연구지도 및 연구윤리 13주차 보고서

(Chapter6. 배치 정규화, CNN)

경북대학교 전자공학부
2016113566 김남영

1) 첨부된 배치정규화 논문 분석

심층 신경망을 트레이닝 하는 것은 이전 레이어의 파라미터가 변경될 때 각 레이어의 입력 분포가 바뀐다는 사실 때문에 매우 복잡한 일이다. 이것을 Internal Covariate Shift라고 부르며, 이는 낮은 학습률을 요구하고, 파라미터 초기화를 신중하게 해야하므로 학습을 느리게 만든다. 이러한 Internal Covariate Shift를 없애는 메카니즘을 배치 정규화라고 부르며 레이어 입력의 mean을 0으로, variance를 1로 고정함으로써(whitening) 실현이 가능하다. 배치 정규화는 Internal Covariate Shite를 줄여 훈련 속도를 가속화 시키며, 높은 학습률을 가지더라도 발산을 덜 할 수 있게 만든다. 게다가, Dropout의 필요성을 줄이고 오버피팅에 빠지는 것을 예방하여 비선형적인 성질의 실현을 가능하게 한다.

올바른 배치 정규화를 실현하기 위해서 데이터 전체에 대한 mean과 variance를 정규화 하는 것이 아니라 mini-batch 단위로 각각 mean과 variance를 정규화 한다. Fig. 1에서 배치 정규화 알고리즘을 확인할 수 있다. 마지막 줄의 γ는 확대(scale), β는 이동(shift)을 담당한다. 이 두 값은 학습하면서 적합한 값으로 조정된다.

전통적인 심층 신경망에서 너무 높은 학습률은 기울기 소실이나 발산을 야기할 수 있었다. 일반적으로 높은 학습률은 레이어 파라미터의 scale을 증가시키고 이것은 역전파시 기울기의 증폭을 유발하지만, 배치 정규화를 사용하면 역전파는 그것의 파라미터의 scale에 영향을 받지 않으므로 학습률을 높여도 괜찮다. 따라서 빠른 학습이 가능해 진다. 또한 배치 정규화에서도 오버피팅을 줄이는 효과를 발견할 수 있으므로 Dropout을 줄이거나 제거할 수 있다. 이러한 내용에 대한 실험적 결과는 Section 4에서 확인할 수 있다.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Fig. 1. Batch Normalizing Transform

2) LeNet과 AlexNet 구현 및 결과 분석(코드 한 줄씩 설명)

아래 Fig. 2와 Fig 3의 architecture를 참고해서 여태껏 배운 layer들을 결합하여 network를 구현하려 했으나 입력으로 받는 파라미터 값을 제대로 설정하지 못해서 구현에 실패했습니다.

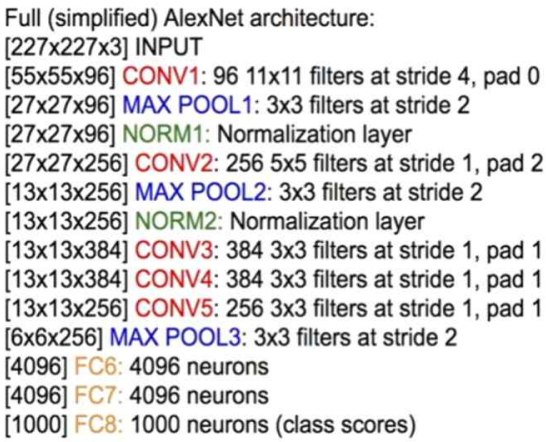

Fig. 2. AlexNet architecture

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

Fig. 3. LeNet-5 architecture

3) 고찰

1. 책의 코드를 참고해서 network를 구현하다가 혼자서 구현하려니 어려움이 많았다.
2. 특히 입출력으로 어떤 값이 전달되는지가 문제가 되었다.
3. 7장 CNN에 관련된 내용이 전체적으로 숙지가 덜 되었다는 느낌이 든다.
4. 갈수록 중요한 부분을 배우는데, 다른 전공 공부에 시간을 할애하느라 깊게 공부하지 못했다. 종강을 하면 부족했던 부분들을 다시 한 번 복습해야겠다.