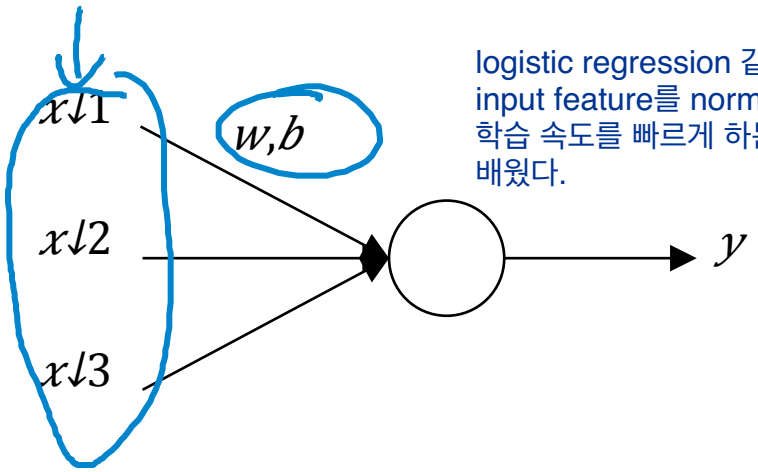
deeplearning.ai

Batch Normalization

Normalizing activations in a network

Normalizing inputs to speed up learning



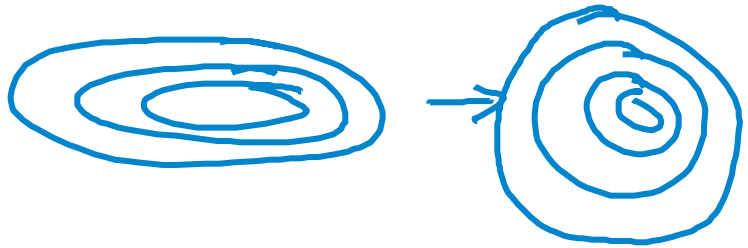
logistic regression 같은 모델을 학습할 때 input feature를 normalization하는 것이 학습 속도를 빠르게 하는데 도움이 된다는걸 배웠다.

$$\mu = \frac{1}{m}\sum_i x^{(i)}$$

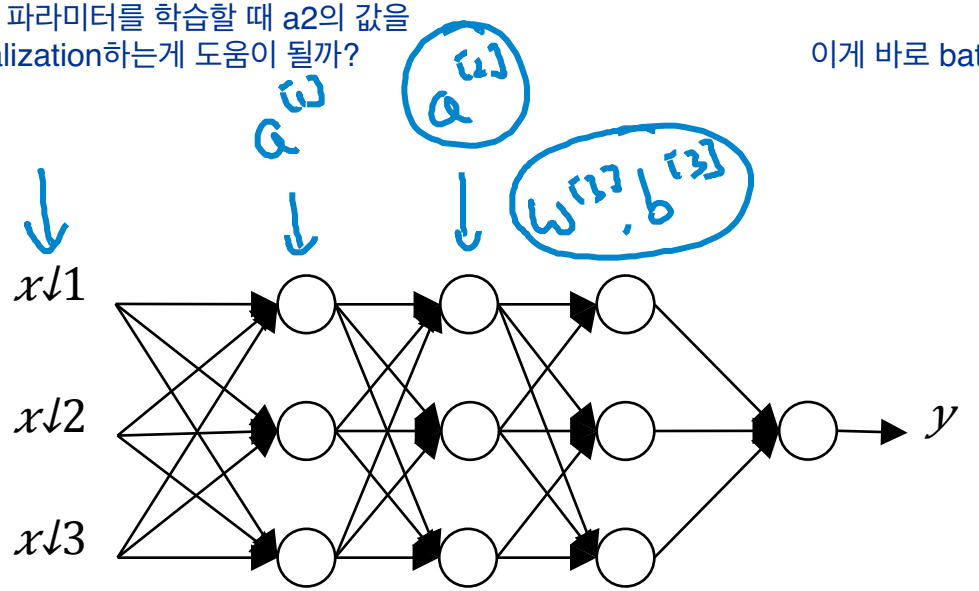
$$X = X - \mu \quad \leftarrow \text{element-wise}$$

$$\sigma^2 = \frac{1}{m}\sum_i x^{(i)2}$$

$$X = X / \sigma^2$$



그렇다면 deeper model에서는 어떨까? 즉, w3,b3 파라미터를 학습할 때 a2의 값을 normalization하는게 도움이 될까?



이게 바로 batch normalization의 기본 아이디어다

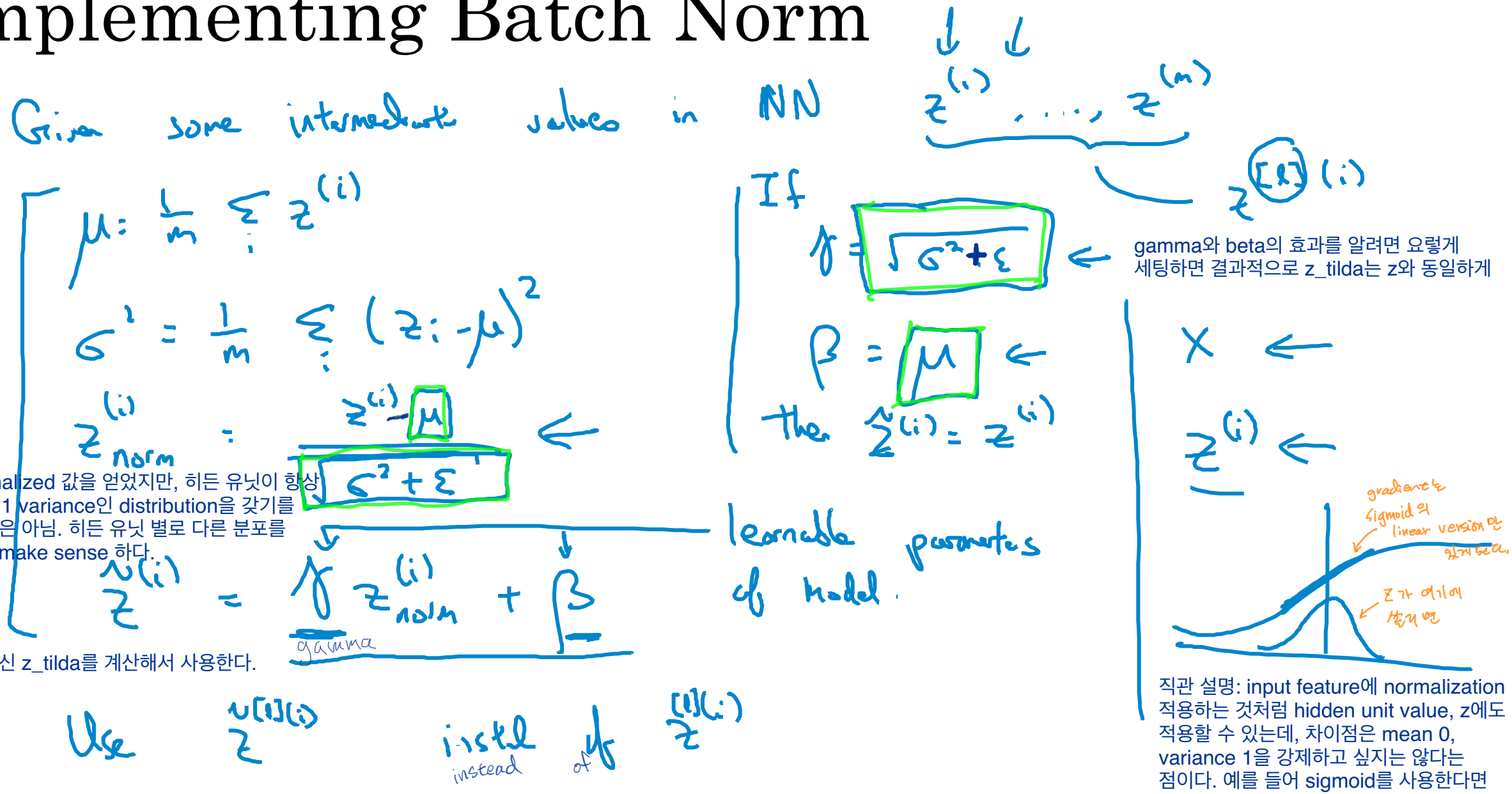
Can we normalize $a^{[2]}$ so as to train $w^{[3]}, b^{[3]}$ faster?

Normalize $z^{[2]}$

실제로는 activation function 적용하기 전 z의 값을 normalize한다. (a를 normalize 할 것인지 z를 normalize 할 것인지에 대한 논쟁은 있음)

Andrew Ng

Implementing Batch Norm



z의 normalized 값을 얻었지만, 히든 유닛이 항상 0 mean, 1 variance인 distribution을 갖기를 원하는 것은 아님. 히든 유닛 별로 다른 분포를 가지는게 make sense 하다.

그래서 대신 z_tilda를 계산해서 사용한다.

gamma와 beta의 효과를 알려면 요렇게 세팅하면 결과적으로 z_tilda는 z와 동일하게

직관 설명: input feature에 normalization 적용하는 것처럼 hidden unit value, z에도 적용할 수 있는데, 차이점은 mean 0, variance 1을 강제하고 싶지는 않다는 점이다. 예를 들어 sigmoid를 사용한다면 non-linearity of the sigmoid function의 이점을 활용하고 싶다는 것이다.

Andrew Ng