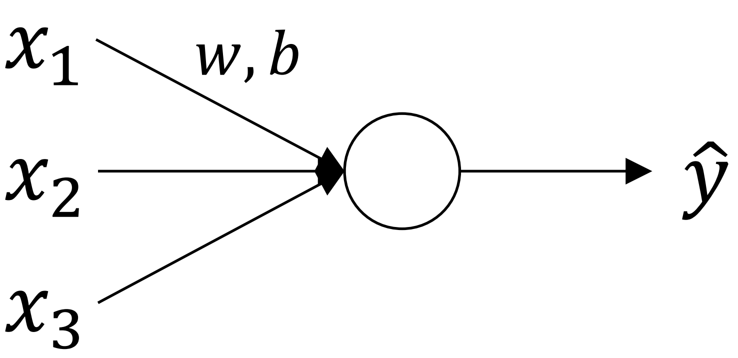
Batch Normalization

deeplearning.ai

Normalizing activations in a network

(Allows you to set hyper params more easily - compared to manual selection)

Normalizing inputs to speed up learning



Till now, we were only normalizing the i/p layer,

ie, $\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$

$X = X - \mu$

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

$X = X / \sigma^2$

What if we could normalize the Activation inputs ($a^{[1]}, a^{[2]} \ldots$) as well?

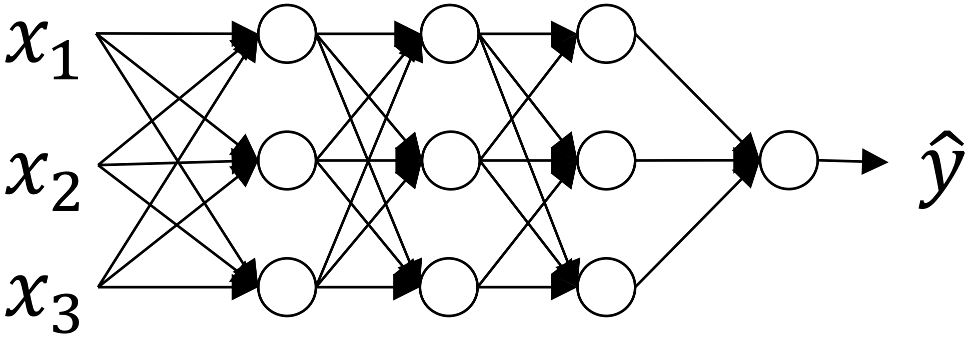
Why?

↳ so that the training of inner w & bs, say $W^{[3]}, b^{[3]}$ would be more efficient

to

$a^{[1]}$ $a^{[2]}$ $a^{[3]}$

→ Normalize $a^{[2]}$ to make training of $W^{[3]}, b^{[3]}$ more efficient



↳ $z^{[3]} = W^{[3]} \underbrace{a^{[2]}}_{\text{Normalize this}} + b^{[3]}$

Normalizing $a^{[2]}$ ≈ Normalizing $z^{[2]}$

So lets Normalize $Z^{[2]}$

Andrew Ng

Implementing Batch Norm

Given some intermediate hidden unit values in the NN in layer l

$\hookrightarrow z^{(1)}, z^{(2)} \dots z$

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

$$\sigma^2 = \frac{1}{m} \sum_i (z_i - \mu)^2$$

$$z^{(i)}_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

$\hookrightarrow$ ε to avoid $\sqrt{0}$

- This gives mean=0, var=1 for z
- But maybe we dont want mean=0, var=1 for z

$$\therefore \tilde{z}^{(i)} = \gamma \cdot z^{(i)}_{norm} + \beta$$

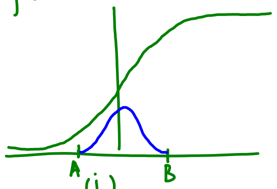
γ, β: learnable params of the model (just like the weight/bias)

Why write z like this?

- It allows z to have whatever mean we want it to have

Why?

Imagine you have sigmoid Activation func



for $z^{(i)}$

If all of z's in all layers had mean=0, var=1, the model's prediction may become limited as you will only capture activations from A to B - why?

Because all zs lie b/w A & B

$\Rightarrow g(z) \in (.2, .8)$ for example

which limits predictability, y

this rule is applied to all layers

If $\gamma = \sqrt{\sigma^2 + \varepsilon}$

& $\beta = \mu$

then, we would get back

$\tilde{z}^{(i)}$ = original $z^{(i)}$ with mean μ, var = σ^2

Now, because $\tilde{z}^{(i)}$ is more flexible than $z^{(i)}$ $\therefore$ we will use $\tilde{z}^{(i)}$ in ALL FUTURE COMPUTATIONS of z in layer l

Andrew Ng