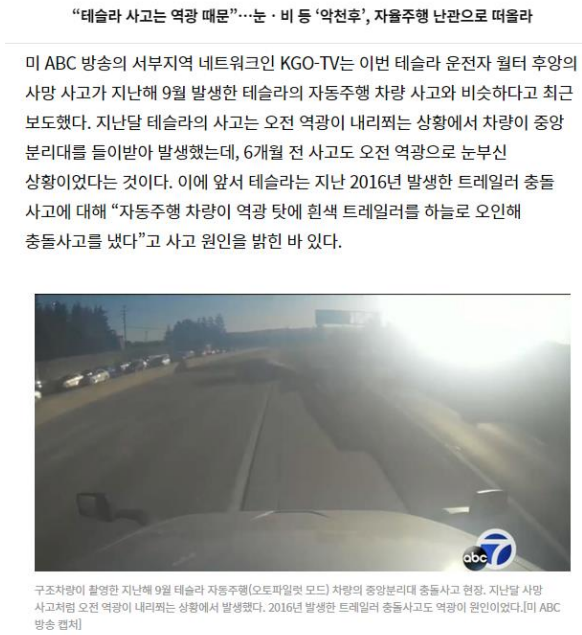
Introduction

Types of Uncertainty

❖ Aleatoric uncertainty (data uncertainty)

- 데이터에 내재된 노이즈로 인해 이해하지 못하는 정도

  (e.g. measurement noise, randomness inherent)
- 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- 측정 정밀도를 높이면 줄일 수 있음



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

미 ABC 방송의 서부지역 네트워크인 KGO-TV는 이번 테슬라 운전자 월터 후앙의 사망 사고가 지난해 9월 발생한 테슬라의 자동주행 차량 사고와 비슷하다고 최근 보도했다. 지난달 테슬라의 사고는 오전 역광이 내리쬐는 상황에서 차량이 중앙 분리대를 들이받아 발생했는데, 6개월 전 사고도 오전 역광으로 눈부신 상황이었다는 것이다. 이에 앞서 테슬라는 지난 2016년 발생한 트레일러 충돌 사고에 대해 "자동주행 차량이 역광 탓에 흰색 트레일러를 하늘로 오인해 충돌사고를 냈다"고 사고 원인을 밝힌 바 있다.

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]

❖ Why Aleatoric uncertainty?

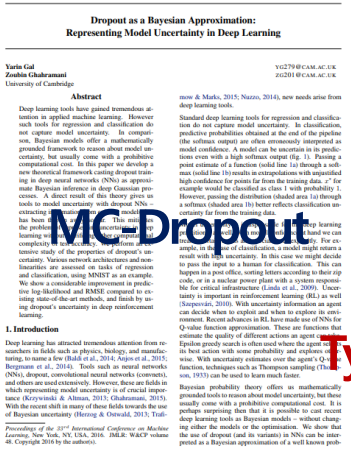
- Aleatoric uncertainty는 실제 상황에서와 같이 일부 데이터의 노이즈가 높게 존재하는 경우 중요
- 노이즈가 큰 데이터에 대해 학습과정에서 제약을 부여할 수 있으므로, 예측 성능 안정화 과정에 기여

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?. *Advances in Neural Information Processing Systems*, *30*, 5574-5584.

Uncertainty Quantification

예측에 대한 불확실성 잘 정량화하자

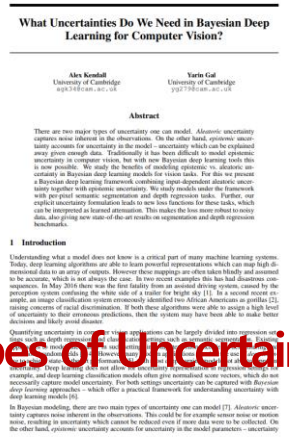
Bayesian-based Approach



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

Yarin Gal
Zoubin Ghahramani
University of Cambridge

MC Dropout



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

Alex Kendall
University of Cambridge

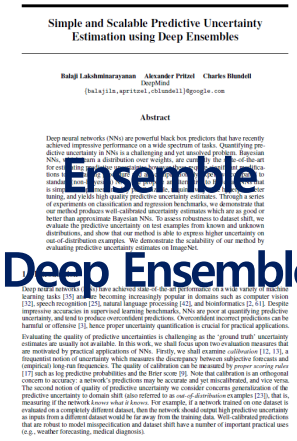
Yarin Gal
University of Cambridge

Types of Uncertainty

Epistemic

Epistemic+Aleatoric

Ensemble-based Approach



Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

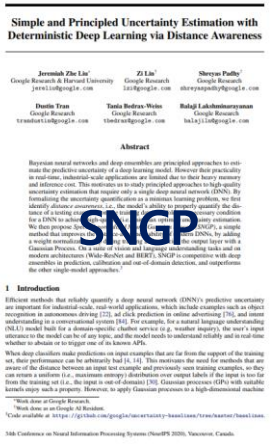
Balaji Lakshminarayanan Alexander Pritzel Charles Blundell
DeepMind

Ensemble

Deep Ensemble

Aleatoric

GP-based Approach



Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness

Jeremiah Zhe Liu
Google Research & Harvard University

Zi Lin
Google Research

Shreyas Padhy
Google Research

Dustin Tran
Google Research

Tania Bedrax-Weiss
Google Research

Balaji Lakshminarayanan
Google Research

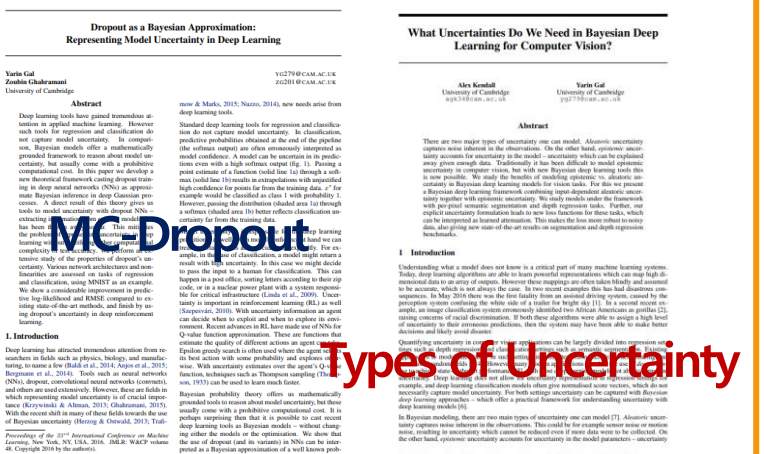
SNGP

Out-of-distribution
(calibrated probability)

Uncertainty Quantification

예측에 대한 불확실성 잘 정량화하자

Bayesian-based Approach

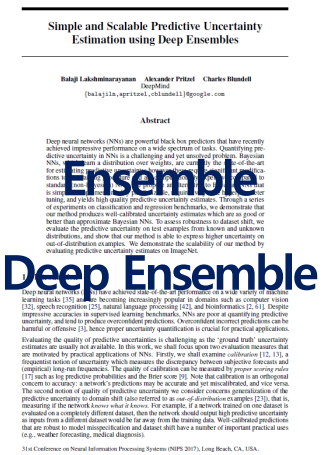

Dropout as a Bayesian Approximation:
Representing Model Uncertainty in Deep Learning

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

MC Dropout

Types of Uncertainty

Ensemble-based Approach


Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Ensemble

Deep Ensemble

GP-based Approach


Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness

SNGP

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ **ICML 2016 (22년 1월 기준 4670건 인용)**

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 learning without sacrificing either computational complexity or test accuracy. We perform an extensive study of the properties of dropout's uncertainty. Various network architectures and nonlinearities are assessed on tasks of regression and classification, using MNIST as an example. We show a considerable improvement in predictive log-likelihood and RMSE compared to existing state-of-the-art methods, and finish by using dropout's uncertainty in deep reinforcement learning.

1. Introduction

Deep learning has attracted tremendous attention from researchers in fields such as physics, biology, and manufacturing, to name a few (Baldi et al., 2014; Anjos et al., 2015; Bergmann et al., 2014). Tools such as neural networks (NNs), dropout, convolutional neural networks (convnets), and others are used extensively. However, these are fields in which representing model uncertainty is of crucial importance (Krzywinski & Altman, 2013; Ghahramani, 2015).

With the recent shift in many of these fields towards the use of Bayesian uncertainty (Herzog & Ostwald, 2013; Trafimow & 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 practitioner as well. With model confidence at hand we can treat uncertain inputs and special cases explicitly. For example, in the case of classification, a model might return a result with high uncertainty. In this case we might decide to pass the input to a human for classification. This can happen in a post office, sorting letters according to their zip code, or in a nuclear power plant with a system responsible for critical infrastructure (Linda et al., 2009). Uncertainty is important in reinforcement learning (RL) as well (Szepesvári, 2010). With uncertainty information an agent can decide when to exploit and when to explore its environment. Recent advances in RL have made use of NNs for Q-value function approximation. These are functions that estimate the quality of different actions an agent can take. Epsilon greedy search is often used where the agent selects its best action with some probability and explores otherwise. With uncertainty estimates over the agent's Q-value function, techniques such as Thompson sampling (Thompson, 1933) can be used to learn much faster.

Bayesian probability theory offers us mathematically grounded tools to reason about model uncertainty, but these usually come with a prohibitive computational cost. It is perhaps surprising then that it is possible to cast recent

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

Yarin Gal

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

Machine Learning Artificial Intelligence Probability Theory Statistics

팔로우

제목	인용	연도
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)	4721	2015
What uncertainties do we need in Bayesian deep learning for computer vision? A Kendall, Y Gal Advances in neural information processing systems, 5574-5584	2403	2017
A theoretically grounded application of dropout in recurrent neural networks Y Gal, Z Ghahramani Advances in neural information processing systems 29, 1019-1027	1499	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 ...	1409	2018
Uncertainty in Deep Learning Y Gal University of Cambridge	1211	2016
Deep Bayesian Active Learning with Image Data Y Gal, R Islam, Z Ghahramani International Conference on Machine Learning (ICML), 1183-1192	819	2017
Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference Y Gal, Z Ghahramani 4th International Conference on Learning Representations (ICLR) workshop track	545	2015

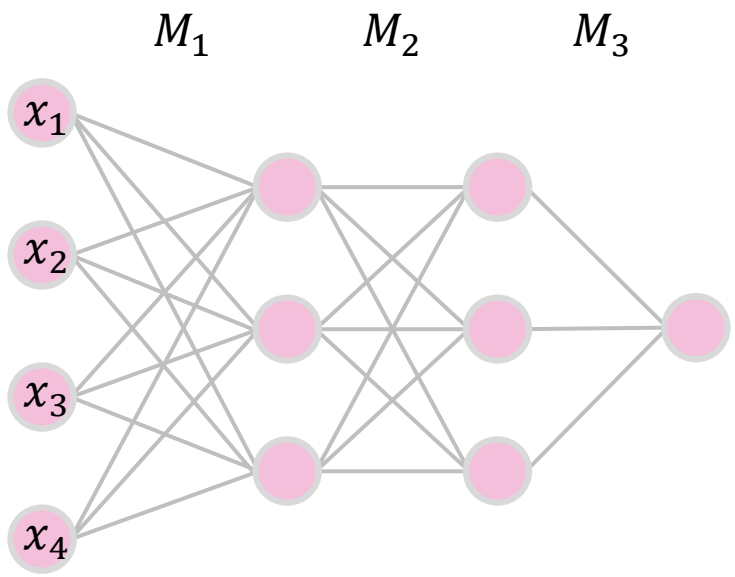
Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." international conference on machine learning. PMLR, 2016.

Bayesian-based Approach

Frequentist way & Bayesian way

❖ Frequentist : Standard Deep Learning / Deterministic Deep Learning

- 학습 후, 동일한 입력 값에 대해서는 동일한 예측 값이 도출

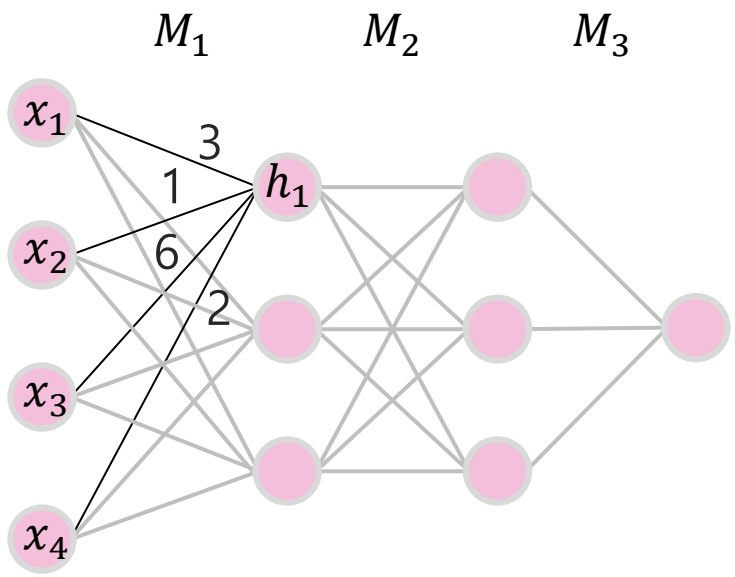


Bayesian-based Approach

Frequentist way & Bayesian way

❖ Frequentist : Standard Deep Learning / Deterministic Deep Learning

- 학습 후, 동일한 입력 값에 대해서는 동일한 예측 값이 도출



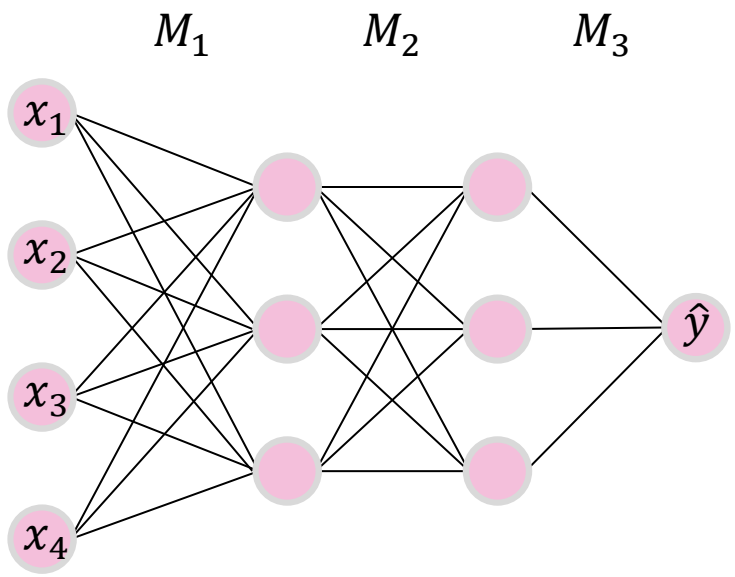
$$h_1 = 3\,x_1 + 1\,x_2 + 6\,x_3 + 2\,x_4$$

Bayesian-based Approach

Frequentist way & Bayesian way

❖ **Frequentist : Standard Deep Learning / Deterministic Deep Learning**

- 학습 후, 동일한 입력 값에 대해서는 동일한 예측 값이 도출



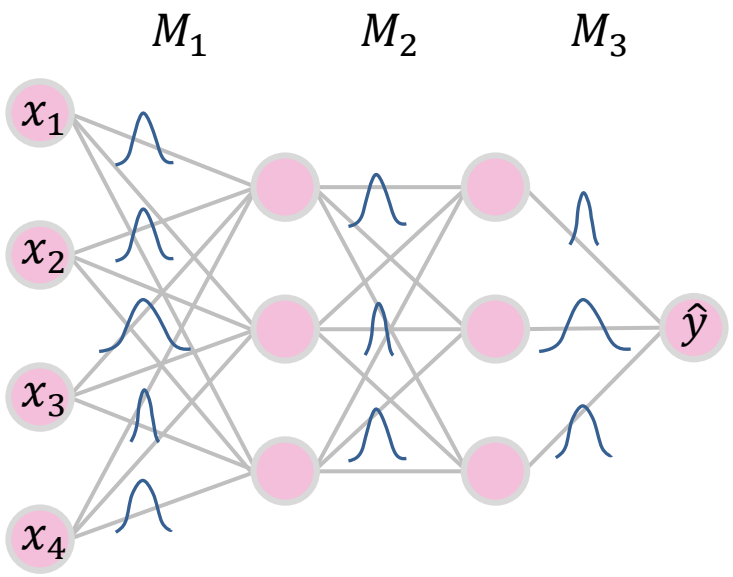
T=1	T=2	T=3	…	T=T
48	48	48	…	48

Bayesian-based Approach

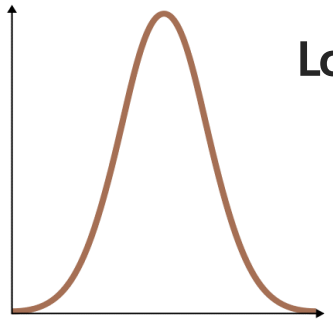
Frequentist way & Bayesian way

❖ Bayesian : Bayesian Deep learning / Stochastic Deep Learning

- 학습 후, 동일한 입력 값에 대해 서로 다른 예측 값이 도출



T=1	T=2	T=3	…	T=T	
48	52	46	…	47	$\hat{y} \sim N(50, 2^2)$



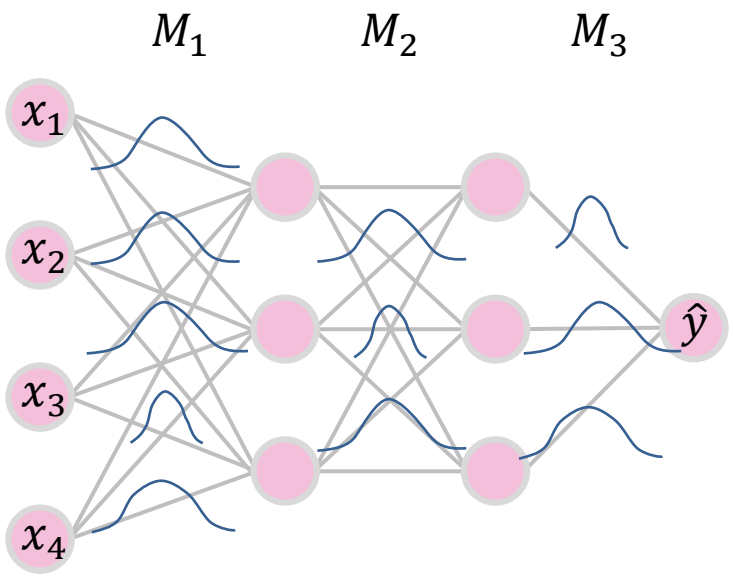
Low variance

Bayesian-based Approach

Frequentist way & Bayesian way

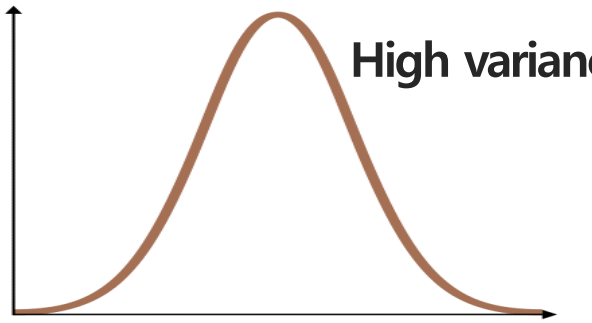
❖ **Bayesian : Bayesian Deep learning / Stochastic Deep Learning**

- 학습 후, 동일한 입력 값에 대해 서로 다른 예측 값이 도출



T=1	T=2	T=3	…	T=T
59	46	32	…	50

$\hat{y} \sim N(50, 10^2)$



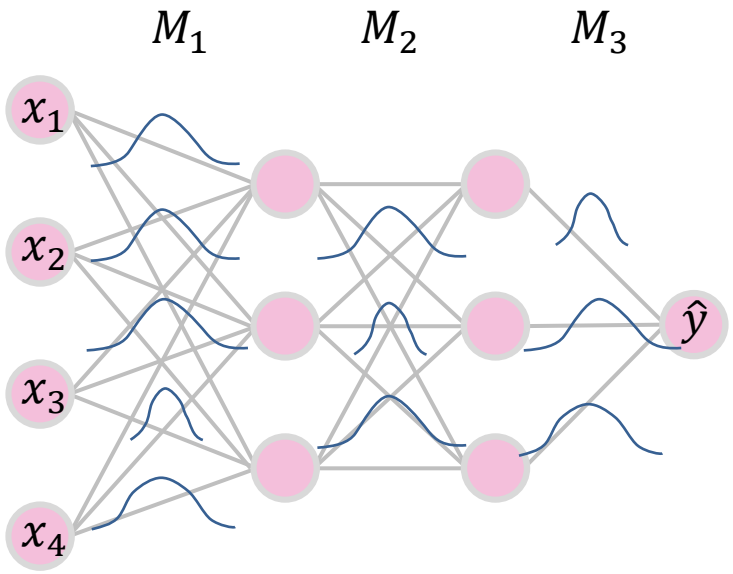
어떻게 파라미터 분포를 추정할까?

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ **Bayesian : Bayesian Deep learning / Stochastic Deep Learning**

- MC dropout과 weight에 L2 regularization을 적용시키는 것으로 동일한 효과임을 증명



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

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

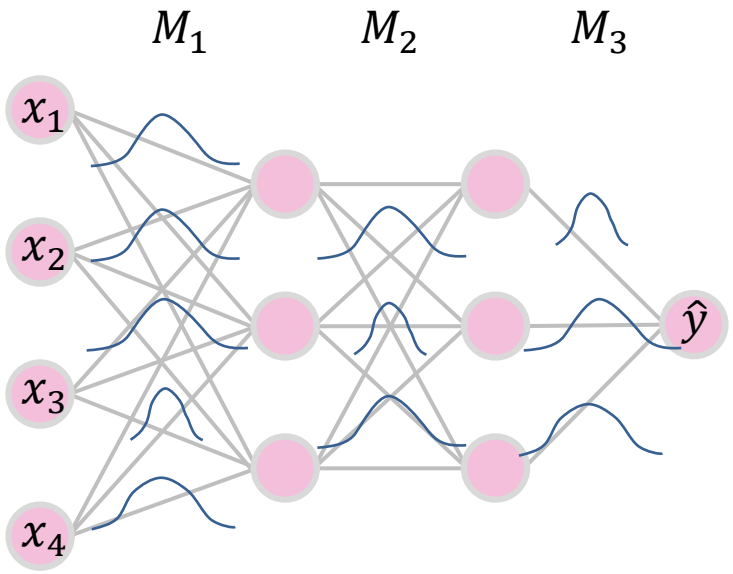
This integration is not computable in general

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ **Bayesian : Bayesian Deep learning / Stochastic Deep Learning**

- MC dropout과 weight에 L2 regularization을 적용시키는 것으로 동일한 효과임을 증명



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

임의로 분포를 가정하고,
이를 posterior와 비슷하게 근사

$$q_\theta(W)$$

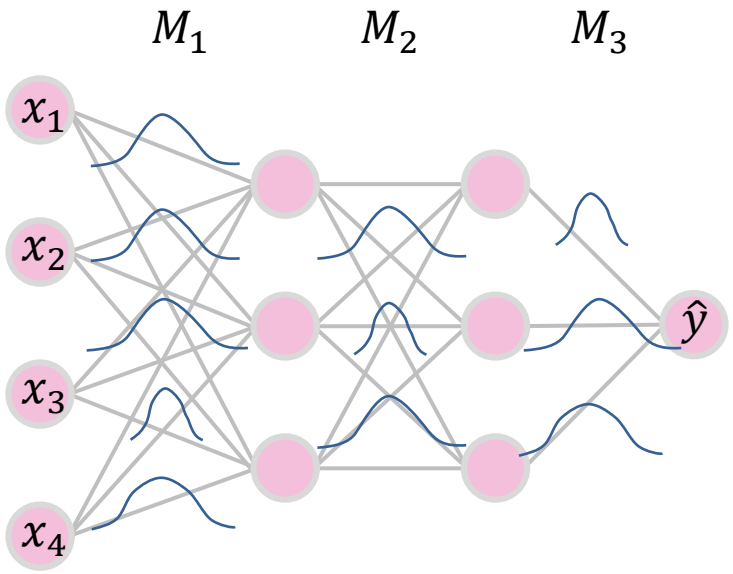
Variational distribution

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Bayesian : Bayesian Deep learning / Stochastic Deep Learning

- MC dropout과 weight에 L2 regularization을 적용시키는 것으로 동일한 효과임을 증명



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

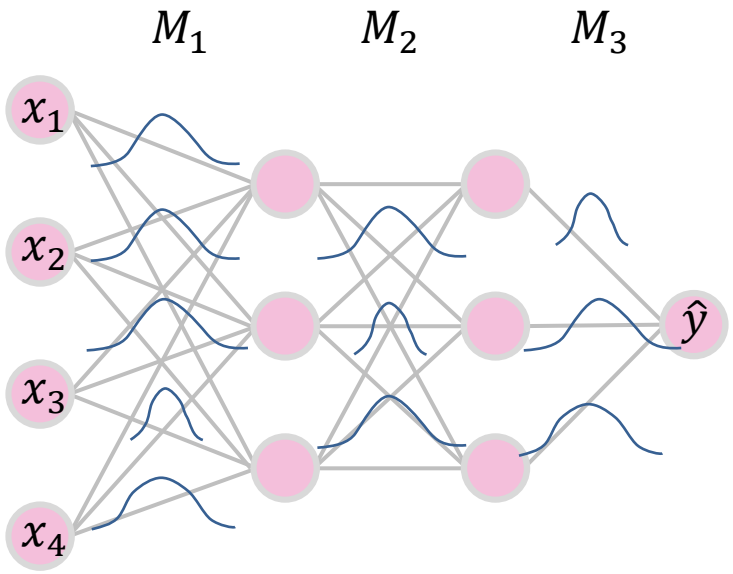
Variational distribution

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ **Bayesian : Bayesian Deep learning / Stochastic Deep Learning**

- MC dropout과 weight에 L2 regularization을 적용시키는 것으로 동일한 효과임을 증명



MC dropout with L2 regularization

Variational inference

Kullback-Leibler Divergence
(두 확률분포의 차이를 계산)

Posterior

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

Variational distribution

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ **Loss function 정의 (Appendix 참고)**

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

$$= \quad Maximize\ ELBO$$

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

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

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

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

Regression: MSE

Classification: Softmax cross entropy

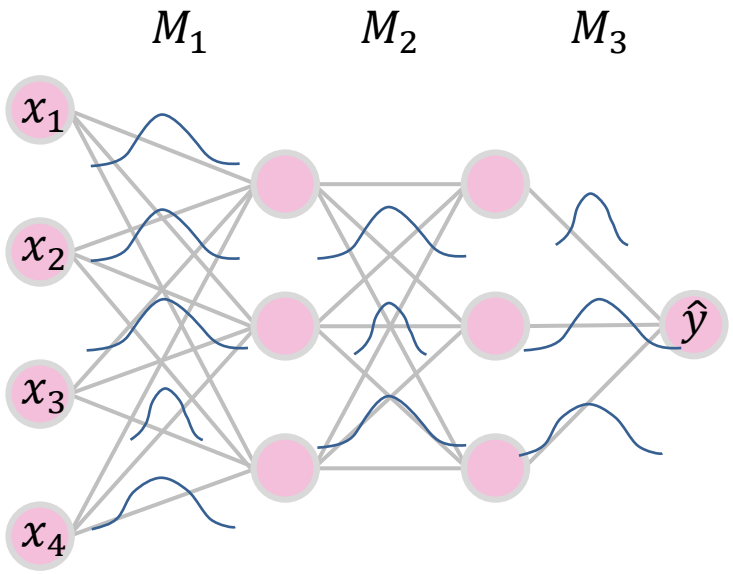
Gal, Y. (2016). Uncertainty in deep learning.

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ **Bayesian : Bayesian Deep learning / Stochastic Deep Learning**

- MC dropout과 weight에 L2 regularization을 적용시키는 것으로 동일한 효과임을 증명



MC dropout with L2 regularization

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

Variational distribution

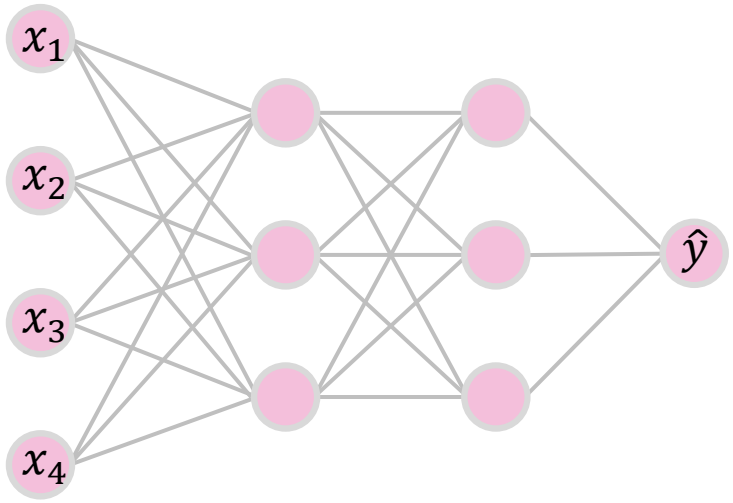
Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Dropout

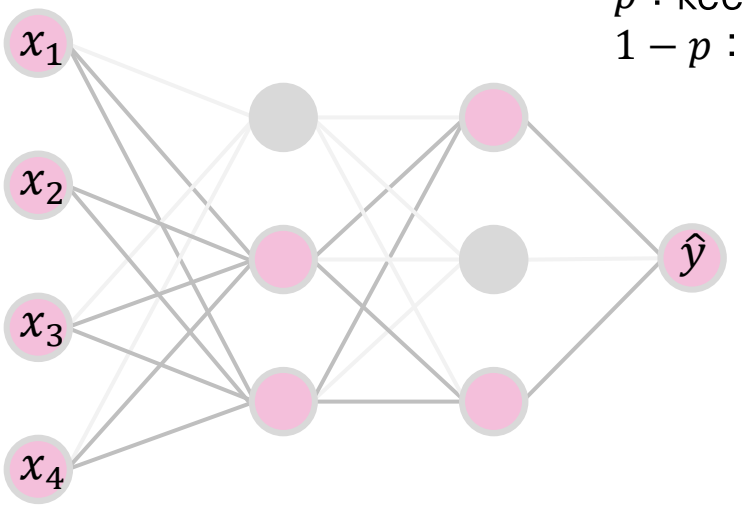
- 학습데이터에 대한 오버피팅을 방지하기 위한 다양한 정규화 방법론들이 있음
- Dropout은 대표적인 모델 정규화 방법으로 배치마다 무작위로 노드 연결을 끊음
- 이때, dropout을 적용할 비율$(1-p)$은 사용자가 정의

Standard Neural Network



After applying dropout

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



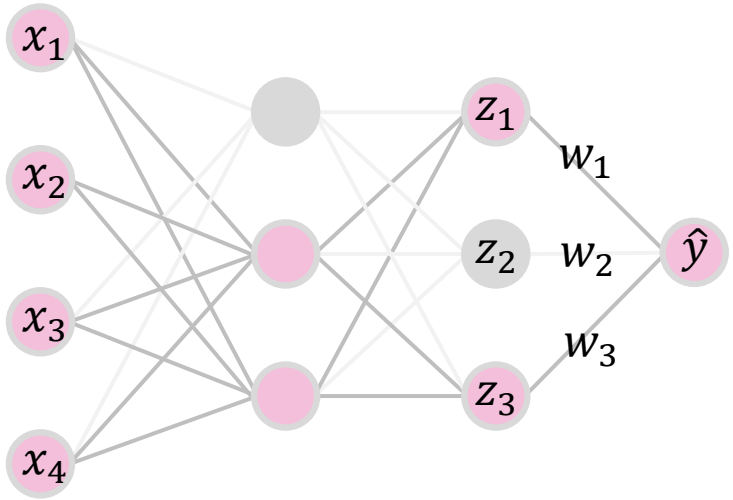
Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Dropout

- Dropout은 추론(inference / test)단계에서는 파라미터를 고정적으로 모델링 (deterministic)
- 고정적인 파라미터에 가중치 p를 곱하여 최종적인 예측 수행 → 결과는 고정적

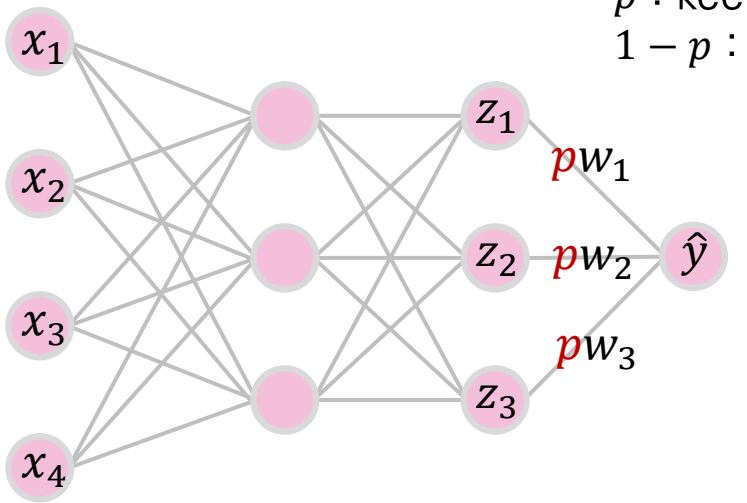
Training Phase



$$\hat{y} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Testing Phase

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



$$\hat{y} = pz_1 w_1 + pz_2 w_2 + pz_3 w_3$$

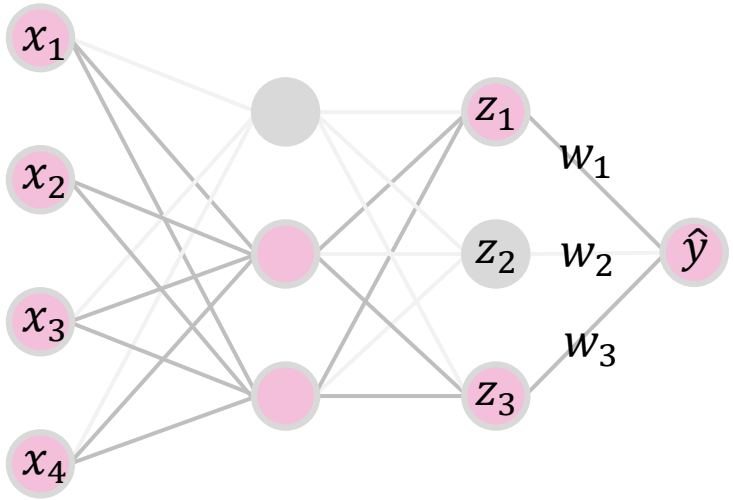
Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Monte Carlo Dropout (MC dropout)

- MC dropout은 추론(inference / test)단계에서도 dropout 적용하여 파라미터 확률적으로 모델링 (Stochastic)
- 매번 추론할 때(stochastic forward pass, T)마다 상이한 예측이 수행 → 결과는 확률적

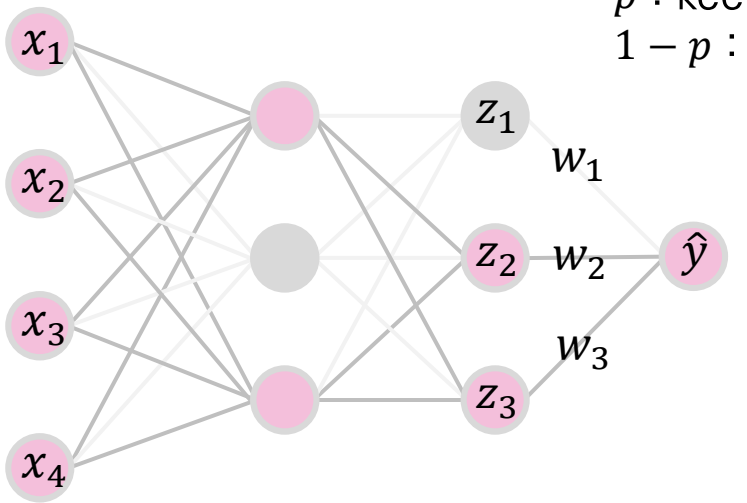
Training Phase



$$\hat{y} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Testing Phase (T=1)

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



$$\widehat{y_1} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

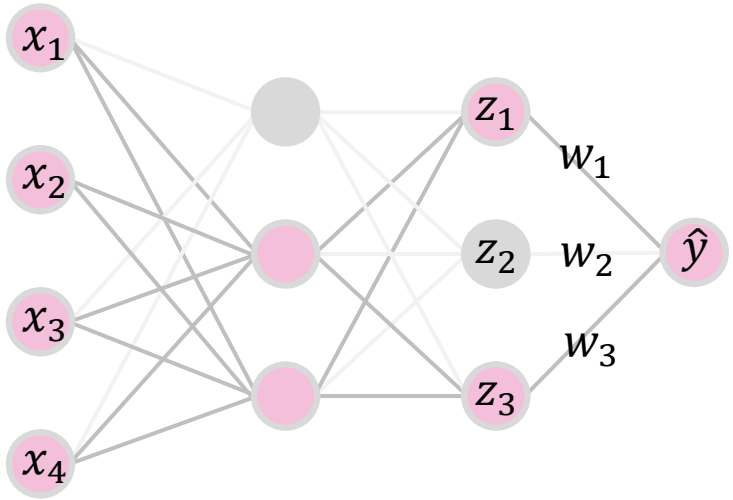
Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Monte Carlo Dropout (MC dropout)

- MC dropout은 추론(inference / test)단계에서도 dropout 적용하여 파라미터 확률적으로 모델링 (Stochastic)
- 매번 추론할 때(stochastic forward pass, T)마다 상이한 예측이 수행 → 결과는 확률적

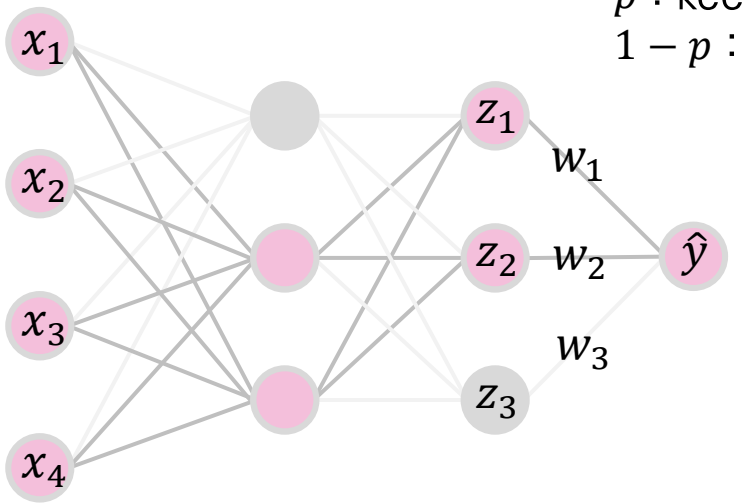
Training Phase



$$\hat{y} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Testing Phase (T=2)

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



$$\widehat{y_2} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

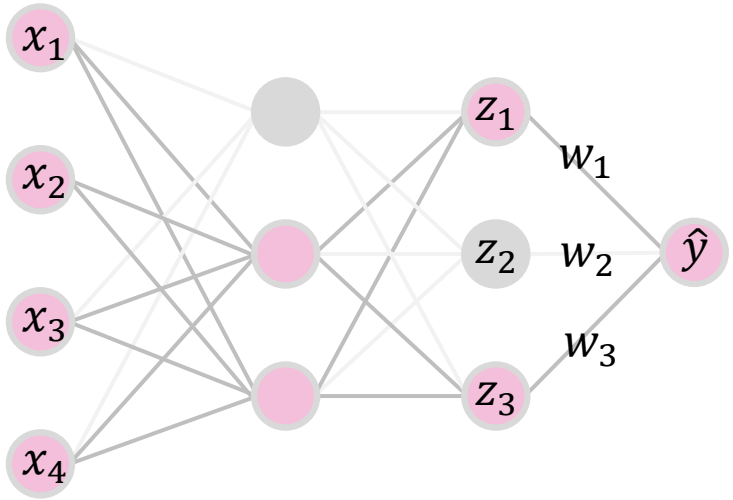
Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Monte Carlo Dropout (MC dropout)

- MC dropout은 추론(inference / test)단계에서도 dropout 적용하여 파라미터 확률적으로 모델링 (Stochastic)
- 매번 추론할 때(stochastic forward pass, T)마다 상이한 예측이 수행 → 결과는 확률적

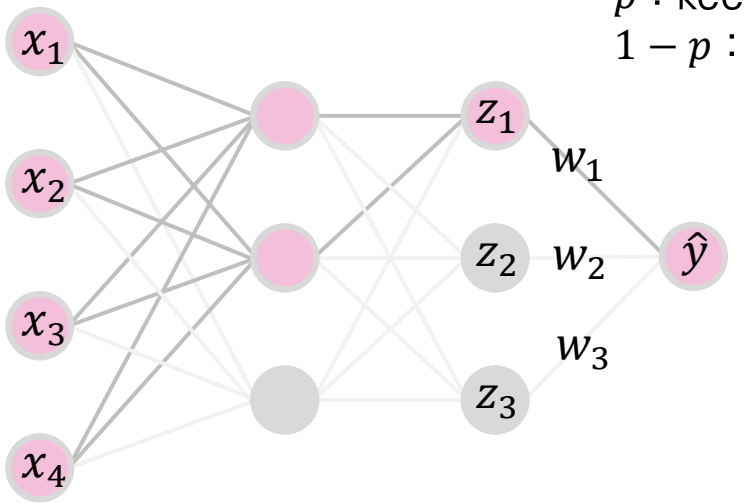
Training Phase



$$\hat{y} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Testing Phase (T=3)

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



$$\widehat{y_3} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Monte Carlo Dropout (MC dropout)

- MC dropout은 추론(inference / test)단계에서도 dropout 적용하여 파라미터 확률적으로 모델링 (Stochastic)
- 매번 추론할 때(stochastic forward pass, T)마다 상이한 예측이 수행 → 결과는 확률적

Testing Phase

$$\widehat{y_1} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

$$\widehat{y_2} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

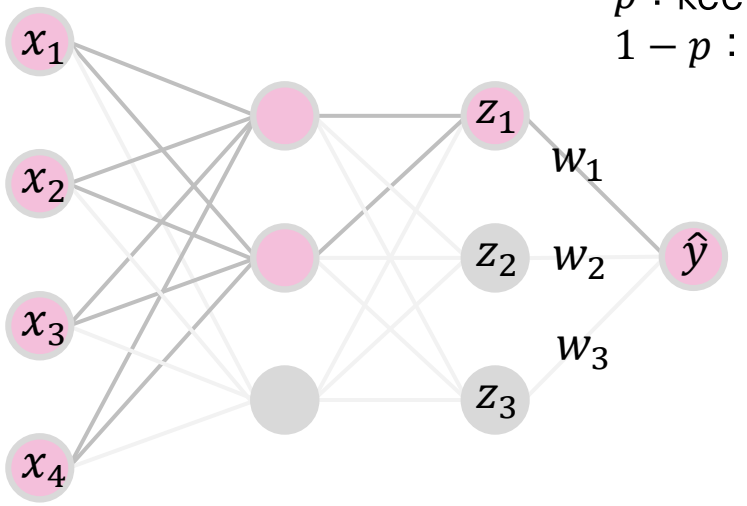
$$\widehat{y_3} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

$$\vdots$$

$$\widehat{y_T} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Testing Phase (T=3)

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



$$\widehat{y_3} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

Bayesian-based Approach

Dropout as Bayesian Approximation

❖ Monte Carlo Dropout (MC dropout)

- MC dropout은 추론(inference / test)단계에서도 dropout 적용하여 파라미터 확률적으로 모델링 (Stochastic)
- 매번 추론할 때(stochastic forward pass, T)마다 상이한 예측이 수행 → 결과는 확률적

Testing Phase

$$\widehat{y_1} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

$$\widehat{y_2} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

$$\widehat{y_3} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

$$\vdots$$

$$\widehat{y_T} = z_1 w_1 + z_2 w_2 + z_3 w_3$$

After T stochastic forward passes

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

Prediction

$$Var(y) \approx \tau^{-1}\mathrm{I_D} + \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t{}^{\mathrm{T}}\, \hat{y}_t - E(y)^T E(y)$$

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

Epistemic uncertainty

Bayesian-based Approach

Bayesian Neural Networks for Computer Vision

- **NeurIPS 2017 (22년 1월 기준 2403건 인용)**
  - Uncertainty 정량화를 세분화함 (Epistemic uncertainty, Aleatoric uncertainty)
  - Computer vision tasks에 적용

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.

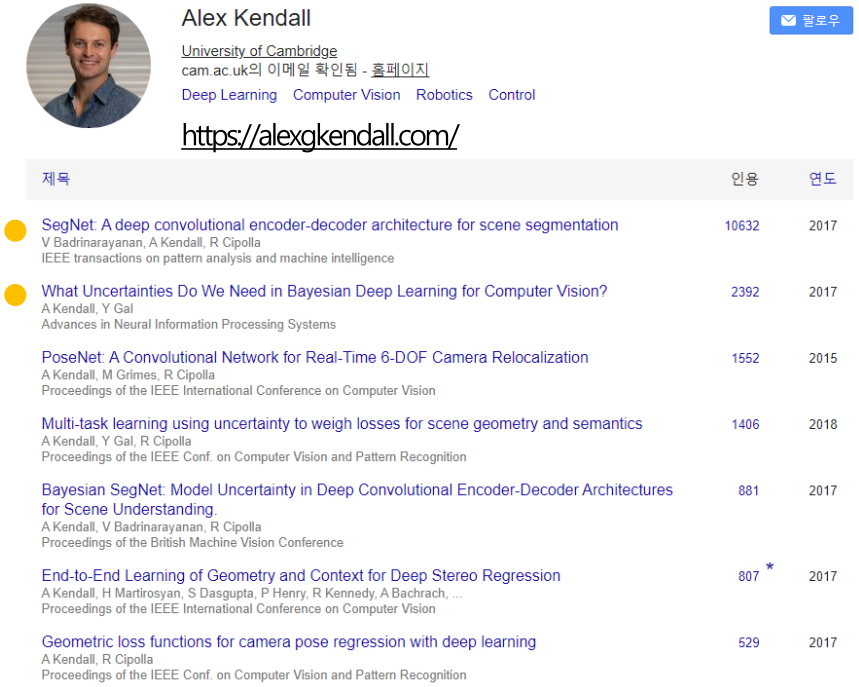
1 Introduction

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.



Alex Kendall

팔로우

University of Cambridge
cam.ac.uk의 이메일 확인됨 - 홈페이지
Deep Learning Computer Vision Robotics Control

https://alexgkendall.com/

제목	인용	연도
SegNet: A deep convolutional encoder-decoder architecture for scene segmentation V Badrinarayanan, A Kendall, R Cipolla IEEE transactions on pattern analysis and machine intelligence	10632	2017
What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? A Kendall, Y Gal Advances in Neural Information Processing Systems	2392	2017
PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization A Kendall, M Grimes, R Cipolla Proceedings of the IEEE International Conference on Computer Vision	1552	2015
Multi-task learning using uncertainty to weigh losses for scene geometry and semantics A Kendall, Y Gal, R Cipolla Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition	1406	2018
Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. A Kendall, V Badrinarayanan, R Cipolla Proceedings of the British Machine Vision Conference	881	2017
End-to-End Learning of Geometry and Context for Deep Stereo Regression A Kendall, H Martirosyan, S Dasgupta, P Henry, R Kennedy, A Bachrach, ... Proceedings of the IEEE International Conference on Computer Vision	807 *	2017
Geometric loss functions for camera pose regression with deep learning A Kendall, R Cipolla Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition	529	2017

Kendall, Alex, and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?." *NIPS*. 2017.

Introduction

Types of Uncertainty

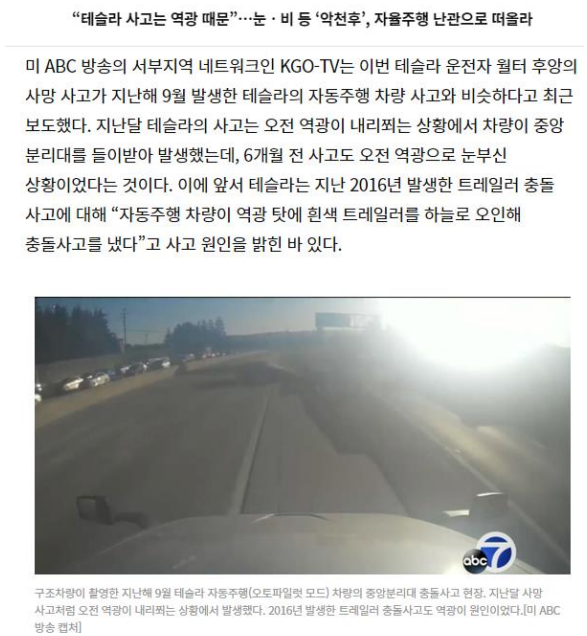
❖ Epistemic uncertainty (model uncertainty)

- 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)
- 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- 측정 정밀도를 높이면 줄일 수 있음



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

미 ABC 방송의 서부지역 네트워크인 KGO-TV는 이번 테슬라 운전자 웰터 후앙의 사망 사고가 지난해 9월 발생한 테슬라의 자동주행 차량 사고와 비슷하다고 최근 보도했다. 지난달 테슬라의 사고는 오전 역광이 내리쬐는 상황에서 차량이 중앙 분리대를 들이받아 발생했는데, 6개월 전 사고도 오전 역광으로 눈부신 상황이었다는 것이다. 이에 앞서 테슬라는 지난 2016년 발생한 트레일러 충돌 사고에 대해 "자동주행 차량이 역광 탓에 흰색 트레일러를 하늘로 오인해 충돌사고를 냈다"고 사고 원인을 밝힌 바 있다.

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]

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?. *Advances in Neural Information Processing Systems*, *30*, 5574-5584.

Introduction

Types of Uncertainty

❖ Aleatoric uncertainty (data uncertainty)

- 데이터에 내재된 노이즈로 인해 이해하지 못하는 정도

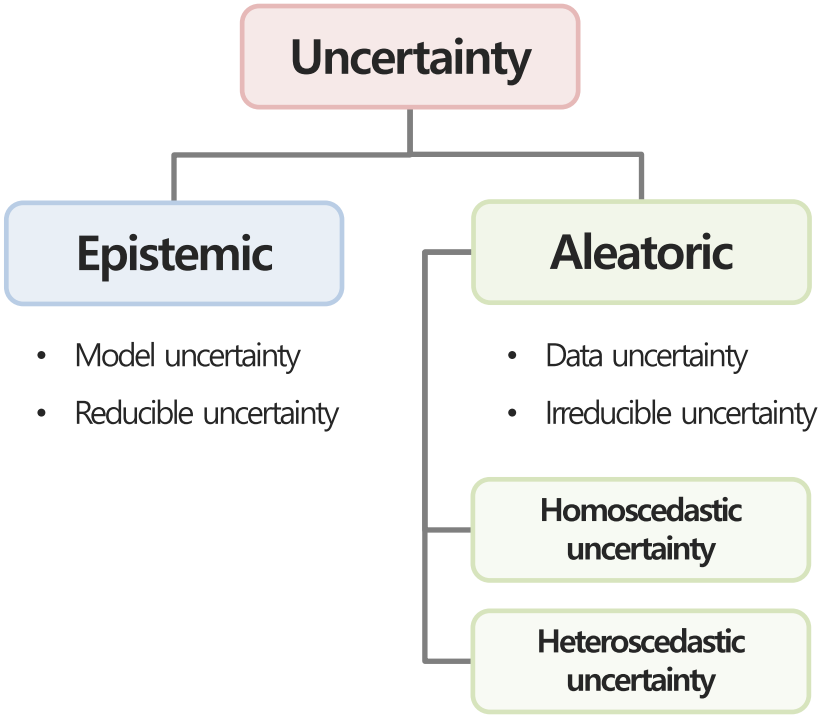
  (e.g. measurement noise, randomness inherent)
- 더 많은 데이터가 학습되더라도 줄일 수 없음, irreducible uncertainty
- 측정 정밀도를 높이면 줄일 수 있음

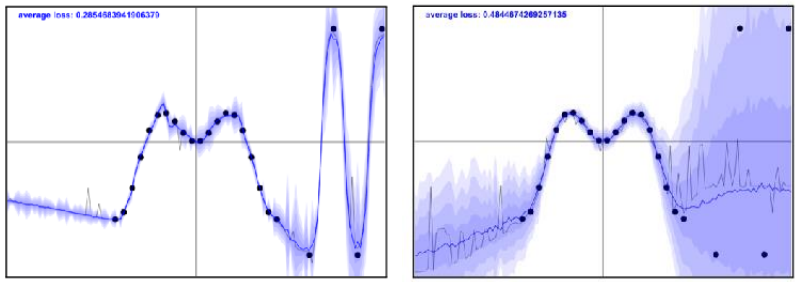
❖ Homoscedastic uncertainty

- 서로 다른 입력 값에 대해서 동일한 상수값을 지님

❖ Heteroscedastic uncertainty

- 서로 다른 입력 값에 대해서 다른 값을 지님, input-dependent uncertainty
- 적절하게 정량화 되는 경우 outlier의 지표로 활용할 수 있음



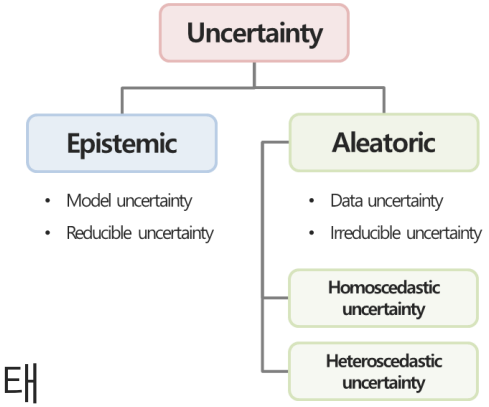


Homoscedastic uncertainty | Heteroscedastic uncertainty

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?. *Advances in Neural Information Processing Systems*, *30*, 5574-5584.

Bayesian-based Approach

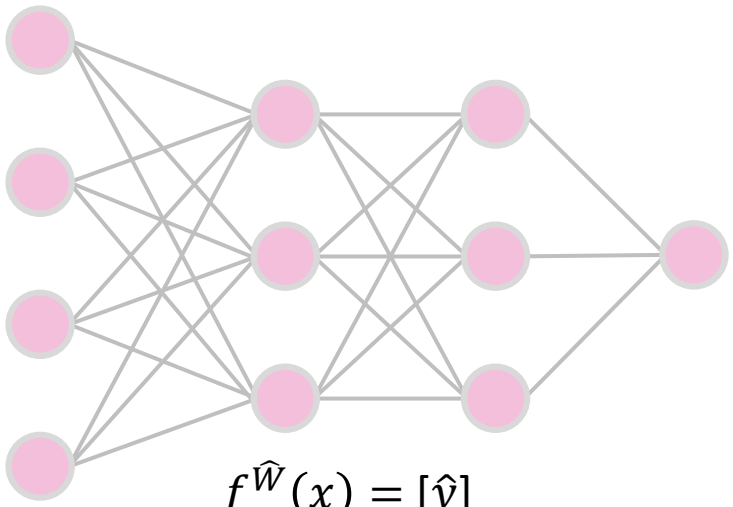
Bayesian Neural Networks for Computer Vision



❖ Density Network Architecture

- MC dropout구조와 유사하며, aleatoric uncertainty를 추정하기 위한 output node가 추가된 형태

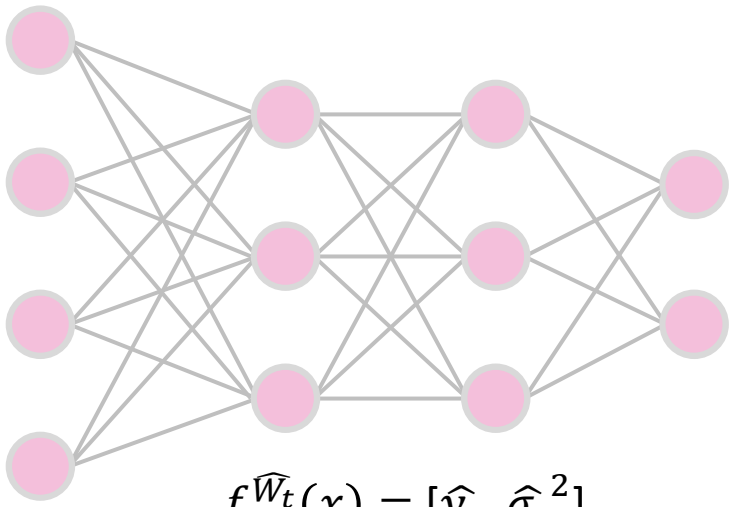
Standard Neural Network



$\hat{y}_{\hat{W}}(x)$

$f^{\hat{W}}(x) = [\hat{y}]$

Density Network



$\hat{y}_{t\,\hat{W}}(x)$

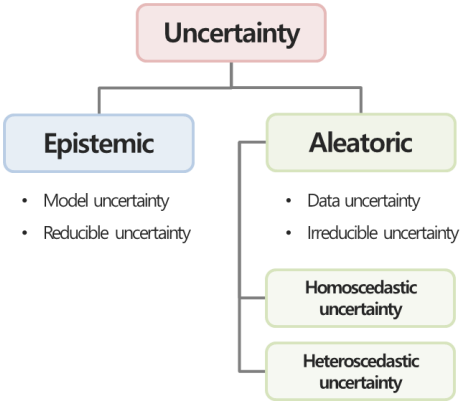
$\hat{\sigma}_t{}^2{}_{\hat{W}}(x)$

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

$\hat{W}_t \sim q^*(W)$

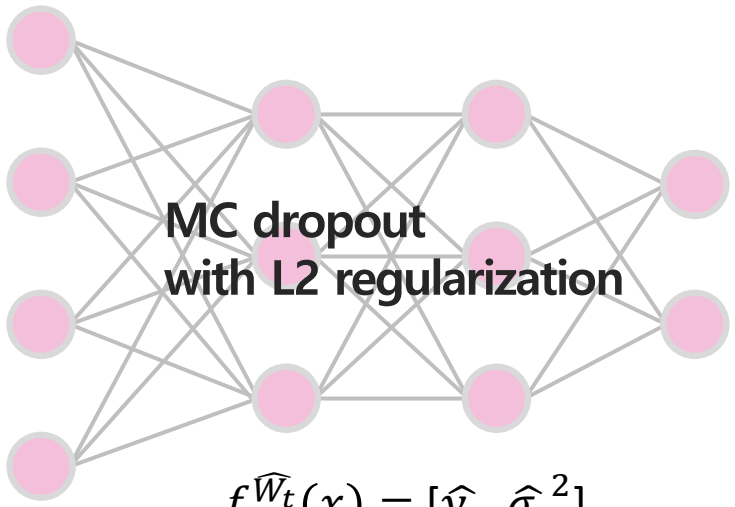
Bayesian-based Approach

Bayesian Neural Networks for Computer Vision



❖ Density Network Architecture

- Loss attenuation: heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축
- 불확실성이 큰 입력에 대해서는 loss에 영향을 적게 반영하기 위한 가중치를 적용



Epistemic uncertainty

$\hat{y}_{t\,\widehat{W}}(x)$

$\hat{\sigma}_t^{\,2}{}_{\widehat{W}}(x)$

Aleatoric uncertainty

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

$$\widehat{W}_t \sim q^*(W)$$

After T stochastic forward passes

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

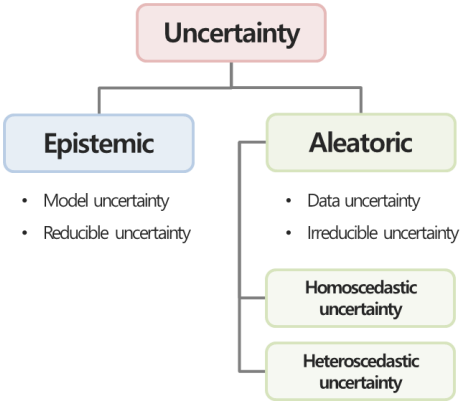
Prediction

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

Total uncertainty | Epistemic uncertainty | Aleatoric uncertainty

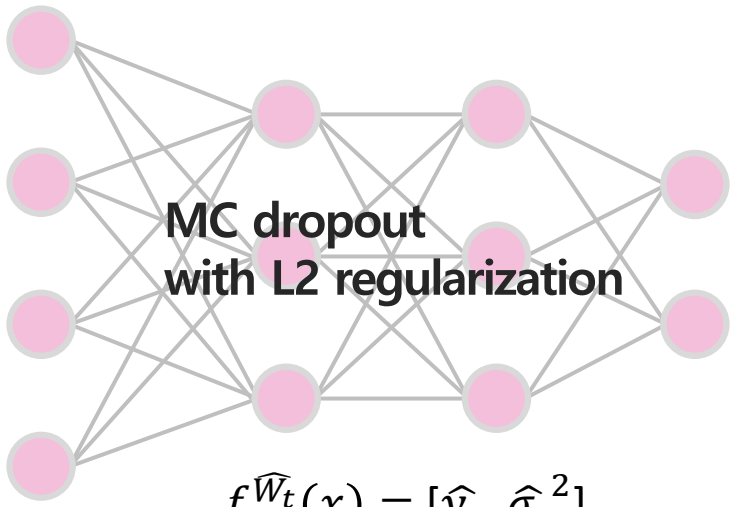
Bayesian-based Approach

Bayesian Neural Networks for Computer Vision



❖ Density Network Architecture

- Loss attenuation: heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축
- 불확실성이 큰 입력에 대해서는 loss에 영향을 적게 반영하기 위한 가중치를 적용



Epistemic uncertainty

$\hat{y}_{t\,\widehat{W}}(x)$

$\hat{\sigma}_t^{\,2}{}_{\widehat{W}}(x)$

Aleatoric uncertainty

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

$$\widehat{W}_t \sim q^*(W)$$

Heteroscedastic uncertainty as learned loss attenuation

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

MSE

Residual's weight

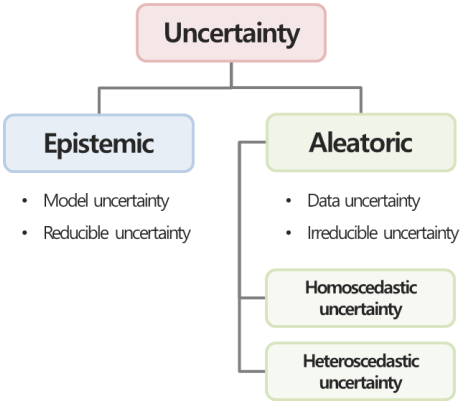
Uncertainty regularization

- Residual's weight
  Aleatoric heteroscedastic uncertainty가 큰 예측 값에 대해서는 loss(residual)를 적게 반영

- Uncertainty regularization
  Aleatoric uncertainty가 모든 데이터에 대해 무한히 커지는 것을 제약

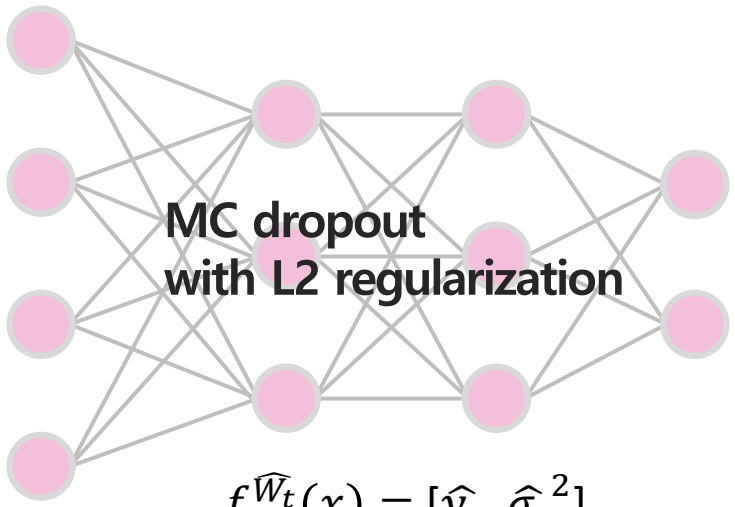
Bayesian-based Approach

Bayesian Neural Networks for Computer Vision



❖ Density Network Architecture

- Loss attenuation: heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축
- 불확실성이 큰 입력에 대해서는 loss에 영향을 적게 반영하기 위한 가중치를 적용



Epistemic uncertainty

$\hat{y}_{t\,\widehat{W}}(x)$

$\hat{\sigma}_t^{\,2}{}_{\widehat{W}}(x)$

Aleatoric uncertainty

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

$$\widehat{W}_t \sim q^*(W)$$

Heteroscedastic uncertainty as learned loss attenuation

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

MSE

Residual's weight

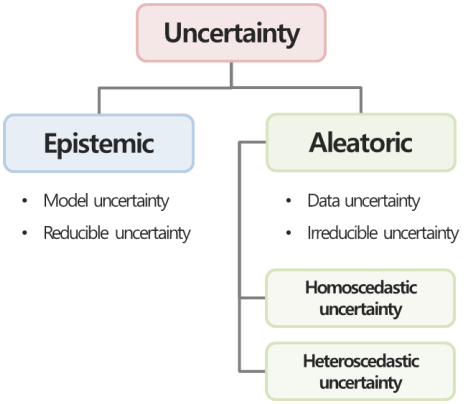
Uncertainty regularization

- Residual's weight
  Aleatoric heteroscedastic uncertainty가 큰 예측 값에 대해서는 loss(residual)를 적게 반영

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

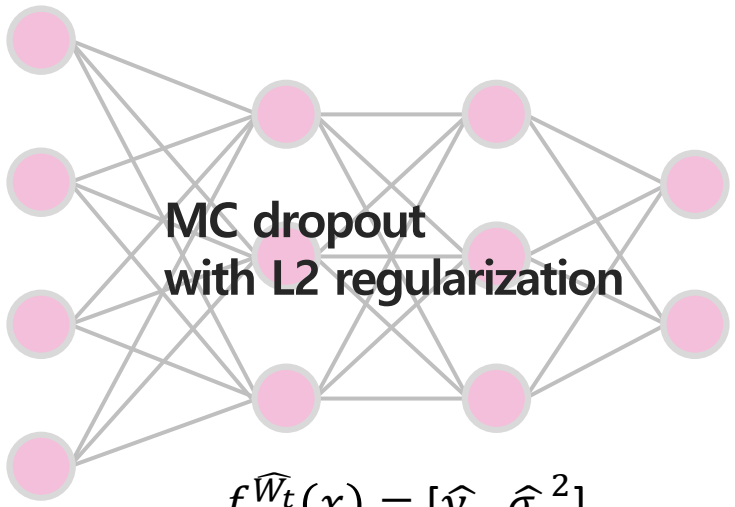
Bayesian-based Approach

Bayesian Neural Networks for Computer Vision



❖ Density Network Architecture

- Loss attenuation: heteroscedastic uncertainty 반영하여 더욱 강건한 모델 구축
- 불확실성이 큰 입력에 대해서는 loss에 영향을 적게 반영하기 위한 가중치를 적용



Epistemic uncertainty

$\hat{y}_{t\,\widehat{W}}(x)$

$\hat{\sigma}_t^{\,2}{}_{\widehat{W}}(x)$

Aleatoric uncertainty

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

$$\widehat{W}_t \sim q^*(W)$$

Heteroscedastic uncertainty as learned loss attenuation

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

MSE

Residual's weight

Uncertainty regularization

- Residual's weight
  Aleatoric heteroscedastic uncertainty가 큰 예측 값에 대해서는 loss(residual)를 적게 반영

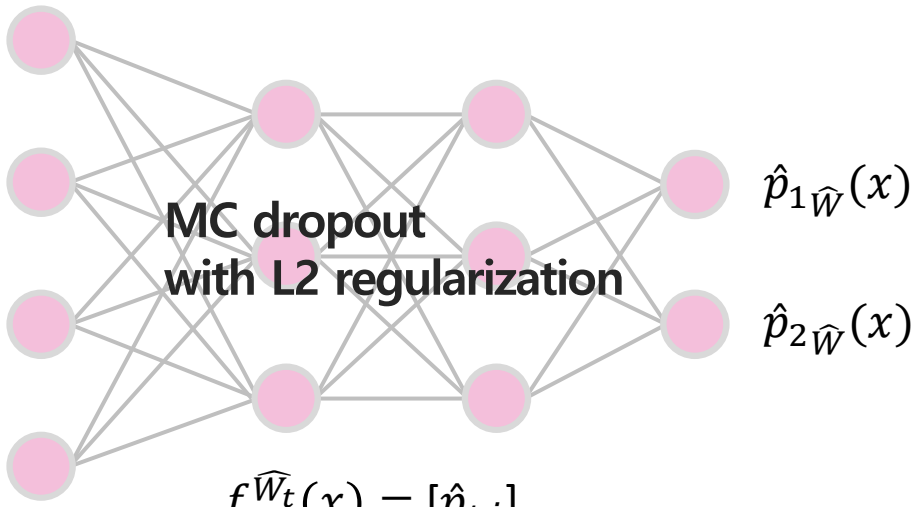
- 노이즈가 적은 데이터(낮은 heteroscedastic uncertainty가 예측된 값)에 대해서는 loss에 크게 반영

Bayesian-based Approach

Bayesian Neural Networks for Computer Vision

❖ Density Network Architecture for classification

- 후속 연구들에 classification에 적합하도록 수식을 정리하고자 하는 시도가 있음
- $\hat{\sigma}^2 = \hat{p}(1-\hat{p}) = \hat{p} - \hat{p}^2$ 으로 도출할 수 있음을 활용하여 별도의 node로 구분하지 않음
  Aleatoric uncertainty



$$f^{\widehat{W}_t}(x) = [\hat{p}_{t,i}]$$

$$\widehat{W}_t \sim q^*(W)$$

After T stochastic forward passes

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

Total uncertainty Epistemic uncertainty Aleatoric uncertainty

$$Var(y^*) \approx \frac{1}{T}\sum_{t=1}^{T} \hat{p}_{t,i}{}^2 - \left(\frac{1}{T}\sum_{t=1}^{T} \hat{p}_{t,i}\right)^2 + \frac{1}{T}\sum_{t=1}^{T} \hat{p}_{t,i} - \hat{p}_{t,i}{}^2$$

Total uncertainty Epistemic uncertainty Aleatoric uncertainty

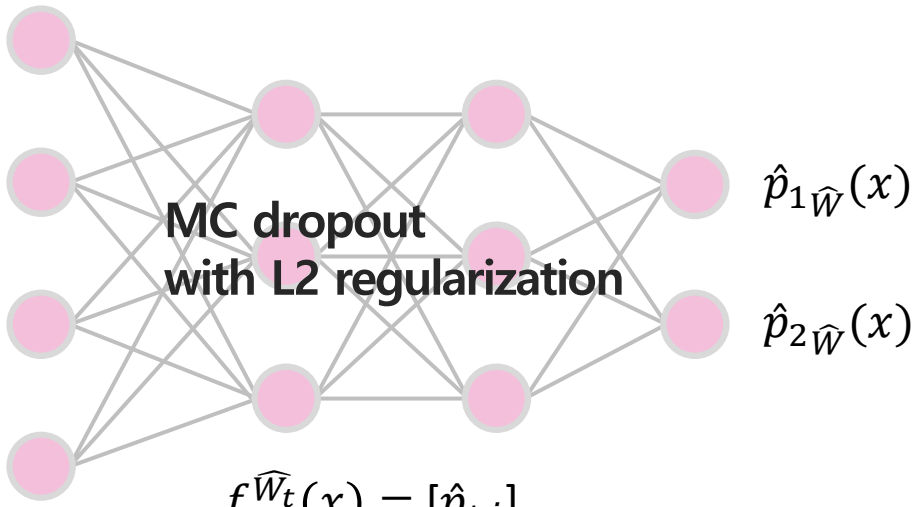
Deodato, Giacomo, Christopher Ball, and Xian Zhang. "Bayesian neural networks for cellular image classification and uncertainty analysis." bioRxiv (2020): 824862.

Bayesian-based Approach

Bayesian Neural Networks for Computer Vision

❖ Density Network Architecture for classification

- 후속 연구들에 classification에 적합하도록 수식을 정리하고자 하는 시도가 있음
- $\hat{\sigma}^2 = \hat{p}(1-\hat{p}) = \hat{p} - \hat{p}^2$ 으로 도출할 수 있음을 활용하여 별도의 node로 구분하지 않음
  Aleatoric uncertainty



$$f^{\widehat{W}_t}(x) = [\hat{p}_{t,i}]$$

$$\widehat{W}_t \sim q^*(W)$$

After T stochastic forward passes

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

Total uncertainty Epistemic uncertainty Aleatoric uncertainty

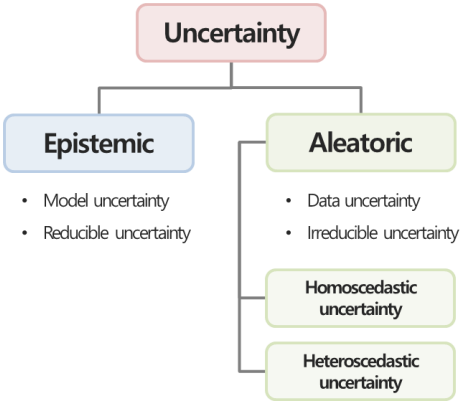
$$Var(y^*) \approx \frac{1}{T}\sum_{t=1}^{T}(\hat{p}_{t,i} - \bar{p}_{t,i})^2 + \frac{1}{T}\sum_{t=1}^{T}\hat{p}_{t,i} - \hat{p}_{t,i}{}^2$$

Total uncertainty Epistemic uncertainty Aleatoric uncertainty

Deodato, Giacomo, Christopher Ball, and Xian Zhang. "Bayesian neural networks for cellular image classification and uncertainty analysis." bioRxiv (2020): 824862.

Bayesian-based Approach

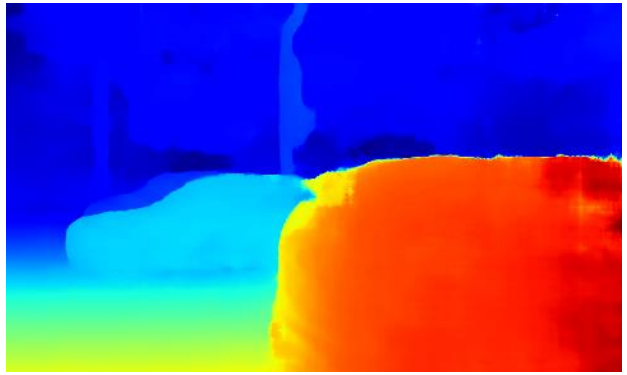
Bayesian Neural Networks for Computer Vision Results



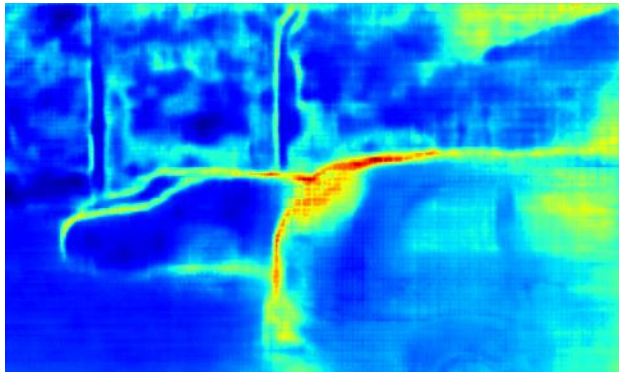
❖ Computer vision tasks

- Depth regression (regression task)
- Semantic segmentation (classification task)

Original image



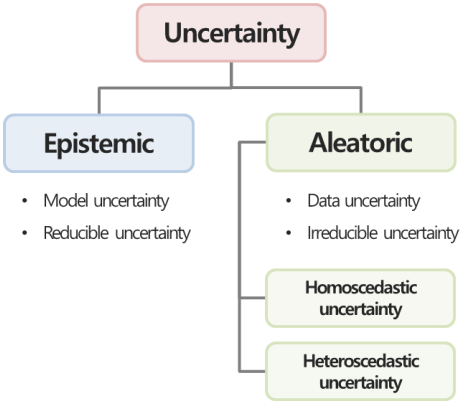
Depth regression



Semantic segmentation

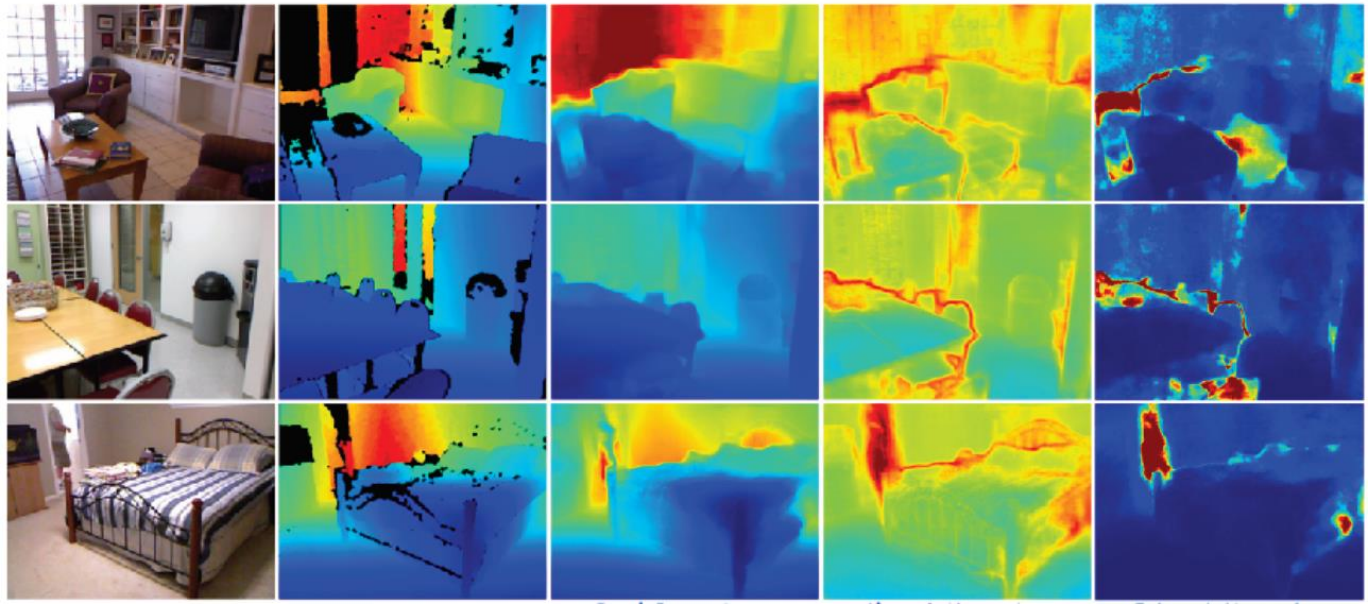
Bayesian-based Approach

Bayesian Neural Networks for Computer Vision Results



❖ Depth regression

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

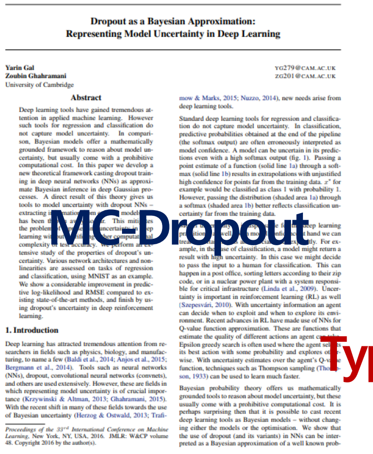


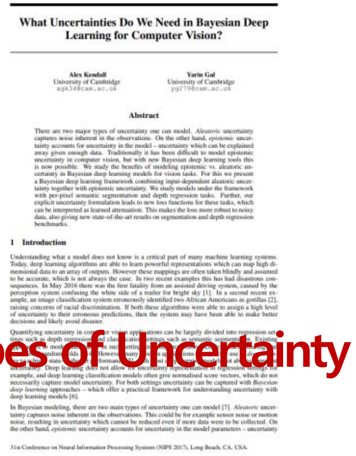
Input Image | Ground Truth | Depth Regression | Aleatoric Uncertainty | Epistemic Uncertainty

Uncertainty Quantification

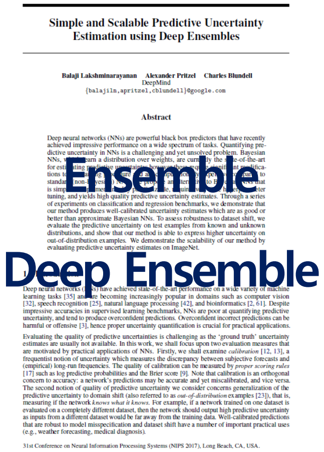
예측에 대한 불확실성 잘 정량화하자

Bayesian-based Approach





Ensemble-based Approach



GP-based Approach

Ensemble-based Approach

Simple and Scalable Deep Ensembles

❖ **NeurIPS 2017 (22년 1월 기준 2257건 인용)**

- 기존의 BNN의 경우, 모델 구조가 한정적, 계산량多
- Ensemble을 이용하여 간단하게 uncertainty 모델링

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.

Balaji Lakshminarayanan

팔로우

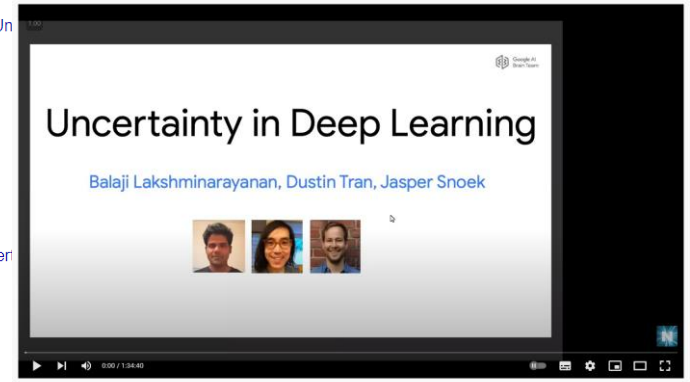
Staff Research Scientist at Google Brain
google.com의 이메일 확인됨 - 홈페이지

Machine Learning

http://www.gatsby.ucl.ac.uk/~balaji/

제목	인용	연도
Simple and scalable predictive uncertainty estimation using deep ensembles B Lakshminarayanan, A Pritzel, C Blundell Advances in Neural Information Processing Systems, 6393-6395	2257	2017
Clinically applicable deep learning for diagnosis and referral in retinal disease J De Fauw, JR Ledsam, B Romera-Paredes, S Nikolov, N Tomasev, ... Nature medicine 24 (9), 1342-1350	1283	2018
Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Un Shift Y Ovadia, E Fertig, J Ren, Z Nado, D Sculley, S Nowozin, JV Dillon, ... NeurIPS 2019		
Normalizing Flows for Probabilistic Modeling and Inference G Papamakarios, E Nalisnick, DJ Rezende, S Mohamed, ... Journal of Machine Learning Research 22 (57), 1-64		
Do Deep Generative Models Know What They Don't Know? E Nalisnick, A Matsukawa, YW Teh, D Gorur, B Lakshminarayanan ICLR 2019		
AugMix: A Simple Data Processing Method to Improve Robustness and Uncer D Hendrycks, N Mu, ED Cubuk, B Zoph, J Gilmer, B Lakshminarayanan ICLR 2020		
Learning in Implicit Generative Models S Mohamed, B Lakshminarayanan arXiv preprint arXiv:1610.03483		

NeurIPS 2020 Tutorials



Week 5 - Uncertainty and Out-of-Distribution Robustness in Deep Learning

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles." *Advances in Neural Information Processing Systems* 30 (2017).

Ensemble-based Approach

Simple and Scalable Deep Ensembles

❖ **NeurIPS 2017 (22년 1월 기준 2257건 인용)**

- 기존의 BNN의 경우, 모델 구조가 한정적, 계산량多 → 모델 구조의 제약 없애서 Simple
- Ensemble을 이용하여 간단하게 uncertainty 모델링 → 병렬연산을 수행하여 Scalable

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.

Balaji Lakshminarayanan

팔로우

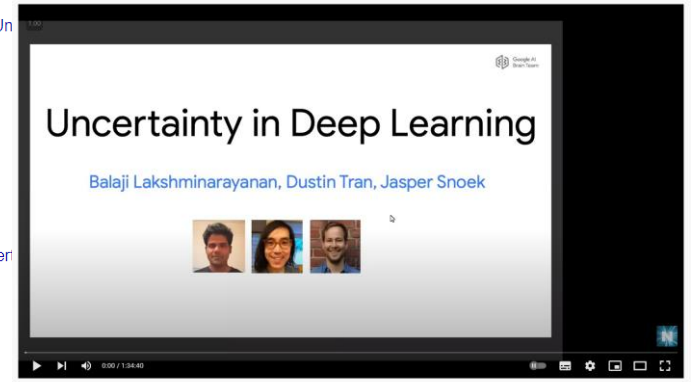
Staff Research Scientist at Google Brain
google.com의 이메일 확인됨 - 홈페이지

Machine Learning

http://www.gatsby.ucl.ac.uk/~balaji/

제목	인용	연도
Simple and scalable predictive uncertainty estimation using deep ensembles B Lakshminarayanan, A Pritzel, C Blundell Advances in Neural Information Processing Systems, 6393-6395	2257	2017
Clinically applicable deep learning for diagnosis and referral in retinal disease J De Fauw, JR Ledsam, B Romera-Paredes, S Nikolov, N Tomasev, ... Nature medicine 24 (9), 1342-1350	1283	2018
Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Un Shift Y Ovadia, E Fertig, J Ren, Z Nado, D Sculley, S Nowozin, JV Dillon, ... NeurIPS 2019		
Normalizing Flows for Probabilistic Modeling and Inference G Papamakarios, E Nalisnick, DJ Rezende, S Mohamed, ... Journal of Machine Learning Research 22 (57), 1-64		
Do Deep Generative Models Know What They Don't Know? E Nalisnick, A Matsukawa, YW Teh, D Gorur, B Lakshminarayanan ICLR 2019		
AugMix: A Simple Data Processing Method to Improve Robustness and Uncer D Hendrycks, N Mu, ED Cubuk, B Zoph, J Gilmer, B Lakshminarayanan ICLR 2020		
Learning in Implicit Generative Models S Mohamed, B Lakshminarayanan arXiv preprint arXiv:1610.03483		

NeurIPS 2020 Tutorials



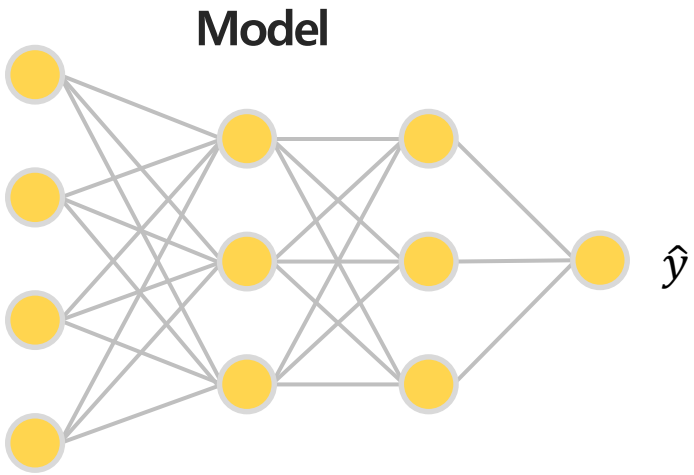
Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles." *Advances in Neural Information Processing Systems* 30 (2017).

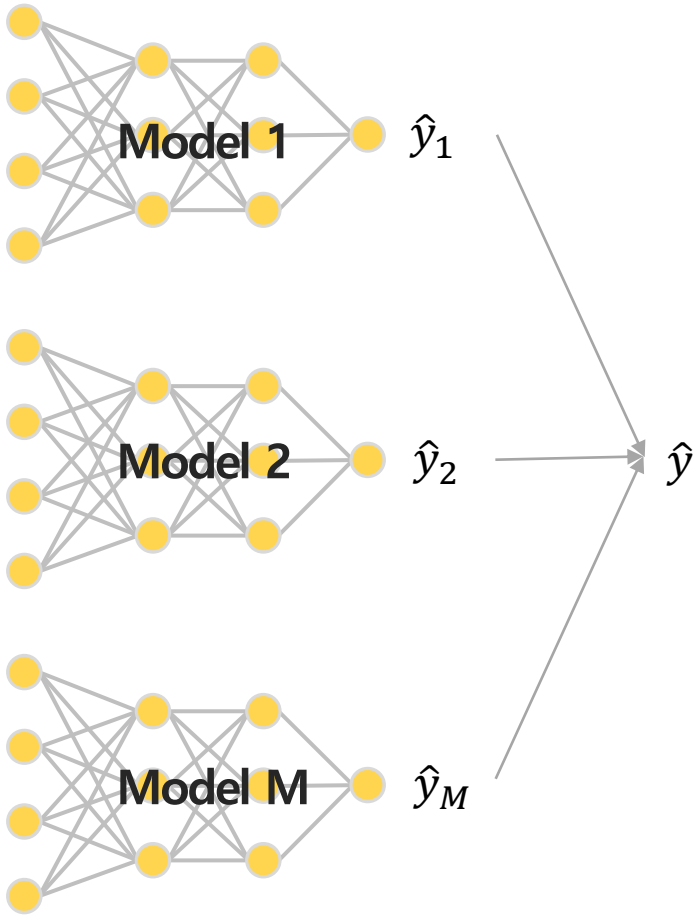
Ensemble-based Approach

Simple and Scalable Deep Ensembles

❖ Ensemble

- 다수의 결과를 종합하여 최종 예측을 수행
- 대표적으로 배깅(Bagging)과 부스팅(Boosting)으로 구분



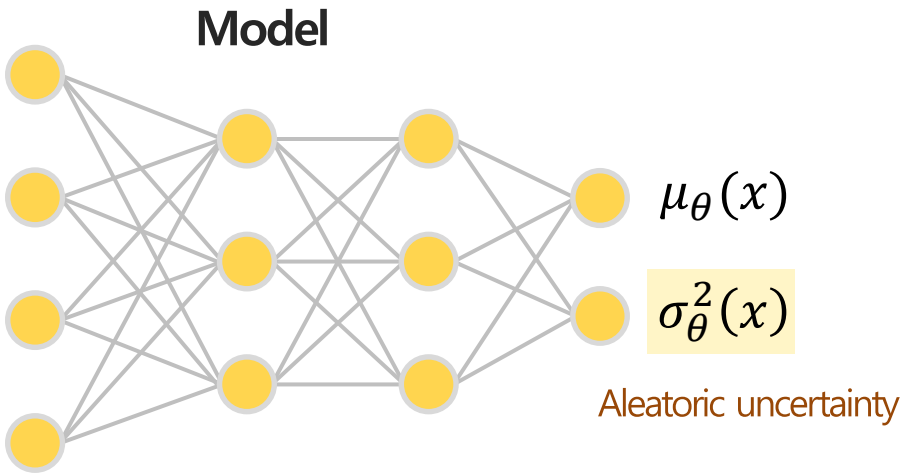


Ensemble-based Approach

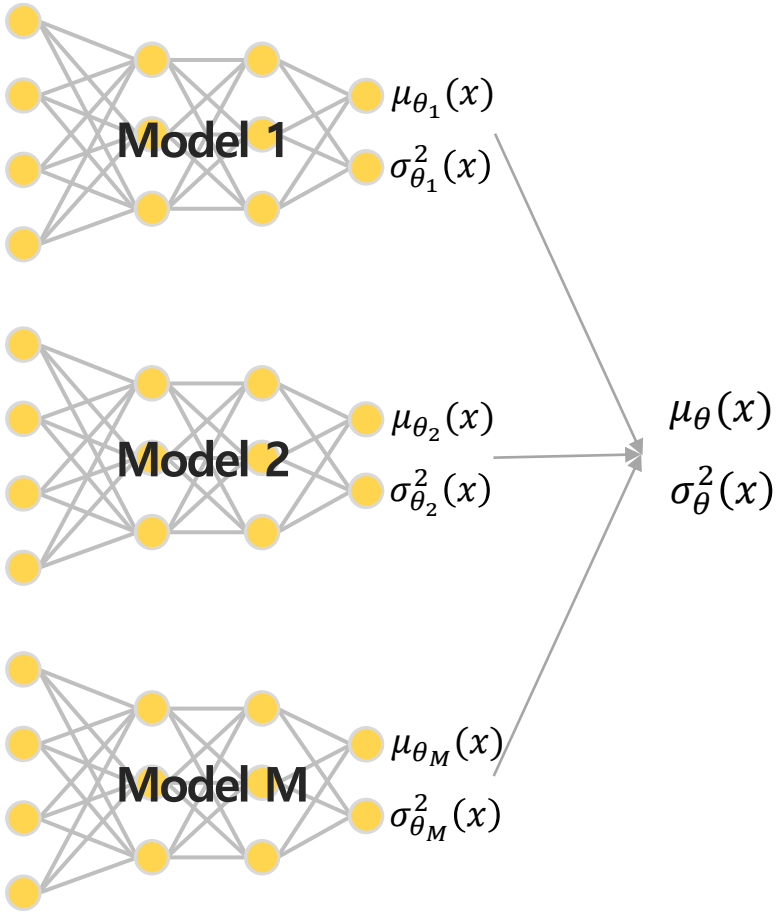
Simple and Scalable Deep Ensembles

❖ **Ensemble + Uncertainty (Aleatoric) = Deep Ensembles**

- 다수의 결과를 종합하여 최종 예측을 수행
- 대표적으로 배깅(Bagging)과 부스팅(Boosting)으로 구분



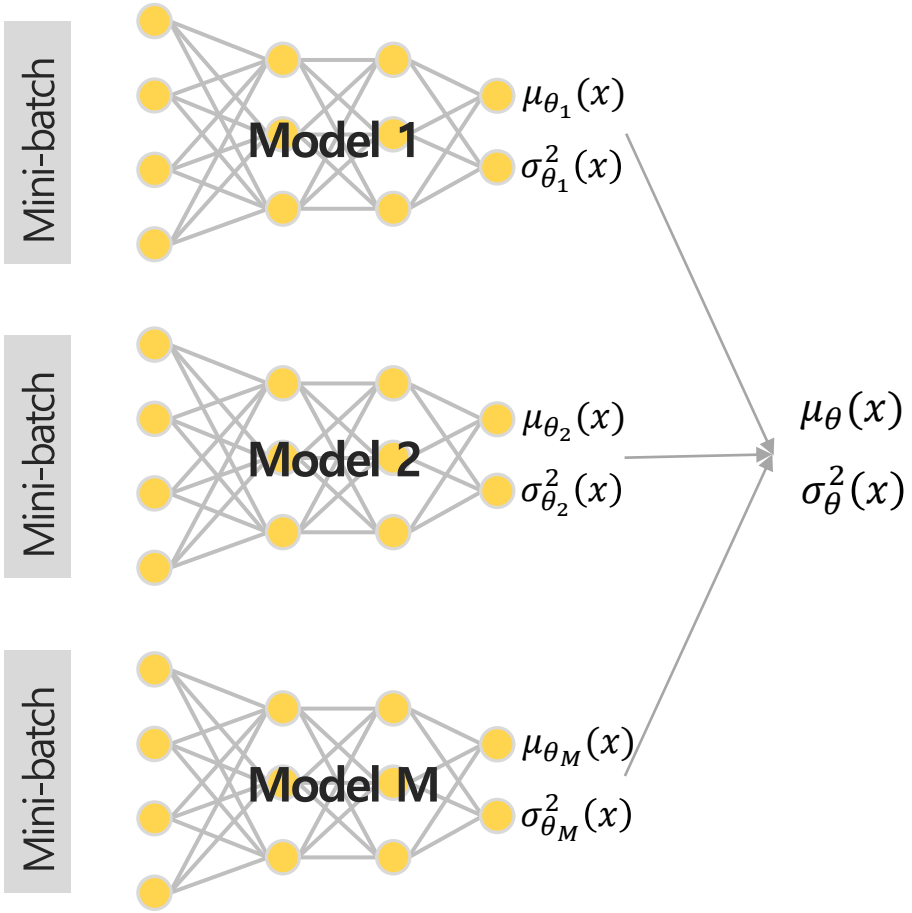
$$f_\theta(x) = [\,\mu_\theta(x), \sigma_\theta(x)\,]$$



Ensemble-based Approach

Simple and Scalable Deep Ensembles

❖ **Ensemble + Uncertainty (Aleatoric) = Deep Ensembles**



M 개의 mini-batch 구성하여 학습 수행 후,

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

Prediction

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

Uncertainty (Aleatoric uncertainty)

Ensemble-based Approach

Simple and Scalable Deep Ensembles

Heteroscedastic uncertainty as learned loss attenuation

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

MSE

Residual's weight

Uncertainty regularization

- Residual's weight
  Aleatoric heteroscedastic uncertainty가 큰 예측 값에 대해서는 loss(residual)를 적게 반영
- Uncertainty regularization
  Aleatoric uncertainty가 모든 데이터에 대해 무한히 커지는 것을 제약

❖ Proper scoring rules

- 모델의 학습 기준으로 scoring rule을 활용
- 일반적인 loss function (cross entropy, Brier score)은 scoring rule을 만족
- Regression에서는 불확실성 정량화가 가능한 새로운 scoring rule을 제안

Classification

- Brier score: 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 $[0.8, 0.2]$

Regression

- $\sigma_\theta^2(x)$ 을 반영하여 MSE 보정
- Negative Log−likelihood(NLL)

Uncertainty regularization

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

Residual's weight

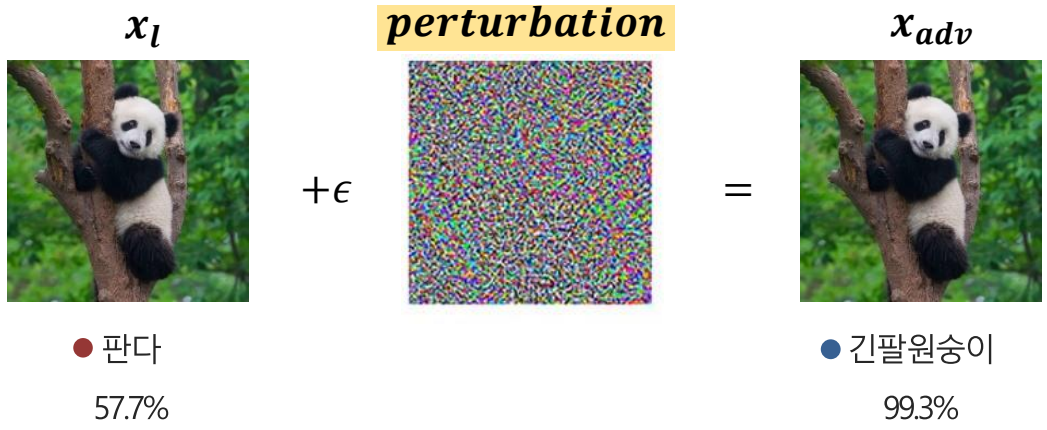
Ensemble-based Approach

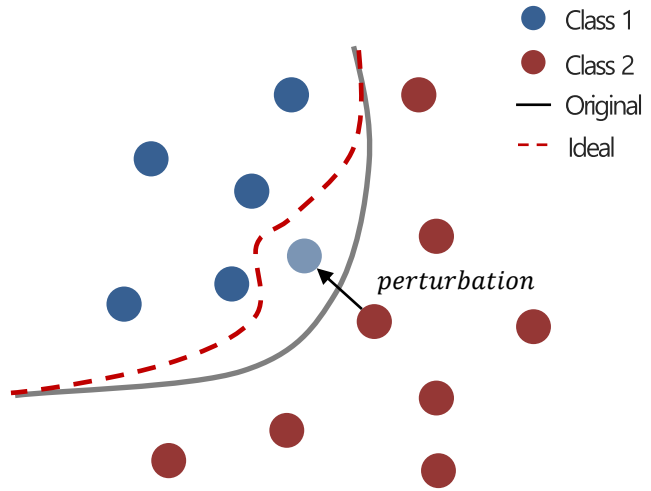
Simple and Scalable Deep Ensembles

❖ Adversarial Training

- Adversarial training은 adversarial example을 포함하여 모델의 과적합을 방지하는 기법
- Adversarial example은 일종의 augmentation기법
- 사람의 눈에는 동일해 보이지만, 모델은 헷갈려 하는 데이터를 perturbation을 더함으로써 생성함

손실(loss)을 최대화 하는 방향





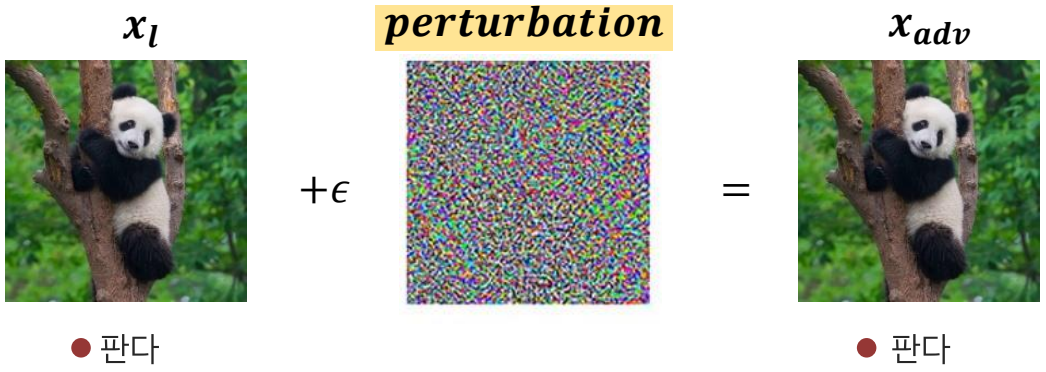
Ensemble-based Approach

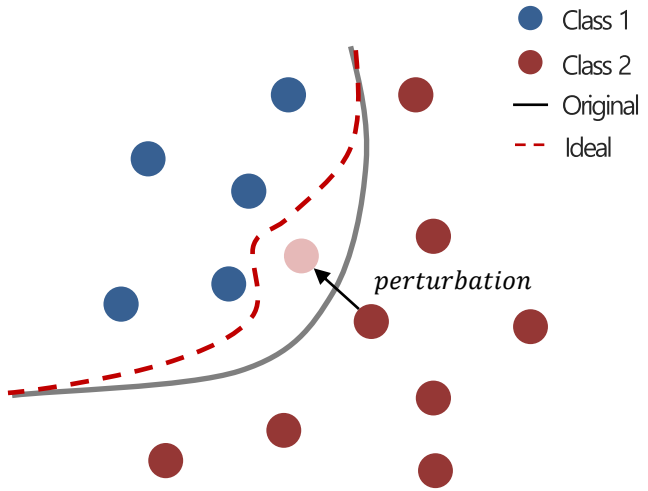
Simple and Scalable Deep Ensembles

❖ Adversarial Training

- Adversarial training은 adversarial example을 포함하여 모델의 과적합을 방지하는 기법
- Adversarial example은 일종의 augmentation기법
- 사람의 눈에는 동일해 보이지만, 모델은 헷갈려 하는 데이터를 perturbation을 더함으로써 생성함

손실(loss)을 최대화 하는 방향





Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

Ensemble-based Approach

Simple and Scalable Deep Ensembles

❖ Deep Ensemble 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 $\mathbf{x}'_{n_m} = \mathbf{x}_{n_m} + \epsilon \operatorname{sign}\left(\nabla_{\mathbf{x}_{n_m}} \ell(\theta_m, \mathbf{x}_{n_m}, y_{n_m})\right)$
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

Ensemble-based Approach

Simple and Scalable Deep Ensembles Results

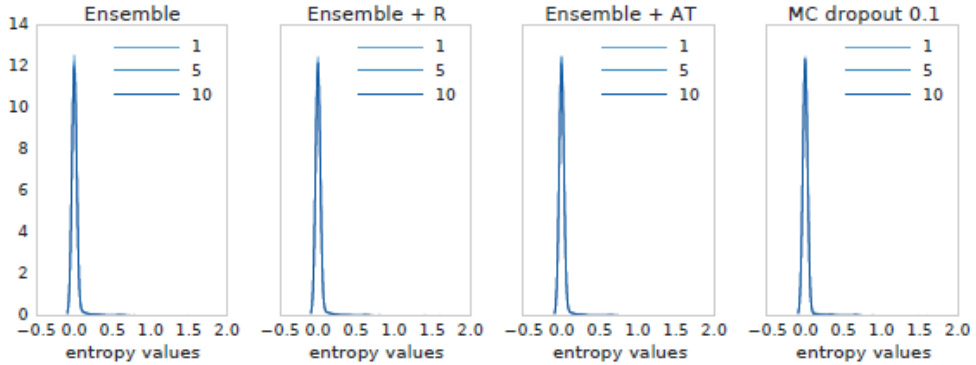
MNIST

Not-MNIST

❖ **학습에 활용하지 않은 Out-of-distribution데이터에 대한 불확실성 검증**

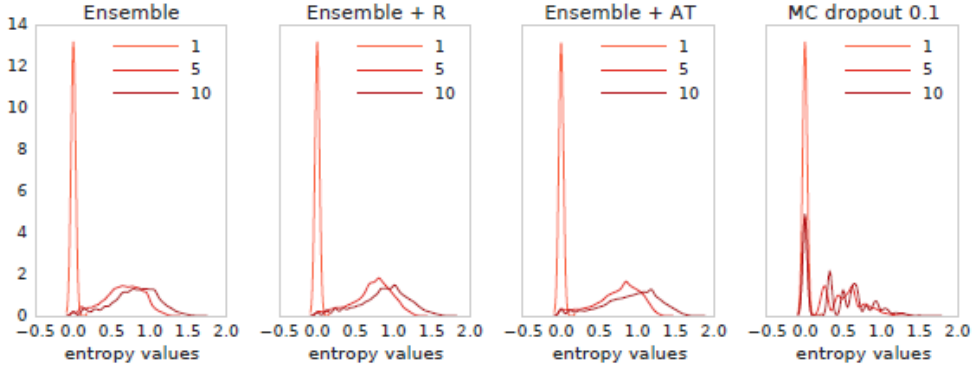
- 분류 문제상황에서 불확실성에 대한 지표로 entropy를 확인
- R: random noise / AT: adversarial training
- MC dropout (첫번째 논문과 비교)

Train: MNIST
Test: MNIST



Out-of-distribution

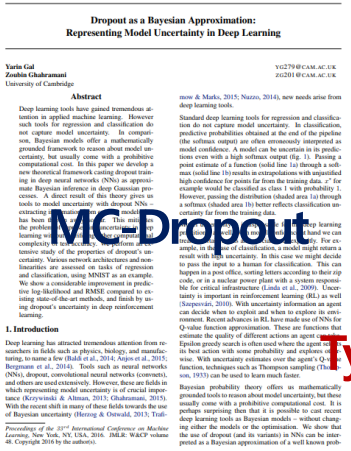
Train: MNIST
Test: Not-MNIST



Uncertainty Quantification

예측에 대한 불확실성 잘 정량화하자

Bayesian-based Approach


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

Yarin Gal
Zoubin Ghahramani
University of Cambridge

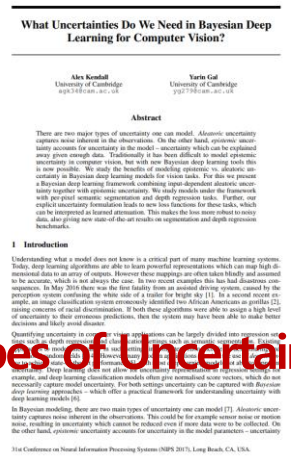
Abstract

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 learning without sacrificing either computational complexity or test accuracy. We perform an extensive study of the properties of dropout's uncertainty. Various network architectures and non-linearities are assessed on tasks of regression and classification, using MNIST as an example. We show a considerable improvement in predictive log-likelihood and RMSE compared to existing state-of-the-art methods, and finish by using dropout's uncertainty in deep reinforcement learning.

1. Introduction

Deep learning has attracted tremendous attention from researchers in fields such as physics, biology, and manufacturing, to name a few (Baldi et al., 2014; Anjos et al., 2015; Bergmann et al., 2014). Tools such as neural networks (NNs), dropout, convolutional neural networks (convnets), and others are used extensively. However, these are fields in which representing model uncertainty is of crucial importance (Krzywinski & Altman, 2013; Ghahramani, 2015). With the recent shift in many of these fields towards the use of Bayesian uncertainty (Herzog & Orrwall, 2013; Trafi-

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

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.

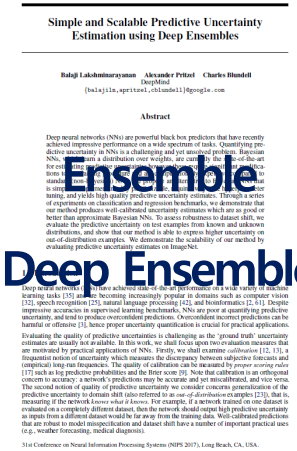
1 Introduction

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 such as depth regression and classification such as semantic segmentation. Existing

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

Ensemble-based Approach


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) NN training. 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.

Deep neural networks (NNs) have achieved state-of-the-art performance on a wide variety of supervised learning tasks 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 dataset 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.

GP-based Approach


Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness

Jeremiah Zhe Liu[1]
Google Research & Harvard University
jereliu@google.com

Zi Lin[2]
Google Research
lzi@google.com

Shreyas Padhy[3]
Google Research
shreyaspadhy@google.com

Dustin Tran
Google Research
trandustin@google.com

Tania Bedrax-Weiss
Google Research
tdweiss@google.com

Balaji Lakshminarayanan
Google Research
balajiln@google.com

Abstract

Bayesian neural networks and deep ensembles are principled approaches to estimate the predictive uncertainty of a deep learning model. However their practicality in real-time, industrial-scale applications are limited due to their heavy memory and inference cost. This motivates us to study principled approaches to high-quality uncertainty estimation that require only a single deep neural network (DNN). By formalizing the uncertainty quantification as a minimax learning problem, we first identify distance awareness, the model's ability to quantify the distance of a testing example to the training data, as a necessary condition for a DNN to achieve high-quality uncertainty estimation. We then propose spectral-normalized neural Gaussian process (SNGP), a simple method that improves the distance awareness of a deep residual network by adding a weight normalization step during training and replacing the output layer with a Gaussian process. On a suite of vision and language understanding tasks and on modern architectures (Wide-ResNet and BERT), SNGP is competitive with deep ensembles in prediction, calibration and out-of-domain detection, and outperforms the other single-model approaches.[1]

1 Introduction

Efficient methods that reliably quantify a deep neural network (DNN)'s predictive uncertainty are important for industrial-scale, real-world applications, which include examples such as object recognition in autonomous driving [22], ad click prediction in online advertising [76], and intent understanding in a conversational system [84]. For example, for a natural language understanding (NLU) model built for a domain-specific chatbot service (e.g. weather inquiry), the user's input utterance to the model can be of any topic, and the model needs to understand reliably and in real-time whether to abstain or to trigger one of its known APIs.

When deep classifiers make predictions on input examples that are far from the support of the training set, their performance can be arbitrarily bad [4, 14]. This motivates the need for methods that are aware of the distance between an input test example and previously seen training examples, so they can return a uniform (i.e., maximum entropy) distribution over output labels if the input is too far from the training set (i.e., the input is out-of-domain) [30]. Gaussian processes (GPs) with suitable kernels enjoy such a property. However, to apply Gaussian processes to a high-dimensional machine

[1]Work done at Google Research.
[2]Work done as an Google AI Resident.
[3]Code available at https://github.com/google/uncertainty-baselines/tree/master/baselines.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

MC-Dropout — Types of Uncertainty

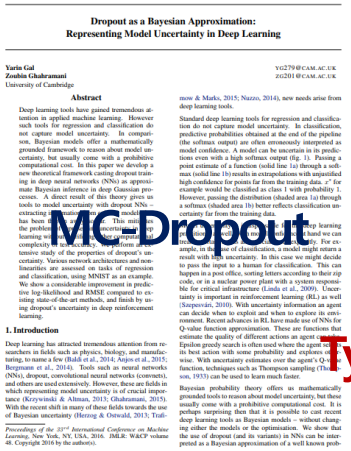
Ensemble — **Deep Ensemble**

SNGP

Uncertainty Quantification

예측에 대한 불확실성 잘 정량화하자

Bayesian-based Approach


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

Yarin Gal
Zoubin Ghahramani
University of Cambridge

YG279@CAM.AC.UK
ZG201@CAM.AC.UK

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 learning without sacrificing either computational complexity or test accuracy. We perform an extensive study of the properties of dropout's uncertainty. Various network architectures and non-linearities are assessed on tasks of regression and classification, using MNIST as an example. We show a considerable improvement in predictive log-likelihood and RMSE compared to existing state-of-the-art methods, and finish by using dropout's uncertainty in deep reinforcement learning.

1. Introduction

Deep learning has attracted tremendous attention from researchers in fields such as physics, biology, and manufacturing, to name a few (Baldi et al., 2014; Anjos et al., 2015; Bergmann et al., 2014). Tools such as neural networks (NNs), dropout, convolutional neural networks (convnets), and others are used extensively. However, these are fields in which representing model uncertainty is of crucial importance (Krzywinski & Altman, 2013; Ghahramani, 2015). With the recent shift in many of these fields towards the use of Bayesian uncertainty (Herzog & Orwault, 2013; Trafi-

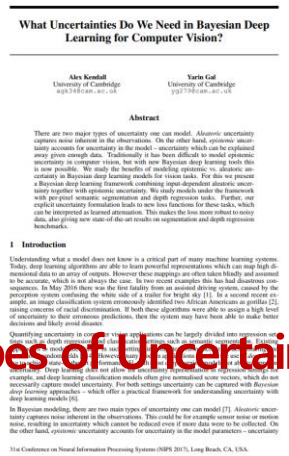
Proceedings of the 31^{st} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

now & Marks, 2015; Nazzo, 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.

Bayesian probability theory offers us mathematically grounded tools to reason about model uncertainty, but usually come with a prohibitive computational cost. It is perhaps surprising then that it is possible to cast recent deep learning tools as Bayesian models – without changing either the models or the optimisation. We show that the use of dropout (and its variants) in NNs can be interpreted as a Bayesian approximation of a well known prob-

MC Dropout


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 things such as depth regression and classification things such as semantic segmentation. Existing

its best action with some probability and explores it

wise. With uncertainty estimates over the agent's Q

function, techniques such as Thompson sampling (Thom-

son, 1933) can be used to learn much faster.

Bayesian probability theory offers us mathematically grounded tools to reason about model uncertainty, but usually come with a prohibitive computational cost. It is perhaps surprising then that it is possible to cast recent deep learning tools as Bayesian models – without changing either the models or the optimisation. We show that the use of dropout (and its variants) in NNs can be interpreted as a Bayesian approximation of a well known prob-

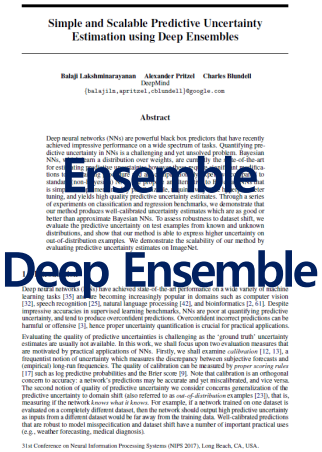
31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Types of Uncertainty

Epistemic

Epistemic+Aleatoric

Ensemble-based Approach


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) NN training. 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.

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.

Ensemble

Deep Ensemble

Aleatoric

GP-based Approach


Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness

Jeremiah Zhe Liu[1]
Google Research & Harvard University
jeremiah@google.com

Zi Lin[2]
Google Research
lzi@google.com

Shreyas Padhy[3]
Google Research
shreyaspadhy@google.com

Dustin Tran
Google Research
trandustin@google.com

Tania Bedrax-Weiss
Google Research
tdedrax@google.com

Balaji Lakshminarayanan
Google Research
balajiln@google.com

Abstract

Bayesian neural networks and deep ensembles are principled approaches to estimate the predictive uncertainty of a deep learning model. However their practicality in real-time, industrial-scale applications are limited due to their heavy memory and inference cost. This motivates us to study principled approaches to high-quality uncertainty estimation that require only a single deep neural network (DNN). By formalizing the uncertainty quantification as a minimax learning problem, we first identify *distance awareness*, the model's ability to quantify the distance of a testing example to the training data, as a necessary condition for a DNN to achieve high-quality uncertainty estimation. We then propose the Spectral-normalized Neural Gaussian Process (SNGP), a simple method that improves the distance awareness of a deep residual network by adding a weight normalization step during training and replacing the output layer with a Gaussian Process. On a suite of vision and language understanding tasks and on modern architectures (Wide-ResNet and BERT), SNGP is competitive with deep ensembles in prediction, calibration and out-of-domain detection, and outperforms the other single-model approaches.[1]

1 Introduction

Efficient methods that reliably quantify a deep neural network (DNN)'s predictive uncertainty are important for industrial-scale, real-world applications, which include examples such as object recognition in autonomous driving [22], ad click prediction in online advertising [76], and intent understanding in a conversational system [84]. For example, for a natural language understanding (NLU) model built for a domain-specific chatbot service (e.g. weather inquiry), the user's input utterance to the model can be of any topic, and the model needs to understand reliably and in real-time whether to abstain or to trigger one of its known APIs.

When deep classifiers make predictions on input examples that are far from the support of the training set, their performance can be arbitrarily bad [4, 14]. This motivates the need for methods that are aware of the distance between an input test example and previously seen training examples, so they can return a uniform (i.e., maximum entropy) distribution over output labels if the input is too far from the training set (i.e., the input is out-of-domain) [30]. Gaussian processes (GPs) with suitable kernels enjoy such a property. However, to apply Gaussian processes to a high-dimensional machine

[1]Work done at Google Research.
[2]Work done as an Google AI Resident.
[3]Code available at https://github.com/google/uncertainty-baselines/tree/master/baselines.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

SNGP

Out-of-distribution
(calibrated probability)

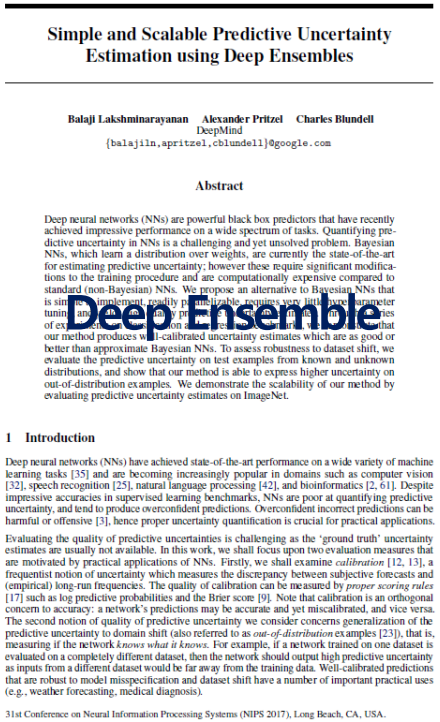
GP-based Approach

Spectral-normalized Neural Gaussian Process

[1] Liu, Jeremiah Zhe, et al. "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness." *arXiv preprint arXiv:2006.10108* (2020).
[2] Abdar, Moloud, et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges." *Information Fusion* (2021).
[3] Nado, Zachary, et al. "Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning." *arXiv preprint arXiv:2106.04015* (2021).

❖ NeurIPS 2020 Tutorials (22년 1월 기준 68건 인용)

- 최근 Uncertainty quantification 리뷰 문헌[2], Benchmark 정리 문헌[3] 에서 SNGP가 주로 언급
- 거리를 반영하여 Gaussian process를 설계한 방법론



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 experimental evaluations on various classification benchmarks, we show 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.

Deep Ensemble

Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness

Jeremiah Zhe Liu* Google Research & Harvard University jereliu@google.com
Zi Lin† Google Research lzi@google.com
Shreyas Padhy† Google Research shreyaspadhy@google.com
Dustin Tran Google Research trandustin@google.com
Tania Bedrax-Weiss Google Research tbedrax@google.com
Balaji Lakshminarayanan Google Research balajiln@google.com

Abstract

Bayesian neural networks and deep ensembles are principled approaches to estimate the predictive uncertainty of a deep learning model. However their practicality in real-time, industrial-scale applications are limited due to their heavy memory and inference cost. This motivates us to study principled approaches to high-quality uncertainty estimation that require only a single deep neural network (DNN). By formalizing the uncertainty quantification as a minimax learning problem, we first identify *distance awareness*, i.e., the model's ability to properly quantify the distance of a testing example from the training data manifold, as a necessary condition for a DNN to achieve high-quality (i.e., minimax optimal) uncertainty estimation. We then propose *Spectral-normalized Neural Gaussian Process (SNGP)*, a simple method that improves the distance-awareness ability of modern DNNs, by adding a weight normalization step during training and replacing the output layer with a Gaussian Process. On a suite of vision and language understanding tasks and on modern architectures (Wide-ResNet and BERT), SNGP is competitive with deep ensembles in prediction, calibration and out-of-domain detection, and outperforms the other single-model approaches.[3]

1 Introduction

Efficient methods that reliably quantify a deep neural network (DNN)'s predictive uncertainty are important for industrial-scale, real-world applications, which include examples such as object recognition in autonomous driving [22], ad click prediction in online advertising [76], and intent understanding in a conversational system [84]. For example, for a natural language understanding (NLU) model built for a domain-specific chatbot service (e.g, weather inquiry), the user's input utterance to the model can be of any topic, and the model needs to understand reliably and in real-time whether to abstain or to trigger one of its known APIs.

When deep classifiers make predictions on input examples that are far from the support of the training set, their performance can be arbitrarily bad [4, 14]. This motivates the need for methods that are aware of the distance between an input test example and previously seen training examples, so they can return a uniform (i.e., maximum entropy) distribution over output labels if the input is too far from the training set (i.e., the input is out-of-domain) [30]. Gaussian processes (GPs) with suitable kernels enjoy such a property. However, to apply Gaussian processes to a high-dimensional machine

*Work done at Google Research.
†Work done as an Google AI Resident
[3]Code available at https://github.com/google/uncertainty-baselines/tree/master/baselines.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Balaji Lakshminarayanan

Staff Research Scientist at Google Brain
google.com의 이메일 확인됨 - 홈페이지
Machine Learning

팔로우

제목	인용	연도
Simple and scalable predictive uncertainty estimation using deep ensembles B Lakshminarayanan, A Pritzel, C Blundell Advances in Neural Information Processing Systems, 6393-6395	2257	2017
Clinically applicable deep learning for diagnosis and referral in retinal disease J De Fauw, JR Ledsam, B Romera-Paredes, S Nikolov, N Tomasev, ... Nature medicine 24 (9), 1342-1350	1283	2018
Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift Y Ovadia, E Fertig, J Ren, Z Nado, D Sculley, S Nowozin, JV Dillon, ... NeurIPS 2019	586	2019
Normalizing Flows for Probabilistic Modeling and Inference G Papamakarios, E Nalisnick, DJ Rezende, S Mohamed, ... Journal of Machine Learning Research 22 (57), 1-64	381	2021
Do Deep Generative Models Know What They Don't Know? E Nalisnick, A Matsukawa, YW Teh, D Gorur, B Lakshminarayanan ICLR 2019	340	2019
AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty D Hendrycks, N Mu, ED Cubuk, B Zoph, J Gilmer, B Lakshminarayanan ICLR 2020	319	2020
Learning in Implicit Generative Models S Mohamed, B Lakshminarayanan arXiv preprint arXiv:1610.03483	309	2016

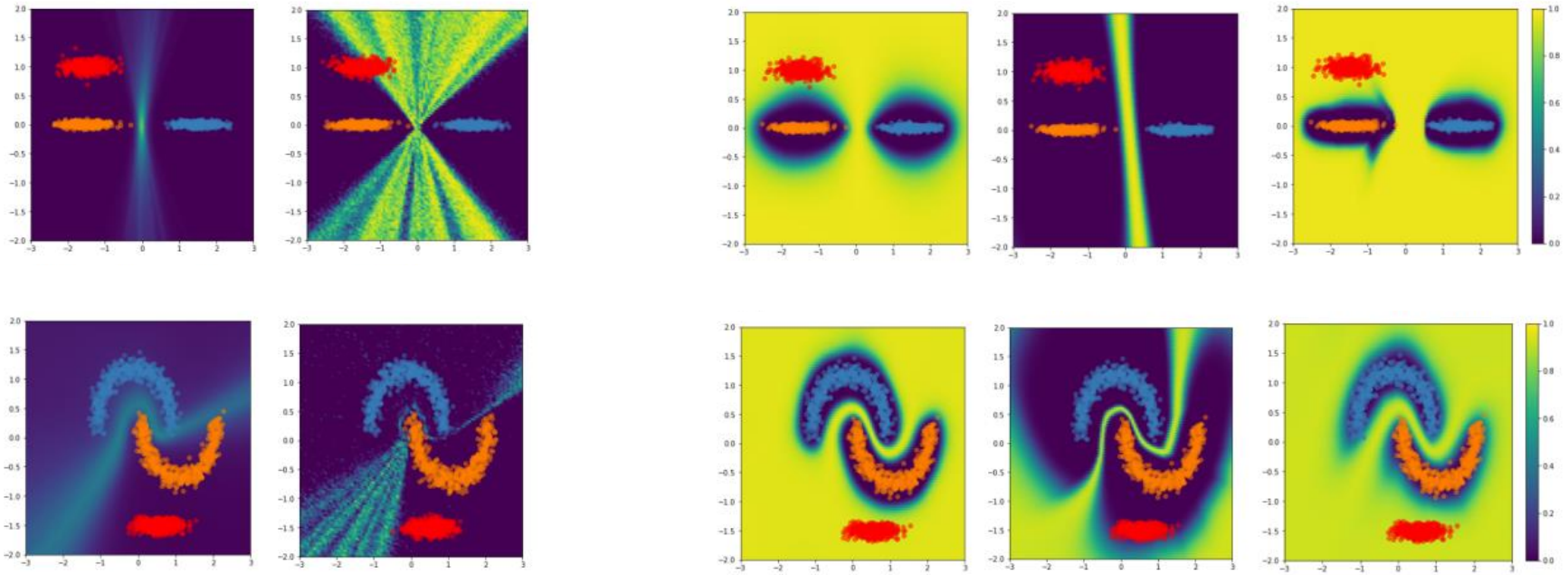
GP-based Approach

Spectral-normalized Neural Gaussian Process

❖ **불확실성 정량화 도메인에서 Bayesian-approach, Ensemble-approach 연구들이 대표적으로 수행**

- 하지만, 해당 방법론들은 out-of-distribution에 대해 적절한 불확실성 정량화가 되지 않는 한계가 있음
- Out-of-distribution: 학습에 사용한 데이터와는 '거리(차이)'가 있는 데이터

- Class 1
- Class 2
- Out-of-distribution



Deep Ensemble MC dropout Gaussian Process DNN-GP SNGP(proposed)

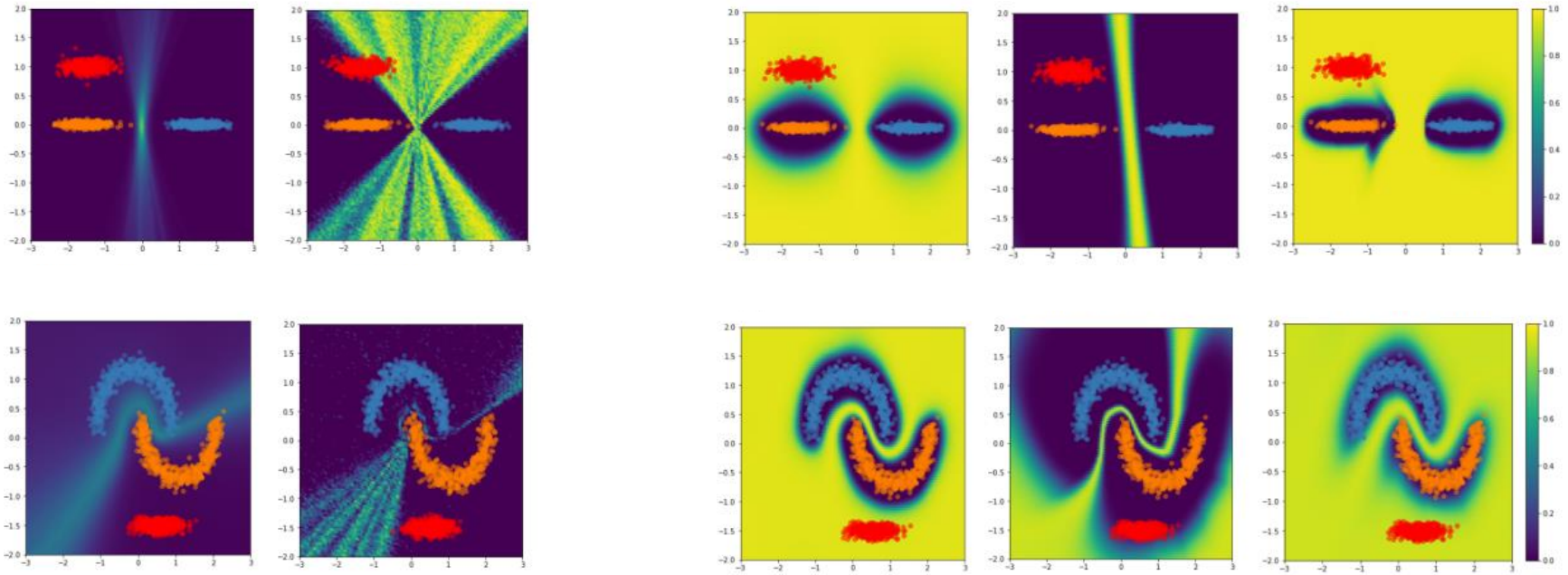
GP-based Approach

Spectral-normalized Neural Gaussian Process

❖ Gaussian Process를 효과적으로 구축하기 위한 접근

- GP는 불확실성 정량화가 적절하지만, 연산량이 매우 크므로 제약이 있으며 일반적으로는 차원축소 값이 입력에 활용
- GP layer를 활용하기 위해 '거리'정보가 유지되도록 차원축소를 적절히 수행하는 것이 중요

- Class 1
- Class 2
- Out-of-distribution



Deep Ensemble | MC dropout | Gaussian Process | DNN-GP | SNGP(proposed)

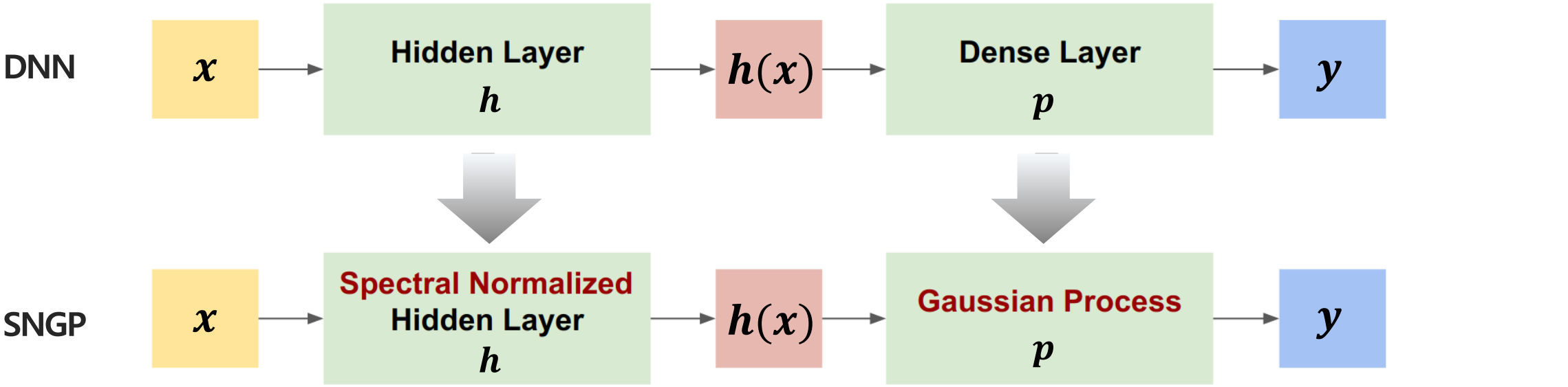
Distance - aware

GP-based Approach

Spectral-normalized Neural Gaussian Process

❖ **'거리'정보가 유지되도록 차원축소를 수행하여 Gaussian process layer 적용**

- GP는 불확실성 정량화가 적절하지만, 연산량이 매우 크므로 제약이 있으며 일반적으로는 차원축소 값이 입력에 활용
- GP layer를 활용하기 위해 '거리'정보가 유지되도록 차원축소를 적절히 수행하는 것이 중요



https://www.tensorflow.org/tutorials/understanding/sngp

GP-based Approach

Spectral-normalized Neural Gaussian Process

❖ **'거리'정보가 유지되도록 차원축소를 수행하여 Gaussian process layer 적용**

- GP는 불확실성 정량화가 적절하지만, 연산량이 매우 크므로 제약이 있으며 일반적으로는 차원축소 값이 입력에 활용
- GP layer를 활용하기 위해 '거리'정보가 유지되도록 차원축소를 적절히 수행하는 것이 중요

Algorithm 1 SNGP Training

Input:
Minibatches $\{D_i\}_{i=1}^{N}$ for $D_i = \{y_m, \mathbf{x}_m\}_{m=1}^{M}$.
Initialize:

$$\hat{\Sigma} = \mathbf{I}, \mathbf{W}_L \overset{iid}{\sim} N(0,1), \mathbf{b}_L \overset{iid}{\sim} U(0, 2\pi)$$

for train_step $= 1$ **to** max_step **do**
  SGD update $\left\{\beta, \{\mathbf{W}_l\}_{l=1}^{L-1}, \{\mathbf{b}_l\}_{l=1}^{L-1}\right\}$
  Spectral Normalization $\{\mathbf{W}_l\}_{l=1}^{L-1}$ (10).
  if final_epoch **then**
    Update precision matrix $\{\hat{\Sigma}_k^{-1}\}_{k=1}^{K}$ (9).
  end if
end for
Compute posterior covariance $\hat{\Sigma}_k = inv(\hat{\Sigma}_k^{-1})$.

Algorithm 2 SNGP Prediction

Input: Testing example $\mathbf{x}$.
Compute Feature:

$$\Phi_{D_L \times 1} = \sqrt{2/D_L} * \cos(\mathbf{W}_L h(\mathbf{x}) + \mathbf{b}_L),$$

Compute Posterior Mean:

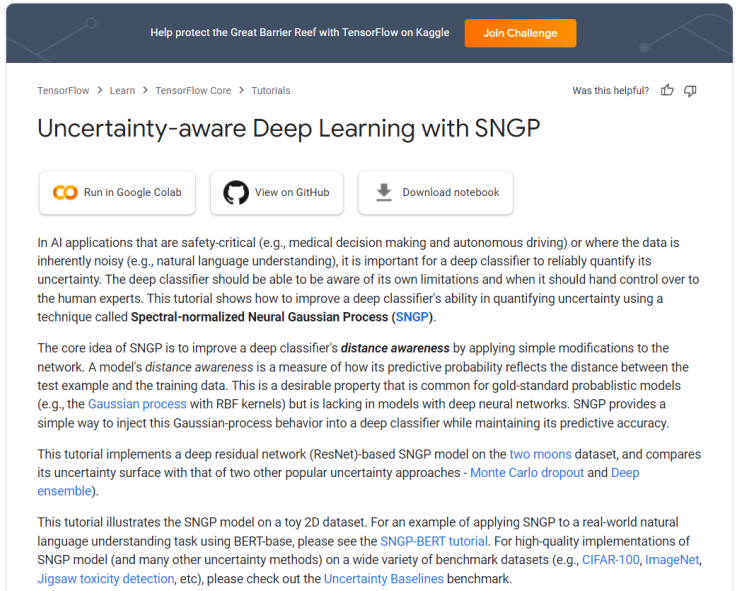
$$\text{logit}_k(\mathbf{x}) = \Phi^\top \beta_k$$

Compute Posterior Variance:

$$\text{var}_k(\mathbf{x}) = \Phi^\top \hat{\Sigma}_k \Phi.$$

Compute Predictive Distribution:

$$p(y|\mathbf{x}) = \int_{m \sim N(\text{logit}(\mathbf{x}), \text{var}(\mathbf{x}))} \text{softmax}(m)$$



https://www.tensorflow.org/tutorials/understanding/sngp

GP-based Approach

Spectral-normalized Neural Gaussian Process Results

Methods	Additional Regularization	Output Layer	Ensemble Training	Multi-pass Inference
Deterministic	-	Dense	-	-
MC Dropout	Dropout	Dense	-	Yes
Deep Ensemble	-	Dense	Yes	Yes
MCD-GP	Dropout	GP	-	Yes
DUQ	Gradient Penalty	RBF	-	-
DNN-SN	Spec Norm	Dense	-	-
DNN-GP	-	GP	-	-
SNGP	Spec Norm	GP	-	-

❖ **CIFAR-10에 Wide ResNet구조 / CLINC OOS(intent)에 BERT구조**

- Accuracy, ECE, NLL 지표에서 기존에 SOTA방법론인 Deep Ensembles와 유사한 수준의 성능 도출
- 특히, Out-of-distribution을 탐지하는 성능이 가장 우수하였으며, 연산 효율성(latency) 측면에서는 SOTA대비 월등히 개선

Vision task

Method	Accuracy (↑) Clean	Accuracy (↑) Corrupted	ECE (↓) Clean	ECE (↓) Corrupted	NLL (↓) Clean	NLL (↓) Corrupted	OOD AUPR (↑) SVHN	OOD AUPR (↑) CIFAR-100	Latency (↓) (ms / example)
Deterministic	96.0 ± 0.01	72.9 ± 0.01	0.023 ± 0.002	0.153 ± 0.011	0.158 ± 0.01	1.059 ± 0.02	0.781 ± 0.01	0.835 ± 0.01	**3.91**
MC Dropout	96.0 ± 0.01	70.0 ± 0.02	0.021 ± 0.002	0.116 ± 0.009	0.173 ± 0.01	1.152 ± 0.01	0.971 ± 0.01	0.832 ± 0.01	27.10
Deep Ensembles	**96.6 ± 0.01**	**77.9 ± 0.01**	**0.010 ± 0.001**	**0.087 ± 0.004**	**0.114 ± 0.01**	**0.815 ± 0.01**	0.964 ± 0.01	0.888 ± 0.01	38.10
MCD-GP	95.5 ± 0.02	70.0 ± 0.01	0.024 ± 0.004	0.100 ± 0.007	0.172 ± 0.01	1.157 ± 0.01	0.960 ± 0.01	0.863 ± 0.01	29.53
DUQ	94.7 ± 0.02	71.6 ± 0.02	0.034 ± 0.002	0.183 ± 0.011	0.239 ± 0.02	1.348 ± 0.01	0.973 ± 0.01	0.854 ± 0.01	8.68
DNN-SN	96.0 ± 0.01	72.5 ± 0.01	0.025 ± 0.004	0.178 ± 0.013	0.171 ± 0.01	1.306 ± 0.01	0.974 ± 0.01	0.859 ± 0.01	5.20
DNN-GP	95.9 ± 0.01	71.7 ± 0.01	0.029 ± 0.002	0.175 ± 0.008	0.221 ± 0.02	1.380 ± 0.01	0.976 ± 0.01	0.887 ± 0.01	5.58
SNGP (Ours)	95.9 ± 0.01	74.6 ± 0.01	0.018 ± 0.001	0.090± 0.012	0.138 ± 0.01	0.935 ± 0.01	**0.990 ± 0.01**	**0.905 ± 0.01**	6.25

NLP task

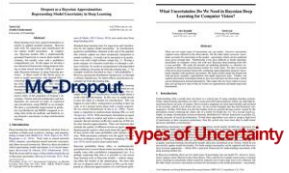
Method	Accuracy (↑)	ECE (↓)	NLL (↓)	OOD AUROC (↑)	OOD AUPR (↑)	Latency (↓) (ms / example)
Deterministic	96.5 ± 0.11	0.024 ± 0.002	3.559 ± 0.11	0.897 ± 0.01	0.757 ± 0.02	**10.42**
MC Dropout	96.1 ± 0.10	0.021 ± 0.001	1.658 ± 0.05	0.938 ± 0.01	0.799 ± 0.01	85.62
Deep Ensemble	**97.5 ± 0.03**	**0.013 ± 0.002**	**1.062 ± 0.02**	0.964 ± 0.01	0.862 ± 0.01	84.46
MCD-GP	95.9 ± 0.05	0.015 ± 0.003	1.664 ± 0.04	0.906 ± 0.02	0.803 ± 0.01	88.38
DUQ	96.0 ± 0.04	0.059 ± 0.002	4.015 ± 0.08	0.917 ± 0.01	0.806 ± 0.01	15.60
DNN-SN	95.4 ± 0.10	0.037 ± 0.004	3.565 ± 0.03	0.922 ± 0.02	0.733 ± 0.01	17.36
DNN-GP	95.9 ± 0.07	0.075 ± 0.003	3.594 ± 0.02	0.941 ± 0.01	0.831 ± 0.01	18.93
SNGP	96.6 ± 0.05	0.014 ± 0.005	1.218 ± 0.03	**0.969 ± 0.01**	**0.880 ± 0.01**	17.36

https://www.tensorflow.org/tutorials/understanding/sngp

Conclusions

Uncertainty Quantification in Deep Learning

Bayesian-based Approach



Ensemble-based Approach

GP-based Approach

❖ 대표적인 uncertainty quantification방법론에 대해 살펴보고, 특징 파악

❖ 각 문헌마다 배경이 되는 지식을 많이 필요로 하며, 이해하는데 필수적으로 선행되어야 함

❖ 최신 uncertainty 연구 동향은 uncertainty 유형 세분화 보다는 'Robustness', 'Calibration', 'Out-of-Distribution' 과 같은 키워드와 연계되어 수행되고 있음 (예측 확률을 잘 정의하자)

❖ 최근 benchmark에 대한 프로토콜을 정의하고자 하는 시도가 있음

❖ 향후 다양한 application에서 적용 가능할 것이라 기대

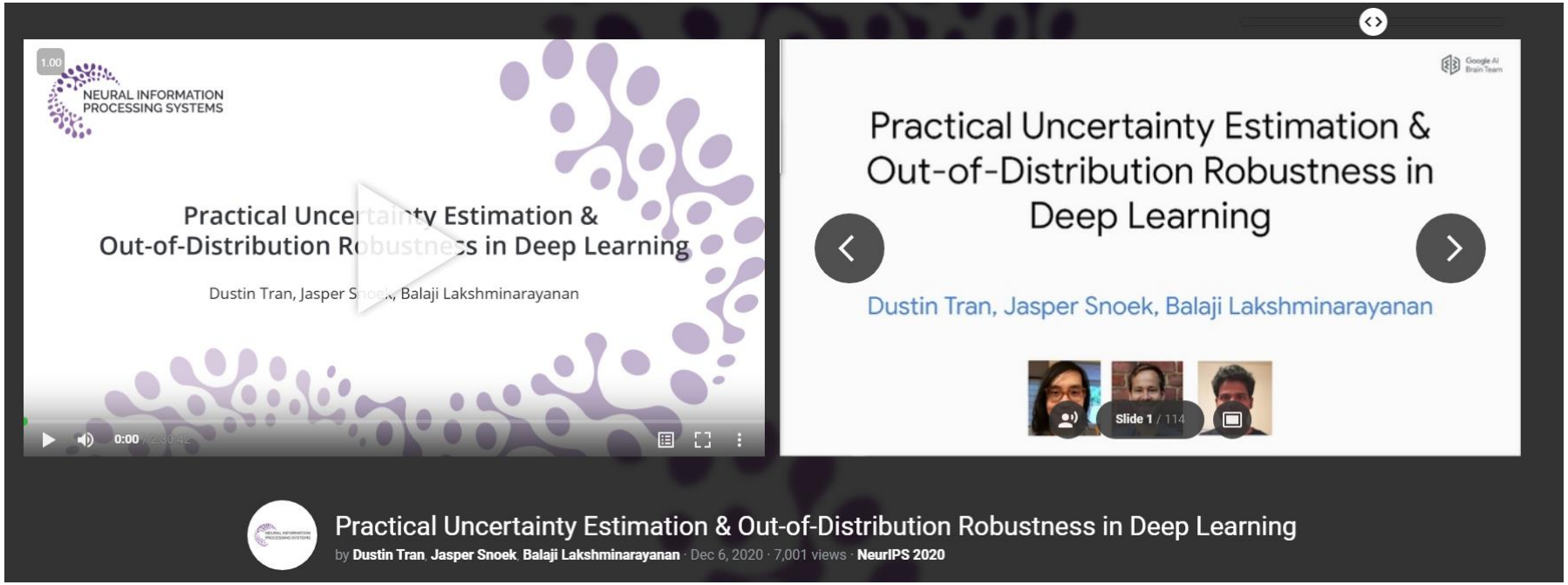
- Explainable AI, Medical imaging, Autonomous vehicle, Active learning, Out-of-distribution detection

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. Machine Learning, 109(2), 373-440.

Materials

Tutorials

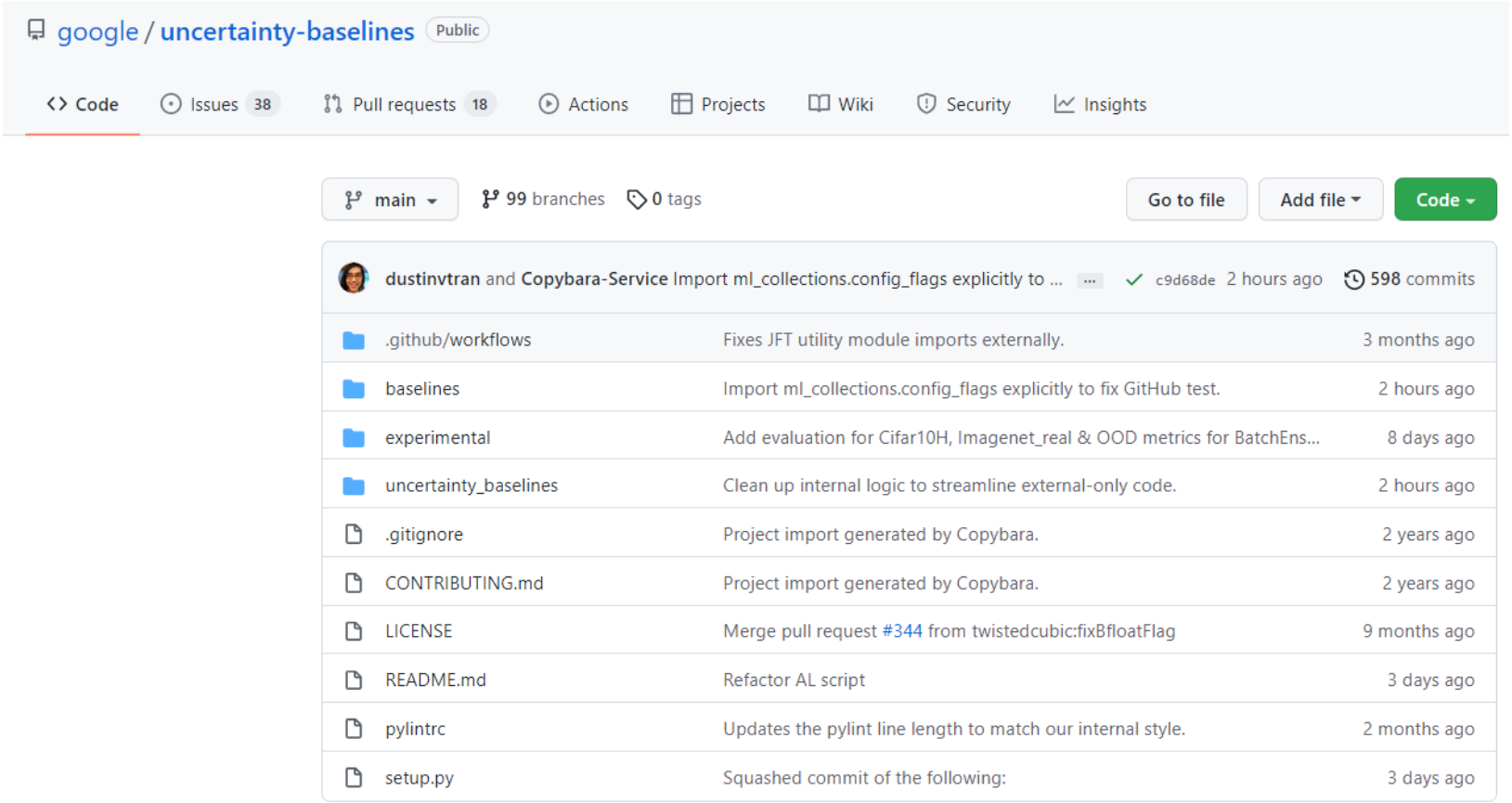
- https://neurips.cc/media/neurips-2020/Slides/16649.pdf
- https://slideslive.com/38935801/practical-uncertainty-estimation-outofdistribution-robustness-in-deep-learning
- https://www.youtube.com/watch?v=ssD7jNDIL2c



Materials

Github

- https://github.com/google/uncertainty-baselines Smoothness assumption



References

❖ Bayesian Neural Nets

- https://www.edwith.org/bayesiandeeplearning
- 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

Thank you

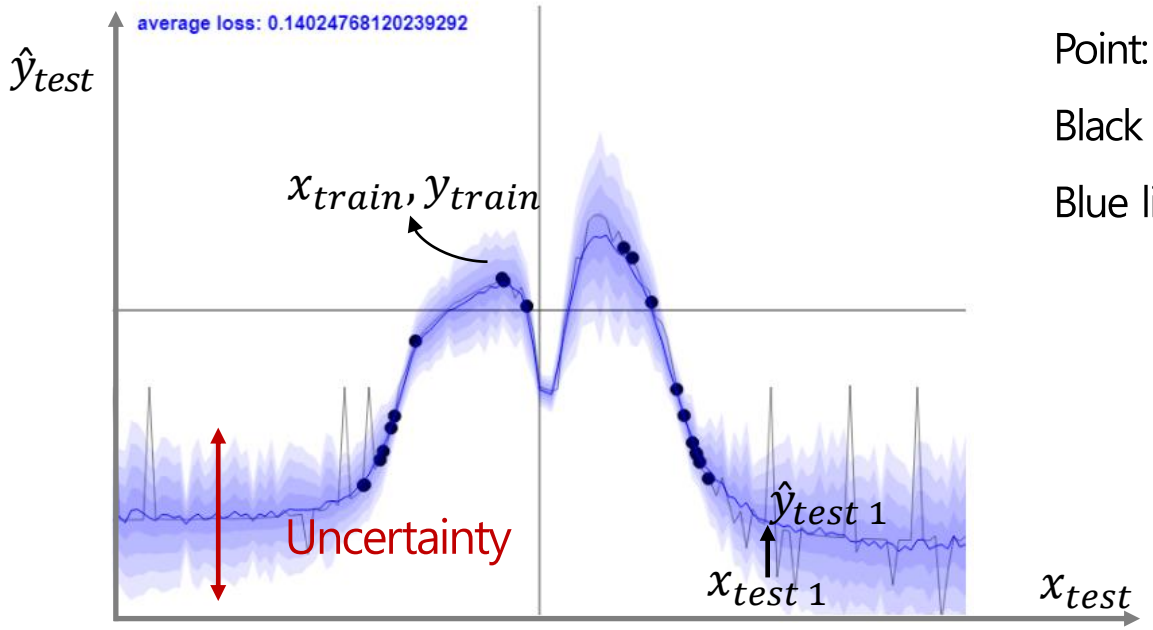
Appendix

Introduction

Uncertainty

❖ Bayesian Neural Networks

- http://mlg.eng.cam.ac.uk/yarin/blog_3d801aa532c1ce.html
- Parameter w 에 분포를 가정하여, 예측값의 형태가 분포를 추정하기 때문에 구간의 형태로 도출할 수 있음
- 이때, 예측값의 분산 ≈ Confidence Interval ≈ Uncertainty



Point: train data points

Black line: $\hat{y}_{test}$

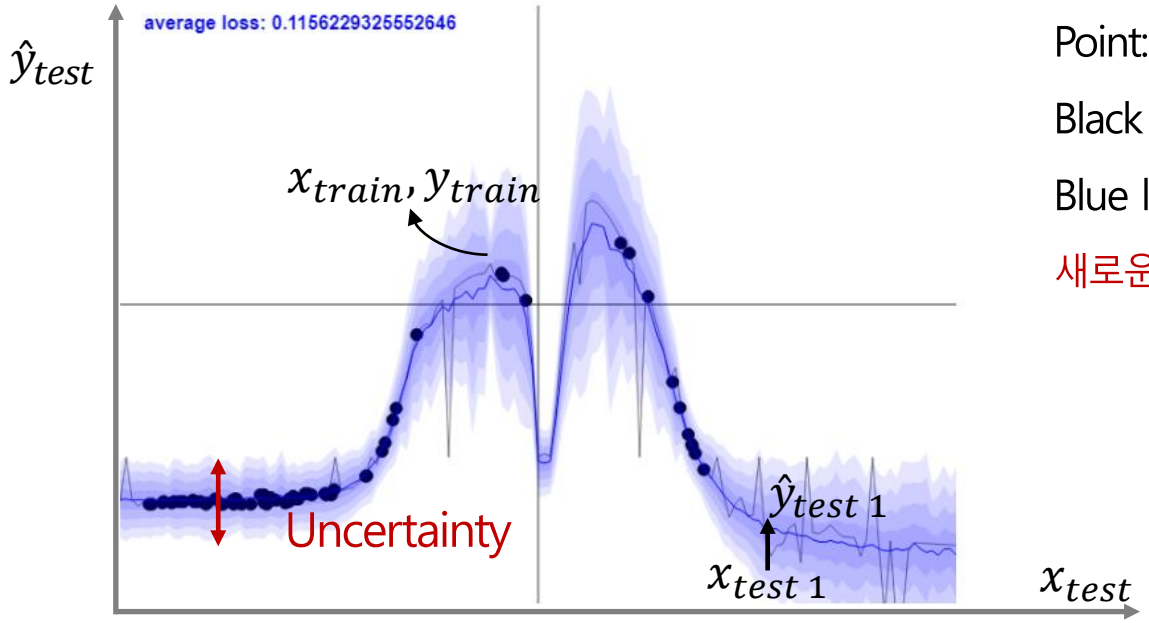
Blue line: $E(\hat{y}_{test})$, Blue shade: $Var(\hat{y}_{test})$

Introduction

Uncertainty

❖ Bayesian Neural Networks

- http://mlg.eng.cam.ac.uk/yarin/blog_3d801aa532c1ce.html
- Parameter w 에 분포를 가정하여, 예측값의 형태가 분포를 추정하기 때문에 구간의 형태로 도출할 수 있음
- 이때, 예측값의 분산 ≈ Confidence Interval ≈ Uncertainty



Point: train data points

Black line: $\hat{y}_{test}$

Blue line: $E(\hat{y}_{test})$, Blue shade: $Var(\hat{y}_{test})$

새로운 train data가 주어지는 경우, Uncertainty를 줄일 수 있음

How do we measure?

Quality of uncertainty

- ❖ **Confidence score를 정의하는 연구에서 활용되는 지표를 활용**
- ❖ **Calibration: 모델의 예측 확률(confidence)이 실제 정확도(accuracy)를 얼마나 반영하는지**
  - Perfect calibration: 모델의 예측 확률(confidence) = 정확도(accuracy)
  - 이에 대한 정량화 지표로 expected calibration error (ECE) 가 대표적

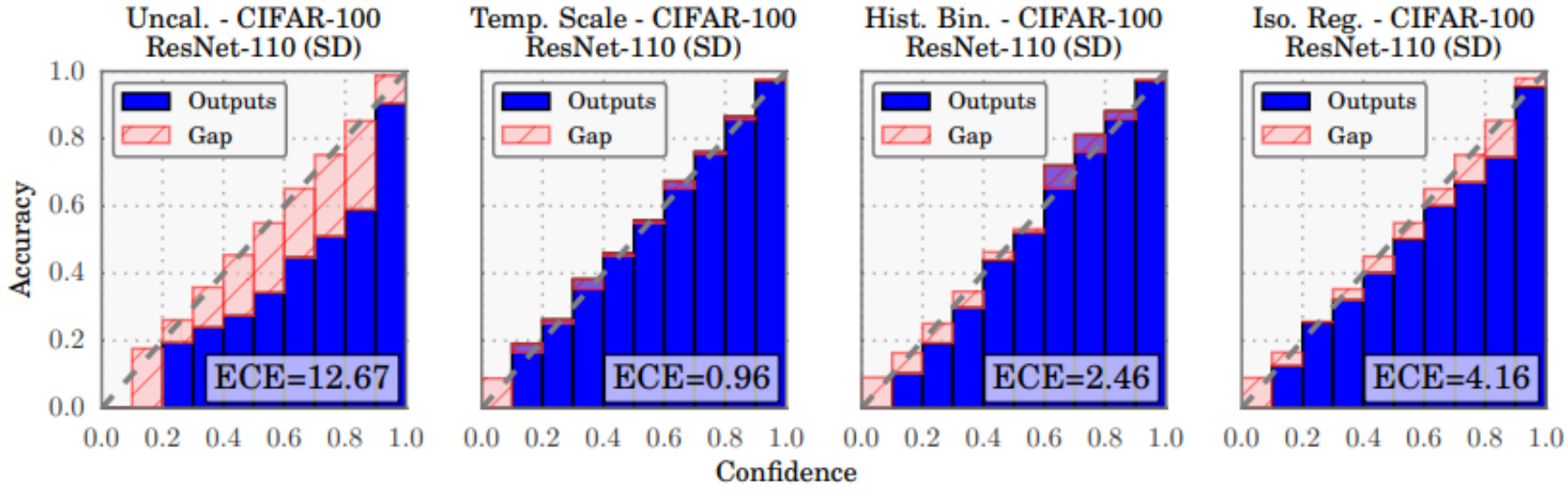


Figure 4. Reliability diagrams for CIFAR-100 before (far left) and after calibration (middle left, middle right, far right).

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In *International Conference on Machine Learning* (pp. 1321-1330). PMLR.

How do we measure?

Quality of uncertainty

- **Calibration: 모델의 예측 확률(confidence)이 실제 정확도(accuracy)를 얼마나 반영하는지**
- **ECE=** $\sum_{b=1}^{B} \frac{n_b}{N} |acc(b) - conf(b)|$
  - 각 bin마다 calibration error를 반영하며, 이는 정확도와는 차이가 있음

Bin 1

Model Prediction	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	**Accuracy**	**Calibration**
True Label	0	0	0	0	0	0	0	1	1	1	Bad (70%)	perfect

Bin 2

Model Prediction	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	**Accuracy**	**Calibration**
True Label	0	0	0	1	1	0	1	1	1	1	Bad (60%)	perfect

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In *International Conference on Machine Learning* (pp. 1321-1330). PMLR.