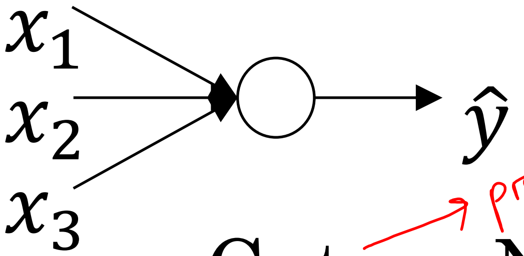
deeplearning.ai

Batch Normalization

Why does Batch Norm work?

We said, it is good practice to normalize hidden units as well as i/p layer, this will keep all values in the same Range & make optimization faster  v/s 

Learning on shifting input distribution



problem - we trained all data w/ black cats

Cat — $y = 1$

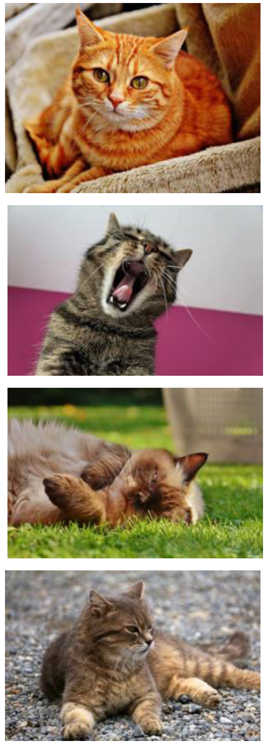
Non-Cat — $y = 0$

This phenomena of data distribution changing from train to test is called "covariance shift"

testing w/ colored cats.

Q why is this Imp? How does this apply to DNN or hidden unit normalization?

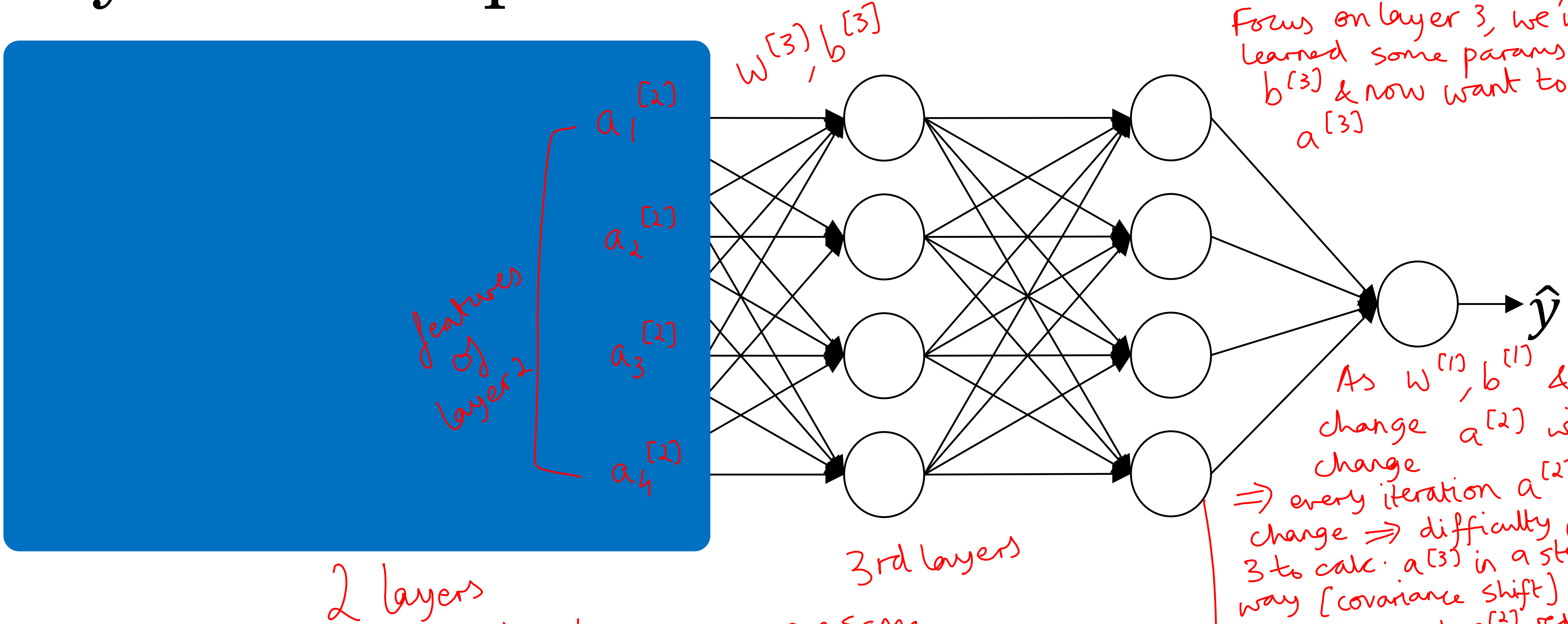
$y = 1$



$y = 0$

Andrew Ng

Why this is a problem with neural networks?



Focus on layer 3, we've learned some params $w^{[3]}$, $b^{[3]}$ & now want to find $a^{[3]}$

As $w^{[1]}, b^{[1]}$ & $w^{[2]}, b^{[2]}$ change $a^{[2]}$ will also change
⇒ every iteration $a^{[2]}$ will change ⇒ difficulty for layer 3 to calc. $a^{[3]}$ in a steady way [covariance shift]
⇒ Batch norm at $a^{[2]}$ reduces the amount $a^{[2]}$ shifts around
⇒ Batch norm at $z^{[2]}$ ensures mean, var at $z^{[2]}$ remains the same at every iteration of training
⇒ More stable results for $a^{[3]}$ based on some γ and β

⇒ Z changes every iteration, but mean, var are same
⇒ limits the amt that updating params in layer 1 & 2 can affect results of layer 3 ⇒ ↓ change in i/p values $a^{[2]}$, makes the dist. stable ⇒ weakens the coupling b/w what the earlier layers have to do v/s what the later layers have to do to accommodate changes from earlier layers ⇒ each layer learns by itself independently ⇒ speeded up learning in the DNN

Andrew Ng

Batch Norm as regularization

(Generally, not used as a Regularizer, use dropout, if need to Regularize)

$X^{\{t\}}$, Say Size = 64, 128, 256

to compute $Z^{[l]\{t\}}$

- Each mini-batch is scaled by the mean/variance computed on just that mini-batch.

  Because we use 64, 128 samples to find $Z^{[l]}$, the estimate is noisy ⇒ $Z^{[l]}$ is noisy ⇒ $\tilde{Z}^{[l]}$ is noisy

- This adds some noise to the values $z^{[l]}$ within that minibatch. So similar to dropout, it adds some noise to each hidden layer's activations.

  dropout picks some nodes & drops others ⇒ minibatch is doing the same, picking all nodes for a few Samples (64/128) at a time.

- This has a slight regularization effect.

  ↳ similar to dropout
  - Dropout's job = Add Regularization
  - This is doing Something similar
  - Tells later layers, not to rely on earlier layers as it may be noisy signals

↑ size of mini Batch, ↓ Regularization effect caused by Batch norm (Read Red part 1st to see why)

- Dropout Adds multiplicative noise. You multiply node w/ 0 or 1 w/ prob "p"
- Batch normalization Adds "Additive" & multiplicative noise

$$\tilde{Z}^{[l]} = \gamma^{[l]} Z_{norm}^{[l]} + \beta^{[l]}$$

$$Z_{norm} = \frac{Z^{[l]} - \mu}{\sigma^2}$$

Additive because you Add $-\mu$

Multiplicative → multiply by $\frac{1}{\sigma^2}$

Andrew Ng