Contents

contents

1 1. deep neural network

1.1 1.1. deep L-layer neural network

1.2 1.2. forward propagation in a deep network

$z^{[l](i)}$ 表示第 l 层的第 i 个训练样本 (列向量)，$Z^{[l]}$ 表示将第 l 层的这 m 个训练样本全部水平地放在一起形成的矩阵。以此 vectorization 的方法，可以避免从 1->m 的这个 for-loop。

另外，从第 1 层到第 4 层这个 for-loop 是无法避免的。

1.3 1.3. getting your matrix dimensions right

vectorized 之后，$b^{[l]}$ 仍然是 $(n^{[l]}, 1)$ 维，只是因为 broadcasting，才复制 m 遍，变成了 $(n^{[l]}, m)$ 维。

1.4 1.4. why deep representations?

对于 $(x_1, x_2, ...x_n)$ 的异或（XOR）操作，如果用树型结构，n 个叶子节点，则树深度是 $log_2 n + 1$(深度为 k 的满二叉树的第 i 层上有 2^{i-1} 个结点, 总共至多有 $2^k - 1$ 个结点)，即只需要 $O(log_2 n)$ 层的树就能完成。

而如果采用单隐层，则需要 $2^{(n-1)}$ 个节点

1.5 1.5. building blocks of deep neural networks

forward 时，需要 cache $Z^{[l]}$ 以供 backward 使用。

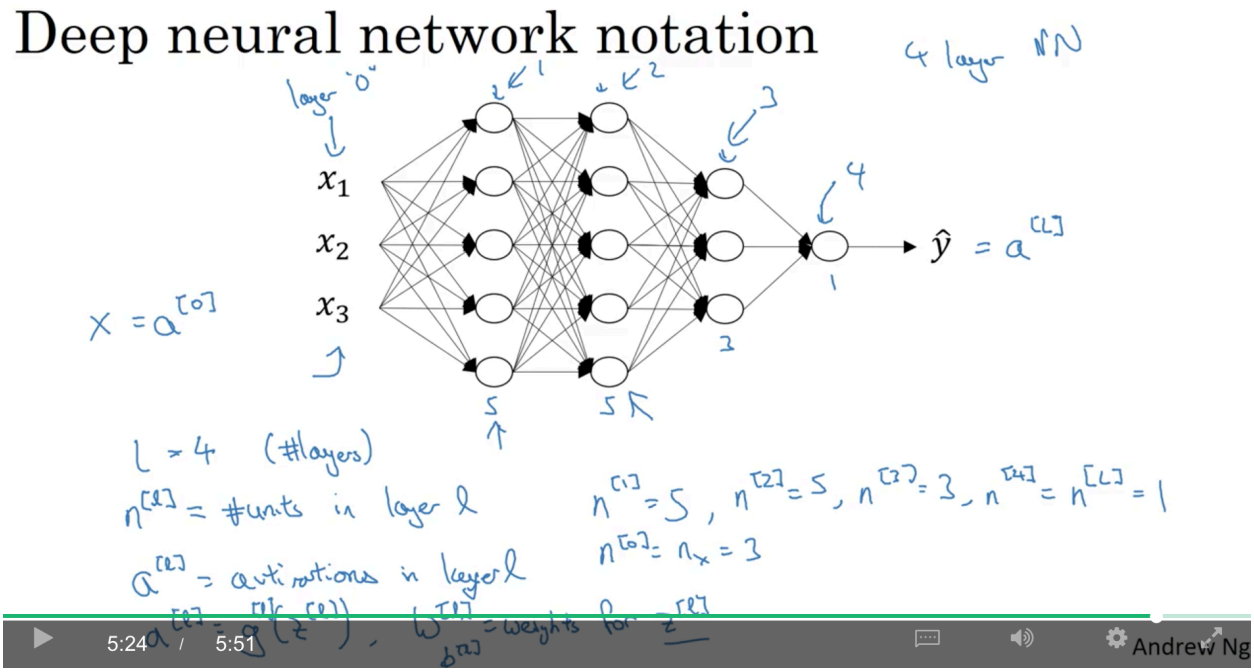

Figure 1: deep-neural-network-notation.png

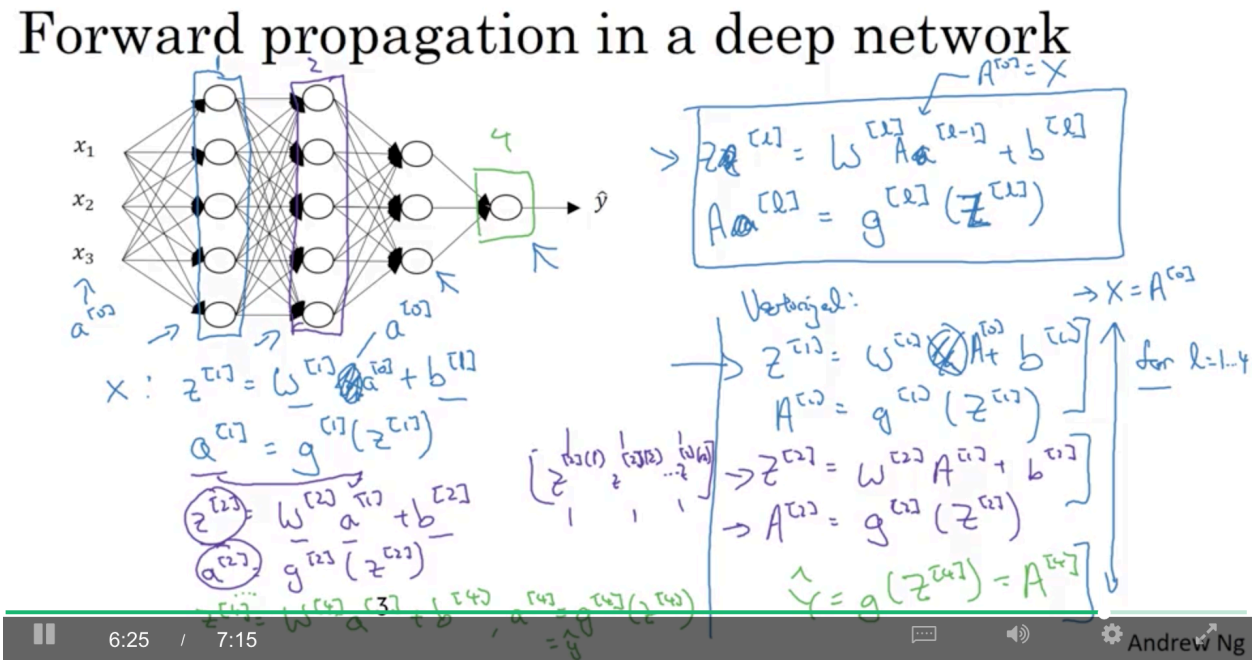

Figure 2: forward-propagation-in-a-deep-network.png

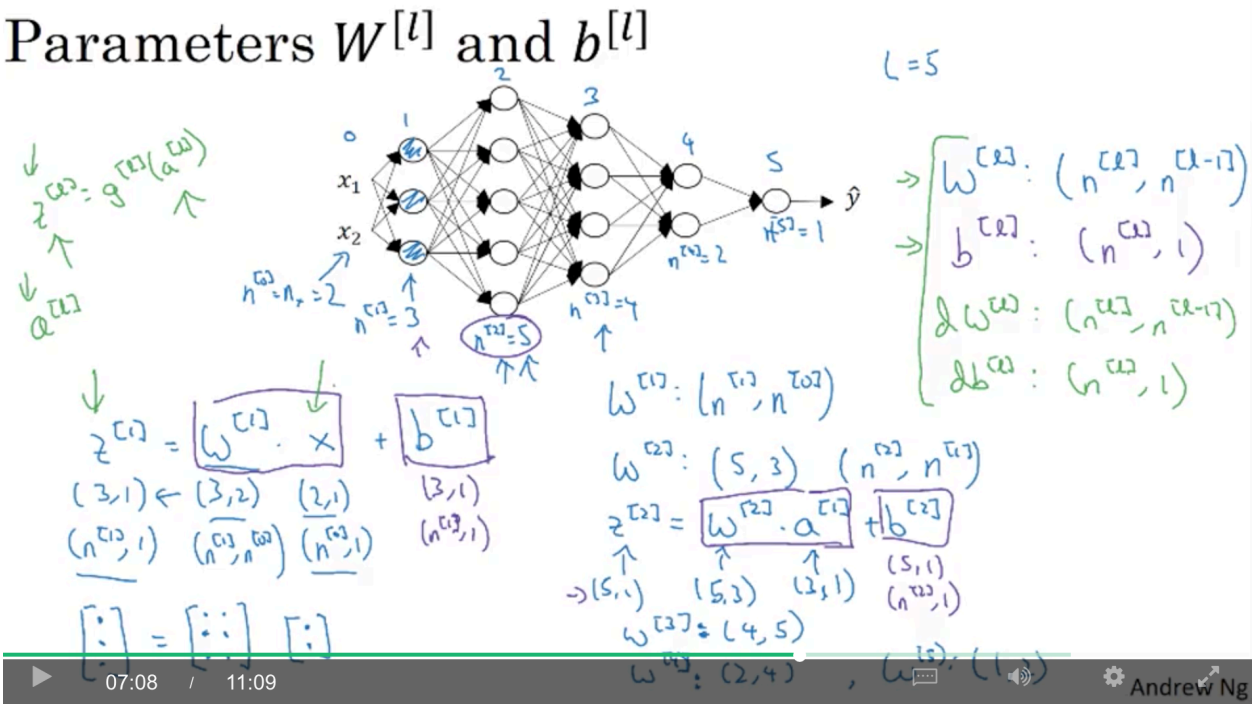

Figure 3: w-and-b-dimension.png

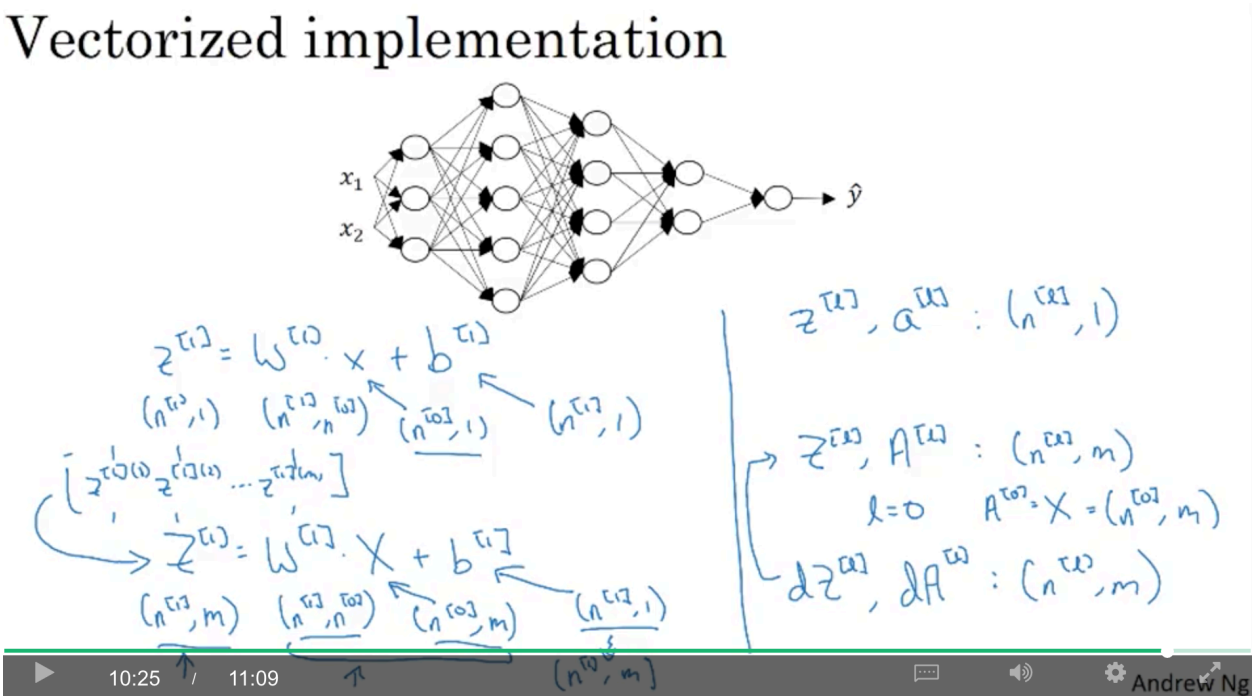

Figure 4: vectorized-implementation.png

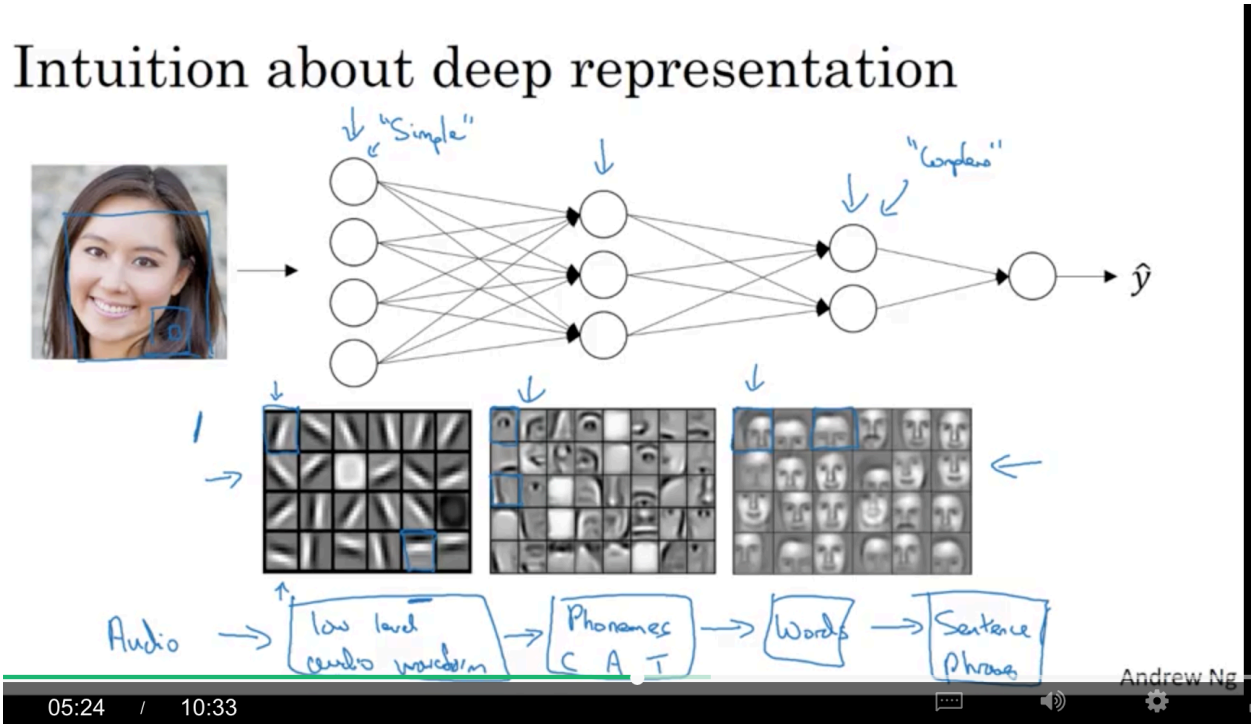

Figure 5: intuition-of-deep-representation.png

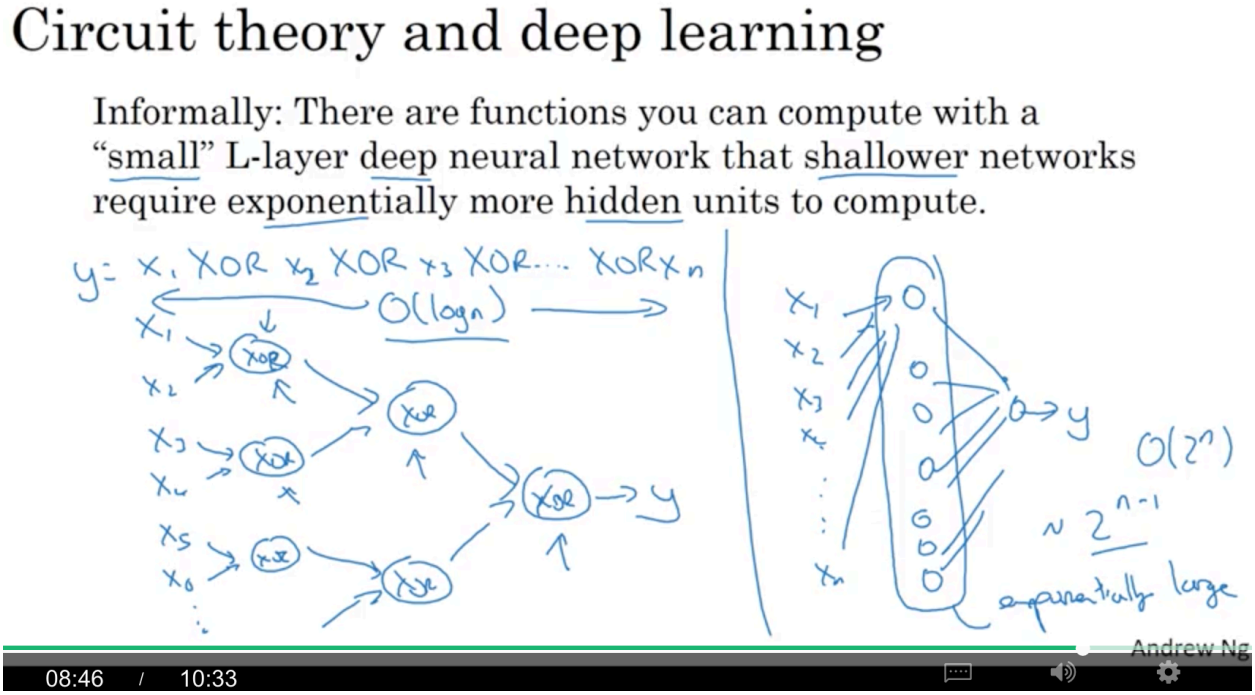

Figure 6: circuit-theory-and-deep-learning.png

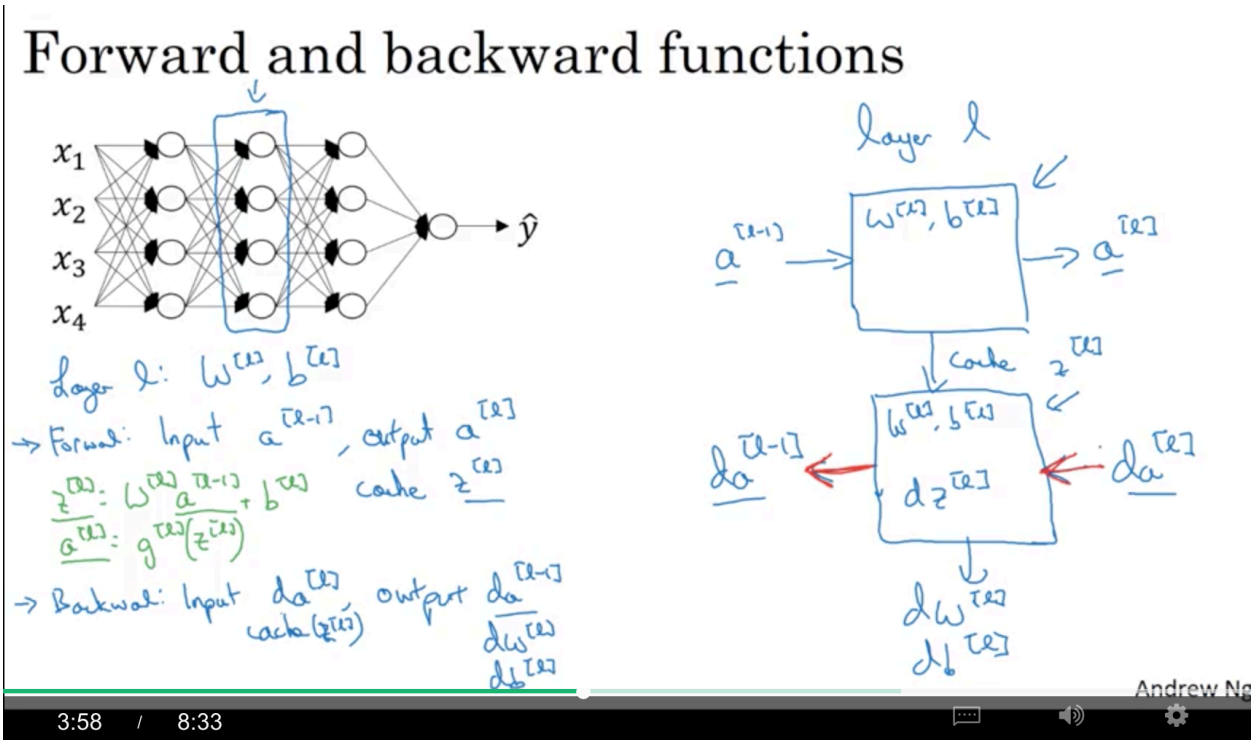

Figure 7: forward-and-backward-functions.png

为了计算 backward，其实需要 cache 的有 $Z^{[l]}$、$W^{[l]}$ 以及 $b^{[l]}$：

1.6 1.6. forward and backward propagation

forward propagation for layer l:

backward propagation for layer l:

参考C1W2 的 backward propagation intuition 部分：注意：$da^{[l]} * g^{[l]'}(z^{[l]})$ 是 element-wise product。

对于最后一层 L，如果是 sigmoid 并采用 logloss，那么：

1.7 1.7. parameters vs hyperparameters

图中下方是：momentum，mini-batch size，regularization

1.8 1.8. what does this have to do with the brain?

1.9 1.9. others

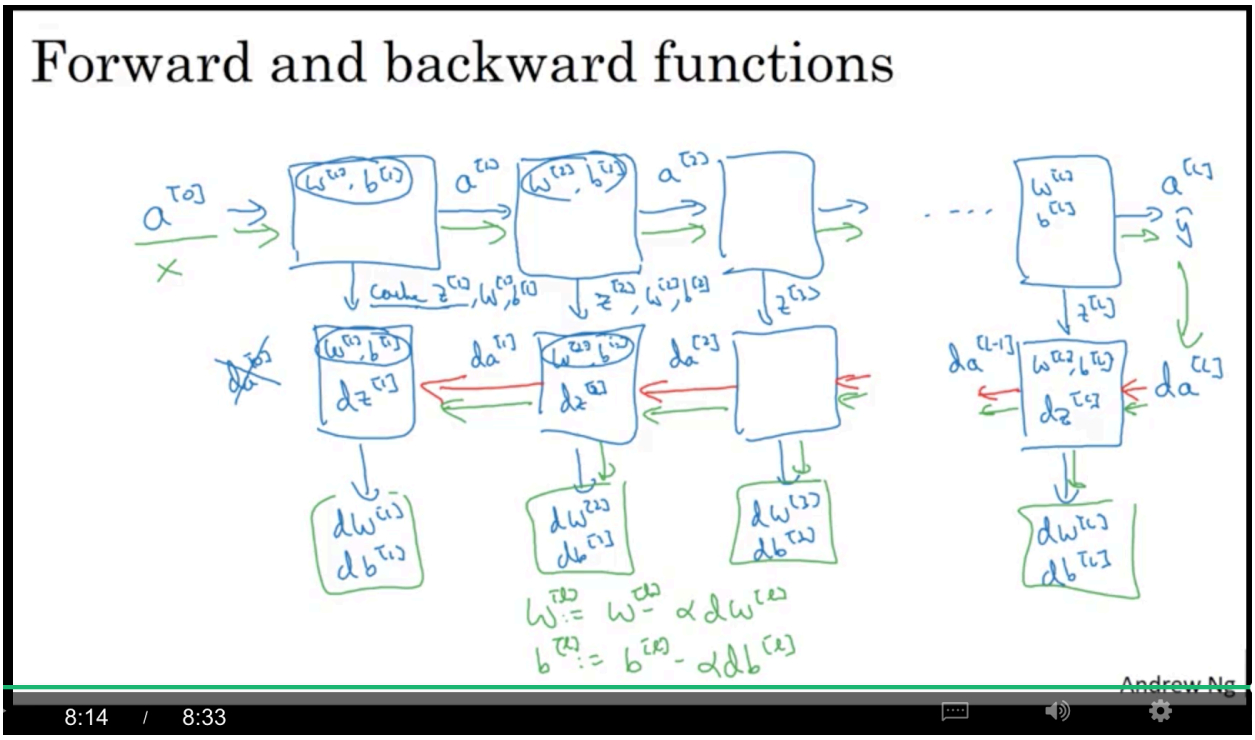

Figure 8: forward-backward-functions.png

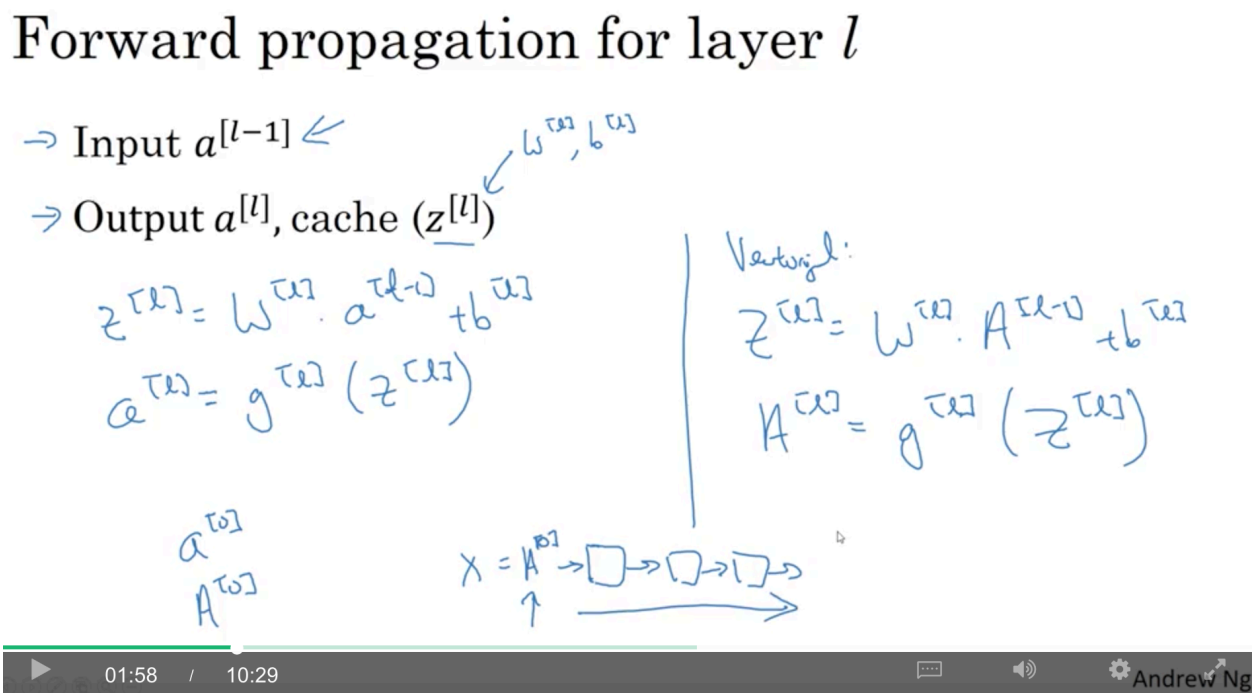

Figure 9: forward-propagation-for-layer-l.png

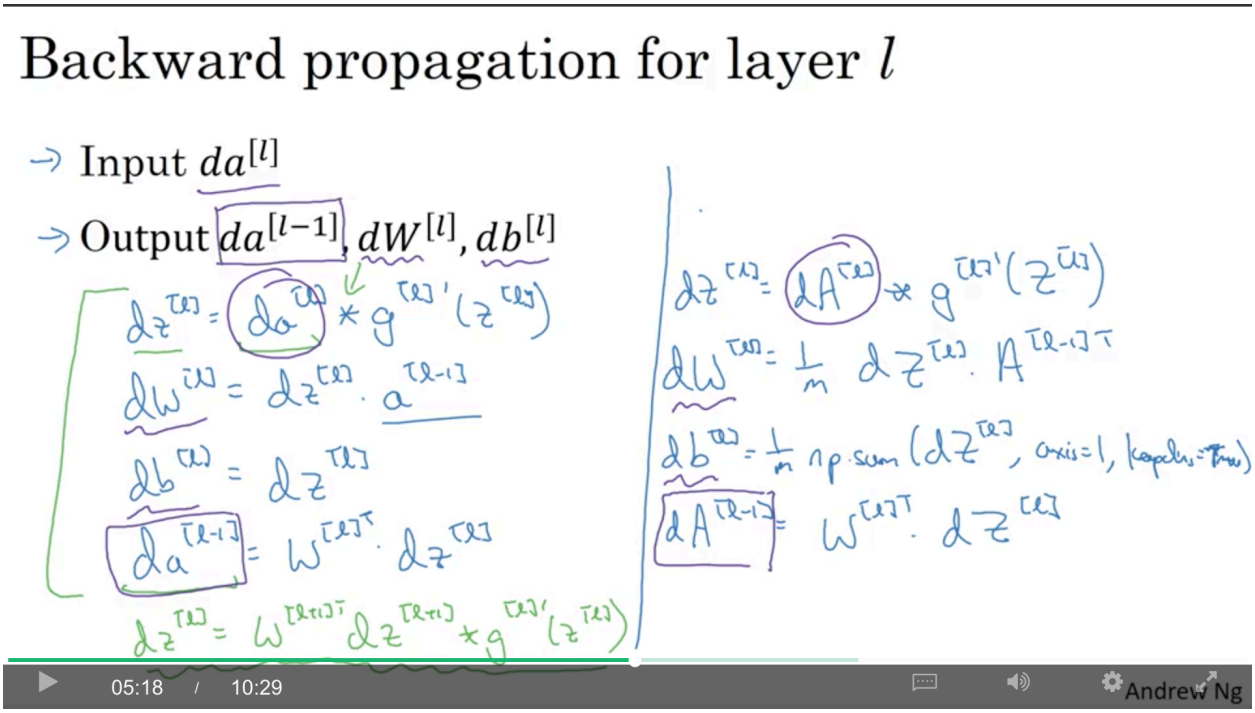

Figure 10: backward-propagation-for-layer-l.png

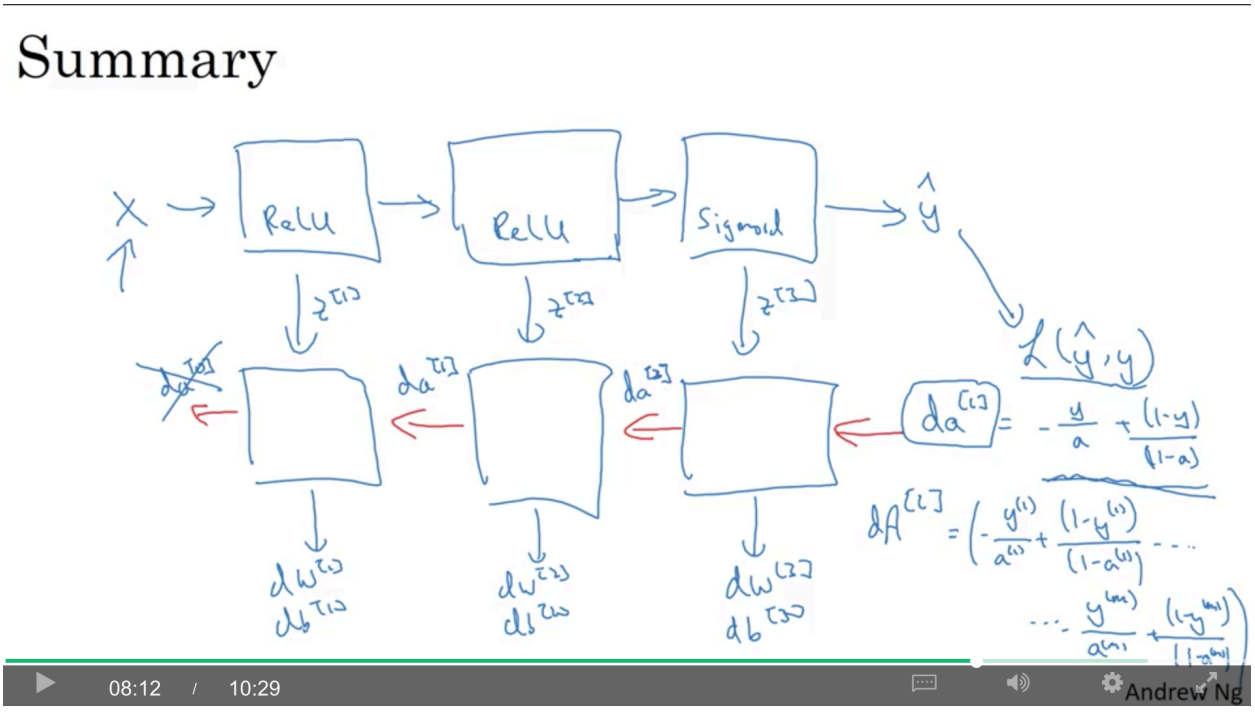

Figure 11: summary-forward-backward-propagation.png

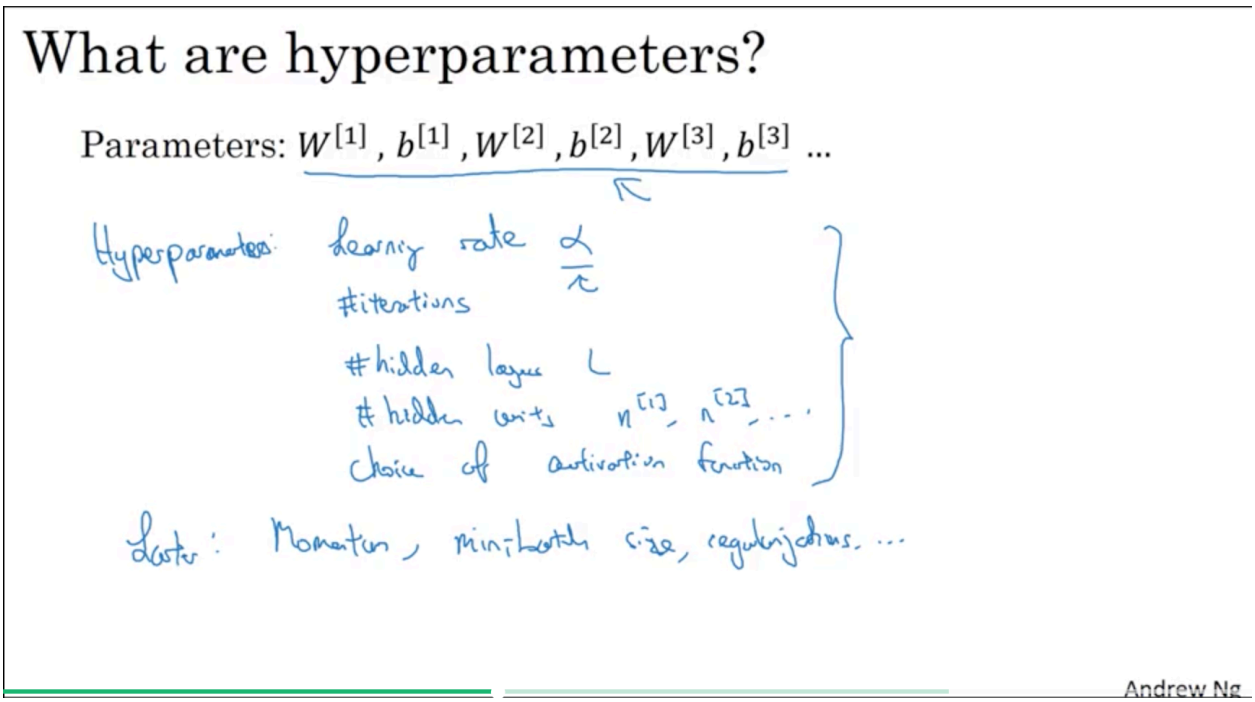

Figure 12: summary-forward-backward-propagation.png

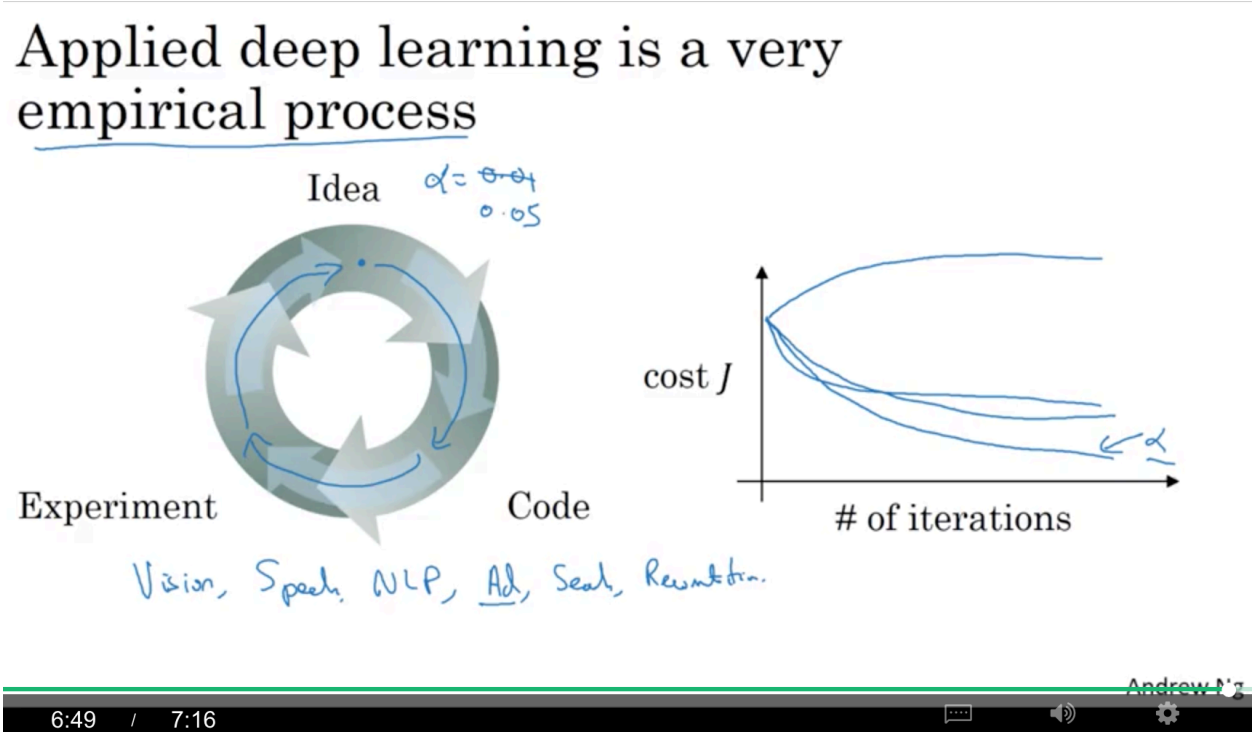

Figure 13: apply-deep-learning-is-a-very-empirical-process.png

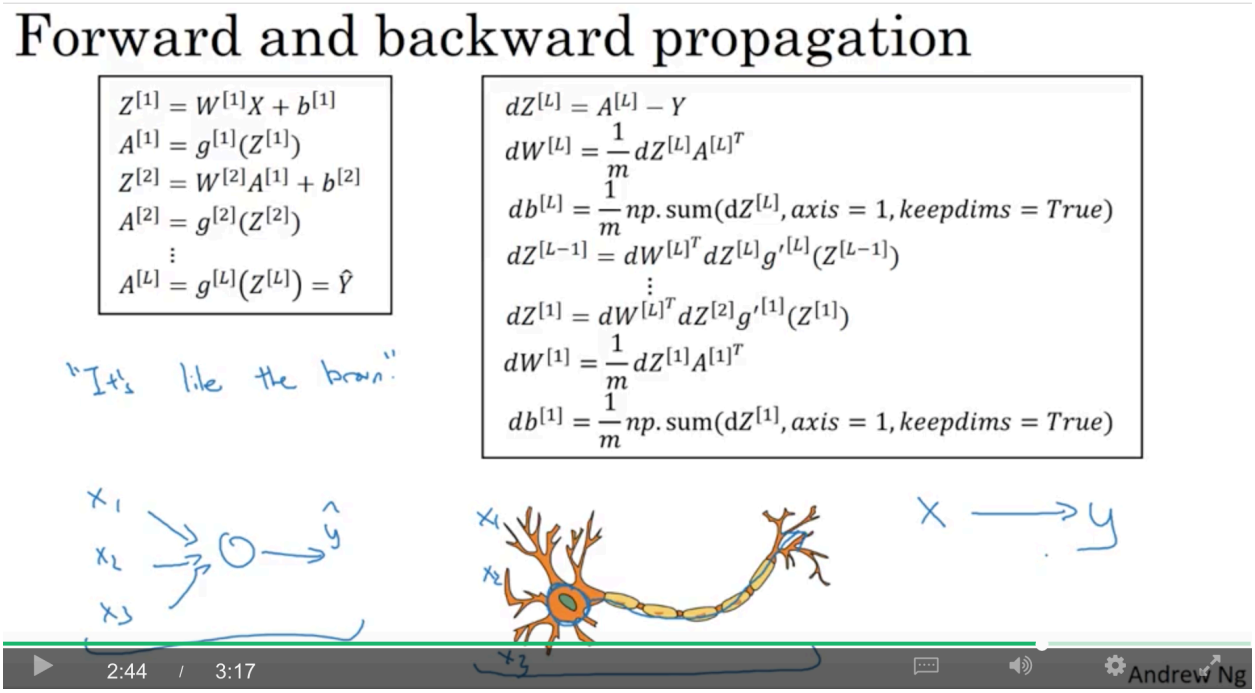

Figure 14: summary-and-brain.png

Thus for example if the size of our input X is $(12288, 209)$ (with $m = 209$ examples) then:

	Shape of W	Shape of b	Activation	Shape of Activation
Layer 1	$(n^{[1]}, 12288)$	$(n^{[1]}, 1)$	$Z^{[1]} = W^{[1]}X + b^{[1]}$	$(n^{[1]}, 209)$
Layer 2	$(n^{[2]}, n^{[1]})$	$(n^{[2]}, 1)$	$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$	$(n^{[2]}, 209)$
⋮	⋮	⋮	⋮	⋮
Layer L-1	$(n^{[L-1]}, n^{[L-2]})$	$(n^{[L-1]}, 1)$	$Z^{[L-1]} = W^{[L-1]}A^{[L-2]} + b^{[L-1]}$	$(n^{[L-1]}, 209)$
Layer L	$(n^{[L]}, n^{[L-1]})$	$(n^{[L]}, 1)$	$Z^{[L]} = W^{[L]}A^{[L-1]} + b^{[L]}$	$(n^{[L]}, 209)$

Figure 15: shape-of-L-layer-nn.png