

Tutorial:

Sparse Variational Dropout

Taewook Nam

MLAI, KAIST

진행방식

- 실제 Variational Dropout을 위한 코드 구현은 간단
 - 다른 수업에서 했던 모델링 + Dropout layer
 - BNN 학습을 위한 전체적 코드 구조는 두번째 시간과 유사
- 왜 이렇게 하고, 어떻게 해야하는지 원리를 이해하는 것이 더 중요
 - 수학적 이해 및 공식 유도가 다소 요구됨
 - 코드 한 줄에 기나긴 수식 유도가 필요
- 수업목표
 - 수학적 디테일을 모두 이해하지 못하더라도, 논리적 흐름을 파악하는 것이 목표
 - 이론 수업에서 보다는 더 자세한 이해

읽어야 할 논문?

Binary Dropout (**BD**)

- Improving neural networks by preventing co-adaptation of feature detectors. Hinton et al. arXiv:1207.0508. 2012. [4002](#)
- Dropout: a simple way to prevent neural networks from overfitting. Srivastava et al. JMLR 2014. [13126](#)

Gaussian Dropout (**GD**)

- Fast dropout training. Wang et al. ICML 2013. [249](#)

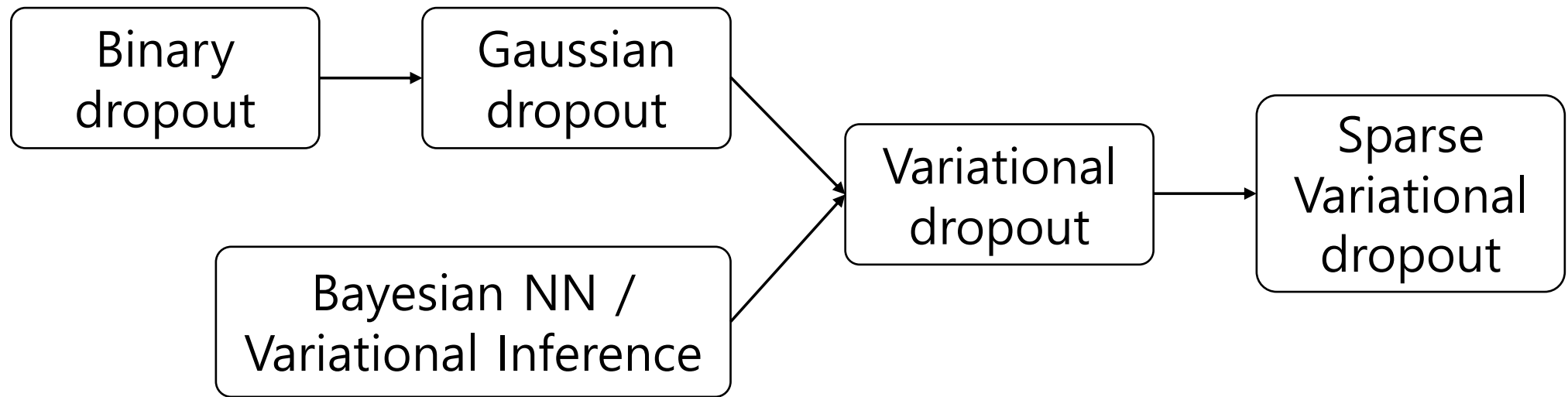
Variational Dropout (**VD**)

- Variational Dropout and the Local Reparameterization Trick. Kingma et al. NIPS 2015. [326](#)

Sparse Variational Dropout (**Sparse VD**) ← Final goal!

- Variational Dropout Sparsifies Deep Neural Networks. Molchanov et al. ICML 2017. [148](#)

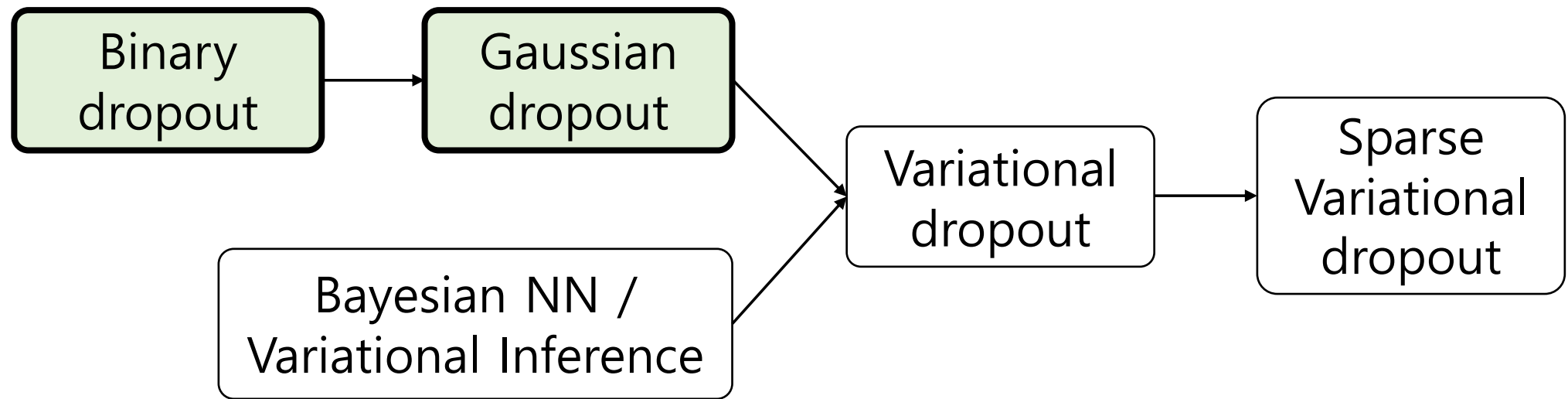
Big Picture



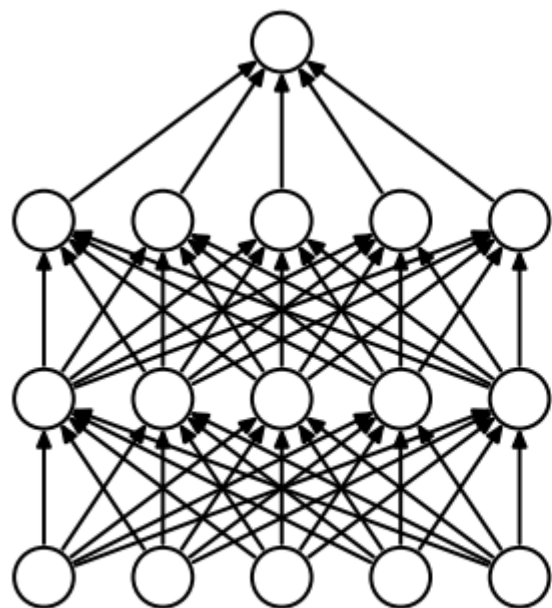
Sparse Variational Dropout

- 기존 'variational dropout'을 약간 변형시켜 **sparse 한 weight** 배워지도록 함
 - Sparse 한 weight => memory efficient한 모델
 - 목표는 Beta Bernolli dropout와 동일
- 'Variational dropout'은 dropout을 (조금 많이) 변형시킨 모델
- Again, why sparse?

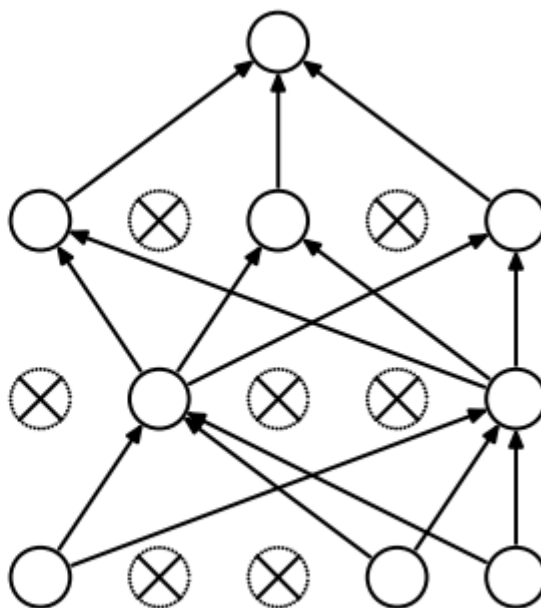
Big Picture



Binary Dropout



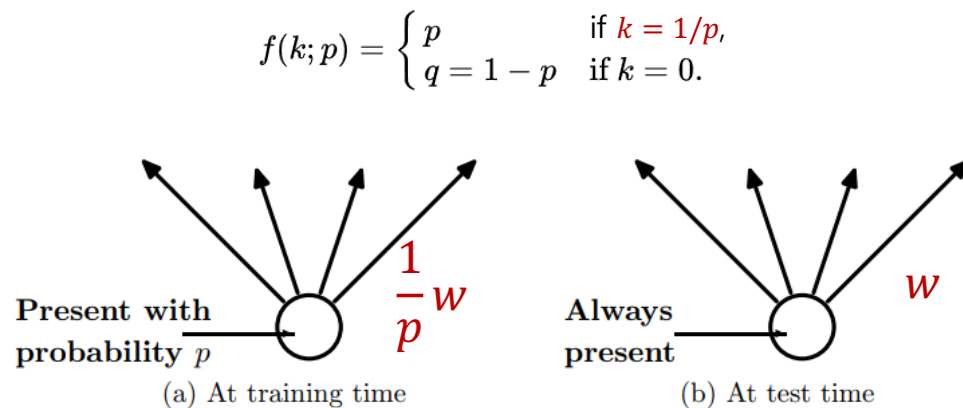
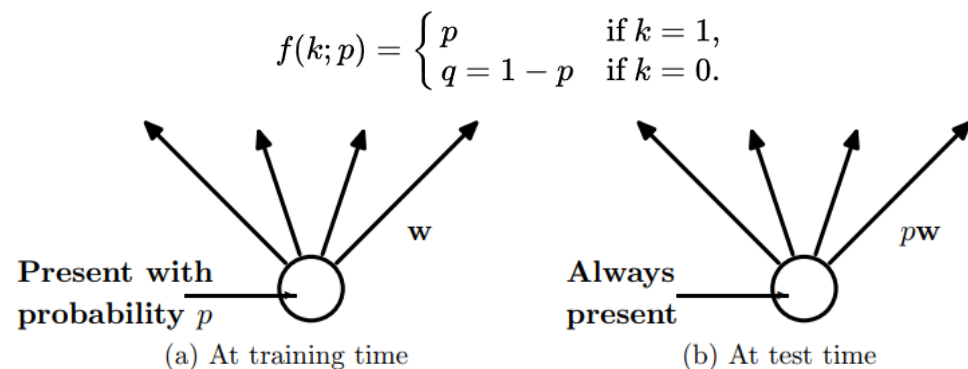
(a) Standard Neural Net



(b) After applying dropout.

- Binary Dropout?
 - 우리가 잘 알고 있는 그 dropout
 - p 의 확률로 **retain**
 - $1-p$ 의 확률로 **drop**
 - 반대로 표기하기도 함
- Multiplicative Bernoulli Noise
 - $h_{drop}^{(l)} = h^{(l)} * \mathbf{z}$
 - $p(\mathbf{z}) = \begin{cases} p & \text{if } z = 1 \\ 1 - p & \text{if } z = 0 \end{cases}$

Binary Dropout

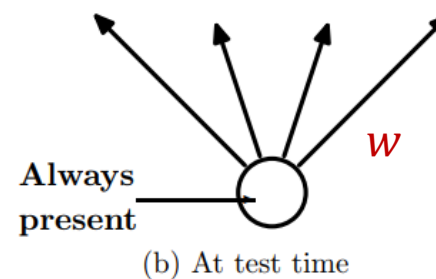
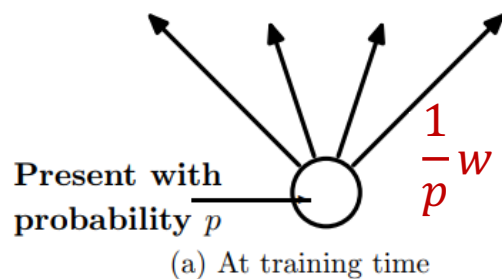


• 학습과 테스트 시의 차이

- Sampling을 여러 번 하기 = dropout을 켜고 여러 번 돌려서 평균 내기
- 각 activation에서 expectation 값을 사용하기 = dropout을 끄되, drop rate로 나눠주기
- Train time에 $\{0, 1\}$ 대신 $\{0, 1/p\}$ 를 곱해 주면 ... $\frac{1}{p}$ $\frac{1}{p}$ $\frac{1}{p}$ $\frac{1}{p}$ 해도 됨

$$p(z) = \begin{cases} p & \text{if } z = 1/p \\ 1 - p & \text{if } z = 0 \end{cases}$$
- Train을 증폭 시킬 것인지, Test를 감쇄 시킬 것인지의 차이

Binary Dropout

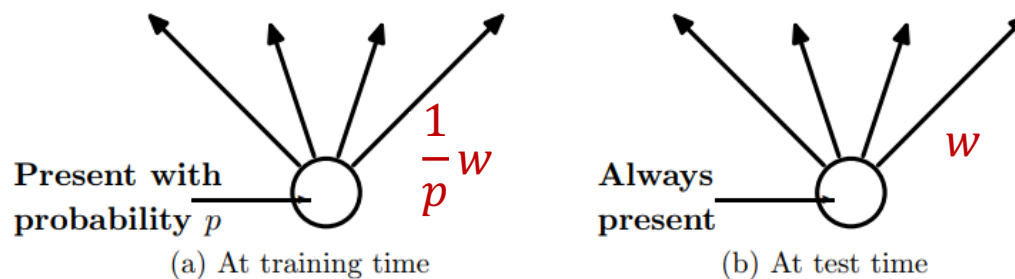


- 두번째 방식을 사용하면,
- Exercise! Noise variable z 의 기댓값과 분산?
- 기대값? $\sum \mathbf{z} p(\mathbf{z})$

$$p(\mathbf{z}) = \begin{cases} p & \text{if } z = 1/p \\ 1 - p & \text{if } z = 0 \end{cases}$$

- 분산? $E[\mathbf{z}^2] - E[\mathbf{z}]^2$

Binary Dropout

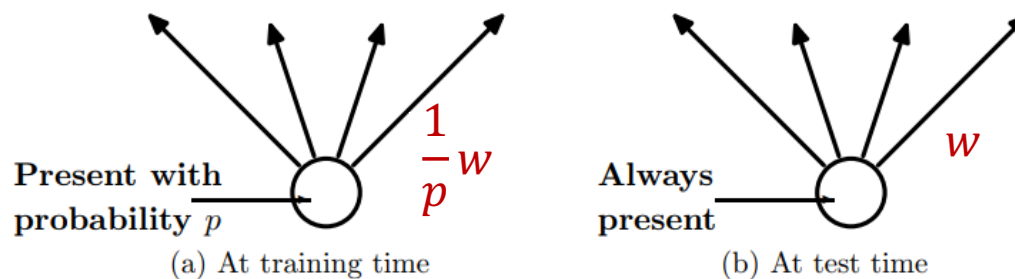


- 두번째 방식을 사용하면,
- Exercise! Noise variable z 의 기댓값과 분산?
- 기대값? $\sum \mathbf{z}p(\mathbf{z})$

$$p(\mathbf{z}) = \begin{cases} p & \text{if } z = 1/p \\ 1 - p & \text{if } z = 0 \end{cases}$$

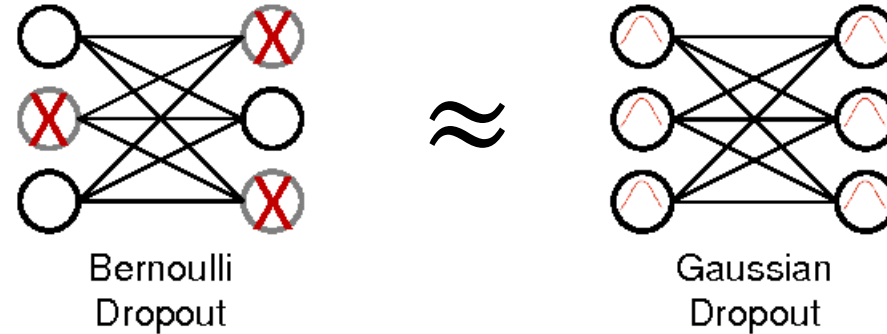
- $E[z] = \frac{1}{p} \cdot \Pr\left(z = \frac{1}{p}\right) + 0 \cdot \Pr(z = 0) = \frac{1}{p} \cdot p + 0 \cdot (1 - p) = 1$
- 분산? $E[z^2] - E[z]^2$
 - $E[z^2] = \left(\frac{1}{p}\right)^2 \cdot p + 0^2 \cdot (1 - p) = \frac{1}{p}$
 - $Var[z] = E[z^2] - E[z]^2 = \frac{1}{p} - 1^2 = \frac{1-p}{p}$

Binary Dropout



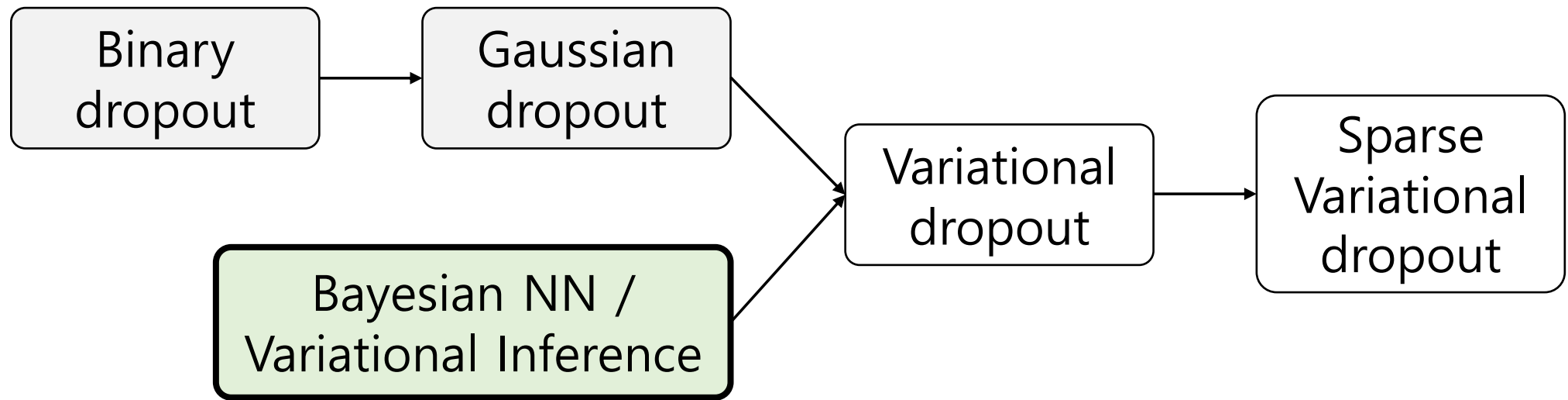
- 두번째 방식을 사용하면,
- Exercise! Noise variable z 의 기댓값과 분산?
- 기대값? $\sum \mathbf{z}p(\mathbf{z})$
 - $E[z] = \frac{1}{p} \cdot \Pr\left(z = \frac{1}{p}\right) + 0 \cdot \Pr(z = 0) = \frac{1}{p} \cdot p + 0 \cdot (1 - p) = 1$
- 분산? $E[z^2] - E[z]^2$
 - $E[z^2] = \left(\frac{1}{p}\right)^2 \cdot p + 0^2 \cdot (1 - p) = \frac{1}{p}$
 - $Var[z] = E[z^2] - E[z]^2 = \frac{1}{p} - 1^2 = \frac{1-p}{p}$

Binary Dropout -> Gaussian Dropout

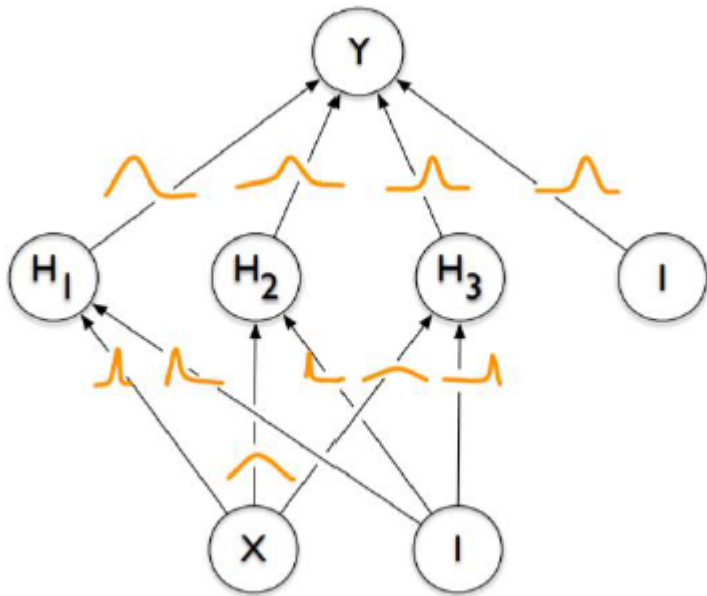


- 같은 평균과 분산을 갖는 Gaussian random variable z 은?
 - $\mu = 1, \sigma = \sqrt{\frac{1-p}{p}}$
 - $z \sim N(\mu, \sigma^2) = N\left(1, \frac{1-p}{p}\right)$
- 새로운 파라미터 $\alpha = \frac{1-p}{p}$ 를 도입
 - $N(1, \alpha) \leftarrow$ 앞으로 계속 보게 될 형태!
 - Exercise! GD에서 큰 noise를 주는 것은 BD에서 어떤 의미일까?
 - Hint : high alpha -> what p?
- Multiplicative Gaussian Noise를 사용한 것이 Gaussian Dropout
 - $h_l^{drop} = h_l * z$
 - $z_i \sim N(1, \alpha) \left(\alpha = \frac{1-p}{p} \right)$

Big Picture



Recap: Bayesian Neural Networks



- **BNN**이란?
 - Weight의 **분포**를 학습하는 네트워크
- 어떻게?
 - Bayes' theorem을 이용
- $p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$, $Posterior = \frac{Likelihood * Prior}{Evidence}$
- 그런데 문제가 있다.
 - 분모를 계산할 수 없음.
$$P(D) = \int_{\theta} P(X|\theta)p(\theta)d\theta$$
- 해결방법?
 - 직접 구할 수 없다면 **근사**하자.
 - 우리가 쓸 방법: **Variational Inference**

Recap: Variational Inference

- **Variational Inference**란?

- 우리의 posterior $p(\theta|D)$ 를 근사하는 기법

Variational distribution

- 어떻게?

- 우리가 쉽게 알 수 있는 분포를 설정하고: $q_\phi(\theta)$
- 이 분포를 $p(\theta|D)$ 와 가깝게 만들자!

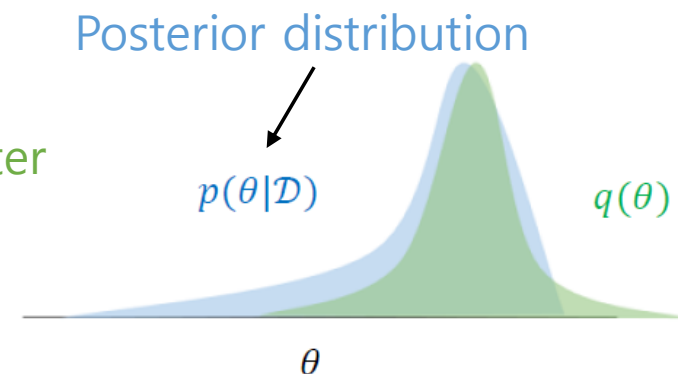
Variational parameter

- 가까움의 기준?

- **KL Divergence** $D_{\text{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$

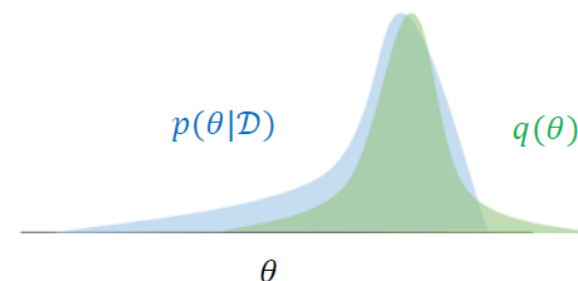
- 우리가 풀어야 할 문제?

- *두 분포의 거리를 줄이는 문제*
- $q_\phi(\theta) = \underset{\phi}{\operatorname{argmin}} KL[q_\phi(\theta) \parallel p(\theta|D)]$
- Inference \rightarrow optimization problem



Recap: Variational Inference

- 유도를 해보면?
 - $KL[q_\phi(\theta)||p(\theta|D)] \downarrow + L(?) \uparrow = \log p(D)$
상수
- 이제 $L(?)$ 를 maximize하면 되는 문제로 치환!
- $L(?)$ 의 실체?
 - $L(?) = \int q_\phi(\theta) \log p(D|\theta) d\theta - KL[q_\phi(\theta)||p(\theta)]$
 - 직관적 해석: Expected Log-likelihood + KL regularization
 - ELBO(Evidence lower bound)라고 불림.
 - 왜? $\log p(D) \geq L(?)$
- 결론: **ELBO**를 maximize하자!

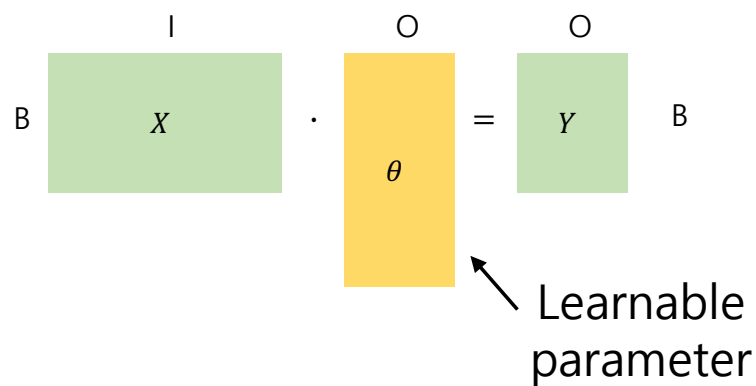


Recap: Variational Inference

- 그래서 어떻게 구현?
- 상황을 가정해보자.
 - 일반적인 classification 태스크 / FC네트워크 & single 레이어
 - Weight가 Gaussian $N(0, I)$ 를 따를 것이 라는 사전(prior) 믿음
 - 자연스럽게 weight의 사후 확률도 Gaussian으로 모델링
 - 데이터에 대한 적절한 사후(posterior) 확률을 학습
- Weight 학습의 기대효과?
 - 우리의 사전 믿음을 기반으로 하되, (min KL term)
 - 데이터를 잘 표현하는 적절한 사후 확률분포를 학습 (min NLL term)

Recap: Variational Inference

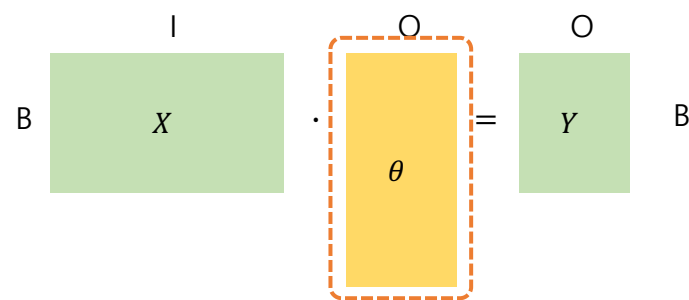
- 그래서 어떻게 구현?



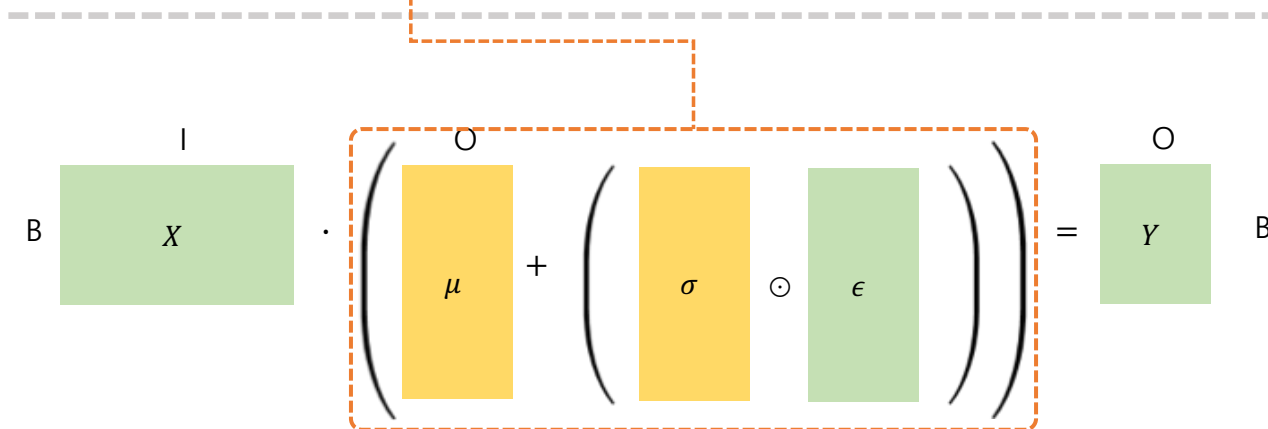
- 우리가 원하는 **posterior** $q_{\phi}(\theta)$:
 - θ 가 Gaussian에서 샘플링: $\theta \sim N(\mu, \sigma^2)$
 - 미분 가능 (back-propagation 위해)

Recap: Variational Inference

- 그래서 어떻게 구현?



- 우리가 원하는 **posterior** $q_\phi(\theta)$:
 - θ 가 Gaussian에서 샘플링: $\theta \sim N(\mu, \sigma^2)$
 - 미분 가능 (back-propagation 위해)



- **Reparametrization Trick(RT)**
 - $\theta \sim q_\phi(\theta) = N(\mu, \sigma^2)$
 - $\rightarrow \theta = f(\phi, \epsilon), \epsilon \sim p(\epsilon)$
 - $\rightarrow \theta = \mu + \sigma \odot \epsilon, \epsilon \sim (0, I)$

Recap: Variational Inference

- 그래서 어떻게 구현?

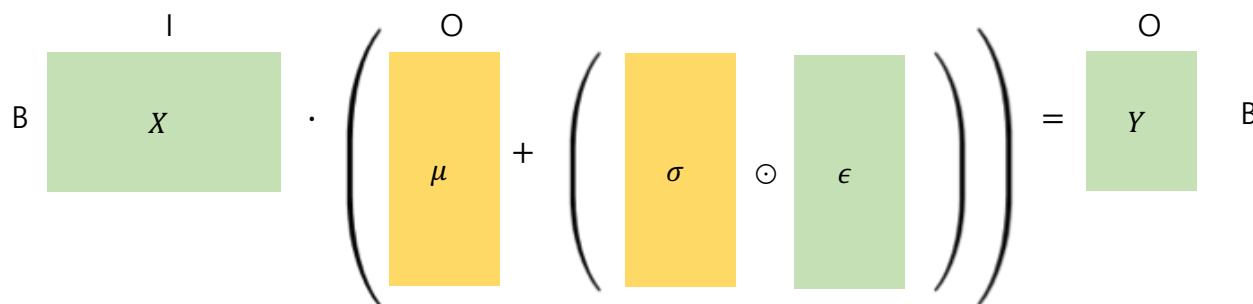
$$B \times I \text{ (X)} \cdot \left(\begin{matrix} O \\ I \end{matrix} \mu + \left(\begin{matrix} O \\ I \end{matrix} \sigma \odot \begin{matrix} O \\ I \end{matrix} \epsilon \right) \right) = \begin{matrix} O \\ B \end{matrix} Y$$

- 이렇게 모델링한 뒤,
- ELBO에 대하여 기존에 하던 것과 동일하게 **minibatch-based** training하면 끝!
 - $\phi = \{\mu, \sigma\}$ 일 때,
 - $\arg\max_{\phi} \int q_{\phi}(\theta) \log p(D|\theta) d\theta - KL[q_{\phi}(\theta)||p(\theta)]$
 - $\approx \arg\max_{\phi} \underbrace{\frac{N}{M} \sum_{i=1}^M \log p(y^i | x^i, f(\phi, \epsilon^i))}_{\text{Minibatch-based MC approximation}} - \underbrace{KL[q_{\phi}(\theta)||p(\theta)]}_{\text{보통 Analytic하게 계산}}$

$N(0, I)$
- 지금까지 한 것:
 - 미분가능한 파이프라인을 만듦(RT)으로써 minibatch 기반 학습을 가능케 함.
 - 이러한 방법을 **Stochastic Gradient Variational Bayes(SGVB)**라고 함.

Recap: Variational Inference

- 그래서 어떻게 구현?


$$B \times I \text{ matrix } X \cdot \left(\begin{matrix} O \times I \text{ matrix } \mu \\ + \\ \left(\begin{matrix} O \times I \text{ matrix } \sigma \\ \odot \\ O \times I \text{ matrix } \epsilon \end{matrix} \right) \end{matrix} \right) = \begin{matrix} O \times B \text{ matrix } Y \end{matrix}$$

- $\operatorname{argmax}_{\phi} \frac{N}{M} \sum_{i=1}^M \log p(y^i | x^i, f(\phi, \epsilon^i)) - KL[q_{\phi}(\theta) || p(\theta)]$
- 해석해보면?
 - **첫번째 항:** 기존 Non-Bayesian과 똑같은 분류 성능 최적화
 - 단, weight에 randomness가 추가된 상황
 - **두번째 항:** prior $N(0, I)$ 와의 KL divergence.
 - 우리의 초기 믿음에서 너무 벗어나지 않도록 regularize.

Recap: Variational Inference

- In tensorflow,
 - Reparameterization은 구현돼있음
 - `W = tf.distribution.Normal(mu, sigma).sample()`
 - Weight를 직접 생성해서 사용하는 방식으로 구현 가능

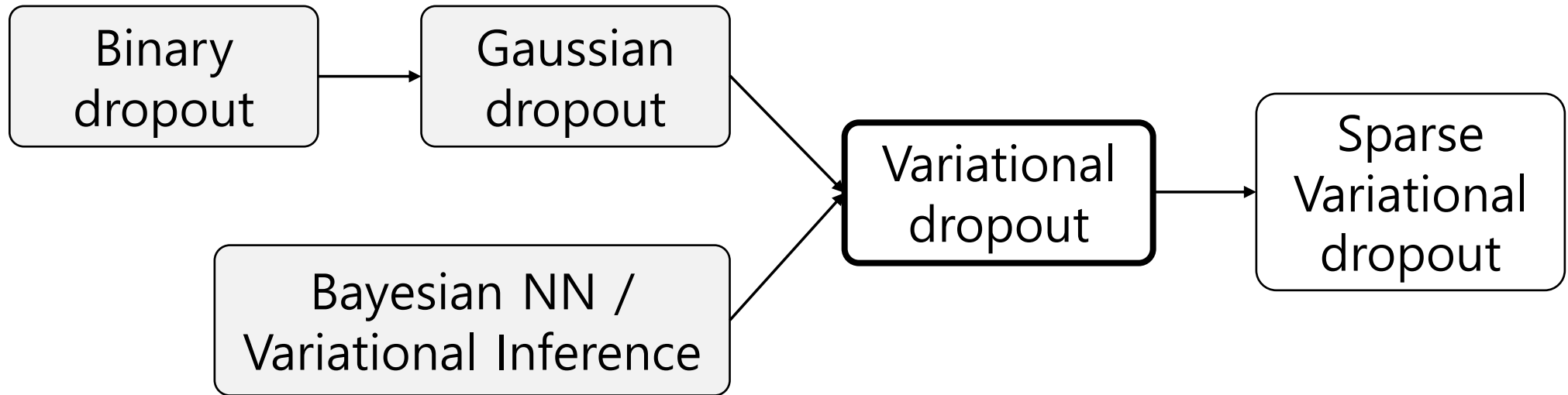
```
mu1 = get_variable('mu1', ...)
rho1 = get_variable('mu2', ...)
w1 = tf.distribution.Normal(mu, softplus(sigma)).sample()
```

- KL은 사용하는 prior, weight에 따라 수식을 표현해야 함
- 직접 weight를 sampling 하면
 - Model output variance가 매우 크고
 - 따라서 gradient variance도 매우 크고
 - Sampling cost도 크다

Recap: Variational Inference

- 마지막으로 생각해볼 것들
 - $\operatorname{argmax}_{\phi} \int q_{\phi}(\theta) \log p(D|\theta) d\theta - KL[q_{\phi}(\theta) || p(\theta)]$
 - SGVB에서의 **Gradient variance**?
 - randomness가 개입되므로 gradient의 variance가 크다!
 - Source: **data** distribution $p(D)$ / **noise** distribution $p(\epsilon)$
 - Variance를 줄이는 것은 학습 안정화에 매우 중요한 요소
 - 두번째 항(KL term)은 가능한 경우, closed-form으로 직접 계산.
 - 계산 가능한데 근사할 필요는 없음
 - 불필요한 gradient variance가 더 증가

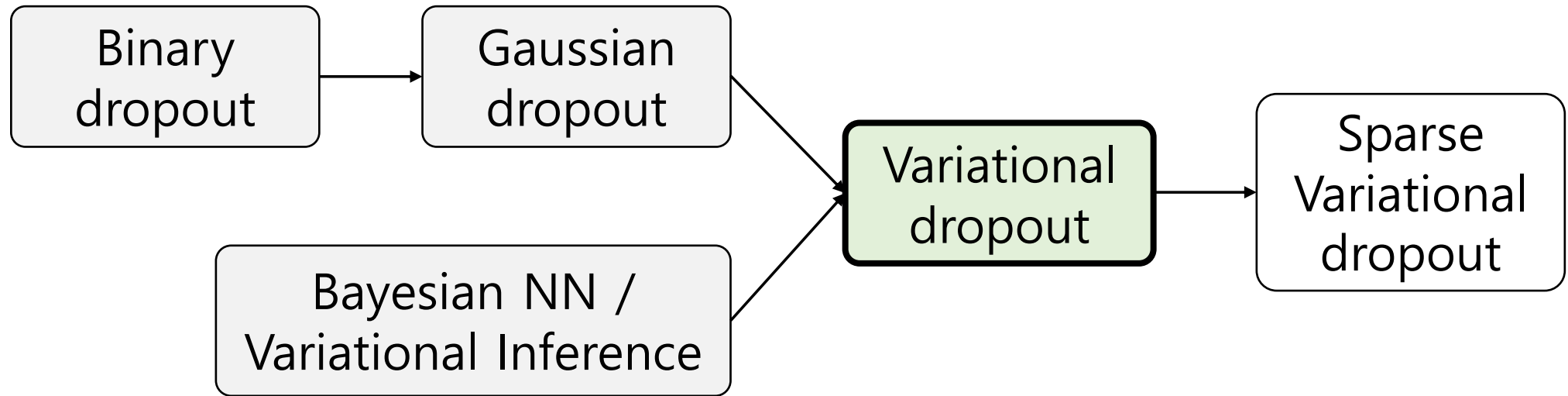
Big Picture



- Dropout : activation 에 multiplicative noise
recall – Binary와 Gaussian에 대해 noise 식?
- Bayesian : parameter의 distribution을 배우자
recall – 어떻게 parameter distribution을 infer?

=> 아이디어 : 둘의 공통점...?

Big Picture



VD: Variational Dropout

- 전체 개요
 - **SGVB**를 효율적으로 개선하려는テクニック을 제안 ← Part 1
 - **Local Reparametrization Trick**(LRT)
 - Gradient variance를 낮추고 더 쉽고 빠르게 계산
 - Dropout과 variational method의 연결점을 탐색 ← Part 2
 - GD + Variational method + LRT = **Variational Dropout**
 - 이를 통해 얻을 수 있는 것?
 - 발전: GD의 성능 향상 (with LRT)
 - 확장: 학습 가능한 dropout rate.
 - 재해석: GD를 Bayesian network로 보았을 때 prior는 무엇일까?

VD-Part 1: Local Reparameterization Trick

- **Local Reparameterization Trick(LRT)**에 대해 알아보자.
 - 목적? SGVB를 효율적으로 개선
 - SGVB의 **gradient variance**를 줄이자!
 - **먼저 해야할 일?** Gradient variance의 요인을 분석
 - 수학적 decomposition을 통해 분석

VD-Part 1: Local Reparameterization Trick

- **SGVB**를 다시 살펴보자. $\int q_{\phi}(\theta) \log p(D|\theta) d\theta$
 - ELBO: $\sum_{(x,y \in D)} E_{q_{\phi}(\theta)} [\log p(y|x, \theta)] - KL[q_{\phi}(\theta) || p(\theta)]$
 - 두번째 KL term은 closed-form으로 계산이 가능하다고 가정.
 - Minibatch approximation:
 - $\sum_{(x,y \in D)} E_{q_{\phi}(\theta)} [\log p(y|x, \theta)] \approx \frac{N}{M} \sum_{i=1}^M \mathbf{logp}(y^i | x^i, f(\phi, \epsilon^i))$
- 즉, SGVB는 $\frac{N}{M} \sum_{i=1}^M L_i$ 의 꼴로 나타낼 수 있음.
 - L_i 는 i 번째 데이터에 대한 **likelihood**를 나타냄을 기억하자.

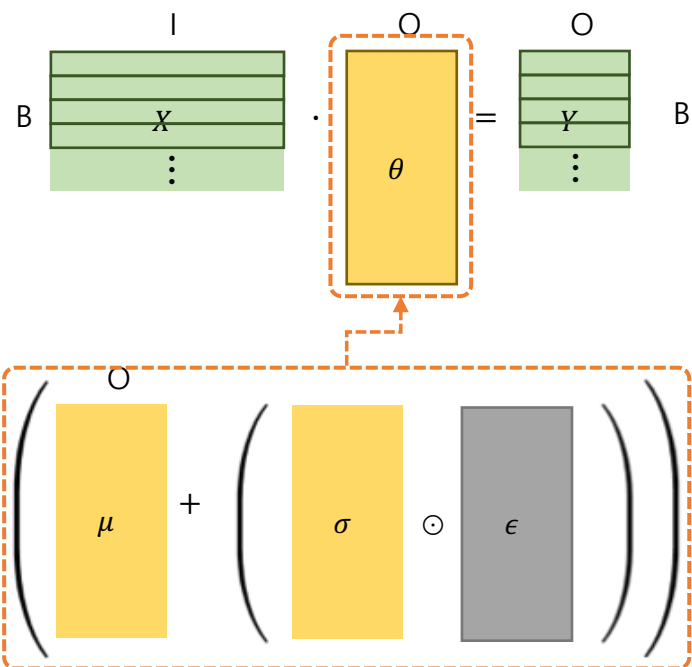
M : Minibatch size N : Data size

VD-Part 1: Local Reparameterization Trick

- 그렇다면 $\frac{N}{M} \sum_{i=1}^M L_i$ 의 **variance**는?
 - $$\begin{aligned} \text{Var} \left[\frac{N}{M} \sum_{i=1}^M L_i \right] &= \frac{N^2}{M^2} \left(\sum_{i=1}^M \text{Var} [L_i] + 2 \sum_{i=1}^M \sum_{j=i+1}^M \text{Cov} [L_i, L_j] \right) \\ &= N^2 \left(\frac{1}{M} \text{Var} [L_i] + \frac{M-1}{M} \text{Cov} [L_i, L_j] \right), \end{aligned}$$
- 알 수 있는 사실?
 - Variance의 영향은 minibatch size M 을 키워서 줄일 수 있음.
 - 반면, **Covariance**의 경우는 **불가능!**
- 우리가 원하는 것?
 - $\text{Cov}[L_i, L_j] = 0$
 - In Korean: Minibatch 안의 데이터들의 log-likelihood를 **종속성을 제거**

VD-Part 1: Local Reparameterization Trick

- 데이터 포인트 사이의 종속성 제거

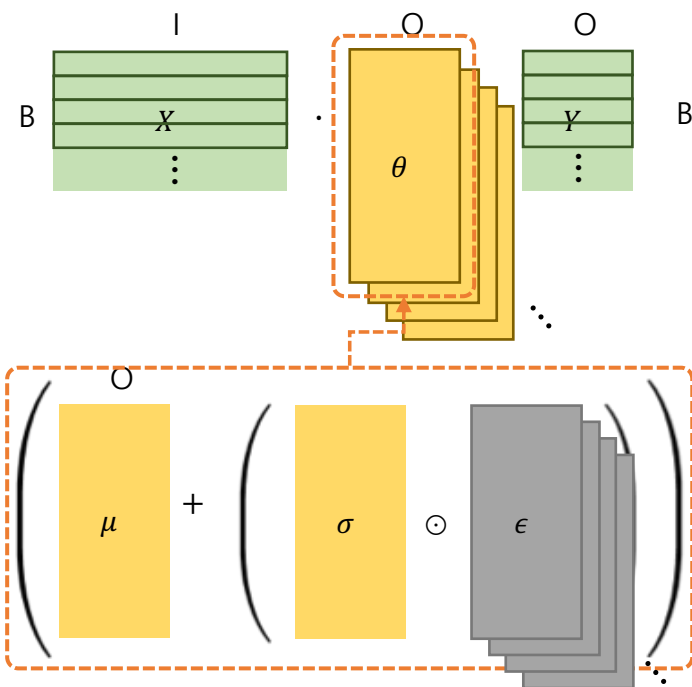


- 기존 상황:

- 배치 안의 모든 데이터 $x_i \in X$ 가 하나의 weight matrix θ 를 공유
- 당연히 θ 는 **하나의** $\epsilon \sim N(0, I)$ 에 dependent
- 모든 데이터가 같은 노이즈를 공유하므로 서로 dependent한 상황
 - $\text{Cov}[L_i, L_j] \neq 0$

VD-Part 1: Local Reparameterization Trick

- 데이터 포인트 사이의 종속성 제거



- 해결 방법?

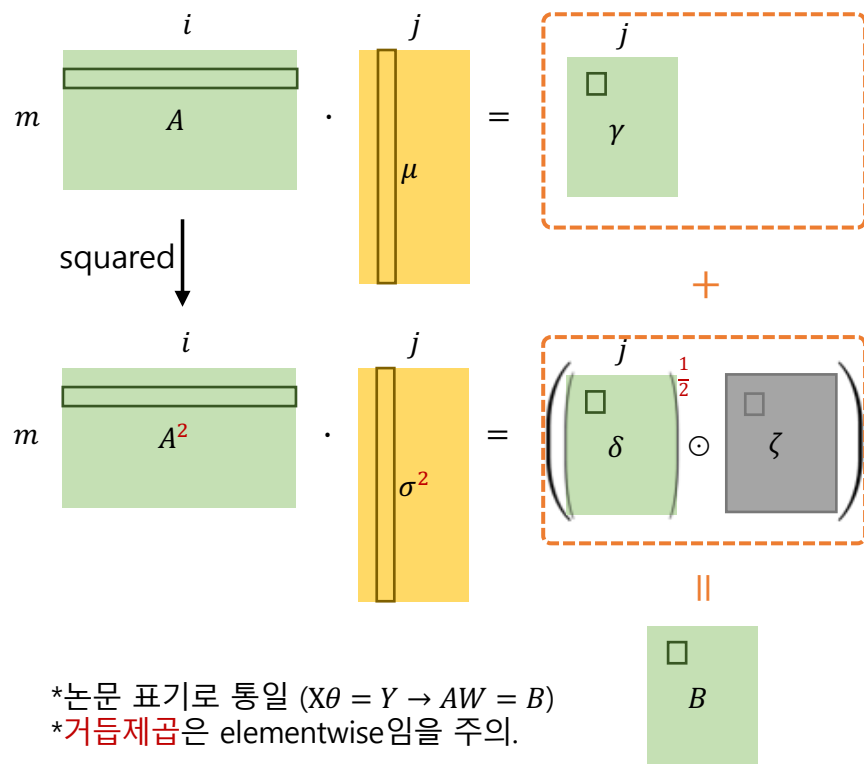
- 배치 안의 모든 데이터 $x_i \in X$ 가 각기 다른 weight matrix θ_i 를 공유
- θ_i 는 각기 다른 $\epsilon_i \sim N(0, I)$ 에 dependent
- 데이터 사이의 dependency가 제거됨
 - $Cov[L_i, L_j] = 0$

- 문제점?

- 계산 비용 증가 (샘플링은 비싼 편)
- 병렬화가 불가능

VD-Part 1: Local Reparameterization Trick

- 데이터 포인트 사이의 종속성 제거



- 더 나은 방법?

- $w_{i,j}$ 가 Gaussian이면, $b_{m,j}$ 도 Gaussian.
 - If X, Y independent and normally distributed, $X+Y$ is also normally distributed.

- 그렇다면 B에서 바로 샘플링해보자. → **LRT!**

$$q_\phi(w_{i,j}) = N(\mu_{i,j}, \sigma_{i,j}^2) \forall w_{i,j} \in \mathbf{W} \implies q_\phi(b_{m,j} | \mathbf{A}) = N(\gamma_{m,j}, \delta_{m,j}),$$

$$\gamma_{m,j} = \sum_{i=1}^{1000} a_{m,i} \mu_{i,j}, \quad \text{and} \quad \delta_{m,j} = \sum_{i=1}^{1000} a_{m,i}^2 \sigma_{i,j}^2.$$

$$b_{m,j} = \gamma_{m,j} + \sqrt{\delta_{m,j}} \zeta_{m,j}, \quad \text{with } \zeta_{m,j} \sim N(0, 1).$$

- 글로벌 noise → 로컬 noise
- weight noise → activation noise

$$X \sim N(\mu_X, \sigma_X^2)$$

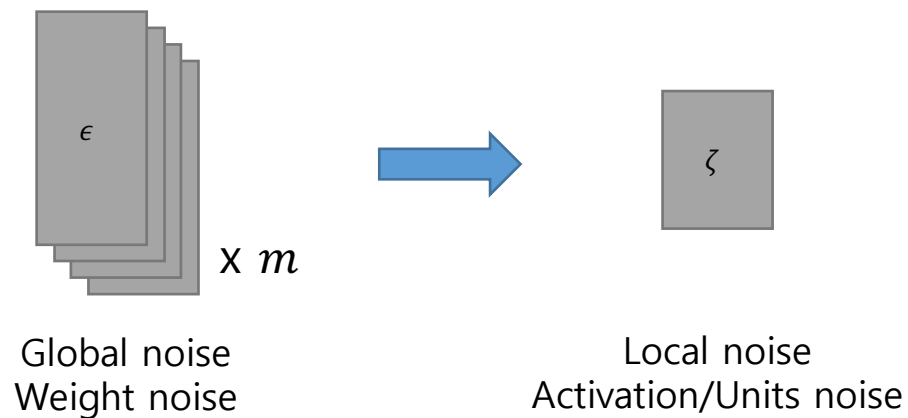
$$Y \sim N(\mu_Y, \sigma_Y^2)$$

$$Z = X + Y,$$

$$Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$$

VD-Part 1: Local Reparameterization Trick

- LRT 의 장점?
 - $\{L_i\}$ 서로 독립적 $\rightarrow \text{Cov}[L_i, L_j] = 0 \rightarrow$ 낮은 gradient variance!
 - 빠른 학습 (in terms of **optimization step**)
 - 더 작은 샘플링 횟수 & 병렬화 가능한 연산
 - 빠른 학습 (in terms of **wall-clock time**)



VD-Part 2

- 지금까지..
 - SGVB에서 사용 가능한 효율적인 테크닉: **LRT**
- 이제부터..
 - Dropout을 variational method로 재해석!
 - **Varational dropout** (with LRT)

VD-Part 2: Reinterpretation of GD as VD

- Dropout과 variational method의 관계

Gaussian dropout

- Multiplicative noise in units
- $B = (A \odot \xi)\theta, \xi \sim N(1, \alpha)$

- LRT:

- $b_{m,j} = \sum_i a_{m,i} \xi_{m,i} \theta_{i,j}$
- $E[b_{m,j}] = \sum_i a_{m,i} \theta_{i,j} E[\xi_{m,i}] = \sum_i a_{m,i} \theta_{i,j}$
- $Var[b_{m,j}] = \sum_i a_{m,i}^2 \theta_{i,j}^2 Var[\xi_{m,i}] = \alpha \sum_i a_{m,i}^2 \theta_{i,j}^2$

If $Cov(X_i, X_j) = 0, \forall (i \neq j)$
then $Var\left(\sum_{i=1}^N X_i\right) = \sum_{i=1}^N Var(X_i)$

Variational Bayesian Inference

- Noise in weights
- $B = AW, W \sim N(\theta, \alpha \theta^2)$

mean Multiplicative noise

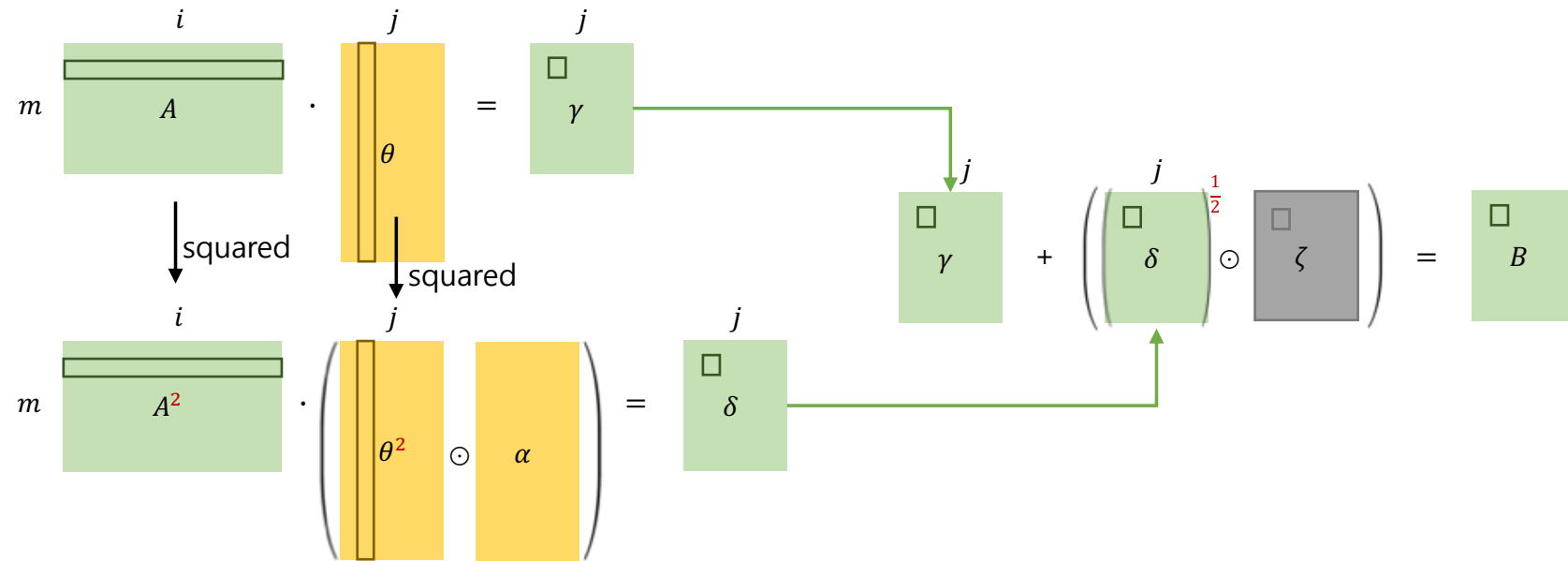
- LRT:

- $b_{m,j} = \sum_i a_{m,i} w_{i,j}$
- $E[b_{m,j}] = \sum_i a_{m,i} E[w_{i,j}] = \sum_i a_{m,i} \theta_{i,j}$
- $Var[b_{m,j}] = \sum_i a_{m,i}^2 Var[w_{i,j}] = \alpha \sum_i a_{m,i}^2 \theta_{i,j}^2$

*직접적 증명은 논문 appendix B 참조.

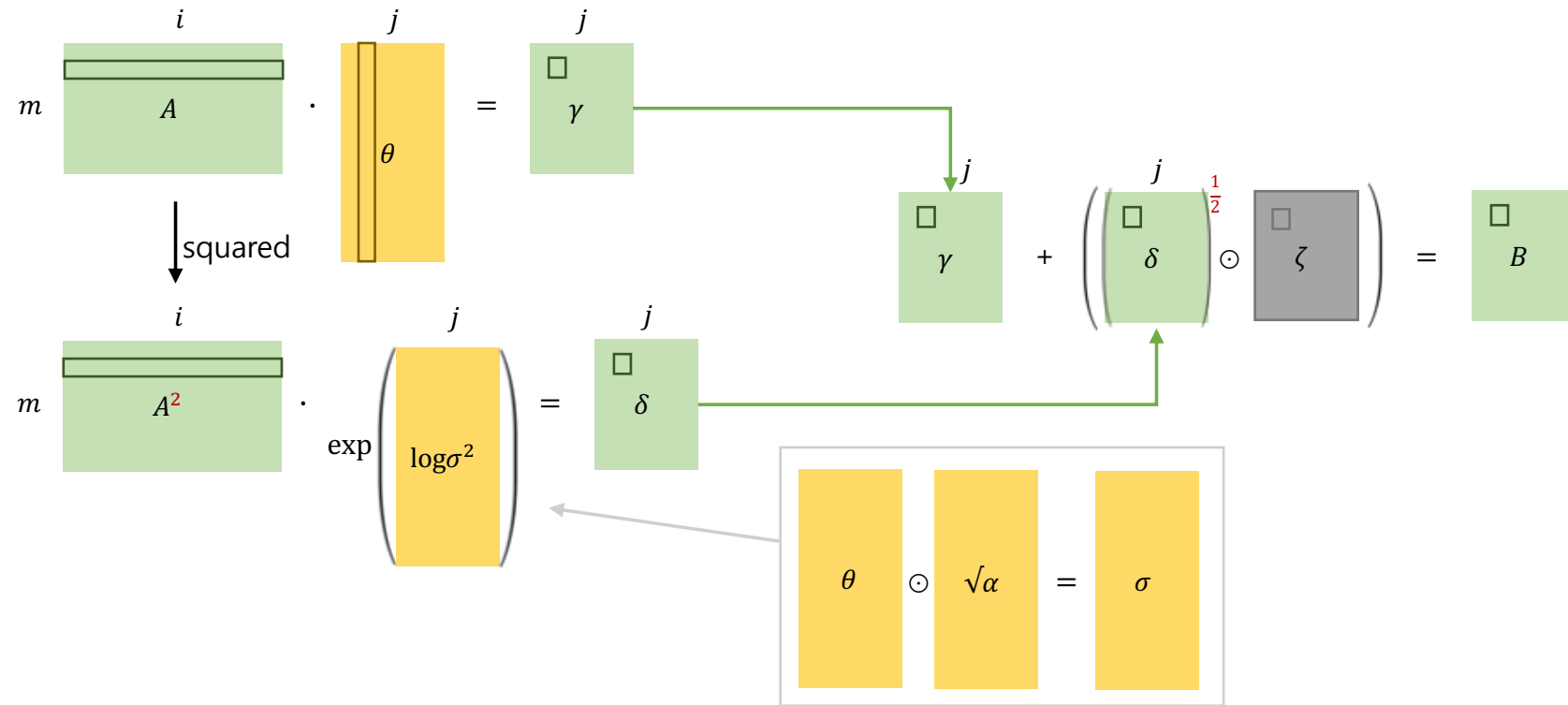
VD-Part 2: Reinterpretation of GD as VD

- VD with LRT



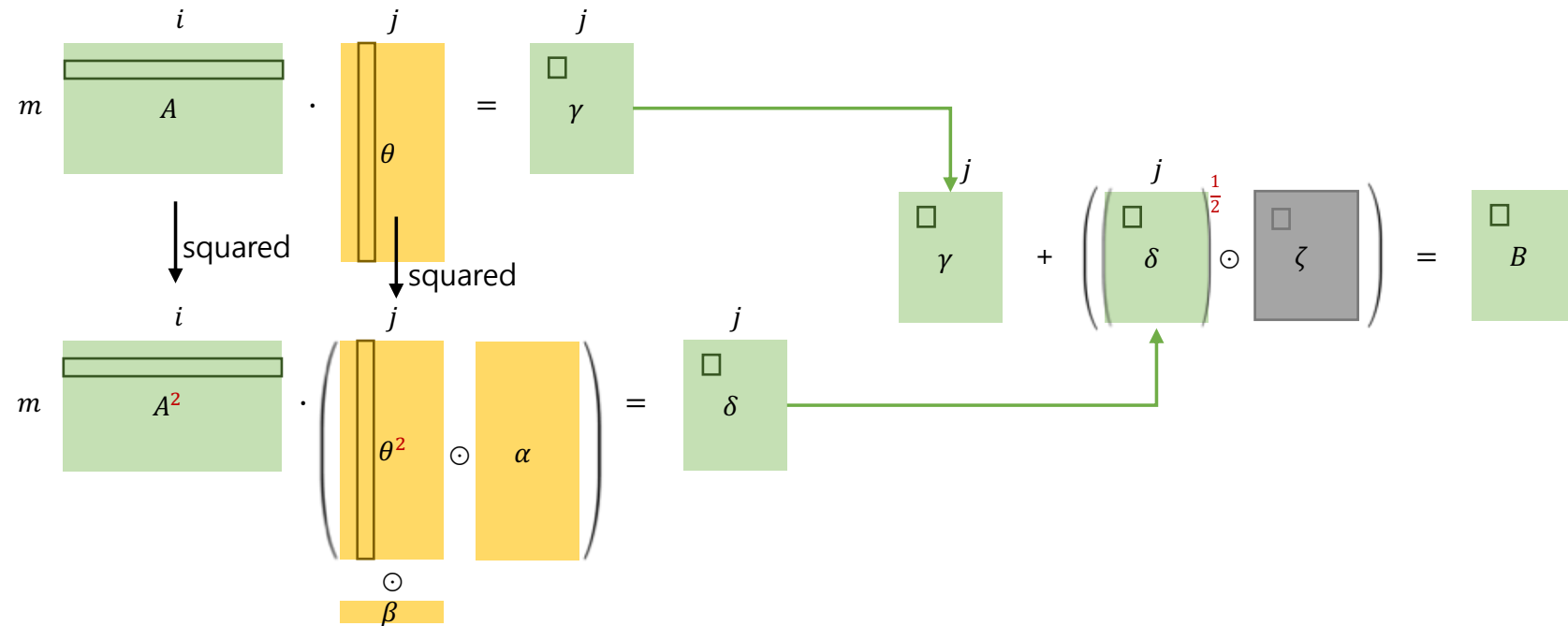
VD-Part 2: Reinterpretation of GD as VD

- VD with LRT



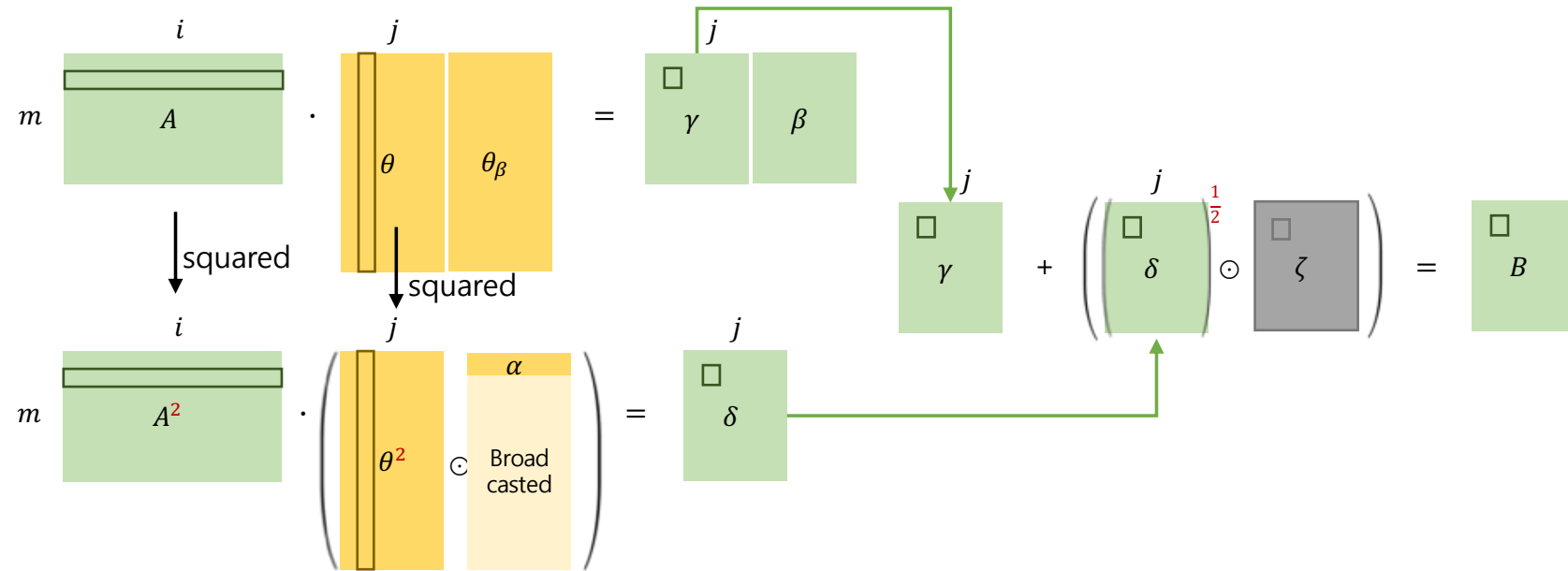
VD-Part 2: Reinterpretation of GD as VD

- VD with LRT



VD-Part 2: Reinterpretation of GD as VD

- VD with LRT



VD-Part 2: Reinterpretation of GD as VD

- Gaussian dropout과 Variational method의 유사성의 의미?
 - **Variational Dropout**을 제안! (드디어)
 - 이를 통해 얻을 수 있는 이점
 - LRT를 이용해 Gaussian drop보다 안정적 학습 가능.
 - 이제 α 를 variational parameter로 놓고 학습할 수 있음.
 - $\min_{\phi} KL[q_{\phi}(W)||p(W|D)]$ 에서 $\phi = \{\theta, \alpha\}$
 - mean
 - Multiplicative noise
 - 또다른 해석 가능: **Prior**는 뭘까?
 - Binary dropout \approx Gaussian Dropout \approx Variational Dropout
 - Binary dropout도 central limit theorem에 의해 근사 가능
 - 참조: Fast dropout training. Wang et al. ICML 2013.

VD-Part 2: Reinterpretation of GD as VD

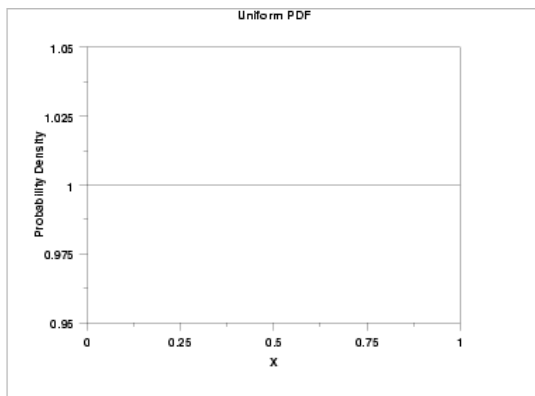
- 그렇다면 prior는? $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$
- Gaussian dropout과의 consistency를 고려(꼭 필요한가?)
 - dropout rate α 는 상수 / weight θ 에 대해서만 학습 $\phi = \{\theta, \alpha\}$
 - ELBO에서 expected log-likelihood term에 대해서만 학습
 - $W \sim N(\theta, \alpha\theta^2)$
 - $\max_{\theta} \sum_{(x,y \in \mathcal{D})} E_{q(W|\theta, \alpha)} [\log p(y|x, W)] - KL[q(W|\theta, \alpha) || p(W)]$
- 이러한 조건을 만족하는 prior?
 - **Log-uniform prior**
 $p(\log |w_{ij}|) = \text{const}$

Has to be Independent to θ (no effect),
when α is fixed.

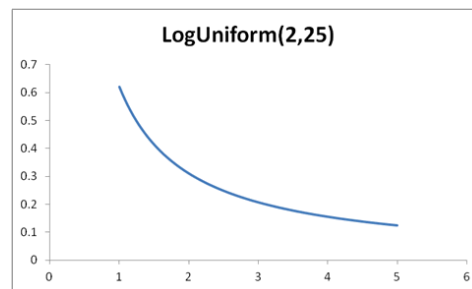
VD-Part 2: Reinterpretation of GD as VD

- **Log-uniform** distribution의 성질

$$p(\log |w_{ij}|) = \text{const} \Leftrightarrow p(|w_{ij}|) \propto \frac{1}{|w_{ij}|}$$



$\log(X)$



X

- Zero 근처에서 높은 density \rightarrow weight에 적용할 경우 sparsity 유도

*MDL(Maximum Description Length) 관점으로 해석:

weight를 floating point format으로 변환 시 log-uniform distribution을 따를 경우,
중요한 digit의 숫자를 최적으로 하여 압축 가능. weight의 크기를 제한하는 효과. (논문참조)

VD-Part 2: Reinterpretation of GD as VD

- **Negative KL term**을 **closed-form**으로 구할 수 있을까?

- $\max_{\phi} \sum_{(x,y \in D)} E_{q_{\phi}(W)} [\log p(y|x, W)] - KL[q_{\phi}(W) || p(W)]$
- Appendix C를 믿는다면,

$$D_{KL}(q(W | \theta, \alpha) || p(W)) = \sum_{ij} D_{KL}(q(w_{ij} | \theta_{ij}, \alpha_{ij}) || p(w_{ij}))$$
$$-D_{KL}(q(w_{ij} | \theta_{ij}, \alpha_{ij}) || p(w_{ij})) = \frac{1}{2} \log \alpha_{ij} - \overbrace{\mathbb{E}_{\epsilon \sim \mathcal{N}(1, \alpha_{ij})} \log |\epsilon|}^{\text{Analytically intractable}} + C \quad \leftarrow \theta \text{에 independent}$$

- 결과적으로 $\mathbb{E}_{\epsilon \sim \mathcal{N}(1, \alpha_{ij})} \log |\epsilon|$ 항 때문에 계산 불가!
 - 그러나, 모든 α 에 대해 쉽게 샘플링 가능

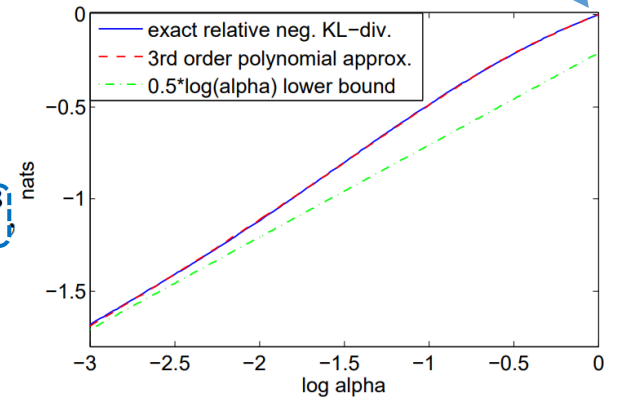
VD-Part 2: Reinterpretation of GD as VD

- 계산할 수 없다면 많이 샘플링해서 근사하자!

- (1) 3차 다항식으로 근사:

$$\begin{aligned}
 -D_{KL}(q(w_{ij} | \theta_{ij}, \alpha_{ij}) \| p(w_{ij})) &= \frac{1}{2} \log \alpha_{ij} - \overbrace{\mathbb{E}_{\epsilon \sim \mathcal{N}(1, \alpha_{ij})} \log |\epsilon|}^{\text{Intractable}} + \underbrace{C}_{\text{Approximated}} \\
 &\approx \text{constant} + 0.5 \log(\alpha) + c_1 \alpha + c_2 \alpha^2 + c_3 \alpha^3 \\
 c_1 &= 1.16145124, \quad c_2 = -1.50204118, \quad c_3 = 0.58629921.
 \end{aligned}$$

$\log \alpha = 0$ 일 때,
KL = 0 이 되도록 C 설정

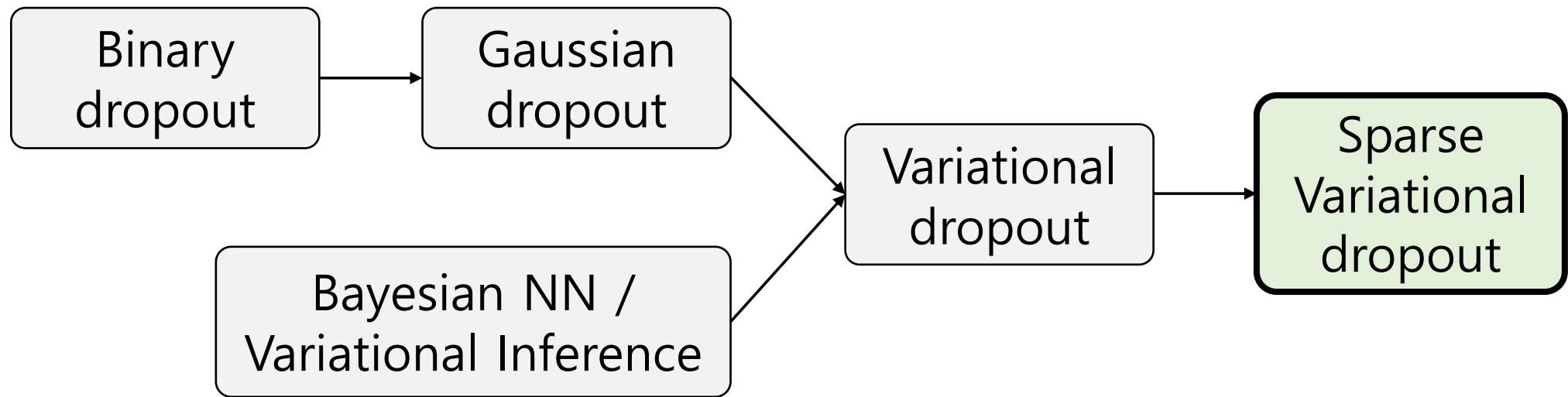


- (2) 더 간단한 lower bound:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(1, \alpha_{ij})} \log |\epsilon| \geq 0 \implies -D_{KL}[q_\phi(w_i) \| p(w_i)] \geq \text{constant} + 0.5 \log(\alpha)$$

- 제한:** $\alpha \leq 1$, $p \leq 0.5$ ($\alpha = \frac{1-p}{p}$) \rightarrow 완전히 drop ($p = 1$) 불가능!
 - 이유? α 가 클때, **large gradient variance** \rightarrow local minima

Big Picture



Sparse VD:

- VD에서 무엇이 추가 되었나?
 - 기본전제: α 에서 $\alpha_{i,j}$ 로 확장 (weight별 독립적인 dropout 학습)
 - **Additive Noise Reparameterization (1)**
 - Gradient variance를 줄이기 위한 새로운 테크닉
 - **Approximation of the KL Divergence (2)**
 - α 의 범위에 제한(e.g. $\alpha \leq 1$) 없이 학습
 - $\alpha \rightarrow \infty$ / $p \rightarrow 1$: 항상 drop / 제거 가능
 - 기타 등등
- 결과적으로?
 - 매우 **sparse**한 network 학습
 - Bayesian pruning으로의 연결

Sparse VD: Additive Noise Reparametrization

- VD에서의 문제점: $q(w_{ij} | \theta_{ij}, \alpha) = \mathcal{N}(w_{ij} | \theta_{ij}, \alpha\theta_{ij}^2)$.
 - Droprate α 가 큰 영역에서 θ 에 대한 **gradient variance**가 매우 큼

$$\frac{\partial \mathcal{L}^{SGVB}}{\partial \theta_{ij}} \uparrow = \frac{\partial \mathcal{L}^{SGVB}}{\partial w_{ij}} \cdot \frac{\partial w_{ij}}{\partial \theta_{ij}} \uparrow$$

$$w_{ij} = \theta_{ij}(1 + \sqrt{\alpha_{ij}} \cdot \epsilon_{ij}),$$

$$\frac{\partial w_{ij}}{\partial \theta_{ij}} \uparrow = 1 + \sqrt{\alpha_{ij}} \cdot \epsilon_{ij},$$

$$\epsilon_{ij} \sim \mathcal{N}(0, 1)$$

- 해결방법:
 - 새로운 변수 도입

$$\theta_{ij} + \theta_{ij} \cdot \sqrt{\alpha_{ij}} \cdot \epsilon_{ij}$$

↓ **New variable**

$$w_{ij} = \theta_{ij}(1 + \sqrt{\alpha_{ij}} \cdot \epsilon_{ij}) = \theta_{ij} + \sigma_{ij} \cdot \epsilon_{ij}$$

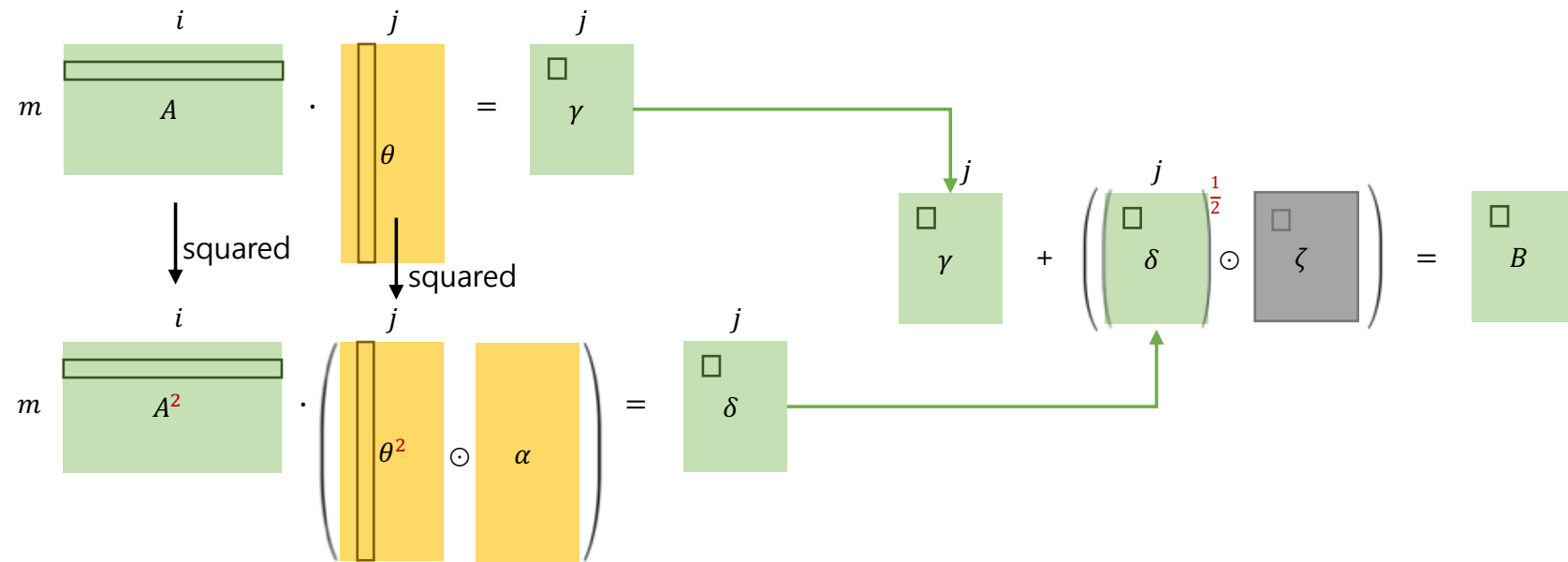
$$\frac{\partial w_{ij}}{\partial \theta_{ij}} = 1, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

α 가 궁금하면 θ 와의 관계에서 역으로 계산.
 σ 값 자체는 θ 와 무관!

- 실제로는 α 대신에 **$\log \sigma^2$** 를 학습 \rightarrow 학습 안정
 - 네트워크 output 자체를 **$\log \sigma^2$** 값으로 해석
 - $w = \theta + \exp(\log \sigma^2) \cdot \epsilon$

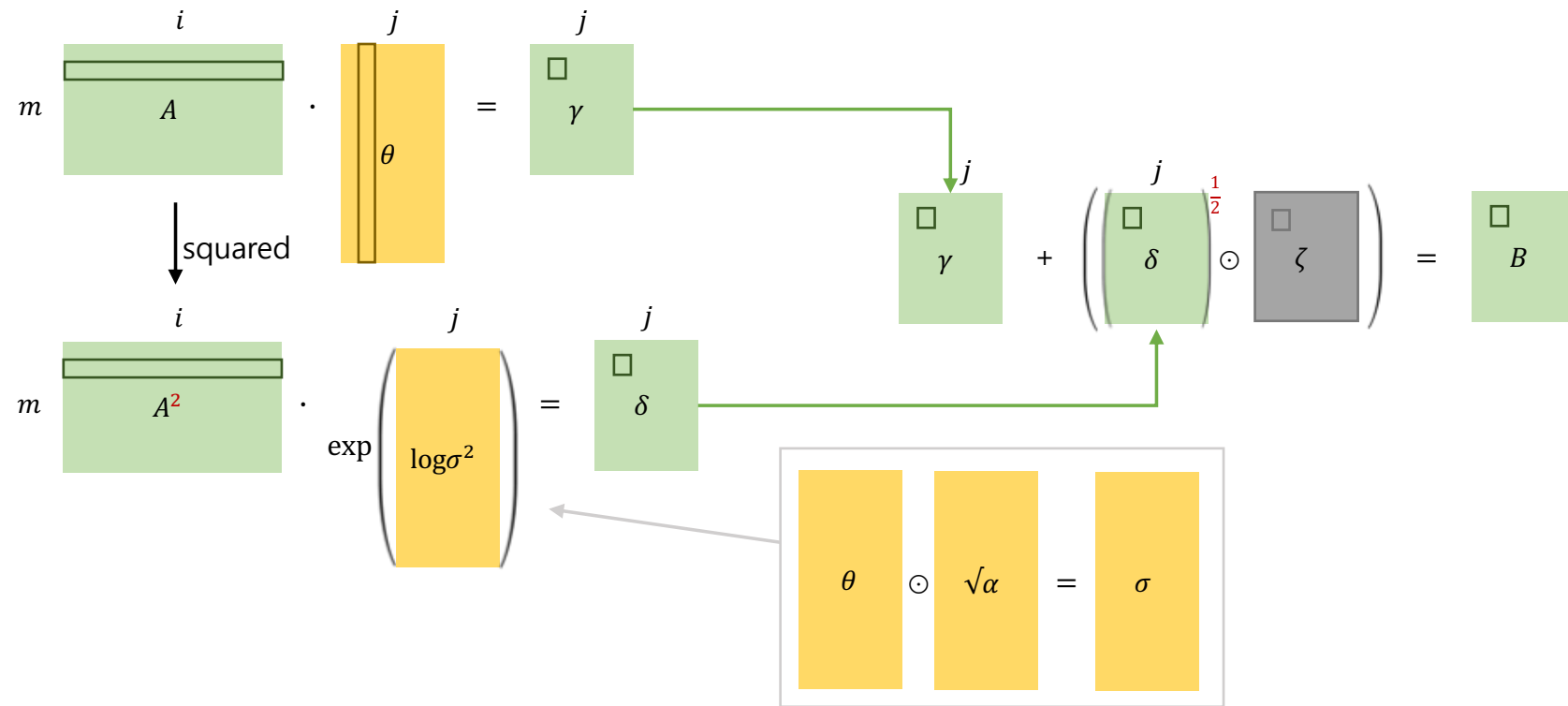
Sparse VD: Additive Noise Reparametrization

- VD with LRT



Sparse VD: Additive Noise Reparametrization

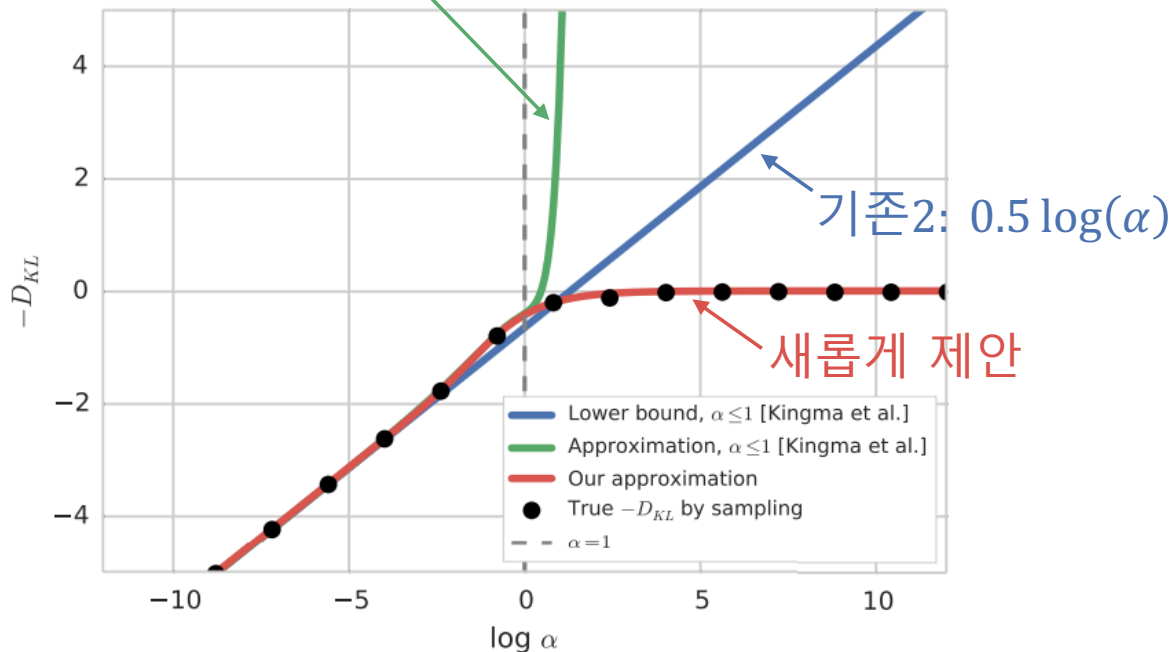
- VD with LRT + ANRT



Sparse VD: Approximation of the KL term

- KL term approximation: 모든 α 영역에서 더 정확한 근사

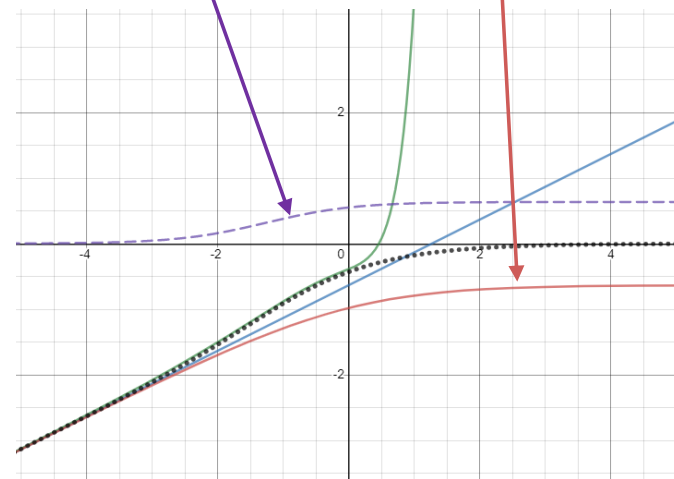
기존1: $0.5 \log(\alpha) + c_1 \alpha + c_2 \alpha^2 + c_3 \alpha^3$



$$-D_{KL}(q(w_{ij} | \theta_{ij}, \alpha_{ij}) \| p(w_{ij})) \approx$$

$$\approx k_1 \sigma(k_2 + k_3 \log \alpha_{ij}) - 0.5 \log(1 + \alpha_{ij}^{-1}) + C$$

$k_1 = 0.63576 \quad k_2 = 1.87320 \quad k_3 = 1.48695$



- 사실상 Heuristic한 방법을 사용
 - $-0.5 \log(1 + \alpha^{-1})$ 를 먼저 설정
 - 남은 차이가 sigmoid와 비슷하다는 점에 착안하여 근사 함수 디자인

Sparse VD: Sparsity

- α 를 dropout rate p 관점에서 본다면?
 - $\alpha \rightarrow \infty : p \rightarrow 1$ 이므로 항상 drop / 제거 가능
- α 를 w_{ij} 에 더해지는 multiplicative noise 관점에서 본다면?
 - $\alpha \rightarrow \infty$: 무한대의 noise / 완전한 random / 상쇄시켜야 함 $\theta_{ij} \rightarrow \mathbf{0}$

$$q(w_{ij} | \theta_{ij}, \alpha) = \mathcal{N}(w_{ij} | \theta_{ij}, \alpha \theta_{ij}^2).$$

Sparse VD: For convolution layers

- Sparse VD for FC layers:

$$b_{mj} \sim \mathcal{N}(\gamma_{mj}, \delta_{mj})$$
$$\gamma_{mj} = \sum_{i=1}^I a_{mi} \theta_{ij}, \quad \delta_{mj} = \alpha_{ij} \sum_{i=1}^I a_{mi}^2 \theta_{ij}^2 = \sum_{i=1}^I a_{mi}^2 \sigma_{ij}^2$$

By additive reparam. trick

$$\alpha_{ij} \theta_{ij}^2 = \sigma_{ij}^2$$

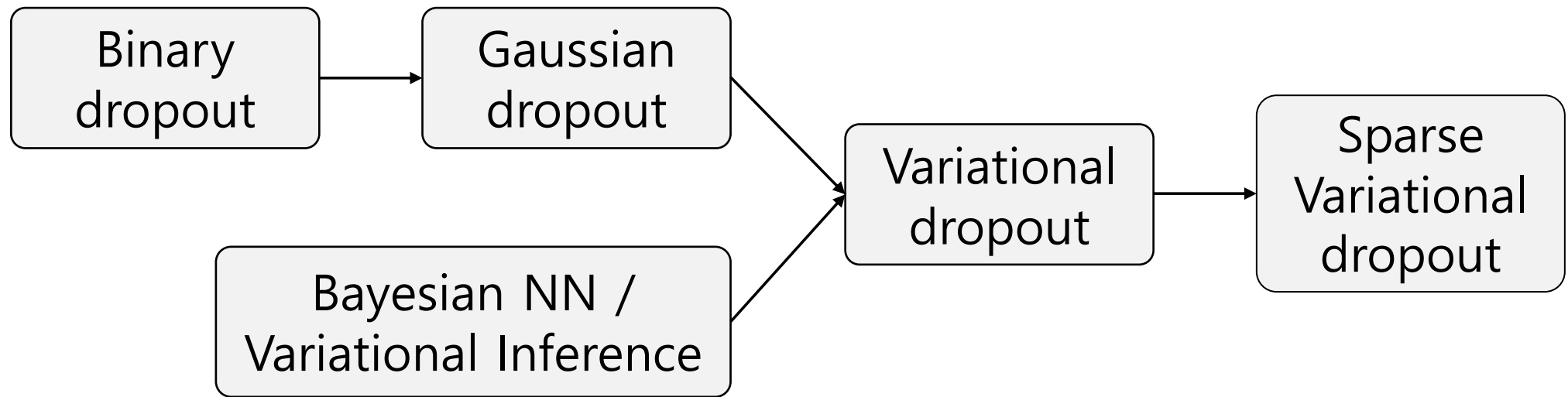
- Sparse VD for Conv layers:

$$\text{vec}(b_{mk}) \sim \mathcal{N}(\gamma_{mk}, \delta_{mk})$$
$$\gamma_{mk} = \text{vec}(A_m * \theta_k), \quad \delta_{mk} = \text{diag}(\text{vec}(A_m^2 * \sigma_k^2))$$

Sparse VD: Empirical Observations

- Test time에는?
 - 실제 완전히 드랍되는 경우는 없으므로 α 에 대한 thresholding이 필요
- Expected log likelihood term보다 KL term이 지배적인 경우가 더 일반적
 - 초반에 급격하게 높은 sparsity로 수렴하여 학습에 실패
 - 해결책? Pretraining or Scaling term 사용
- Prior 없이도 학습이 가능
 - 사전 지식없이 데이터만 보고 variance를 fitting시킬 수 있음

Big Picture



Implementation

논문저자 공개 (Theano, Lasagne)

- <https://github.com/senya-ashukha/variational-dropout-sparsifies-dnn>

다른 논문에서 활용 (TF / 저자 참여 / by Google AI research / 바로 사용하기 어려움)

- https://github.com/google-research/google-research/tree/master/state_of_sparsity

개인 repository (TF / 미검증)

- <https://github.com/cjratcliff/variational-dropout> (in progress)
- <https://github.com/BayesWatch/tf-variational-dropout> (incomplete)

Any questions?