Layer Normalization

Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton

23/03/31

서강대학교 정보통신대학원
배성은

목차

(0) Abstract

✓ 최신 심층 신경망 사용은 계산 비용이 많이 든다.
✓ 시간을 단축시키기 위해 정규화 기술을 사용
✓ 배치 정규화는 미니배치 크기에 따라 달라지는 종속적인 문제가 있음.

✓ 이 논문에선 배치정규화를 레이어 정규화로 변환함.

✓ 배치 정규화와는 달리 레이어 정규화는 훈련과 테스트 시점에 정확히 동일한 계산을 수행함.
✓ 레이어 정규화는 순환신경망의 hidden state 를 안정화시키는데 매우 효과적임
✓ 훈련 시간을 크게 줄일 수 있음을 보여줌.

(1) Introduction

✓ 최근엔 심층 신경망에 정규화 단계를 추가해 훈련 시간을 줄이기 위해 batch normalization이 제안됨
✓ 정규화는 훈련 데이터 전체의 평균과 표준편차를 사용하여 합산된 각 입력을 표준화함.
✓ 배치 정규화 효과는 batch size에 따라 달라지게된다. 또한 매우 큰 규모의 모델에는 적용할 수가 없다.
✓ 데이터가 너무 큰 경우에는 그걸 쪼갠 Mini Batch도 크기 때문.
✓ RNN에서는 시퀀스의 길이가 달라지는 경우가 많아 쓰기 어렵다.

✓ **Layer Normalization은 종속성이 발생하지 않고, RNN에 대해서도 잘 작동한다.**
(훈련 시간과 일반화 성능 모두 향상시킴)

(2) Background

✓ 특정 layer 가중치의 gradient가 이전 layer 출력값들에 영향을 크게 받는다.
✓ Batch normalization은 배치의 모든 샘플에 은닉층의 각 뉴런으로 들어오는 인풋들의 총합을 정규화하여 이 문제를 해결 (“covariate shift”를 줄이기 위해)

$$a_i^l = {w_i^l}^\top h^l \qquad h_i^{l+1} = f(a_i^l + b_i^l)$$

$$\bar{a}_i^l = \frac{g_i^l}{\sigma_i^l}\left(a_i^l - \mu_i^l\right) \qquad \mu_i^l = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[a_i^l\right] \qquad \sigma_i^l = \sqrt{\mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[\left(a_i^l - \mu_i^l\right)^2\right]}$$

✓ a_i^l : *lth* layer의 *ith* hidden unit으로 들어가는 인풋 총합의 정규화 값

✓ g_i^l : 비선형 활성화 함수 이전에 정규화 활성화의 크기를 결정하는 값
μ, σ : mini-batch의 샘플들을 이용해 추정된다.

(3) Layer normalization

✓ Batch Normalization의 단점을 극복하기 위해 구상됨.

✓ **Covariate Shift** : 특정 layer output의 변화가 다음 layer로의 인풋 총합에 correlated 변화를 크게 일으킨다.

✓ 이러한 covariate shift 문제는, 각 layer에서의 인풋 총합의 mean과 variance를 고정시킴으로써 해결할 수 있다.

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \qquad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^l - \mu^l\right)^2}$$

✓ 각 input의 feature들에 대한 평균과 분산을 구해 각 층의 input을 정규화

- Layer Normalization에서는 같은 layer에 있는 모든 은닉층 유닛들이 정규화 통계량(μ, σ)를 공유한다
- 하지만 배치의 각 훈련 샘플은 서로 다른 정규화 통계량을 갖는다
- mini-batch의 크기에 영향을 받지 않기 때문에 batch size 1의 온라인 학습에도 이용될 수 있다

(3) Layer normalization

✓ Layer normalized recurrent neural networks

✓ 자연어 처리작업에선 훈련마다 문장길이가 다 다른데, rnn은 모든 time-step에서 같은 weight를 사용해 이러한 문제를 쉽게 처리함.
✓ 하지만 batch normailzation을 적용하려면 시퀀스의 각 time-step마다 다른 통계량을 계산하고 저장하는게 필요함.
✓ 테스트 문장길이가 학습 문장들보다 길면 통계량을 이용할 수 없는 문제가 발생됨.
✓ 반면 layer 정규화는 현재 time-step에서 특정 layer로 들어오는 input 총 합의 통계량에 기반해 위와같은 문제가 생기지 않음.
✓ 이 방식은 모든 time-step에 대해서 한 쌍의 gain, bias 파라미터만 공유

$$\mathbf{h}^t = f\left[\frac{\mathbf{g}}{\sigma^t} \odot \left(\mathbf{a}^t - \mu^t\right) + \mathbf{b}\right] \qquad \mu^t = \frac{1}{H}\sum_{i=1}^{H} a_i^t \qquad \sigma^t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^t - \mu^t\right)^2}$$

(4) Related work

- ✓ 이전에 Batch Normalization을 순환신경망(RNN)으로 확장했던 연구가 있음.
- ✓ 각 time-step마다 독립적인 normalization통계량을 사용해, gain파라미터를 0.1로 초기화해서 최상의 성능을 얻음.

- ✓ Weight normalization
- ✓ Variance 대신에 앞 단 weight들의 L2 norm이 뉴런 인풋 총합을 정규화하는 것에 이용

- ✓ 위의 두 방식은 기존 feed-forward network에 다른 parameterization을 적용한 것과 같다고 볼 수 있음.

- ✓ 하지만 Layer Normalization은 이러한 re-parameterization이 아닌, 특별한 invariance 성질을 가진다는 것에서 다름.

(5) Analysis

- ✓ 5.1 Invariance under weights and data transformations
- ✓ Layer Normalization이 Batch Normalization, Weight Normalization 관련이 있긴 하다.
- ✓ 뉴런의 인풋을 두 스칼라 값(μ, σ)으로 정규화한다.
- ✓ 정규화 후 뉴런에 대한 adaptive bias b와 gain g를 학습함.

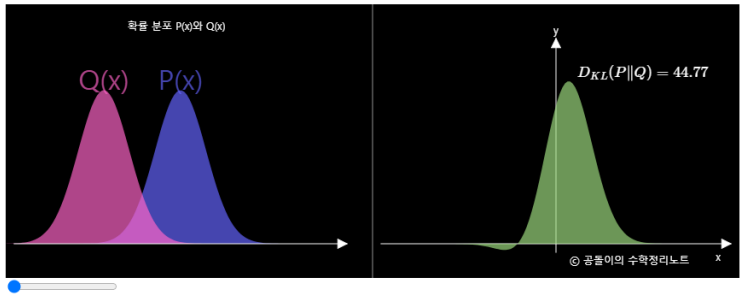
$$h_i = f(\frac{g_i}{\sigma_i}(a_i - \mu_i) + b_i)$$

- ✓ Weight re-scaling and re-centering
  - ✓ Batch, weight Normalization에서 가중치 재조정에 대해 불변이다.
  - ✓ 반면, Layer Normalization은 단일 가중치 벡터 개별 스케일링에 불변하지 않는다.
  - ✓ 대신 전체 weight matrix의 rescaling과 shift에 불변하다.

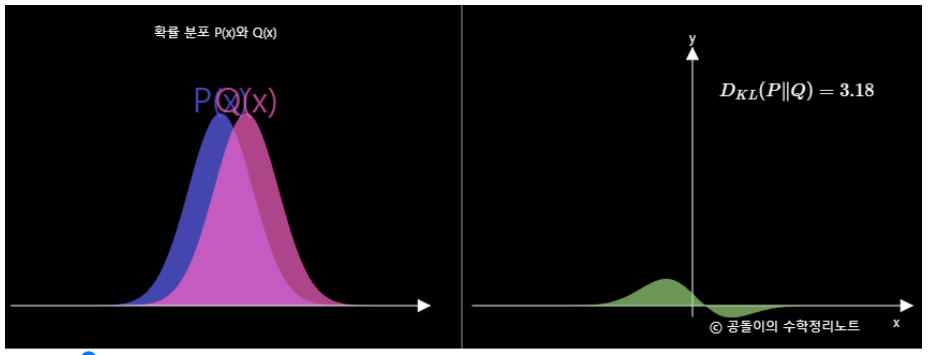
- ✓ Data re-scaling and re-centering
  - ✓ 모든 정규화 기법이 데이터 셋의 re-scaling에 invariance하다.
  - ✓ Layer Normalization은 개별 훈련 샘플의 re-scaling에도 invariance하다.

(5) Analysis

- ✓ 5.2 Geometry of parameter space during learning
- ✓ 이론상으로는 위와 같이 안정되더라도, 실제훈련에서는 매우 다르게 작동할 수 있음
- ✓ 이 섹션에서는 기하학적인 분석을 통해 정규화가 암묵적으로 학습속도를 낮추고 학습을 안정적으로 만들 수 있음을 보여줌.
  - ✓ Riemannian metric
    - ✓ 확률 모델의 학습 가능한 파라미터들은 '가능한 모든 입출력 관계를 지닌' 부드러운 manifold를 만듬
    - ✓ 확률 모델의 manifold 두 점 사이 거리를 측정할 때는 kullback-Leibler divergence를 사용.
    - ✓ 이때의 parameter space를 Riemannian manifold라고 함.
    - ✓ 직관적으로 이 지표는 접선 방향을 따라 모델 출력값 변화를 측정.



KL-divergence의 시각화.
파란색 함수와 빨간색 함수를 각각 $P(x)$, $Q(x)$라고 했을 때, 초록색 함수에 대한 넓이 합이 KL-divergence 값 $D_{KL}(P\|Q)$을 의미한다.



(5) Analysis

✓ 5.2 Geometry of parameter space during learning

✓ The geometry of normalized generalized linear models

✓ 요약

✓ invariance 성질에 의해서 normalized weight matrix를 거쳐도 값이 변하지 않음

✓ Wi방향으로 Fij에서 σ_i.곡률도 함께 반비례하게 됨.

✓ Weight vector norm이 커지면 곡률이 감소 -> learning rate 낮아서 early stopping이 되서 학습이 됨.

✓ Weight vector norm이 작아지면 곡률이 커져서 학습률이 커져서 gradient가 안정화됨.

$\theta = \mathrm{vec}([W, \mathbf{b}, \mathbf{g}]^\top)$:

$$\bar{F}(\theta) = \begin{bmatrix} \bar{F}_{11} & \cdots & \bar{F}_{1H} \\ \vdots & \ddots & \vdots \\ \bar{F}_{H1} & \cdots & \bar{F}_{HH} \end{bmatrix}, \quad \bar{F}_{ij} = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})} \left[\frac{\mathrm{Cov}[y_i,\, y_j \mid \mathbf{x}]}{\phi^2} \begin{bmatrix} \frac{g_i g_j}{\sigma_i \sigma_j} \chi_i \chi_j^\top & \chi_i \frac{g_i}{\sigma_i} & \chi_i \frac{g_i (a_j - \mu_j)}{\sigma_i \sigma_j} \\ \chi_j^\top \frac{g_j}{\sigma_j} & 1 & \frac{a_j - \mu_j}{\sigma_j} \\ \chi_j^\top \frac{g_j (a_i - \mu_i)}{\sigma_i \sigma_j} & \frac{a_i - \mu_i}{\sigma_i} & \frac{(a_i - \mu_i)(a_j - \mu_j)}{\sigma_i \sigma_j} \end{bmatrix} \right] \tag{13}$$

$$\chi_i = \mathbf{x} - \frac{\partial \mu_i}{\partial w_i} - \frac{a_i - \mu_i}{\sigma_i} \frac{\partial \sigma_i}{\partial w_i}. \tag{14}$$

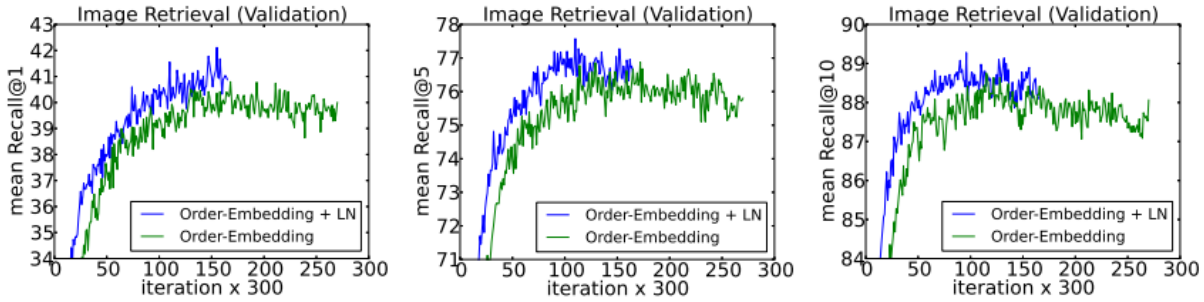
(6) Experimental results

✓ 연구진은 Recurrent Neural Network에 집중하여 다음 6개의 task에 대해서 실험을 진행했다.

- Image-sentence ranking
- question-answering
- contextual language modelling
- generative modelling
- handwriting sequence generation
- MNIST classification

(6) Experimental results

✓ 6.1 Order embeddings of images and language
✓ 이미지와 문장의 joint embedding space를 학습하는 order-embedding모델을 이용한 실험.



(a) Recall@1 (b) Recall@5 (c) Recall@10

Figure 1: Recall@K curves using order-embeddings with and without layer normalization.

	MSCOCO							
	Caption Retrieval				Image Retrieval			
Model	**R@1**	**R@5**	**R@10**	**Mean *r***	**R@1**	**R@5**	**R@10**	**Mean *r***
Sym [Vendrov et al., 2016]	45.4		88.7	5.8	36.3		85.8	9.0
OE [Vendrov et al., 2016]	46.7		88.9	5.7	37.9		85.9	8.1
OE (ours)	46.6	79.3	89.1	5.2	37.8	73.6	85.7	7.9
OE + LN	**48.5**	**80.6**	**89.8**	**5.1**	**38.9**	**74.3**	**86.3**	**7.6**

Table 2: Average results across 5 test splits for caption and image retrieval. **R@K** is Recall@K (high is good). **Mean *r*** is the mean rank (low is good). Sym corresponds to the symmetric baseline while OE indicates order-embeddings.

(6) Experimental results

✓ 6.2 Teaching machines to read and comprehend
✓ Layer Normalization과 Batch Normalization을 비교하기 위한 실험
✓ 실험 내용은 question-answering task로, 질문이 주어지면 빈칸을 채우는 방식으로 답변해야 한다.

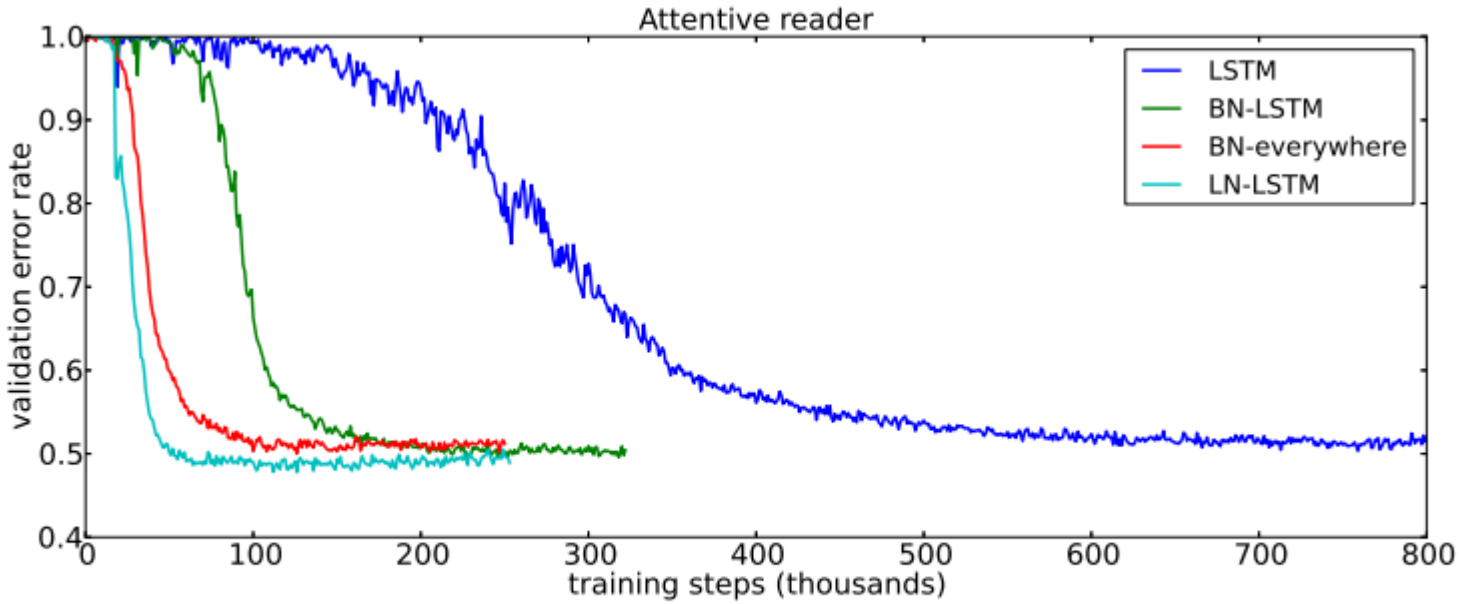
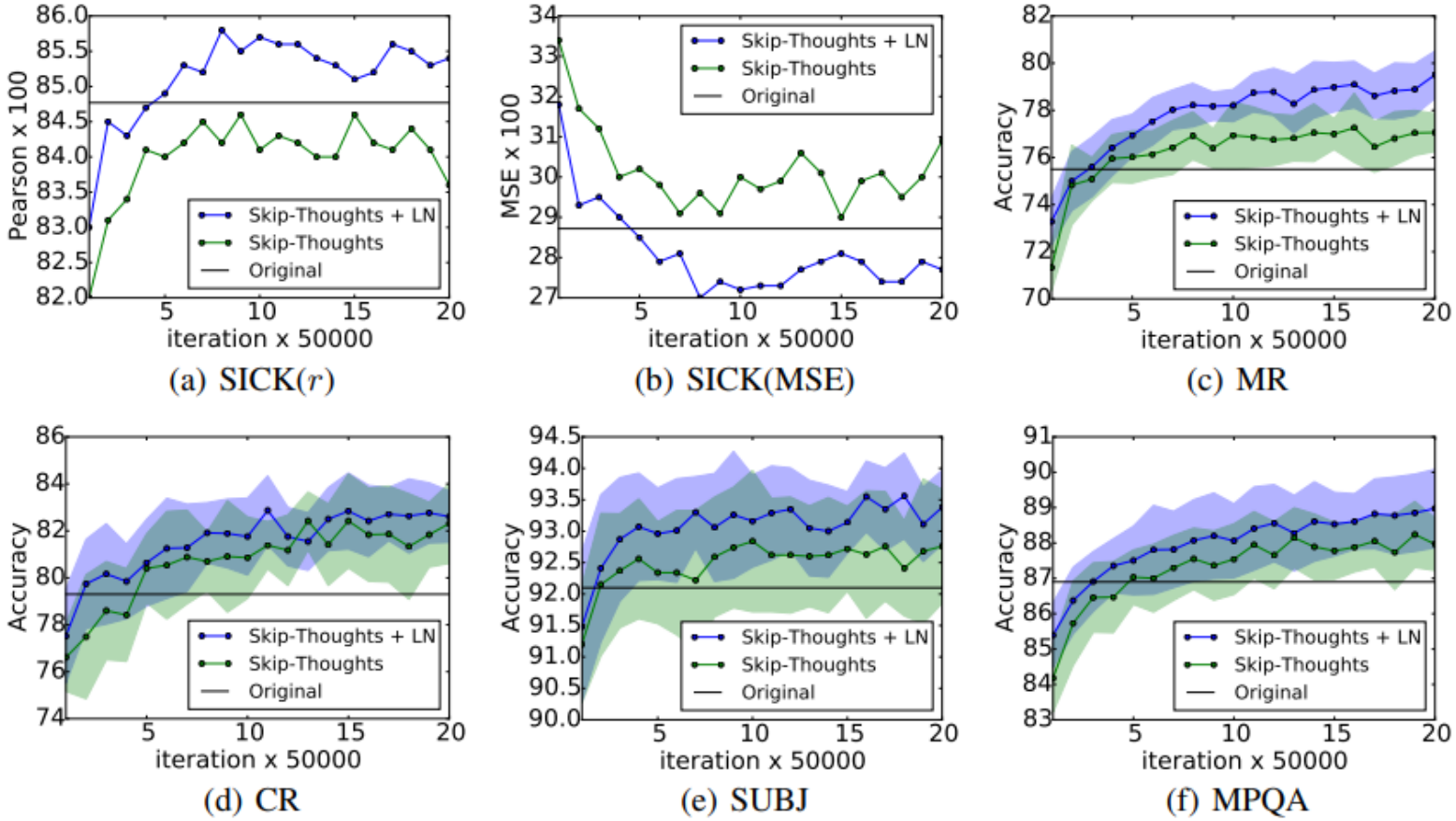


Figure 2: Validation curves for the attentive reader model. BN results are taken from [Cooijmans et al., 2016].

(6) Experimental results

- ✓ 6.3 Skip-thought vectors
- ✓ 인접하는 단어가 주어지면 문장은 encoder RNN에 의해 인코딩되고 decoder RNN이 주변 문장들을 예측하는데 이용



(7) Conclusion

- ✓ Neural Network의 훈련 속도를 가속시킬 수 있는 Layer Normalization을 소개하였다
- ✓ 다른 정규화 기법들과 invariance 성질을 이론적으로 분석하였다
- ✓ Layer Normalization은 단일 훈련 샘플의 shifting, scaling에 invariant함을 보였다
- ✓ 긴 문장과 작은 사이즈의 mini-batch가 주어졌을 때 RNN이 layer normalization의 효능을 더 많이 받음을 실험적으로 보였다