Lecture 7-0

Title	Training Neural Networks II
slide	http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture7.pdf

복습

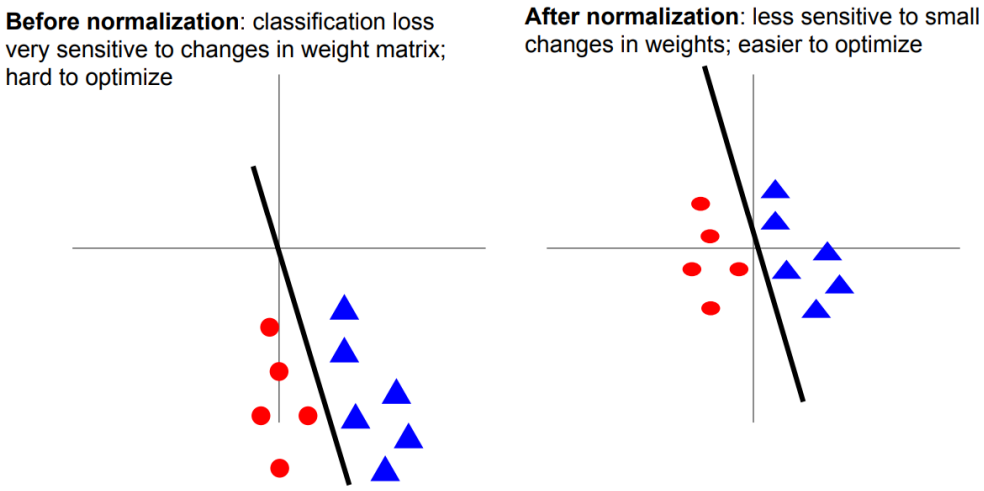
지난 시간에는 Nerural networks를 학습 시킬 때 필요한 여러가지 중요한 것들을 배웠다.

1. Activation Function

   다양한 Activation Function과 각각의 특성을 배웠다. 10년 전에는 sigmoid가 유명했지만, Vanishing gradients가 생기는 문제로 인해 요즘은 대부분 ReLU를 사용한다.

2. 가중치 초기화

   가중치가 지나치게 작으면 작은 값이 여러 번 곱해지기 때문에 점점 0이 되어서 activation이 사라지고, 결국 모든 값이 0이 되고 학습은 일어나지 않는다. 반면에 가중치가 너무 큰 값으로 초기화되면 그 값이 계속 곱해지게 되어 결국 폭발해버려서, 이 경우에도 학습이 일어나지 않는다. Xavier/MSRA(HE) Initialzation 같은 방법으로 초기화를 잘 시켜주면 Activation의 분포를 좋게 유지 시킬 수 있다. 명심해야 할 점은 Network가 깊어질수록 가중치를 더 많이 곱하게 되기 때문에 더욱 중요하다는 것이다.



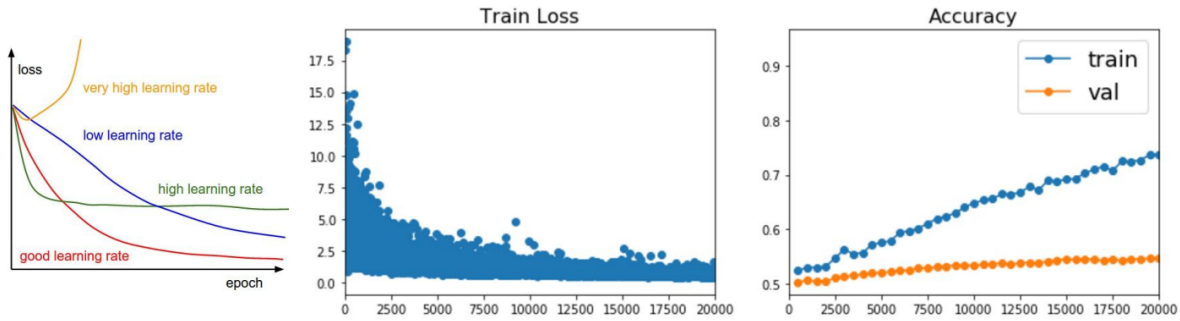
3. 데이터 전처리

   CNN은 zero-mean을 주로 사용하며, 그 밖에 zero-mean, unit variance에 대해서도 배웠다. 우리가 이것을 왜 해야 하는지 좀 더 직관적으로 얘기해보면, 예를 들어 빨간/파란 점들을 나누는 Binary classification 문제를 풀고자 한다고 하자. 왼쪽의 경우 not

normalized/not centered 데이터로, classification이 가능하지만 선을 조금만 움직여도 classification이 잘 되지 않는다. 이것이 의미하는 것은 Loss가 파라미터에 너무 민감하기 때문에 손실 함수가 아주 약간의 가중치 변화에도 엄청 예민해서 동일한 함수를 쓰더라도 학습 시키기 아주 어렵다는 것이다. 반면 오른쪽은 데이터의 중심을 원점에 맞추고(zero-center), Unit variance로 만들어 준 경우이다. 오른쪽에서 손실 함수는 가중치의 변동에 덜 민감하여 최적화가 더 쉽고, 학습이 더 잘된다. 이것은 Linear classification의 경우에만 국한되는 것이 아니라, Neural network 내부에도 다수의 (interleavings) linear classifer가 있다고 생각할 수 있으므로, 이 경우에도 Neural network의 입력이 zero-centered가 아니고 Unit variance가 아닌 경우라면 레이어의 Weight matrix가 아주 조금만 변해도 출력은 엄청 심하게 변하게 되고, 이는 학습을 어렵게 한다.

4. batch normalization

   Normalization이 매우 중요하기 때문에 batch normalization에 대해서도 배웠다. 이는 activations이 zero mean과 unit variance가 될 수 있도록 레이어를 하나 추가하는 방법이다. BN에서는 forward pass 시에 미니 배치에서의 평균과 표준편차를 계산해서 Normalization을 수행하고 레이어의 유연한 표현성(expressivity)을 위해서 scale, shift 파라미터를 추가했다.



5. 학습 과정 다루기

   학습 도중 Loss curve가 어떻게 나타나야 하는지도 배웠다. 가운데 그래프는 시간에 따른 Loss 값을 나타낸다. 네트워크가 Loss를 줄이고 있으면 잘 하고 있는 것이다. 맨 오른쪽 그래프를 보면 X는 시간이고 Y는 성능 지표이다. Training/Validation set의 성능 지표를 나타낸다. Training set의 성능은 계속 올라가고, Loss도 계속 내려간다. 그러나 validation은 침체하고 있으므로, 이런 경우는 overfititing된 것으로 추가적인 regularization이 필요하다.

6. hyperparameter search

   네트워크에는 무수히 많은 하이퍼파라미터가 존재하는데, 이것을 올바르게 잘 선택하는 것은 상당히 중요하다. 성능이 특정 하이퍼파라미터에 의해 크게 좌우될 때 그 파라미터를 좀 더 넓은 범위로 탐색할 수 있기 때문에 이론상 random search가 grid search보

다 더 좋다. 또한 하이퍼파라미터 최적화 할 때는 coarse search 이후에 fine search를 한다. [coarse search]처음에는 하이퍼파라미터를 조금 더 넓은 범위에서 찾고, Interation도 작게 줘서 학습 시킨다. [fine search] 그리고 결과가 좋은 범위로 좁히고, iterations를 조금 더 돌면서 더 작은 범위를 다시 탐색한다. 적절한 하이퍼파라미터를 찾을 때 까지 이 과정을 반복한다. 가장 중요한 점은 coarse range를 설정할 때 가능한 최대한 넓은 범위를 설정해 줘서, 그 범위가 하이퍼파라미터 범위의 끝에서 끝까지 다 살펴볼 수 있도록 해야 한다는 것이다.

이번에는 더 강력한 최적화 알고리즘에 대해서 자세히 알아보는 시간을 갖는다. 지난 강의에 Regularization에 대해서 배웠는데, Regularization은 네트워크의 Train/Test Error간의 격차를 줄이고자 사용하는 추가적인 기법이다. Neural Network에서 실제로 사용하고 있는 Regularization 전략에 대해서 다뤄보고, 원하는 양 보다 더 적은 데이터만을 가지고 있을 때 사용할 수 있는 방법인 Transfer learning에 대해서도 배울 것이다.