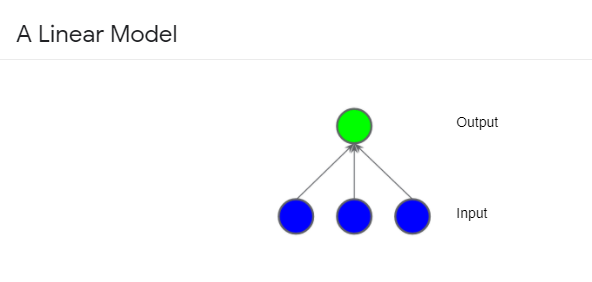
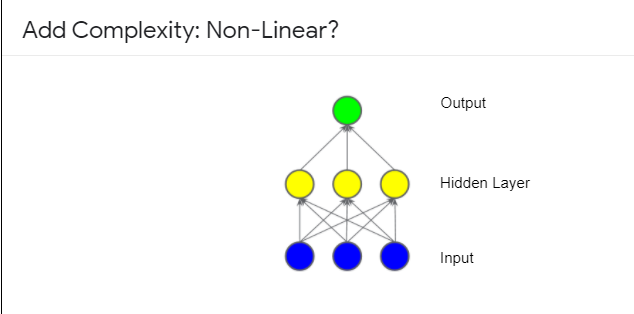
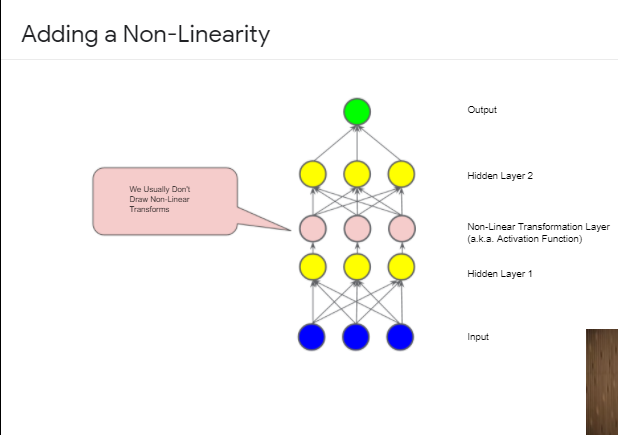
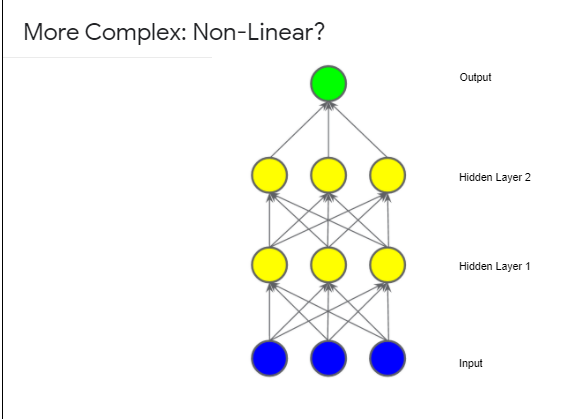
**Neural Networks**

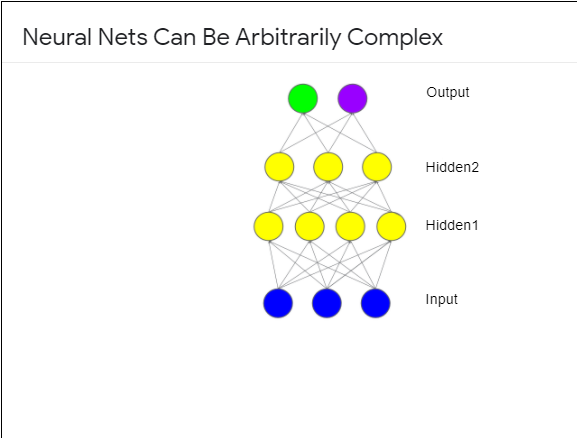
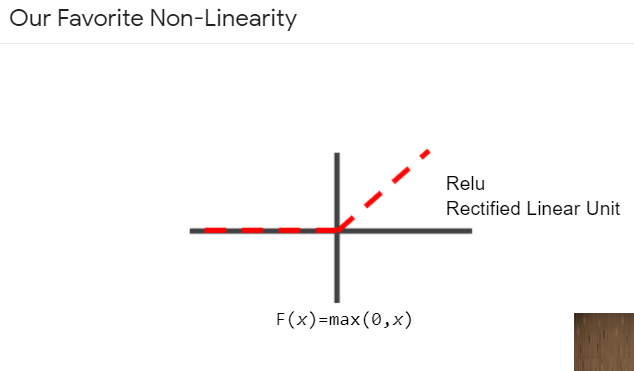


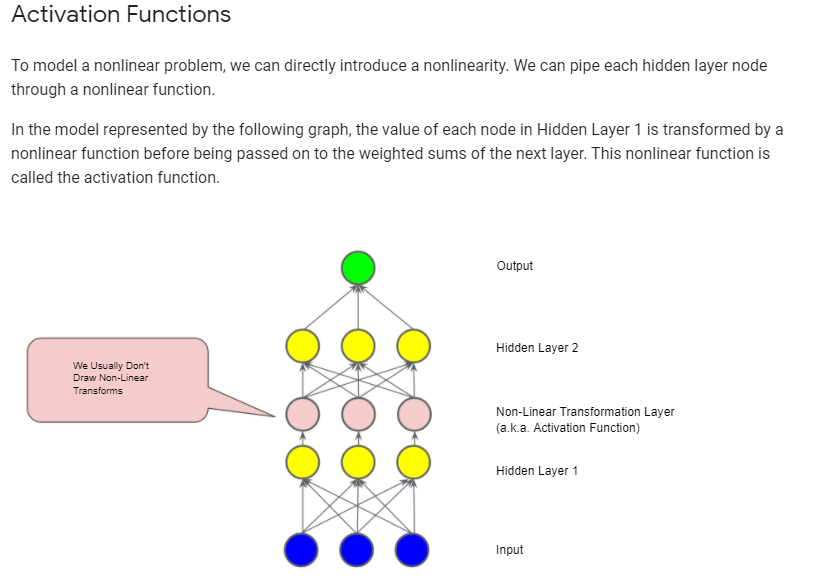
Neural networks are a more sophisticated version of feature crosses. In essence, neural networks learn the appropriate feature crosses for you.











### Common Activation Functions

The following **sigmoid** activation function converts the weighted sum to a value between 0 and 1.

F(x)=11+e−x

Here's a plot:

**Figure 7. Sigmoid activation function.**

The following **rectified linear unit** activation function (or **ReLU**, for short) often works a little better than a smooth function like the sigmoid, while also being significantly easier to compute.

F(x)=max(0,x)

The superiority of ReLU is based on empirical findings, probably driven by ReLU having a more useful range of responsiveness. A sigmoid's responsiveness falls off relatively quickly on both sides.

**Figure 8. ReLU activation function.**

In fact, any mathematical function can serve as an activation function. Suppose that σ represents our activation function (Relu, Sigmoid, or whatever). Consequently, the value of a node in the network is given by the following formula:

σ(w⋅x+b)

TensorFlow provides out-of-the-box support for many activation functions. You can find these activation functions within TensorFlow's [list of wrappers for primitive neural network operations](https://www.tensorflow.org/api_docs/python/tf/nn). That said, we still recommend starting with ReLU.