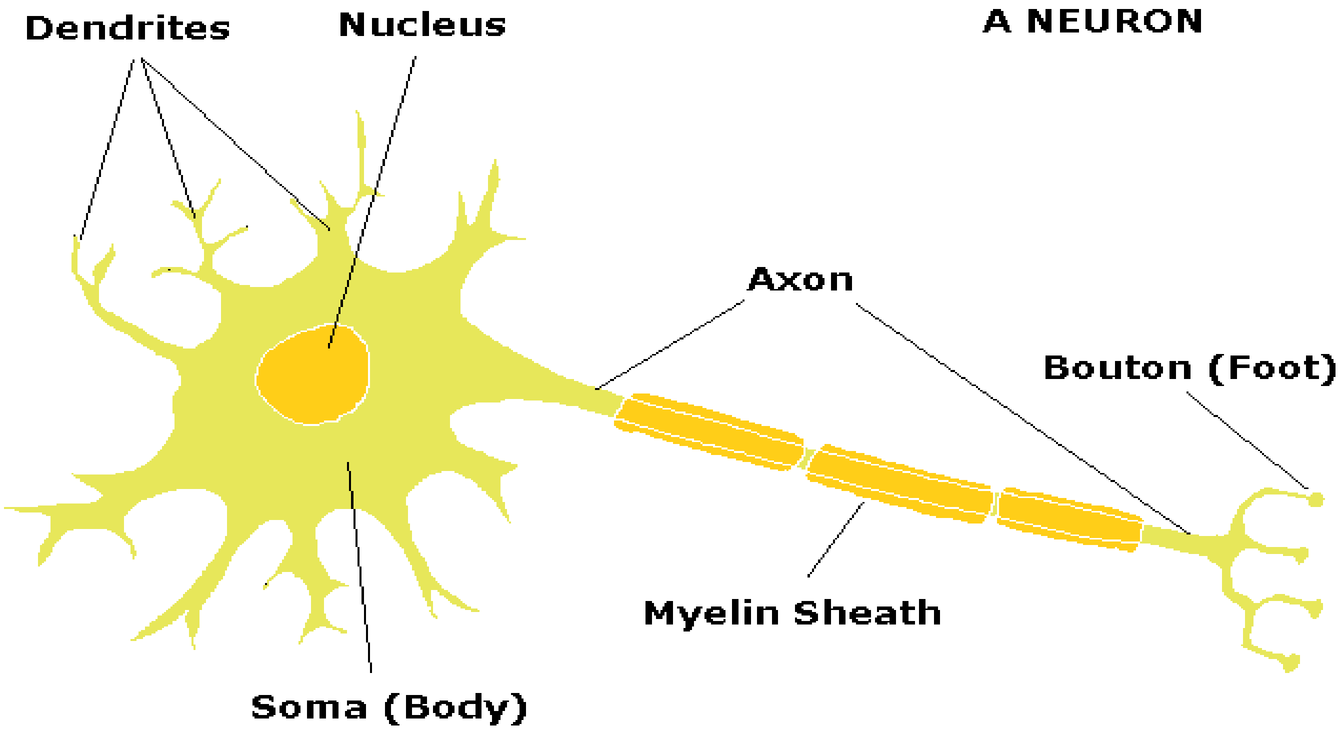
Neural Networks

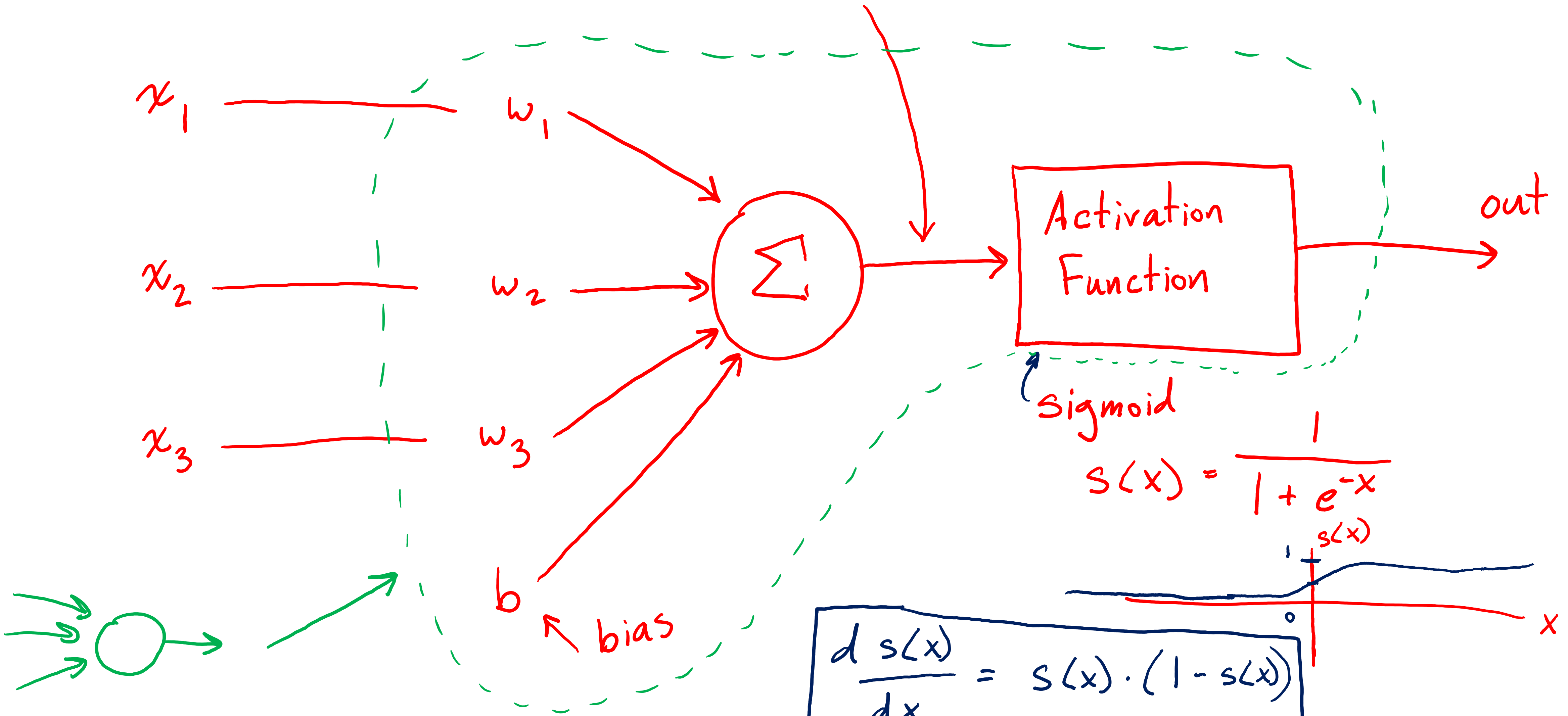
Neural Networks

- An artificial neural network is a mathematical model based on the operation of biological neurons



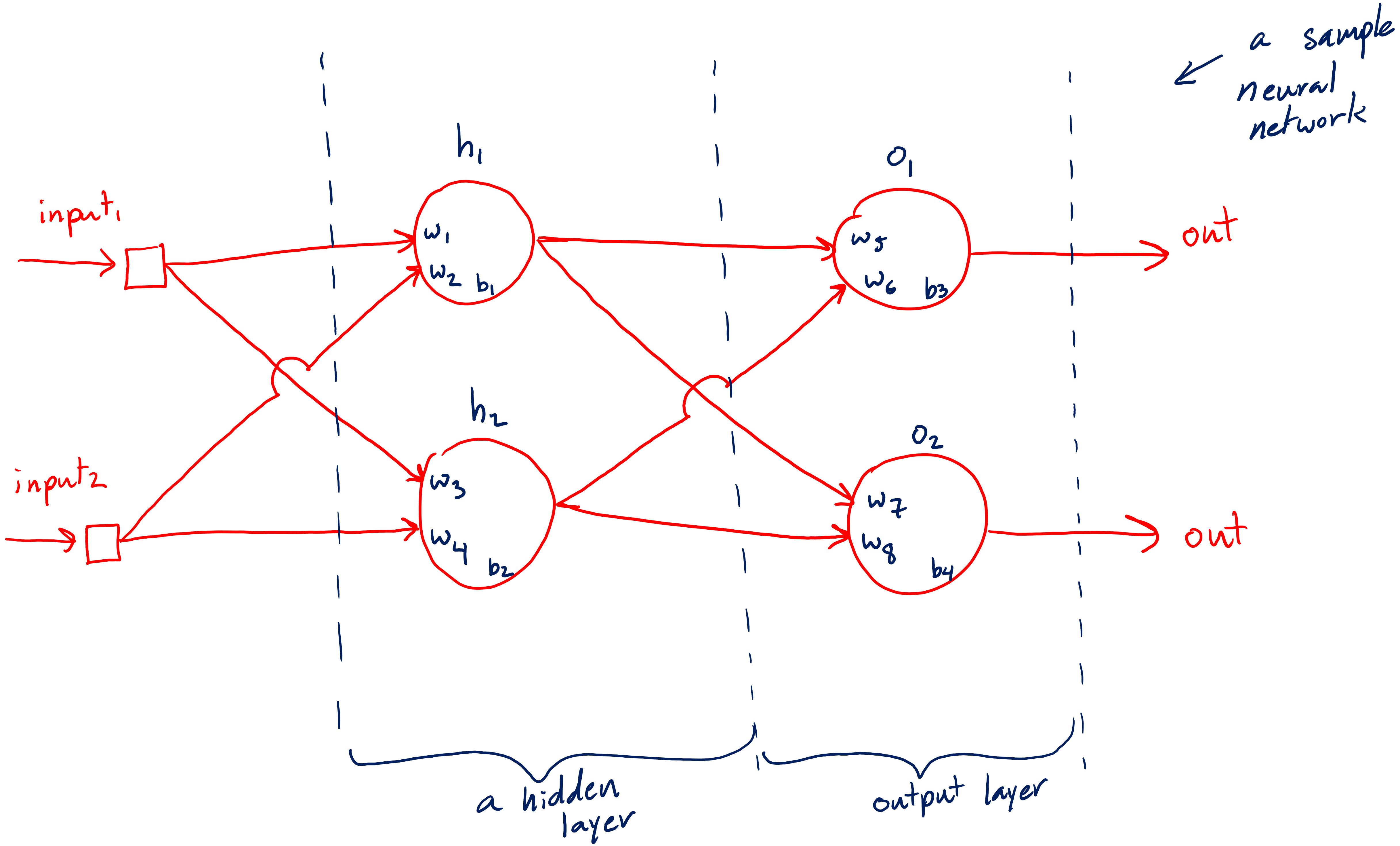
Artificial Neuron Model

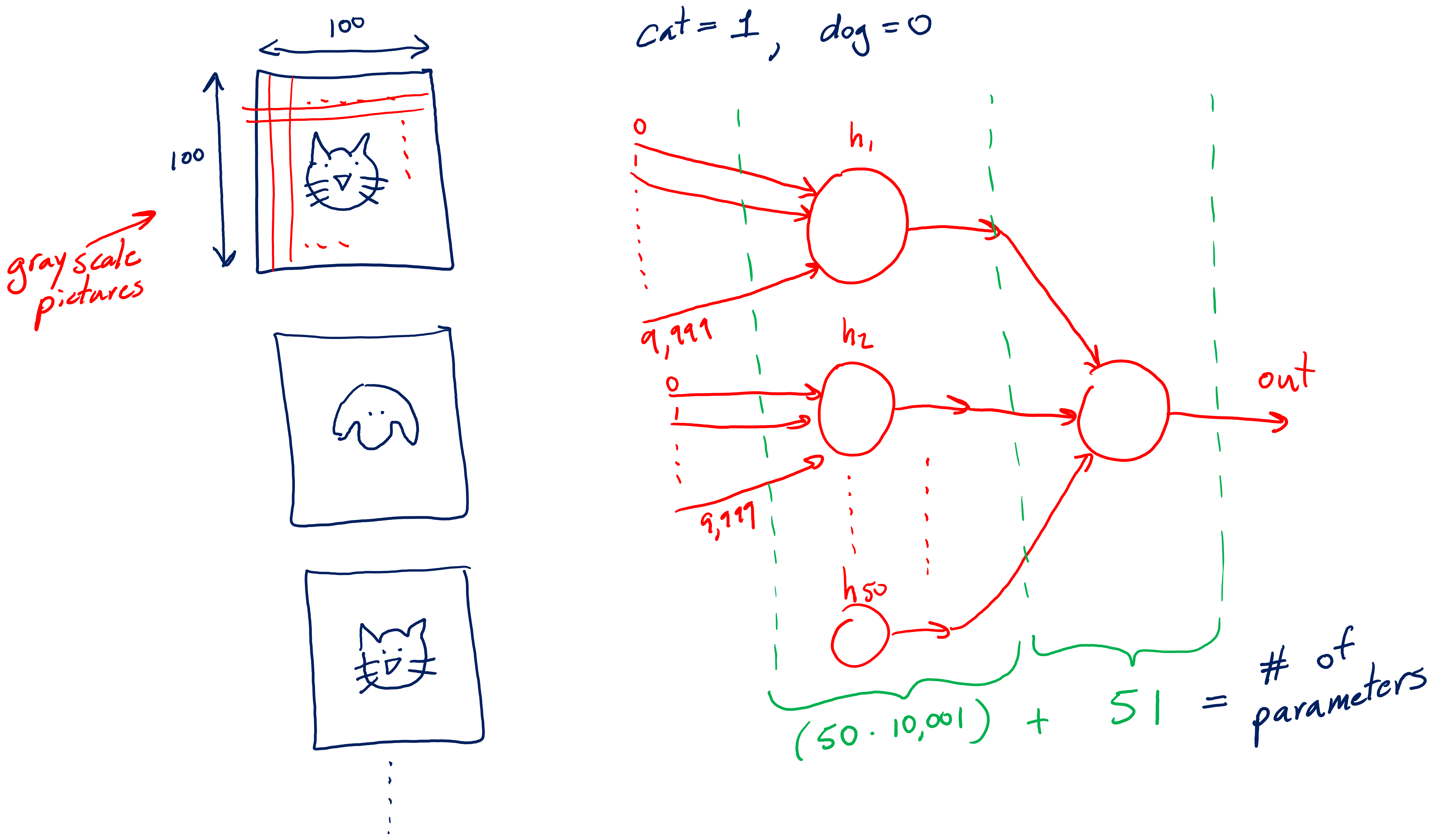
$$net = x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 + b$$



$$s(x) = \frac{1}{1 + e^{-x}}$$

$$\boxed{\frac{d\,s(x)}{dx} = s(x) \cdot (1 - s(x))}$$

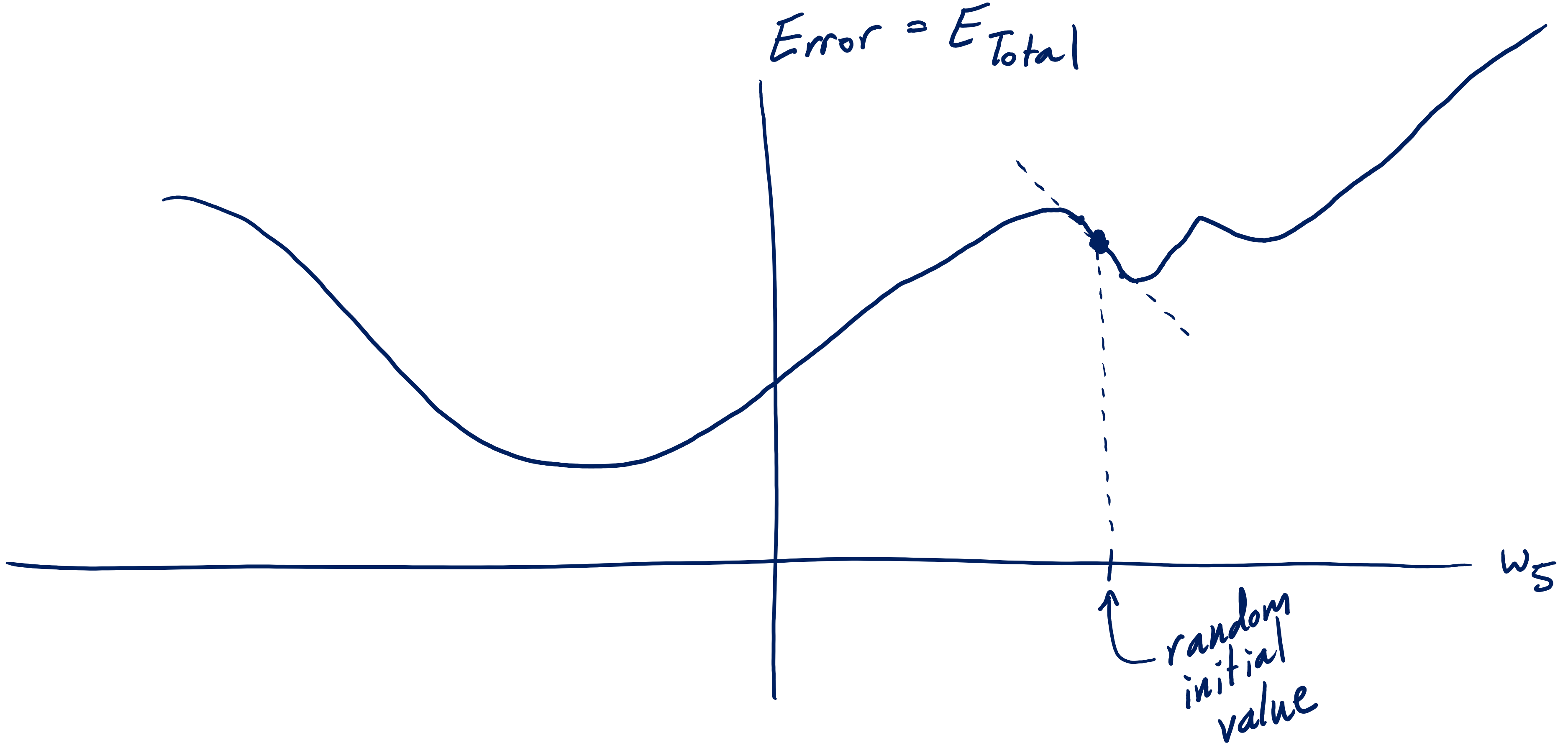

a sample neural network
input1
input2
h1
h2
o1
o2
w1
w2
b1
w3
w4
b2
w5
w6
b3
w7
w8
b4
out
out
a hidden layer
output layer



100
100
gray scale pictures
cat = 1, dog = 0
0
1
9,999
0
1
9,999
h1
h2
h50
out
(50 · 10,001) + 51 = # of parameters

Training - the process of adjusting the weights

Inference - using the trained network



Computing the Total Error

$$E_{Total} = E_{o_1} + E_{o_2}$$

$$E_{o_2} = \frac{1}{2}\left(target_{o_2} - actual_{o_2}\right)^2$$

$$E_{o_1} = \frac{1}{2}\left(target_{o_1} - actual_{o_1}\right)^2$$

Chain Rule

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}$$

Output layer weights

$$\frac{\partial E_{total}}{\partial w_5} = \underbrace{\frac{\partial E_{total}}{\partial out_{o_1}}}_{} \cdot \overbrace{\frac{\partial out_{o_1}}{\partial net_{o_1}}}^{out_{o_1}\cdot(1-out_{o_1})} \cdot \overbrace{\frac{\partial net_{o_1}}{\partial w_5}}^{out_{h_1}}$$

$$E_{total} = \frac{1}{2}(target_{o_1} - out_{o_1})^2 + \frac{1}{2}(target_{o_2} - out_{o_2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o_1}} = 2 \cdot \frac{1}{2}(target_{o_1} - out_{o_1}) \cdot -1$$

$$= (out_{o_1} - target_{o_1})$$

Computing a new weight

$$\text{new } w_5 = \text{old}_{w_5} - \alpha \cdot \frac{\partial E_{total}}{\partial w_5}$$

learning rate (.001 – .05)

$$net_{o_1} = out_{h_1} \cdot w_5 + out_{h_2} \cdot w_6 + b$$

$$\frac{\partial net_{o_1}}{\partial w_5} = out_{h_1}$$

Updating the Hidden Layer Weights

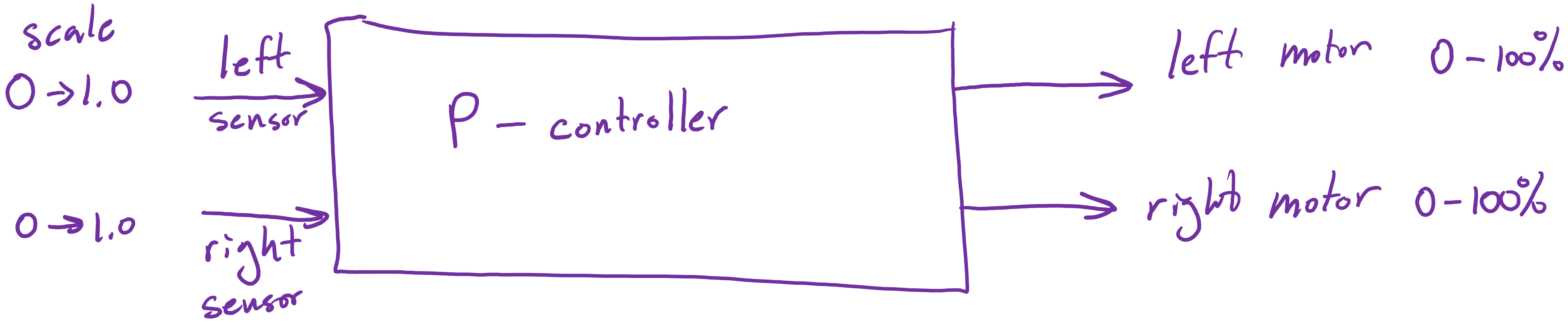
$$\frac{\partial E_{total}}{\partial w_1} = \underbrace{\frac{\partial E_{total}}{\partial out_{h_1}}}_{\frac{\partial E_{total}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial out_{h_1}} + \frac{\partial E_{o_2}}{\partial out_{h_1}}} \cdot \overbrace{\frac{\partial out_{h_1}}{\partial net_{h_1}}}^{\boxed{out_{h_1} \cdot (1 - out_{h_1})}} \cdot \overbrace{\frac{\partial net_{h_1}}{\partial w_1}}^{\text{the input value to } h_1 \text{ (connected to } w_1)}$$

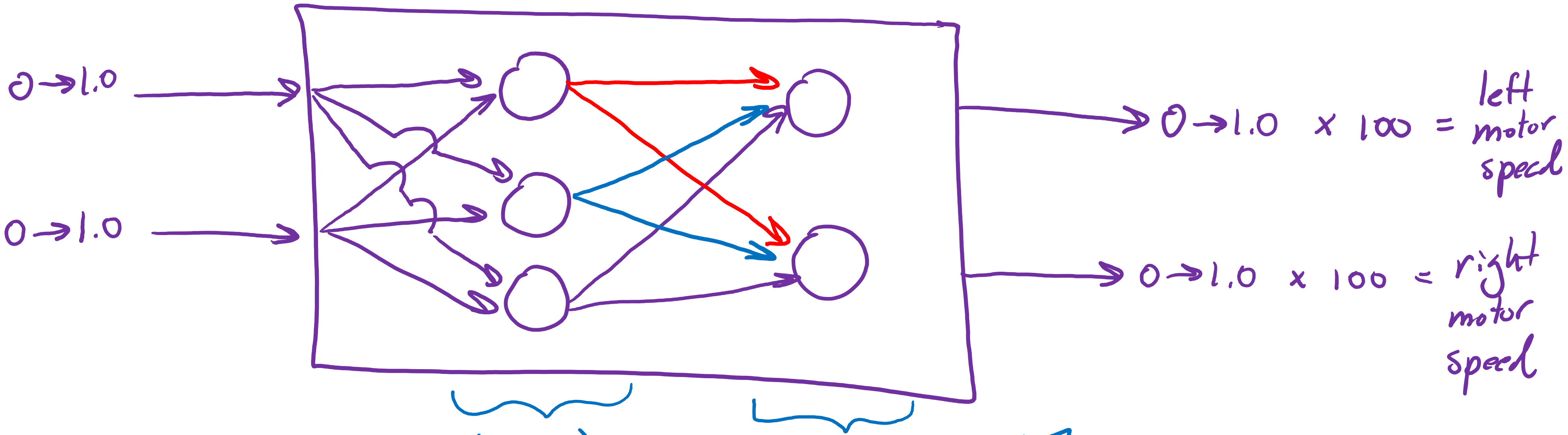
$$\frac{\partial E_{o_1}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial net_{o_1}} \cdot \boxed{\frac{\partial net_{o_1}}{\partial out_{h_1}}}$$

weight connecting h_1 and o_1

$$\frac{\partial E_{o_1}}{\partial net_{o_1}} = \boxed{\frac{\partial E_{o_1}}{\partial out_{o_1}} \cdot \frac{\partial out_{o_1}}{\partial net_{o_1}}}$$

already computed from output layer





$(3 \times 3) + (4 \times 2) = 17$

- when updating the biases, treat them as weights with a constant input of -1

1 epoch is training for all elements in a dataset

For each input example:

- compute new output weights/biases (but don't update yet)
- compute new hidden weights/biases (but don't update yet
- update all weights/biases