

I know what I don't know

Bayesian Neural Networks (Variation Inference)

소프트웨어 끈대 강의

노기섭 교수

(kafa46@cju.ac.kr)

Possible Learning in Bayesian

■ Maximum Likelihood Estimation (MLE)

- Same as frequentist!
- Dataset only!

$$P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)} \approx P(D|\theta)$$

목표: 오직 Likelihood만 최대화

■ Maximum A Posterior (MAP)

- $P(D)$: 알고(given) 있다고 가정
- $P(\theta)$: 정규분포라고 가정

$$P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)} \approx P(D|\theta) \times P(\theta)$$

Likelihood, prior를 동시에 최대화

■ Bayesian Inference (Variation Inference)

- Likelihood, Posterior, Evidence 모두 고려
- Computing $P(D)$ is intractable
- Alternatively, using Variational Inference
- $P(\theta|D)$ 계산이 어렵기 때문에 우리가 알고 있는 함수를 이용하여 잘 모사하도록 접근

$$P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)} \approx Q(\theta|\theta')$$

목표: P 를 잘 흉내내는 Q 의 파라미터 θ' 찾기

Basic Philosophy

■ I know what I don't know

- 모르는 것을 안다는 것
- 딥러닝에서 매우 중요한 요소



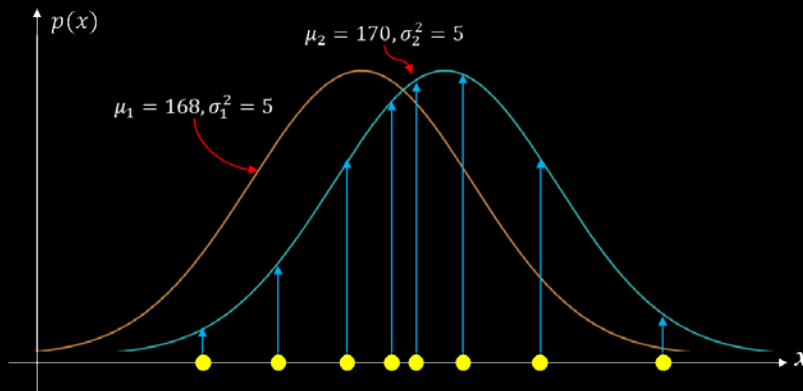
Image source:
<https://www.90daykorean.com/i-dont-know-in-korean/>

Can we know how deep learning gives us answer?

■ 딥러닝이 우리에게 준 답 (answer)

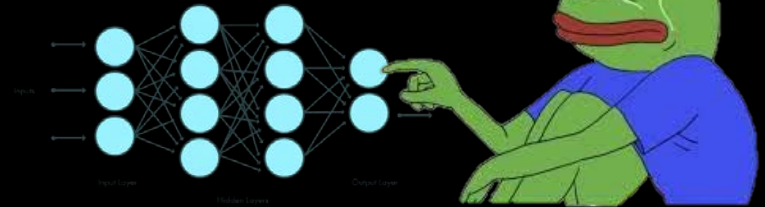
- 어떻게 답을 만들었을까요?
- 그 이유를 알 수 있나?
 - Parameter값이 어떻게 업데이트 되었는지 추적하면 될 것임

요런걸 Explainable AI,
줄여서 XAI 라고 부릅니다.



정규분포: 2개 파라미터 μ, σ
2개 정도는 추적할 수 있지!

1,750억 개 파라미터가
각각 어떻게 업데이트
되었는지 설명하라구요?!!



BERT large: 3억 4천만 개

T5: 110억 개

ChatGPT: 1,750억 개

:

테슬라 AI 첫 사망 사고

■ 2016. 5. 7. 플로리다 윌리스턴 고속도로

- 자율주행 모드 Tesla Model S vs. 흰색 트레일러
- 영상 인공지능: 트럭의 옆면을 '밝게 빛나는 하늘'로 예측 (classification)



이미지 출처:

<https://post.naver.com/viewer/postView.nhn?volumeNo=30957699&memberNo=46914053>

딥러닝도
언제든지
틀릴 수 있다.

틀릴 수 있다는
불확실성을
반영해야 한다.



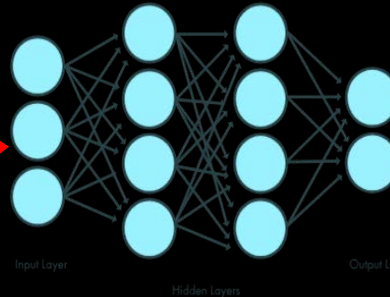
이미지 출처: <https://www.hankyung.com/article/2016070106891>

사고 발생의 근본 원인

■ 딥러닝의 학습 방식

학습방식:

데이터를 기반으로 예측값 \hat{y} 이 정답 y 에
최대한 가깝도록 파라미터 θ 최적화



Answer: Lion 95%

Answer: Dog 98%

Answer: Cat 96%

MLE: 오직 데이터만 고려 (Likelihood only)

MAP: 데이터와 사전 확률을 고려 (Likelihood + Prior)

학습 및 예측 과정을 설명하기 어려움

틀려도 왜, 어떻게 틀렸는지 알 방법이 없음

Gaussian Inference

■ 딥러닝도 틀릴 수 있다!

■ 그러면 틀릴 수 있다는 것을 구현해 주자!

- 어떻게?
- 불확실성(uncertainty) 구현해 주면 된다.
- 불확실성은 확률로 표현해 주면 된다.
- 확률로? 어떻게?

■ Bayesian Theorem

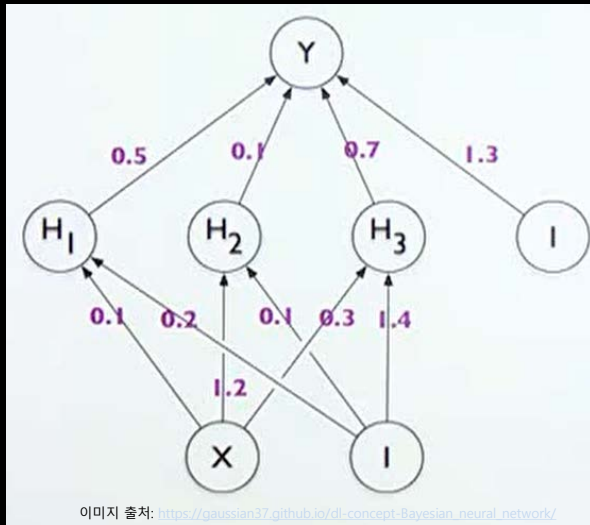
$$\begin{aligned} P(\theta|D) &= \frac{P(D|\theta) \times P(\theta)}{P(D)} \\ &= \frac{P(D|\theta)}{P(D)} \times P(\theta) = \eta \times P(\theta) \end{aligned}$$

데이터셋에 불확실성 존재
MLE, MAP와 다르게
분모까지 고려하자!

Problem & Obstacles

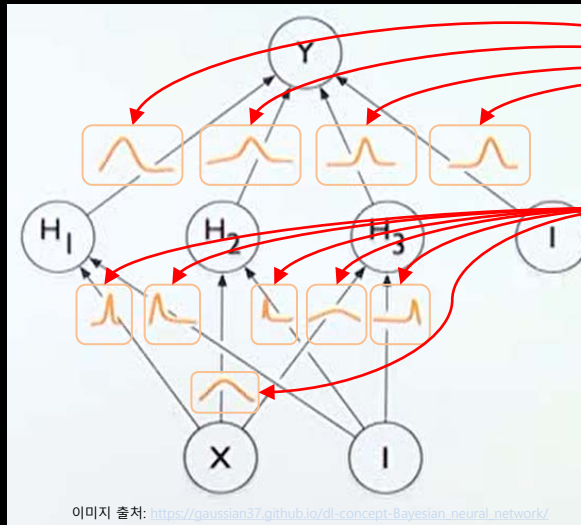
- 간단한 해결책 - Hidden Layer 출력에 확률 분포를 걸어준다!

Frequentist



$$y = w^T x + b$$

Bayesian Network



Weight는 고정 값 아님

확률분포 → 불확실성 내포

$$y = W^L \sigma(\dots (W^2 \sigma(W^1 x + b^1) + b^2) + \dots) + b^L$$

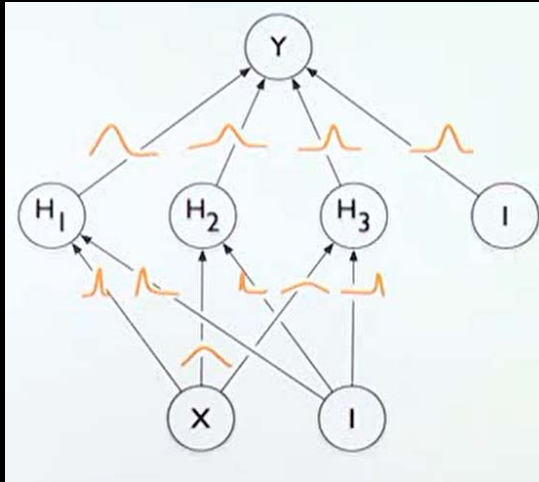
Bayesian

공통점: Prior $P(\theta)$ 를 업데이트 한다는 기본 아이디어

차이점: Prior distribution을 이용해 업데이트

Bayesian Network

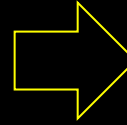
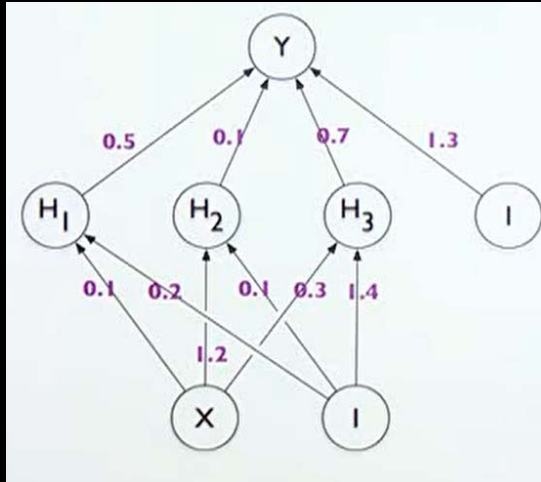
Prior distribution을 이용해 업데이트



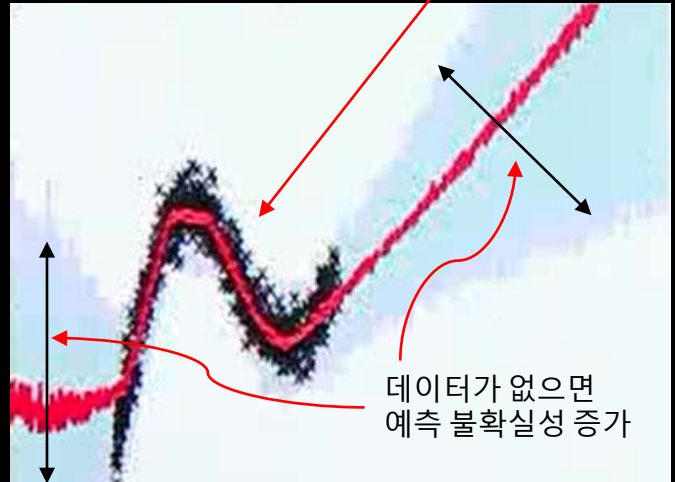
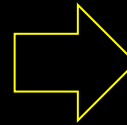
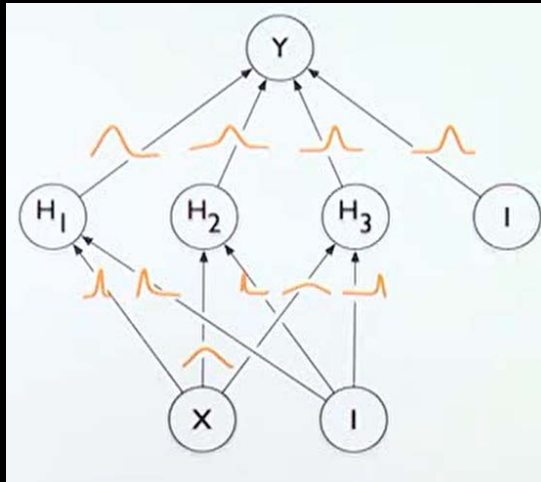
$$\begin{aligned}
 P(\theta) &= \prod_{l=1}^L P(W^l) \times P(b^l) \\
 &= \prod_{l=1}^L \prod_{j,k} N(w_{j,k}^l | 0, \sigma_w^2) \prod_j N(b_j^l | 0, \sigma_b^2)
 \end{aligned}$$

Concept & Effects

Frequentist

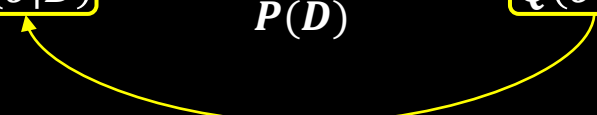


Bayesian Network



이미지 출처 https://gaussian37.github.io/dl-concept-Bayesian_neural_network/

Possible Solution: Variation Inference

$$P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)} \approx Q(\theta|\theta')$$


목표: P 를 잘 흉내내는 Q 의 파라미터 θ' 찾기

각각의 파라미터 θ_i 는 표준정규분포를 따른다고 가정

$$P(\theta_i) \sim N(0, 1^2), \text{ where } \theta = \{\theta_1, \theta_2, \dots, \theta_n\}$$

Q 역시 정규분포를 따른다고 가정

$$Q(\theta_i | \mu_i, \sigma_i) \sim N(\mu_i, \sigma_i^2)$$

μ_i 와 σ_i 먼저 업데이트 $\Rightarrow N(\mu_i, \sigma_i^2)$ 분포에서 θ_i 샘플링

$P(\theta_i) \sim N(0, 1^2)$ 와 $Q(\theta_i | \mu_i, \sigma_i) \sim N(\mu_i, \sigma_i^2)$ 분포의 거리가 최소화 되도록 학습

복잡한 수식은 생략 ^^.

Application

- R. McAlliter, et al., 'Concrete Problems for Autonomous Vehicle Safety: Advantage of Bayesian Deep Learning,' University of Cambridge, UK, presented at International Conference on Artificial Intelligence, 2017

- Paper link: <https://www.ijcai.org/proceedings/2017/0661.pdf>

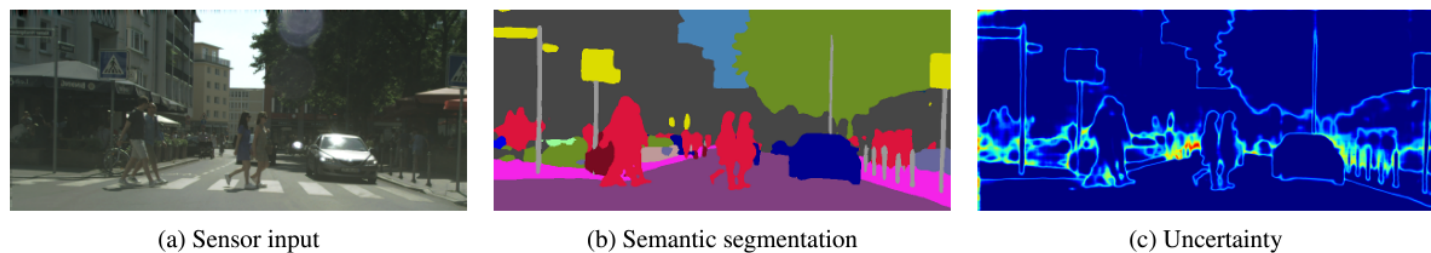


Figure 2: Bayesian deep learning for semantic segmentation. Typically, deep learning models make predictions (b) without considering the uncertainty (c). Our method proposes to estimate uncertainty (c) from each layer to pass down our Bayesian pipeline. (b) shows semantic segmentation, where classes are coloured individually. (c) shows uncertainty where colours other than dark blue indicate a pixel is more uncertain. The model is less confident at class boundaries, distant and unfamiliar objects.

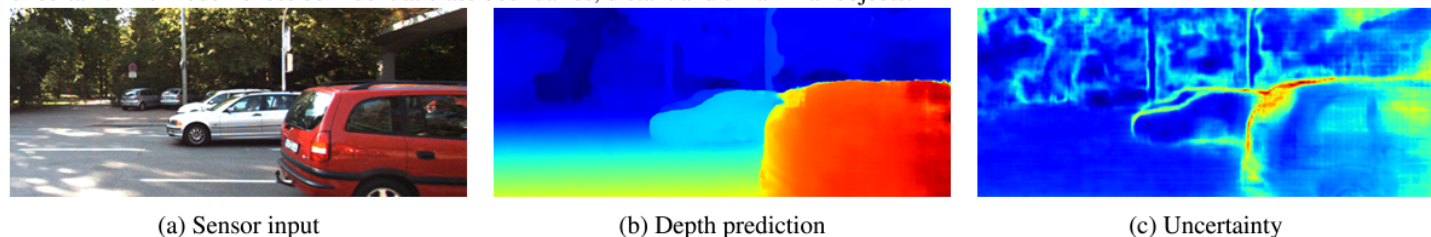


Figure 3: Bayesian deep learning stereo depth estimation algorithm. This example illustrates a common failure case for stereo vision, where we observe that the algorithm incorrectly handles the reflective and transparent window on the red car. Importantly, we observe that the BDL algorithm also predicts a high level of uncertainty (c) with this erroneous output (b).



수고하셨습니다 ..^^..