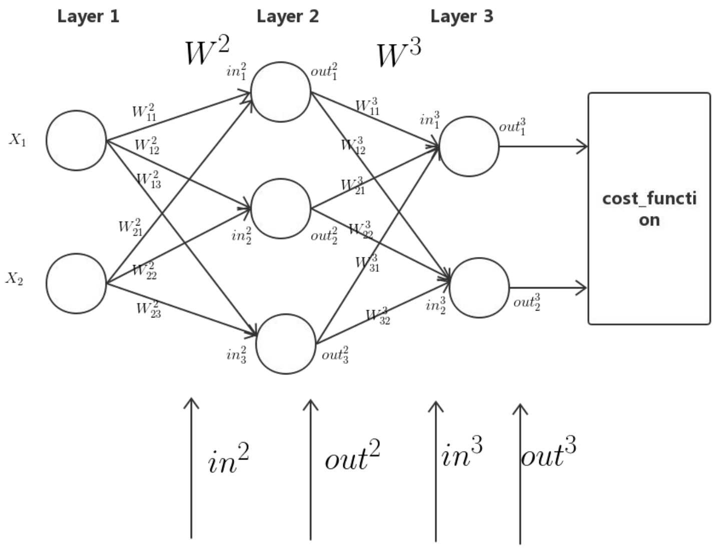
Back propagation

3-layer ANN



$$W^2 = \begin{bmatrix} W_{11}^2 & W_{12}^2 & W_{13}^2 \\ W_{21}^2 & W_{22}^2 & W_{23}^2 \end{bmatrix}$$

$$W^3 = \begin{bmatrix} W_{11}^3 & W_{12}^3 \\ W_{21}^3 & W_{22}^3 \\ W_{31}^3 & W_{32}^3 \end{bmatrix}$$

$$out^2 = \begin{bmatrix} out_1^2 & out_2^2 & out_3^2 \end{bmatrix}$$

$$out^3 = \begin{bmatrix} out_1^3 & out_2^3 \end{bmatrix}$$

$$b^2 = \begin{bmatrix} b_1^2 & b_2^2 & b_3^2 \end{bmatrix}$$

$$b^3 = \begin{bmatrix} b_1^3 & b_2^3 \end{bmatrix}$$

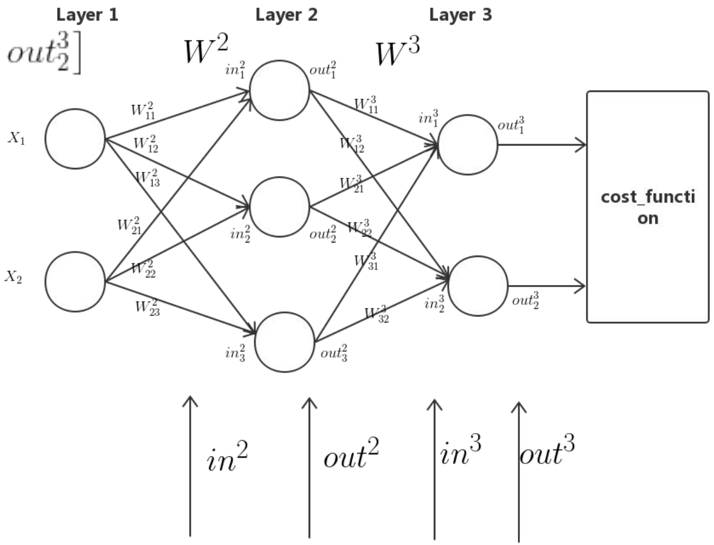
$$in^2 = \begin{bmatrix} in_1^2 & in_2^2 & in_3^2 \end{bmatrix}$$

$$in^3 = \begin{bmatrix} in_1^3 & in_2^3 \end{bmatrix}$$

forward

$$[X_1 \quad X_2] \bullet \begin{bmatrix} W_{11}^2 & W_{12}^2 & W_{13}^2 \\ W_{21}^2 & W_{22}^2 & W_{23}^2 \end{bmatrix} + [b_1^2 \quad b_2^2 \quad b_3^2] \rightarrow [in_1^2 \quad in_2^2 \quad in_3^2] \xrightarrow{sigmoid} [out_1^2 \quad out_2^2 \quad out_3^2]$$

$$[out_1^2 \quad out_2^2 \quad out_3^2] \bullet \begin{bmatrix} W_{11}^3 & W_{12}^3 \\ W_{21}^3 & W_{22}^3 \\ W_{31}^3 & W_{32}^3 \end{bmatrix} + [b_1^3 \quad b_2^3] \rightarrow [in_1^3 \quad in_2^3] \xrightarrow{sigmoid} [out_1^3 \quad out_2^3]$$



$$in_1^2 = W_{11}^2 * X_1 + W_{21}^2 * X_2 + b_1^2$$
$$out_1^2 = sigmoid(in_1^2)$$

$$in_2^2 = W_{12}^2 * X_1 + W_{22}^2 * X_2 + b_2^2$$
$$out_2^2 = sigmoid(in_2^2)$$

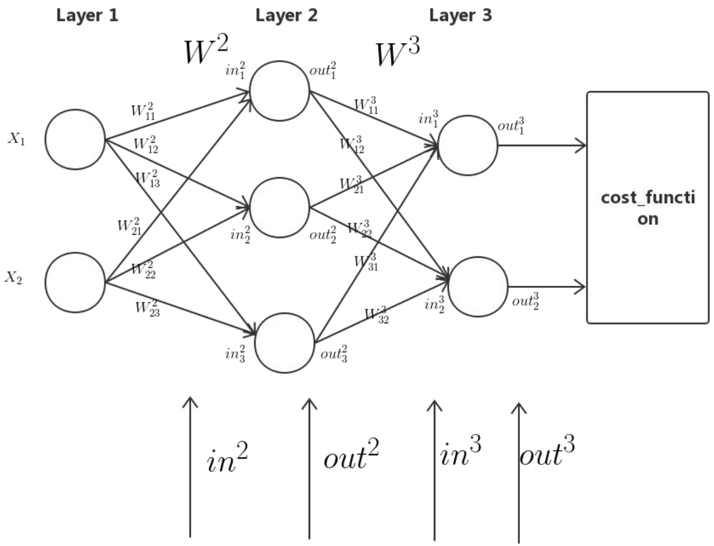
$$in_3^2 = W_{13}^2 * X_1 + W_{23}^2 * X_2 + b_3^2$$
$$out_3^2 = sigmoid(in_3^2)$$

$$in_1^3 = W_{11}^3 * out_1^2 + W_{21}^3 * out_2^2 + W_{31}^3 * out_3^2 + b_1^3$$
$$out_1^3 = sigmoid(in_1^3)$$

$$in_2^3 = W_{12}^3 * out_1^2 + W_{22}^3 * out_2^2 + W_{32}^3 * out_3^2 + b_2^3$$
$$out_2^3 = sigmoid(in_2^3)$$

$$cost - funtion = \frac{1}{2} * ((out_1^3 - y_1)^2 + (out_2^3 - y_2)^2)$$

backpropagation

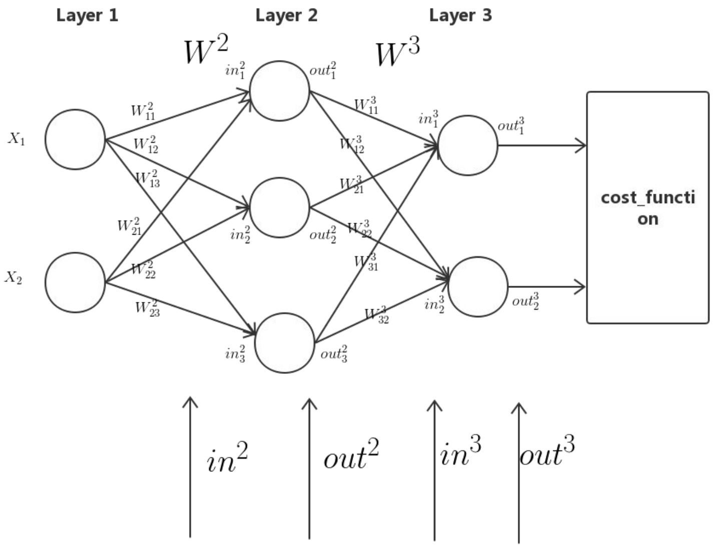


$$\frac{\partial C}{\partial in_1^3} = \frac{\partial C}{\partial out_1^3} * \frac{\partial out_1^3}{\partial in_1^3}$$
$$\frac{\partial C}{\partial in_2^3} = \frac{\partial C}{\partial out_2^3} * \frac{\partial out_2^3}{\partial in_2^3}$$

$$\frac{\partial C}{\partial in_1^2} = \frac{\partial C}{\partial out_1^3} * \frac{\partial out_1^3}{\partial in_1^3} * \frac{\partial in_1^3}{\partial out_1^2} * \frac{\partial out_1^2}{\partial in_1^2} + \frac{\partial C}{\partial out_2^3} * \frac{\partial out_2^3}{\partial in_2^3} * \frac{\partial in_2^3}{\partial out_1^2} * \frac{\partial out_1^2}{\partial in_1^2}$$
$$= \frac{\partial C}{\partial in_1^3} * \frac{\partial in_1^3}{\partial out_1^2} * \frac{\partial out_1^2}{\partial in_1^2} + \frac{\partial C}{\partial in_2^3} * \frac{\partial in_2^3}{\partial out_1^2} * \frac{\partial out_1^2}{\partial in_1^2}$$

$$\frac{\partial C}{\partial in_2^2} = \frac{\partial C}{\partial in_1^3} * \frac{\partial in_1^3}{\partial out_2^2} * \frac{\partial out_2^2}{\partial in_2^2} + \frac{\partial C}{\partial in_2^3} * \frac{\partial in_2^3}{\partial out_2^2} * \frac{\partial out_2^2}{\partial in_2^2}$$
$$\frac{\partial C}{\partial in_3^2} = \frac{\partial C}{\partial in_1^3} * \frac{\partial in_1^3}{\partial out_3^2} * \frac{\partial out_3^2}{\partial in_3^2} + \frac{\partial C}{\partial in_2^3} * \frac{\partial in_2^3}{\partial out_3^2} * \frac{\partial out_3^2}{\partial in_3^2}$$

backpropagation



$$\frac{\partial C}{\partial in^2} = \begin{bmatrix} \frac{\partial C}{\partial in_1^3} & \frac{\partial C}{\partial in_2^3} \end{bmatrix} \bullet \begin{bmatrix} \frac{\partial in_1^3}{\partial out_1^2} & \frac{\partial in_1^3}{\partial out_2^2} & \frac{\partial in_1^3}{\partial out_3^2} \\ \frac{\partial in_2^3}{\partial out_1^2} & \frac{\partial in_2^3}{\partial out_2^2} & \frac{\partial in_2^3}{\partial out_3^2} \end{bmatrix} * \begin{bmatrix} \frac{\partial out_1^2}{\partial in_1^2} & \frac{\partial out_2^2}{\partial in_2^2} & \frac{\partial out_3^2}{\partial in_3^2} \end{bmatrix}$$

$$= \frac{\partial C}{\partial in^3} \bullet \begin{bmatrix} W_{11}^3 & W_{21}^3 & W_{31}^3 \\ W_{12}^3 & W_{22}^3 & W_{32}^3 \end{bmatrix} * \frac{\partial out^2}{\partial in^2}$$

$$= \frac{\partial C}{\partial in^2} = \frac{\partial C}{\partial in^3} \bullet (W^3)^T * \frac{\partial out^2}{\partial in^2}$$

$$\Rightarrow \theta^l = \theta^{l+1} \bullet (W^{l+1})^T * \frac{\partial out^l}{\partial in^l}$$

$$\frac{\partial C}{\partial W^3} = (out^2)^T \bullet \frac{\partial C}{\partial in^3}$$

$$\frac{\partial C}{\partial b^3} = \frac{\partial C}{\partial in^3}$$

$$\frac{\partial C}{\partial W^2} = X^T \bullet \frac{\partial C}{\partial in^2}$$

$$\frac{\partial C}{\partial b^2} = \frac{\partial C}{\partial in^2}$$

code

```
# training samples 2 inputs and 2 outputs
X = np.random.rand(m, 2)
Y = np.random.rand(m, 2)

#layer 2
W2 = np.ones((2, 3))
b2 = np.ones((1, 3))
in2 = np.dot(X, W2) + b2
out2 = sigmoid(in2)

#layer 3
W3 = np.ones((3, 2))
b3 = np.ones((1, 2))
in3 = np.dot(out2, W3) + b3
out3 = sigmoid(in3)

#initial cost
cost = cost_function(out3, Y)
print("start:", cost)
```

```
#find derivative of cost function to in2 in layer3
derivative_c_out3 = np.subtract(out3, Y) / m
derivative_out3_in3 = derivative_sigmoid(in3)
derivative_c_in3 = np.multiply(derivative_c_out3, derivative_out3_in3)
#find derivative of cost function to W3 and b3 in layer3
dw3 = np.dot(out2.T, derivative_c_in3)
db3 = np.sum(derivative_c_in3, axis=0)

#find derivative of cost function to in2 in layer2
derivative_out2_in2 = derivative_sigmoid(in2)
derivative_c_in2 = np.multiply(np.dot(derivative_c_in3, W3.T), derivative_out2_in2)
#find derivative of cost function to W2 and b2 in layer2
dw2 = np.dot(X.T, derivative_c_in2)
db2 = np.sum(derivative_c_in2, axis=0)

#update all variables
W3 = W3 - step * dw3
W2 = W2 - step * dw2
b3 = b3 - step * db3
b2 = b2 - step * db2
```

code

```
('start:', 0.28756482038504333)
('cost:', 0.28755563012340696)
('cost:', 0.28660397594063114)
('cost:', 0.28558324993105377)
('cost:', 0.28448611699978854)
('cost:', 0.28330422587484388)
('cost:', 0.28202803804944432)
('cost:', 0.28064662375474281)
('cost:', 0.27914741820000205)
('cost:', 0.27751593005421449)
('cost:', 0.27573539280932535)
('cost:', 0.27378634841784633)
('cost:', 0.2716461517607095)
('cost:', 0.2692883846989213)
('cost:', 0.26668217084933615)
('cost:', 0.26379138887440773)
('cost:', 0.26057379662312585)
('cost:', 0.25698010711654135)
('cost:', 0.25295311041188734)
('cost:', 0.24842702901392363)
('cost:', 0.24332745233337377)
('cost:', 0.23757244702321492)
('cost:', 0.23107580771449787)
('cost:', 0.22375387466967633)
('cost:', 0.2155377435168222)
('cost:', 0.20639257461205315)
('cost:', 0.19634417596538914)
('cost:', 0.18550895234099596)
('cost:', 0.17411662088367003)
('cost:', 0.16250934319089053)
('cost:', 0.15110409672394695)
('cost:', 0.14032262268150253)
('cost:', 0.1305155910203018)
('cost:', 0.12191304910093054)
('cost:', 0.11461440226006991)
('cost:', 0.10860769529781944)
('cost:', 0.10380034132987744)
('cost:', 0.10004980824387838)
('end:', 0.099984453484645075)
('cnt:', 3603)
```

code

输出层:

$$\frac{\partial C}{\partial in_k} = \frac{\partial}{\partial in_k}\frac{1}{2}\sum_{k}^{-}(out_k - t_k)^2$$

$$= (out_k - t_k) * \frac{\partial out_k}{\partial in_k}$$

$$= (out_k - t_k) * \frac{\partial out_k}{\partial in_k}$$

$$= (out_k - t_k) * out_k * (1 - out_k)$$

$$\frac{\partial C}{\partial W_{jk}} = \frac{\partial C}{\partial in_k} * \frac{\partial in_k}{\partial W_{jk}}$$

$$= (out_k - t_k) * out_k * (1 - out_k) * out_j$$

隐藏层:

$$\frac{\partial C}{\partial in_j} = \frac{\partial}{\partial in_j}\frac{1}{2}\sum_{k}^{-}(out_k - t_k)^2$$

$$= \sum_k (out_k - t_k) * \frac{\partial out_k}{\partial in_j}$$

$$= \sum_k (out_k - t_k) * \frac{\partial out_k}{\partial in_k} * \frac{\partial in_k}{\partial in_j}$$

$$= \sum_k (out_k - t_k) * \frac{\partial out_k}{\partial in_k} * \frac{\partial in_k}{\partial out_j} * \frac{\partial out_j}{\partial in_j}$$

$$= \frac{\partial out_j}{\partial in_j} * \sum_k (out_k - t_k) * out_k * (1 - out_k) * W_{jk}$$

$$= out_j * (1 - out_j) * \sum_k (out_k - t_k) * out_k * (1 - out_k) * W_{jk}$$

$$= out_j * (1 - out_j) * \sum_k \frac{\partial C}{\partial in_k} * W_{jk}$$

$$\frac{\partial C}{\partial W_{ij}} = \frac{\partial C}{\partial in_j} * \frac{\partial in_j}{\partial W_{ij}}$$

$$= out_j * (1 - out_j) * O_i * \sum_k \frac{\partial C}{\partial in_k} * W_{jk}$$

Backpropagation algorithm

The back propagation algorithm

1. Run the network forward with your input data to get the network output
2. For each output node compute

$$\delta_k = \mathcal{O}_k(1 - \mathcal{O}_k)(\mathcal{O}_k - t_k)$$

3. For each hidden node calulate

$$\delta_j = \mathcal{O}_j(1 - \mathcal{O}_j) \sum_{k \in K} \delta_k W_{jk}$$

4. Update the weights and biases as follows
   Given

$$\Delta W = -\eta \delta_\ell \mathcal{O}_{\ell-1}$$
$$\Delta \theta = -\eta \delta_\ell$$

   apply

$$W + \Delta W \rightarrow W$$
$$\theta + \Delta \theta \rightarrow \theta$$

Thank you.