Batch normalization

can we normalize $a^{[l]}$ so as to train $w^{[l]}, b^{[l]}$ faster ?

We normally normalize $z^{[l]}$

Given some intermediate values in NN

$$z^{[l](1)}, \dots, z^{[l](m)}$$

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

$$\sigma^2 = \frac{1}{m}\sum_i \left(z^{[l](i)} - \mu\right)^2$$

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

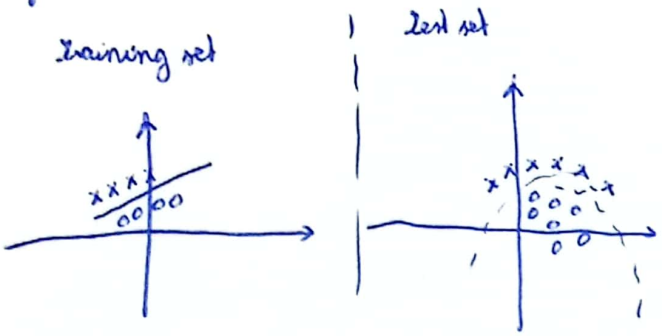
$$\tilde{z}^{[l](i)} = \gamma^{[l]} \cdot z^{[l](i)}_{norm} + \beta^{[l]}$$

($\gamma^{[l]}$, $\beta^{[l]}$: learnable parameters of the model)

instead $\tilde{z}^{[l](i)}$ instead of $z^{[l](i)}$

---

if we have :



data distribution changing from train to test

$$w^{[1]}, b^{[1]} \rightarrow z^{[1]} \xrightarrow[\text{batch norm}]{} \tilde{z}^{[1]} \rightsquigarrow a^{[1]} = g^{[1]}\left(\tilde{z}^{[1]}\right)$$

for t in 1 ..... num Mini batches

- compute forward prop on $X^{\{t\}}$
- In each hidden layer use Batch norm to replace $z^{[l]}$ with $\tilde{z}^{[l]}$
- use backprop to compute $dw^{[l]}, d\beta^{[l]}, d\gamma^{[l]}$
- update parameters : $w^{[l]}, b^{[l]}, \beta^{[l]}, \gamma^{[l]}$

Batch norm as a regularization

- each mini batch is scaled by the mean/variance computed on just that mini-batch
- this adds some noise to the values $z^{[l]}$ within that minibatch.
  - ⇒ similar to dropout, it adds some noise to each hidden layer's activations
- ⇒ this has a slight regularization effect.

Batch norm at test time

we don't have mini batches to add noise when calculating $z^{[i]}$

⇒ we use exponentially weighted average to estimate μ and σ^2