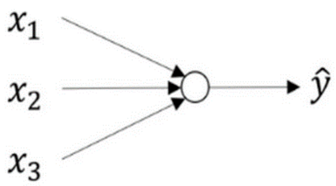
## 3.1 神经网络概述（Neural Network Overview）

现在我们开始快速浏览一下如何实现神经网络。上周我们讨论了逻辑回归，我们了解了这个模型(见图 3.1.1)如何与下面公式 3.1 建立联系。

图 3.1.1 :




公式 3.1：

$$\left.\begin{matrix} x \\ w \\ b \end{matrix}\right\} \Longrightarrow z = w^T x + b$$

如上所示，首先你需要输入特征$x$，参数$w$和$b$，通过这些你就可以计算出$z$，公式 3.2：

$$\left.\begin{matrix} x \\ w \\ b \end{matrix}\right\} \Longrightarrow z = w^T x + b \Longrightarrow \alpha = \sigma(z)$$
$$\Longrightarrow L(a, y)$$

接下来使用$z$就可以计算出$a$。我们将的符号换为表示输出$\hat{y} \Longrightarrow a = \sigma(z)$,然后可以计算出 **loss function** $L(a, y)$

神经网络看起来是如下这个样子（图 3.1.2）。正如我之前已经提到过，你可以把许多 **sigmoid** 单元堆叠起来形成一个神经网络。对于图 3.1.1 中的节点，它包含了之前讲的计算的两个步骤：首先通过公式 3.1 计算出值$z$，然后通过$\sigma(z)$计算值$a$。

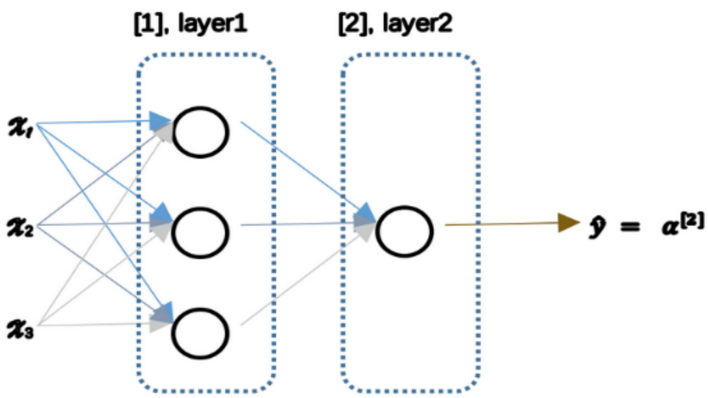



图 3.1.2

在这个神经网络（图 3.1.2）对应的 3 个节点，首先计算第一层网络中的各个节点相关

的数$z^{[1]}$，接着计算$\alpha^{[1]}$，在计算下一层网络同理； 我们会使用符号 $^{[m]}$表示第$m$层网络中节点相关的数，这些节点的集合被称为第$m$层网络。这样可以保证 $^{[m]}$不会和我们之前用来表示单个的训练样本的 $^{(i)}$(即我们使用表示第 i 个训练样本)混淆； 整个计算过程，公式如下：公式 3.3：

$$\left.\begin{array}{r} x \\ W^{[1]} \\ b^{[1]} \end{array}\right\} \implies z^{[1]} = W^{[1]}x + b^{[1]} \implies a^{[1]} = \sigma(z^{[1]})$$

公式 3.4：

$$\left.\begin{array}{r} x \\ dW^{[1]} \\ db^{[1]} \end{array}\right\} \Longleftarrow dz^{[1]} = d(W^{[1]}x + b^{[1]}) \Longleftarrow d\alpha^{[1]} = d\sigma(z^{[1]})$$

类似逻辑回归，在计算后需要使用计算，接下来你需要使用另外一个线性方程对应的参数计算$z^{[2]}$， 计算$a^{[2]}$，此时$a^{[2]}$就是整个神经网络最终的输出，用 $\hat{y}$表示网络的输出。

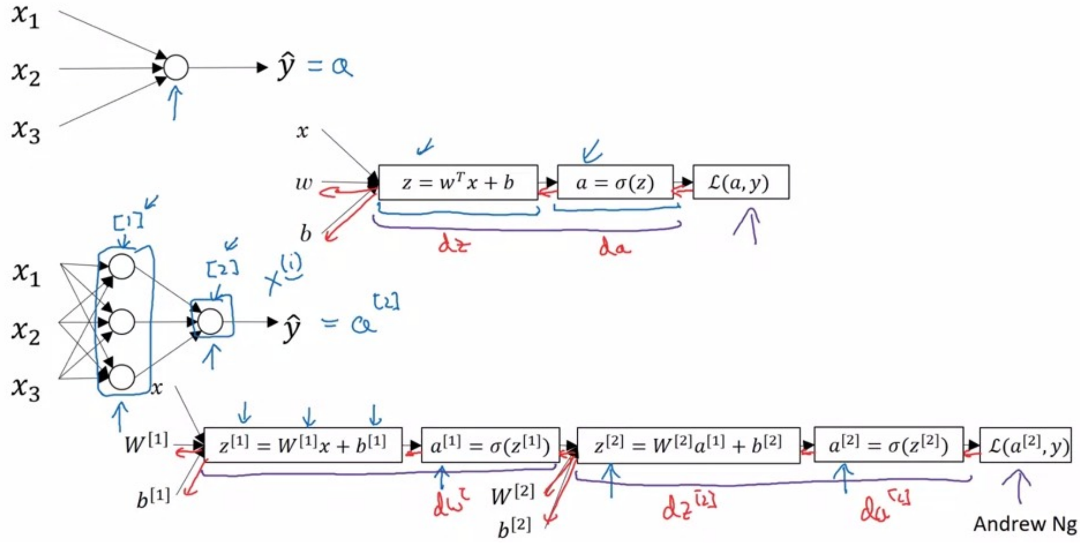
公式 3.5：

$$\left.\begin{array}{r} da^{[1]} = d\sigma(z^{[1]}) \\ dW^{[2]} \\ db^{[2]} \end{array}\right\} \Longleftarrow dz^{[2]} = d(W^{[2]}\alpha^{[1]} + b^{[2]}) \Longleftarrow da^{[2]} = d\sigma(z^{[2]}) \\ \Longleftarrow dL\left(a^{[2]}, y\right)$$

我知道这其中有很多细节，其中有一点非常难以理解，即在逻辑回归中，通过直接计算$z$得到结果$a$。而这个神经网络中，我们反复的计算$z$和$a$，计算$a$和$z$，最后得到了最终的输出 **loss function**。

你应该记得逻辑回归中，有一些从后向前的计算用来计算导数$da$、$dz$。同样，在神经网络中我们也有从后向前的计算，看起来就像这样，最后会计算$da^{[2]}$ 、$dz^{[2]}$，计算出来之后，然后计算计算$dW^{[2]}$、$db^{[2]}$ 等，按公式 3.4、3.5 箭头表示的那样，从右到左反向计算。

# What is a Neural Network?




Andrew Ng