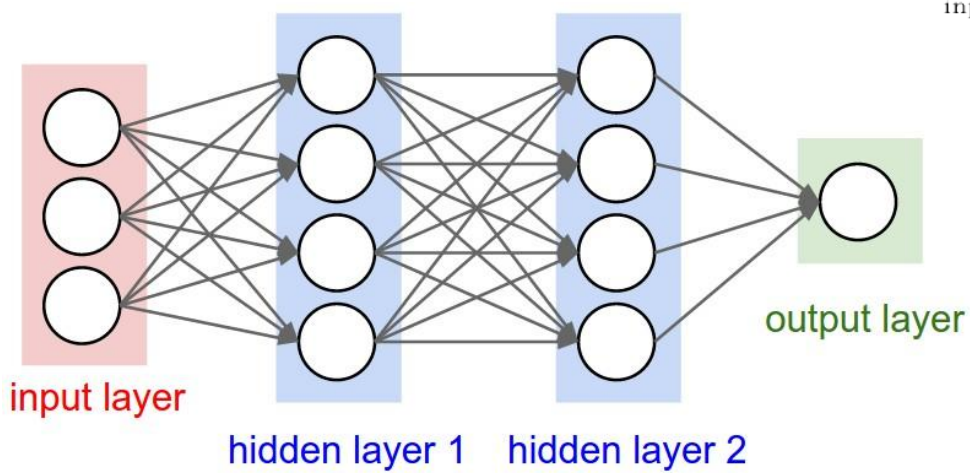
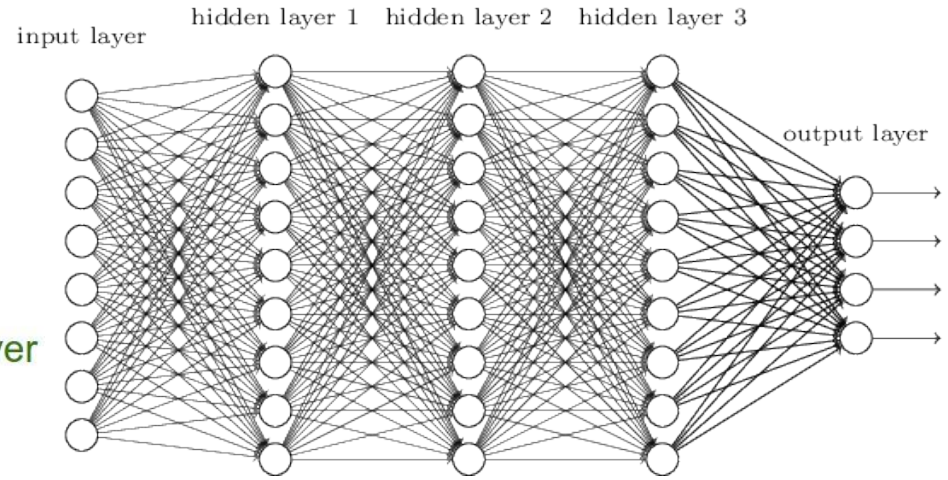
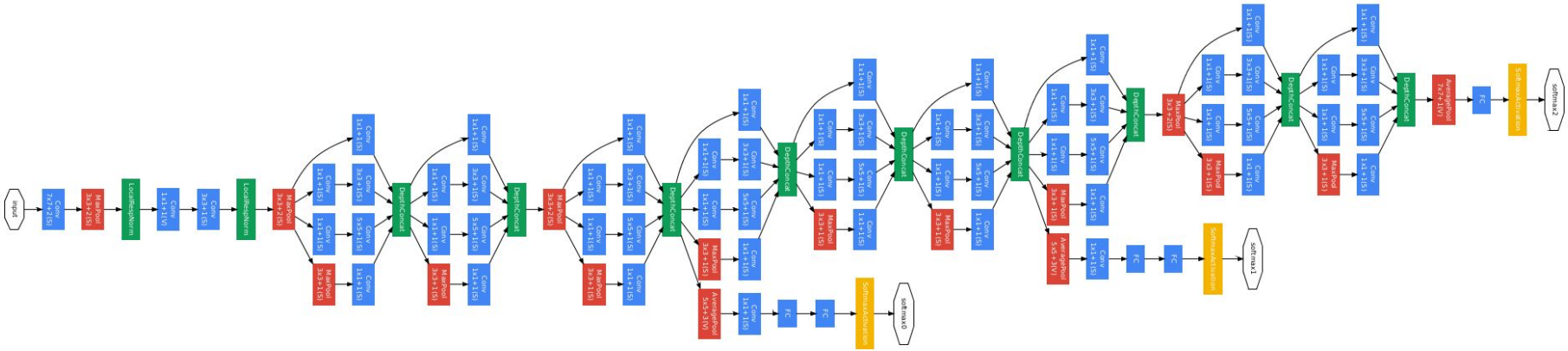
TensorFlow

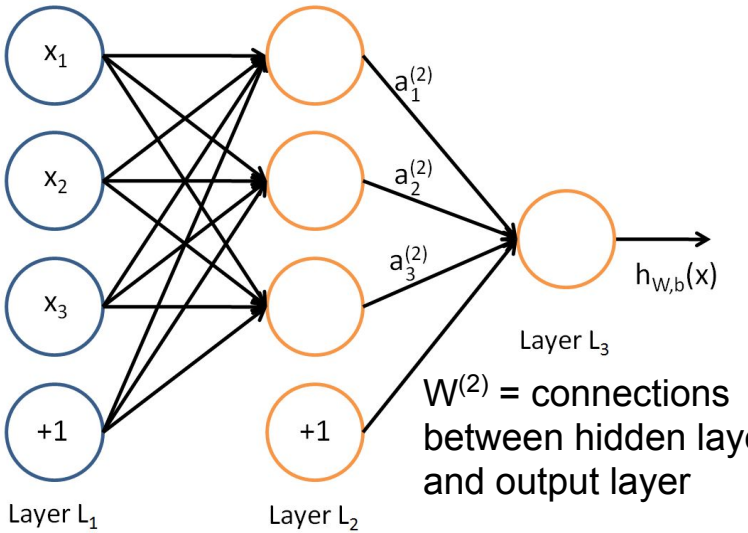

input layer
hidden layer 1
hidden layer 2
output layer


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer



What's a Neural Network?

$W^{(1)}$ = connections between input and hidden layer



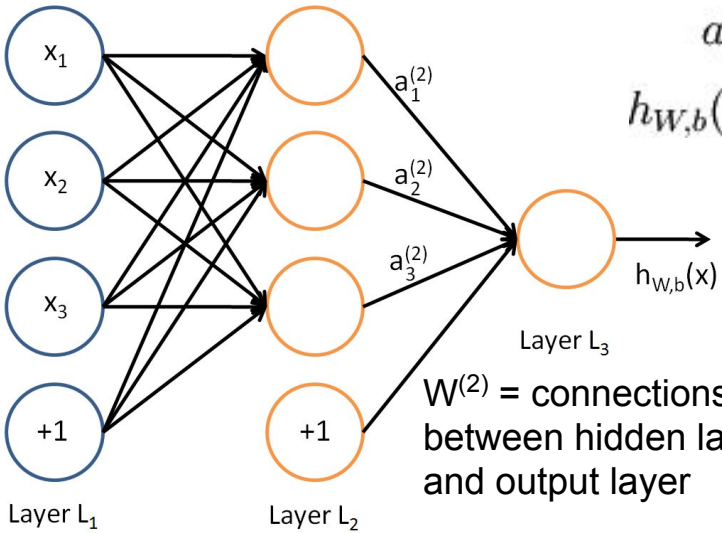
$W^{(2)}$ = connections between hidden layer and output layer

Images stolen from: http://ufldl.stanford.edu/wiki/index.php/Neural_Networks

What's a Neural Network?

$W^{(1)}$ = connections between input and hidden layer

$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$
$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$
$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})$$

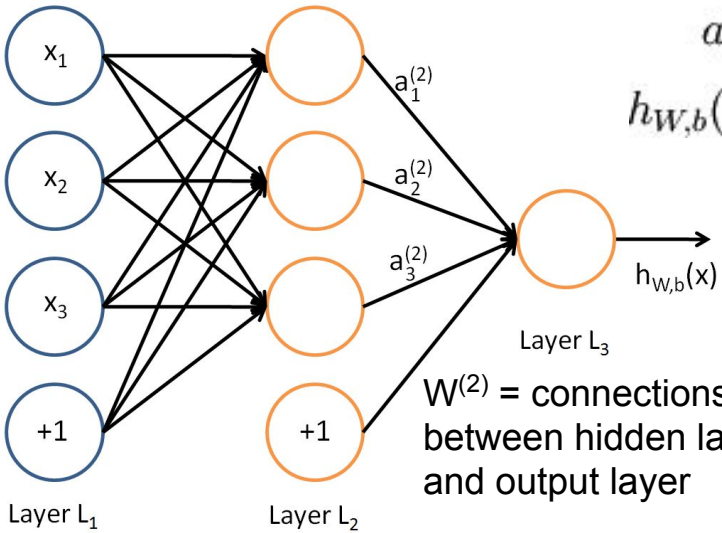


$W^{(2)}$ = connections between hidden layer and output layer

Images stolen from: http://ufldl.stanford.edu/wiki/index.php/Neural_Networks

What's a Neural Network?

$W^{(1)}$ = connections between input and hidden layer



$W^{(2)}$ = connections between hidden layer and output layer

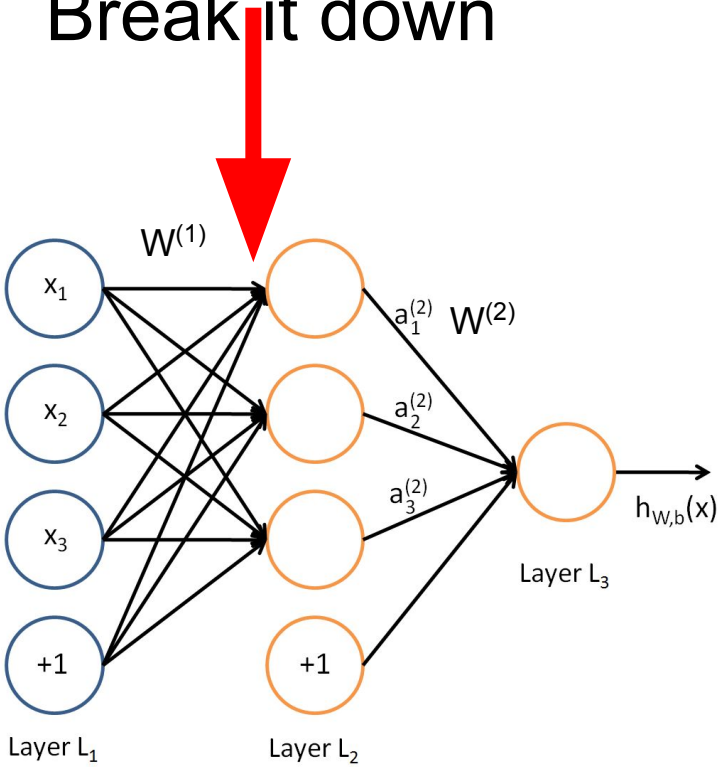
$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$
$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$
$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})$$

$$\boldsymbol{a}^{(2)} = f(W^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$$

$$\boldsymbol{a}^{(3)} = \boldsymbol{h}(\boldsymbol{x}) = f(W^{(2)}\boldsymbol{a}^{(2)} + \boldsymbol{b}^{(2)})$$

Images stolen from: http://ufldl.stanford.edu/wiki/index.php/Neural_Networks

Break it down



N = number of layers
There are N - 1 weight matrices
There are N - 1 **sets** of bias terms

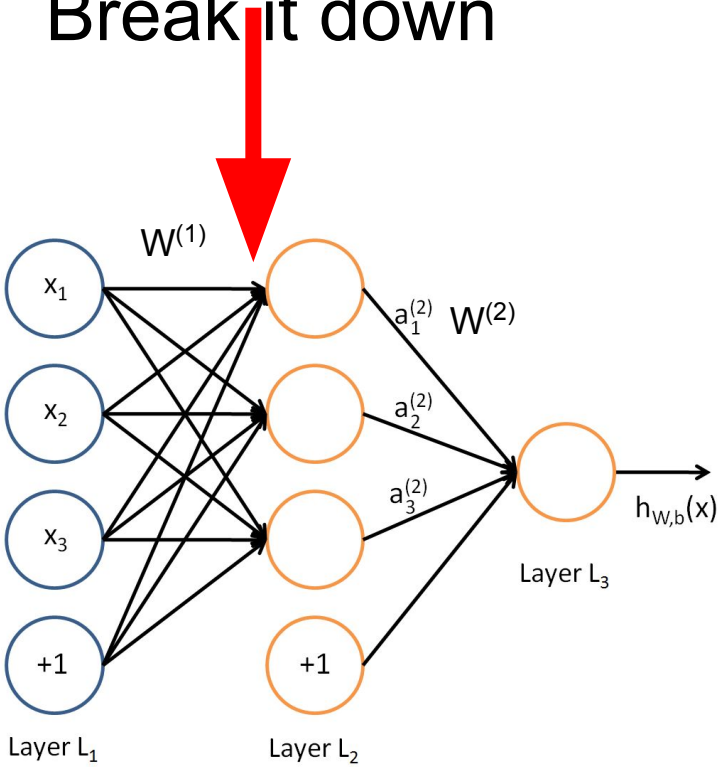
First weight matrix ($W^{(1)}$):

- Between **input layer** and **hidden layer**
- Num_rows = number of **hidden** units
- Num_columns = number of **input** units

First set of bias terms ($\mathbf{b}^{(1)}$):

- Vector, number of elements = number of nodes in next layer

Break it down



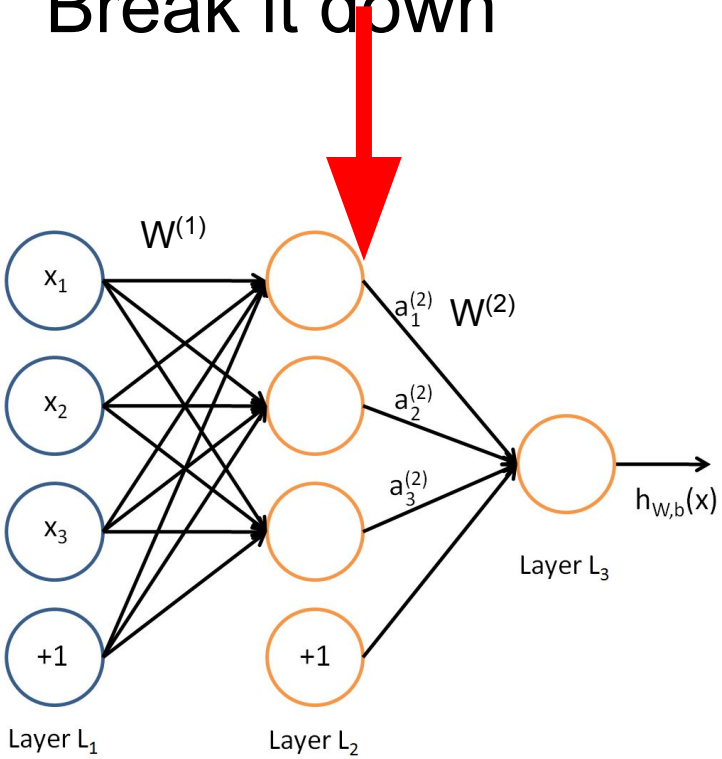
At input of first hidden layer:

- Treat input nodes as elements of a matrix, **x**
- Perform linear transformation of input data using $W^{(1)}$ and $\mathbf{b}^{(1)}$

$$\boldsymbol{a}^{(2)} = f(W^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$$

$$\boldsymbol{a}^{(3)} = \boldsymbol{h}(\boldsymbol{x}) = f(W^{(2)}\boldsymbol{a}^{(2)} + \boldsymbol{b}^{(2)})$$

Break it down



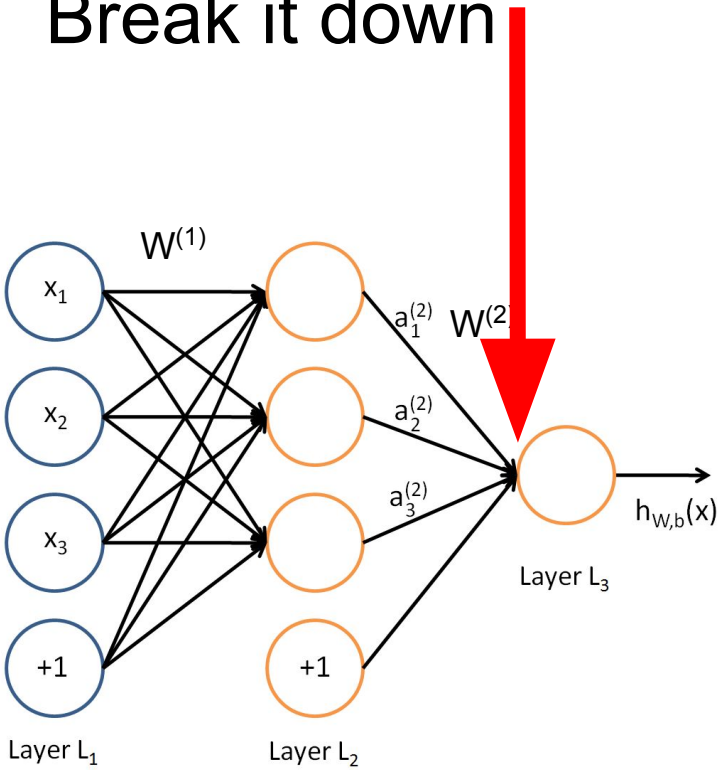
At output of first hidden layer:

- Apply non-linear function (tanh, sigmoid, ReLU) to linear transformation ($W^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$)
- Get values for the vector $\mathbf{a}^{(2)}$ (output of hidden layer)

$$a^{(2)} = f(W^{(1)}x + b^{(1)})$$

$$a^{(3)} = h(x) = f(W^{(2)}a^{(2)} + b^{(2)})$$

Break it down



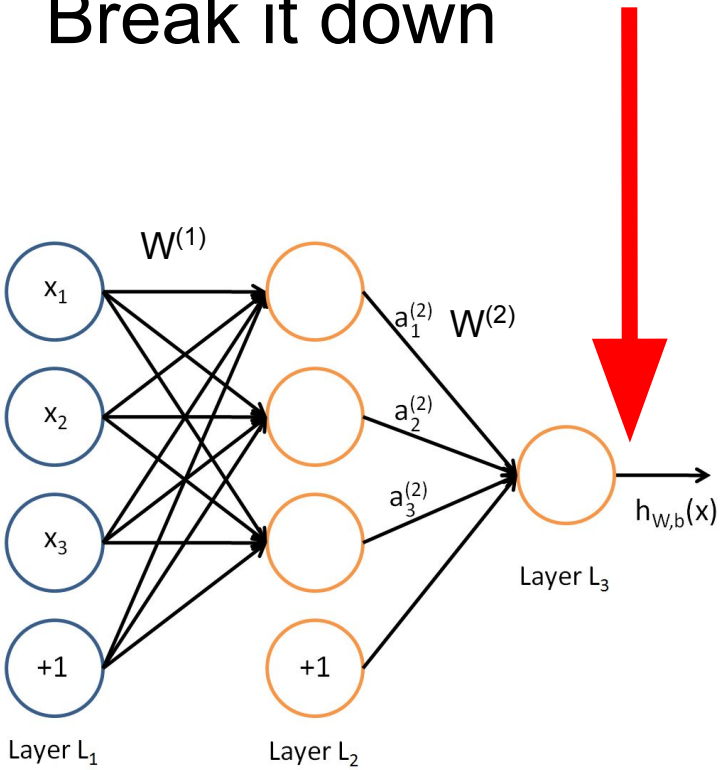
At the input of the last layer:

- Perform linear transformation on outputs of previous layer using $W^{(2)}$ and $\mathbf{b}^{(2)}$

$$\boldsymbol{a}^{(2)} = f(W^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$$

$$\boldsymbol{a}^{(3)} = \boldsymbol{h}(\boldsymbol{x}) = f(W^{(2)}\boldsymbol{a}^{(2)} + \boldsymbol{b}^{(2)})$$

Break it down



At the output of the last layer:

- Perform nonlinear transformation on the inputs to this node to get the output layer vector, $\mathbf{a}^{(3)} = \mathbf{h}(\mathbf{x})$

$$a^{(2)} = f(W^{(1)}x + b^{(1)})$$

$$a^{(3)} = h(x) = f(W^{(2)}a^{(2)} + b^{(2)})$$

How do we pick the W's and b's?

Use random W's and b's initially

Quantify how poorly the network performs (error)

Adjust accordingly (minimize error)

Mean-Square Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Cross-Entropy

$$l(\theta) = \sum_{i=1}^{n} y^{(i)} log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))$$

How do we pick the W's and b's?

Quantify how poorly the network performs (error)

Adjust accordingly (minimize error)

CALC 1 peeps, where you at?!

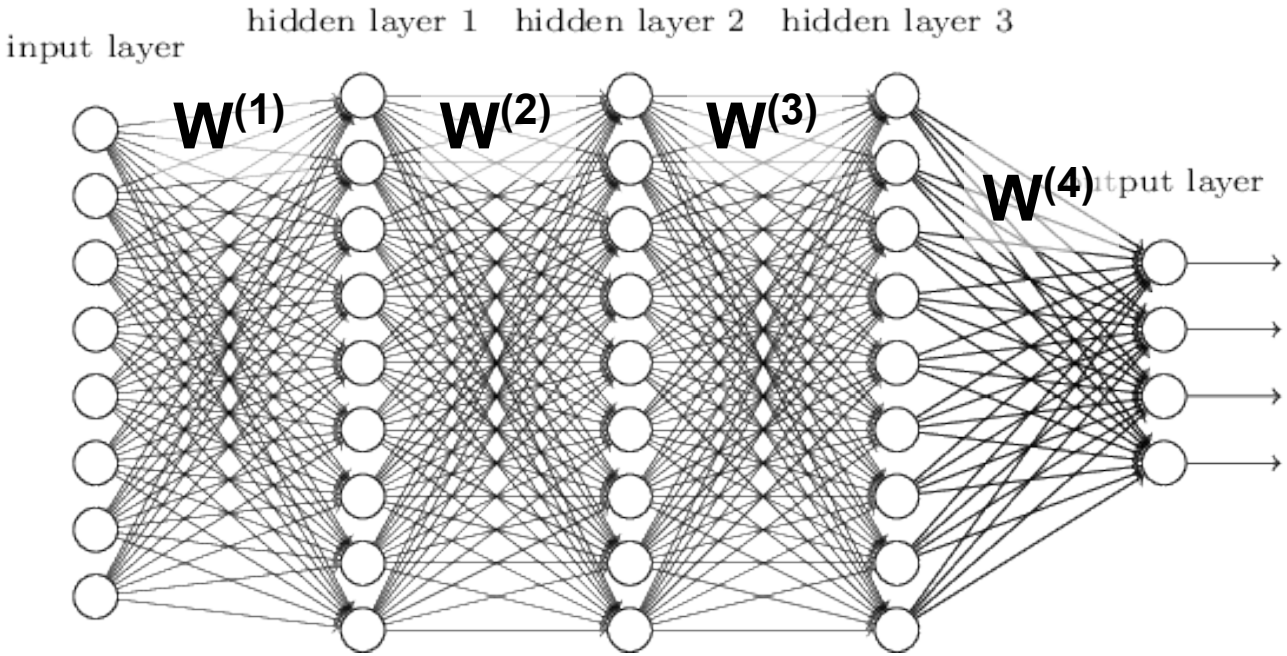
Mean-Square Error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Cross-Entropy

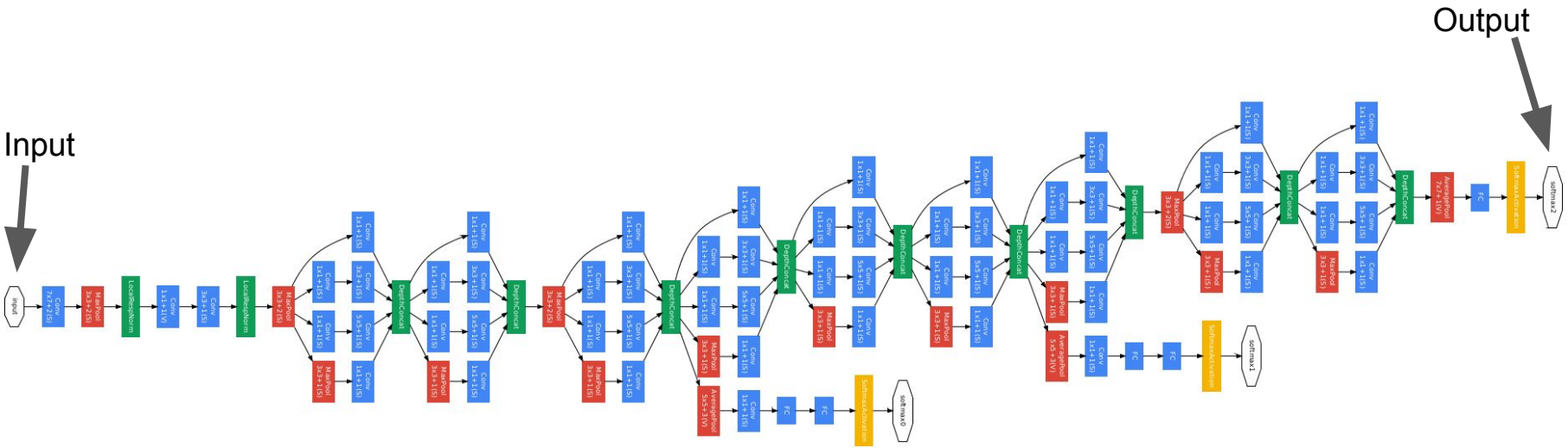
$$l(\theta) = \sum_{i=1}^{n} y^{(i)} log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))$$

What if we had a Deep Neural Network?



Two or more non-linear transformations of the input data (conventionally)

What if we had a Deep Neural Network?



Take lots and LOTS of derivatives...

Outcome depends on:

- Network architecture
- Loss function

Why is TensorFlow so great?

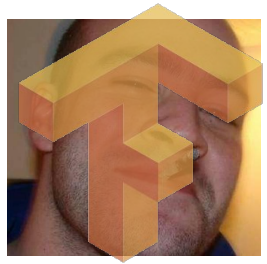
Take lots and LOTS of derivatives...

Outcome depends on:

- Network architecture
- Loss function

Why is TensorFlow so great?

- Pick an architecture (feed-forward, recurrent, ...)
- Pick a loss function (mean square, cross-entropy, …)
- Pick an optimization scheme (SGD, AdaGrad, …)
- TF does the rest*



*Terms and conditions may apply

EXAMPLES PLS