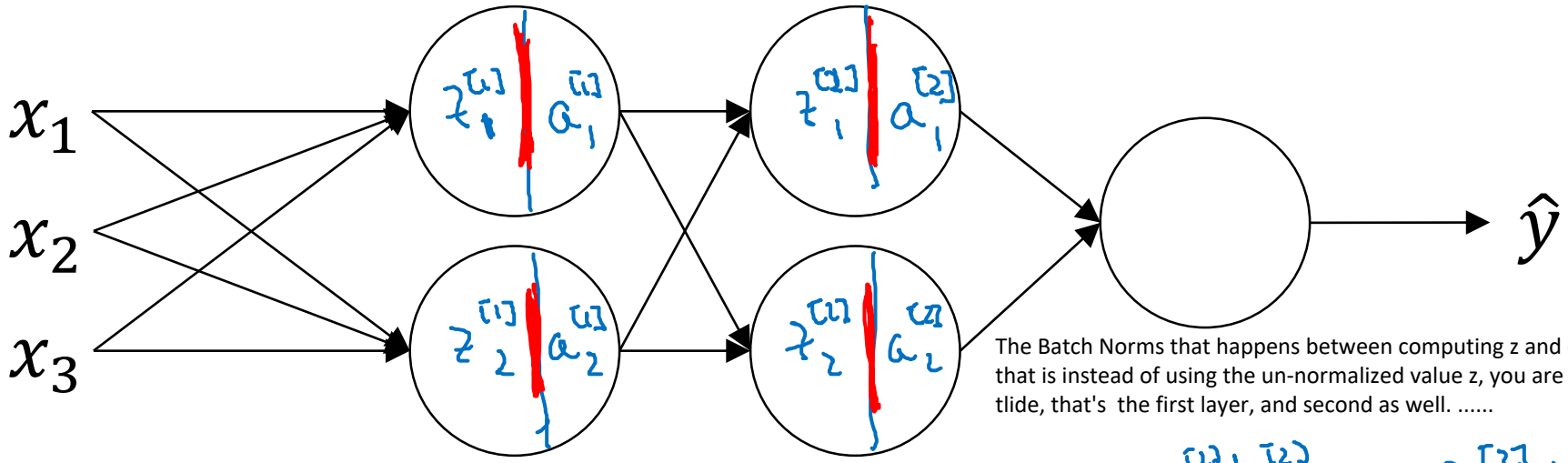
deeplearning.ai

Batch Normalization

Fitting Batch Norm into a neural network

Adding Batch Norm to a network



The Batch Norms that happens between computing z and computing a. And the intuition of that is instead of using the un-normalized value z, you are gonna to use the normalized value z tlide, that's the first layer, and second as well. .....

$X \xrightarrow{W^{[1]}, b^{[1]}} z^{[1]} \xrightarrow[\text{Batch Norm (BN)}]{\beta^{[1]}, \gamma^{[1]}} \tilde{z}^{[1]} \rightarrow a^{[1]} = g^{[1]}(\tilde{z}^{[1]}) \xrightarrow{W^{[2]}, b^{[2]}} z^{[2]} \xrightarrow[BN]{\beta^{[2]}, \gamma^{[2]}} \tilde{z}^{[2]} \rightarrow a^{[2]} \rightarrow \dots$

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, \dots, W^{[L]}, b^{[L]}, \beta^{[1]}, \gamma^{[1]}, \beta^{[2]}, \gamma^{[2]}, \dots, \beta^{[L]}, \gamma^{[L]}$

$d\beta^{[l]}$

$\beta^{[l]} = \beta^{[l]} - \alpha\, d\beta^{[l]}$

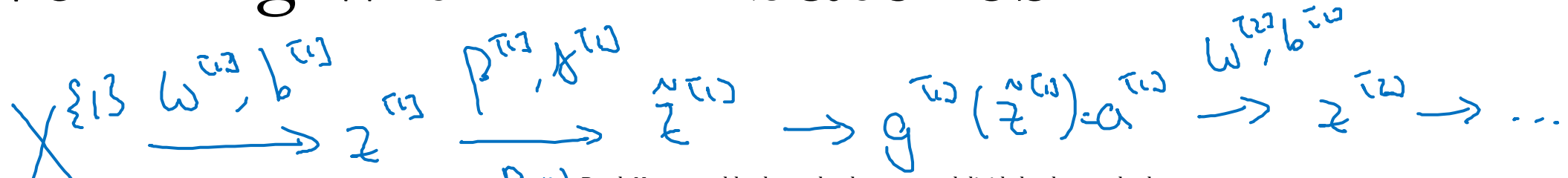
tf.nn.batch_normalization

β: It has nothing to do with the hyperparameter beta that we had for momentum or for computing the variance exponentially weighted average.

Andrew Ng

Working with mini-batches

Batch Norm is usually applied with mini-batches of your training set. So the way you apply Batch Norm is you take your first mini-batch and computer z1...

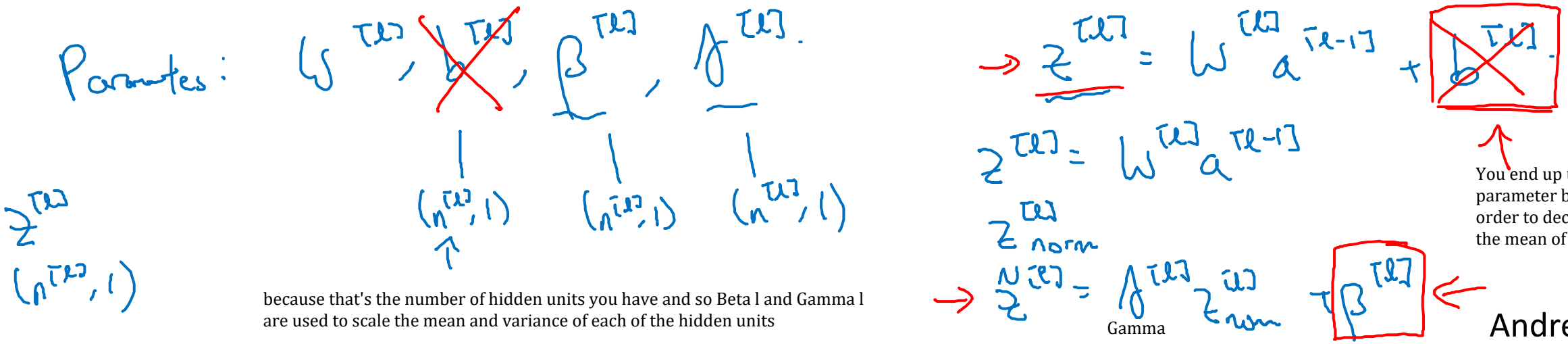


Batch Norm would subtract by the mean and divide by the standard deviation, and rescale by beta1, gamma.

It going to look at the mini-batch and normalize zl first to mean 0 and standard variance, and then rescale by Beta and Gamma.

any constant you add will get canceled out by mean subtractions step. So if you are using batch norm, you can actually eliminate that parameter. Or if you want, think of it as setting it permanently to 0.

kind of replaced with beta l which is the parameter that controls that ends up affecting the shift or the bias terms.



You end up using this parameter beta L in order to decide what's the mean of z tlide l.

because that's the number of hidden units you have and so Beta l and Gamma l are used to scale the mean and variance of each of the hidden units

Gamma

Andrew Ng

Implementing gradient descent

for $t = 1 \ldots$ numMiniBatches

Compute forward prop on $X^{\{t\}}$.

In each hidden layer, use BN to replace $z^{[l]}$ with $\tilde{z}^{[l]}$.

Use backprop to compute $dW^{[l]}$, ~~$db^{[l]}$~~, $d\beta^{[l]}$, $d\gamma^{[l]}$

Update parameters $W^{[l]} := W^{[l]} - \alpha\, dW^{[l]}$

$\beta^{[l]} := \beta^{[l]} - \alpha\, d\beta^{[l]}$

$\gamma^{[l]} := \ldots$

} ← gradient descent

Works w/ momentum, RMSprop, Adam.

And so then it ensures that within that mini-batch, the value z end up with some normalized mean and variance of the version of z tlide l.

you can use the updates given by other optimization algorithms
Some of these other optimization algorithms as well can be used to update the parameters beta and gamma that Batch Norm added to algorithm