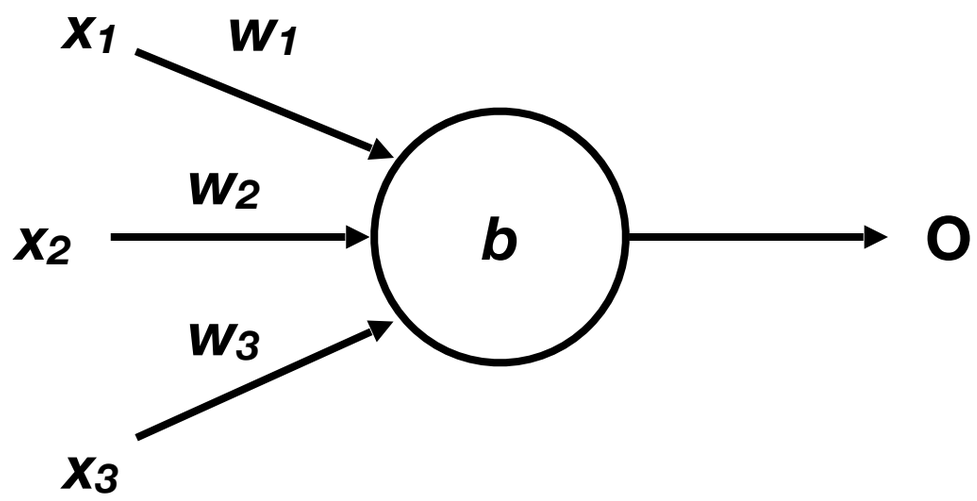
HUDK 4051: LEARNING ANALYTICS: PROCESS & THEORY

4/9/18 5:26 PM

Sigmoid Neurons

Notation

Inputs **Bias** **Output**

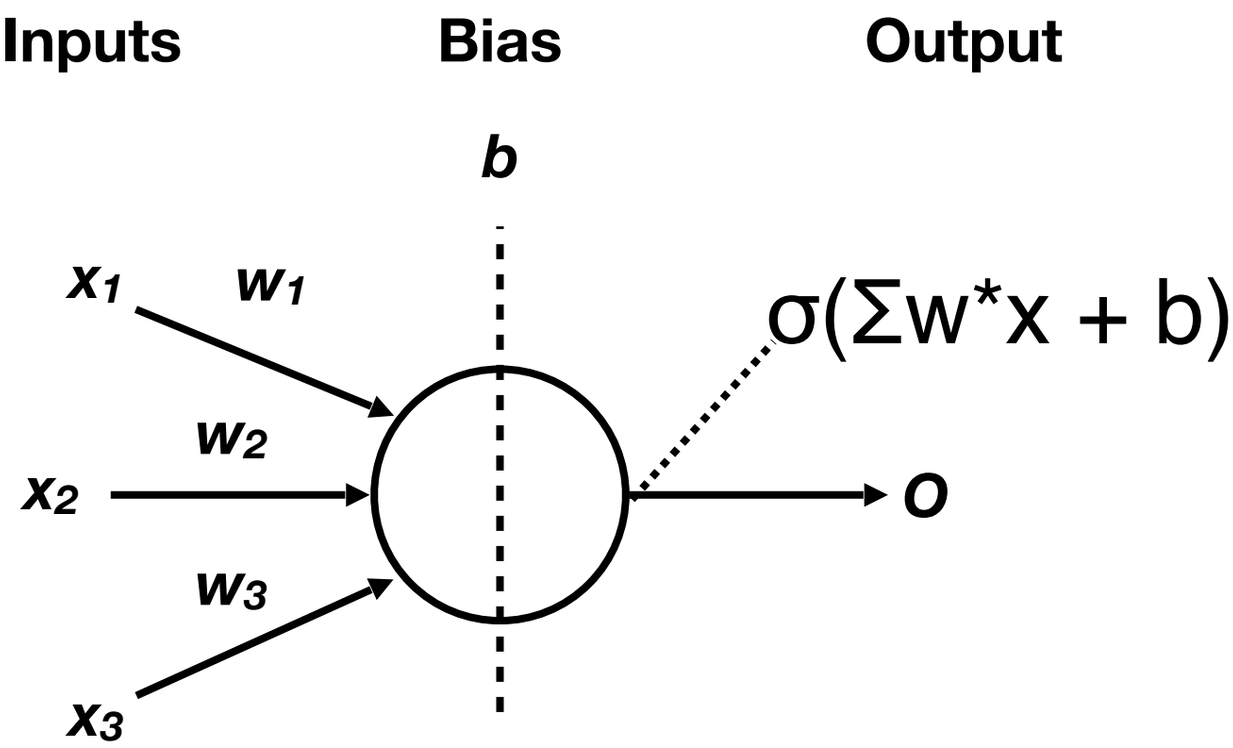


$$x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + b = O$$

Sigmoid Neurons

- Want to build a learning algorithm
- Could change b or w
- BUT that will cause very large changes
- Network will never “fix”
- Solution: “smooth” the output
- Bonus: standardize output

Sigmoid Neurons

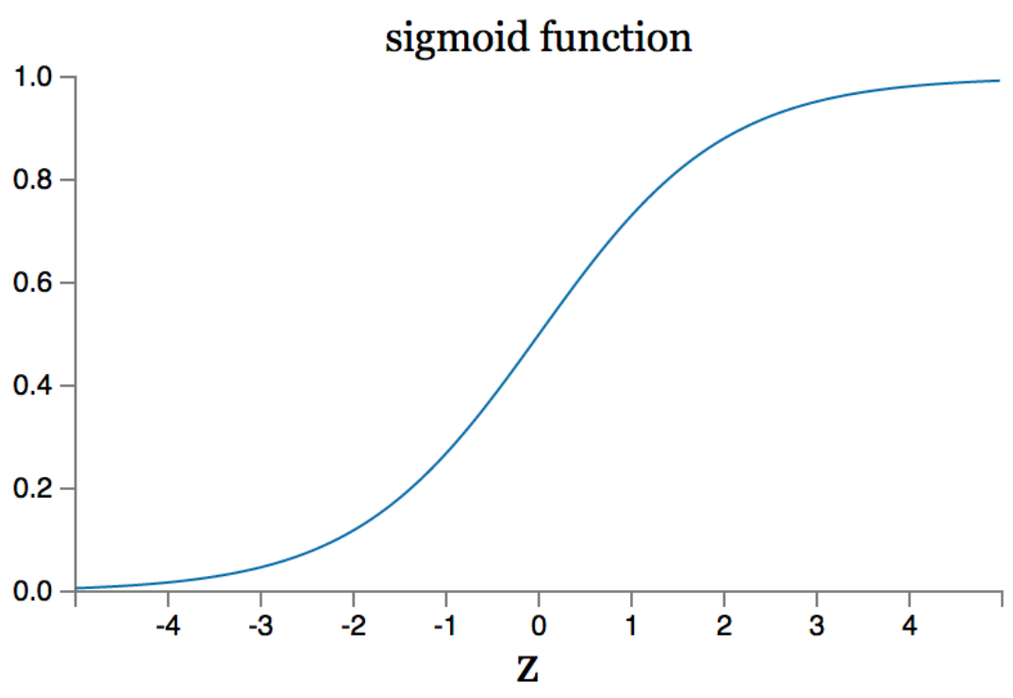


$$\sigma(x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + b) = O$$

Sigmoid Neurons

- Sigmoid (logistic) function "smooths" the output
- Makes changing w *and* b less sudden and more predictable
- Could use lots of other functions…

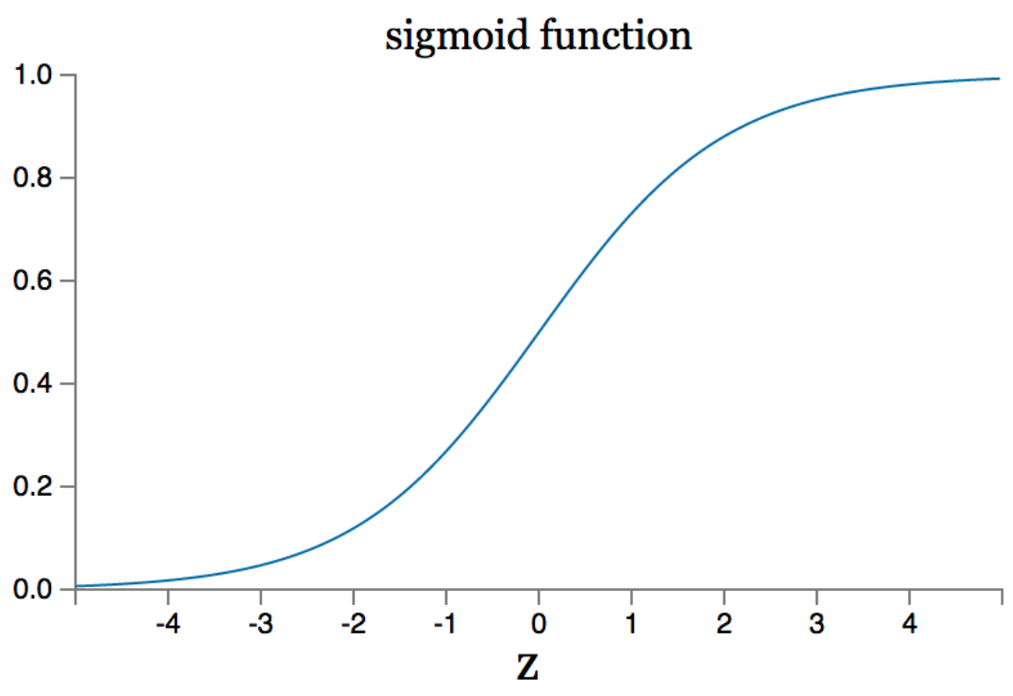
$$\sigma(z) \equiv \frac{1}{1+e^{-z}}.$$



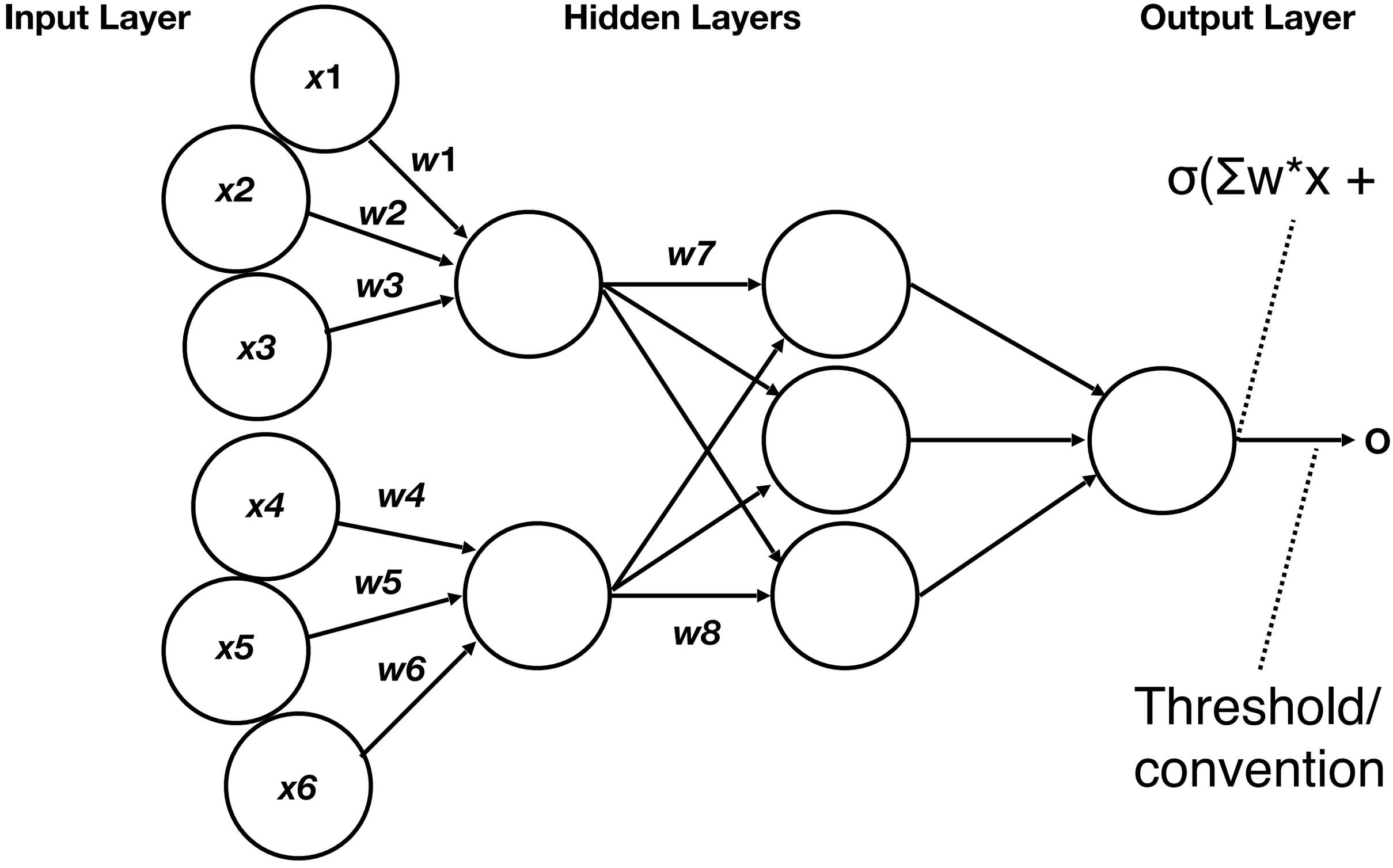
Sigmoid Neurons

- Perceptrons have 0/1 output
- Sigmoid neurons have 0 - 1 output (eg. 0.1, 0.6778, etc.)
- How to interpret sigmoid neuron output?

$$\sigma(z) \equiv \frac{1}{1+e^{-z}}.$$



Complete Feedfoward Network



How many Hidden Layers?

- No foolproof method
- The only method is really trial and error
- Heuristics:
  - Theory based starting point?
  - Number of inputs and outputs?

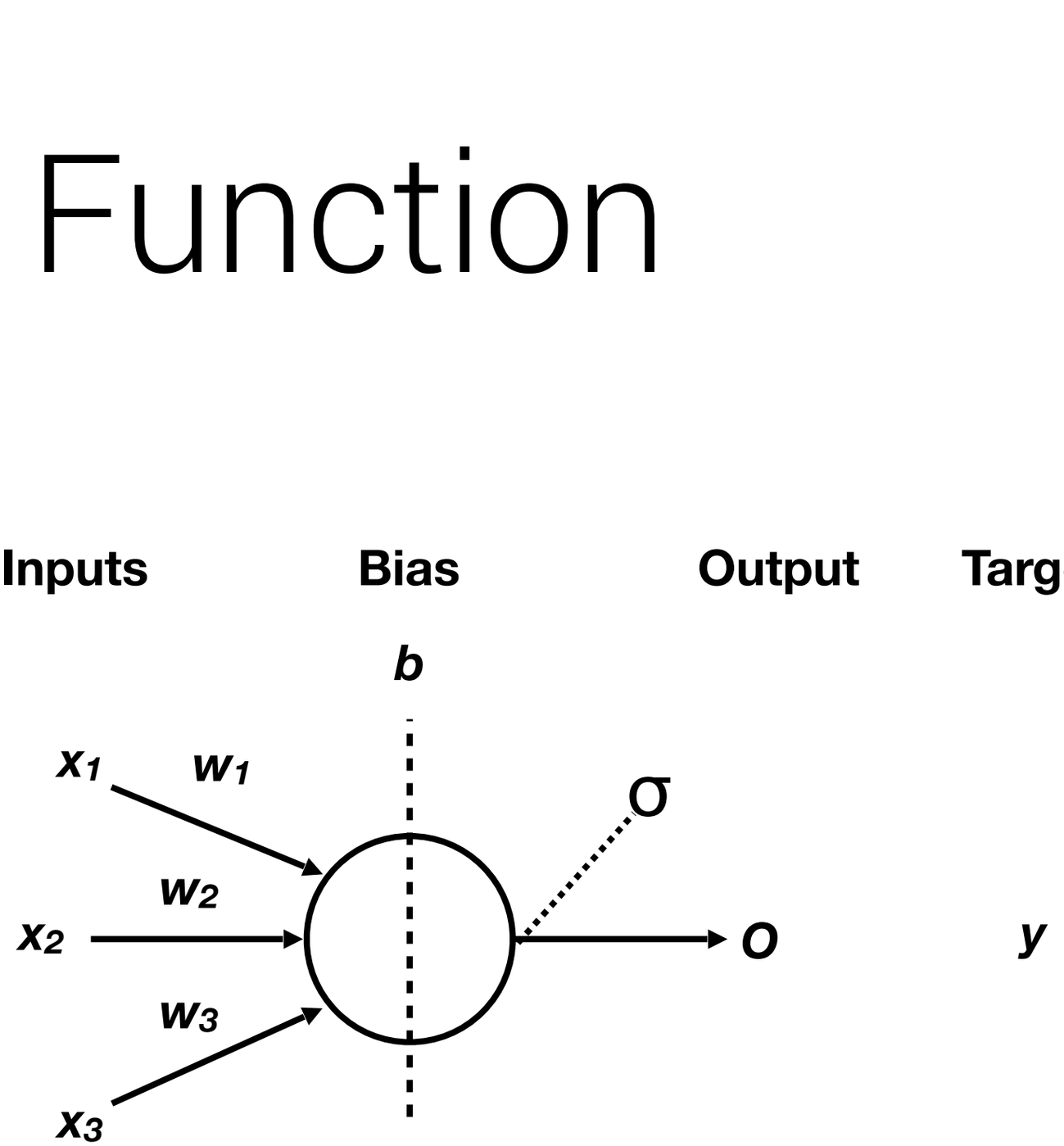
How to make the network learn?

Cost Function

Observation + prediction = error

Want to minimize this number

How to reduce error?

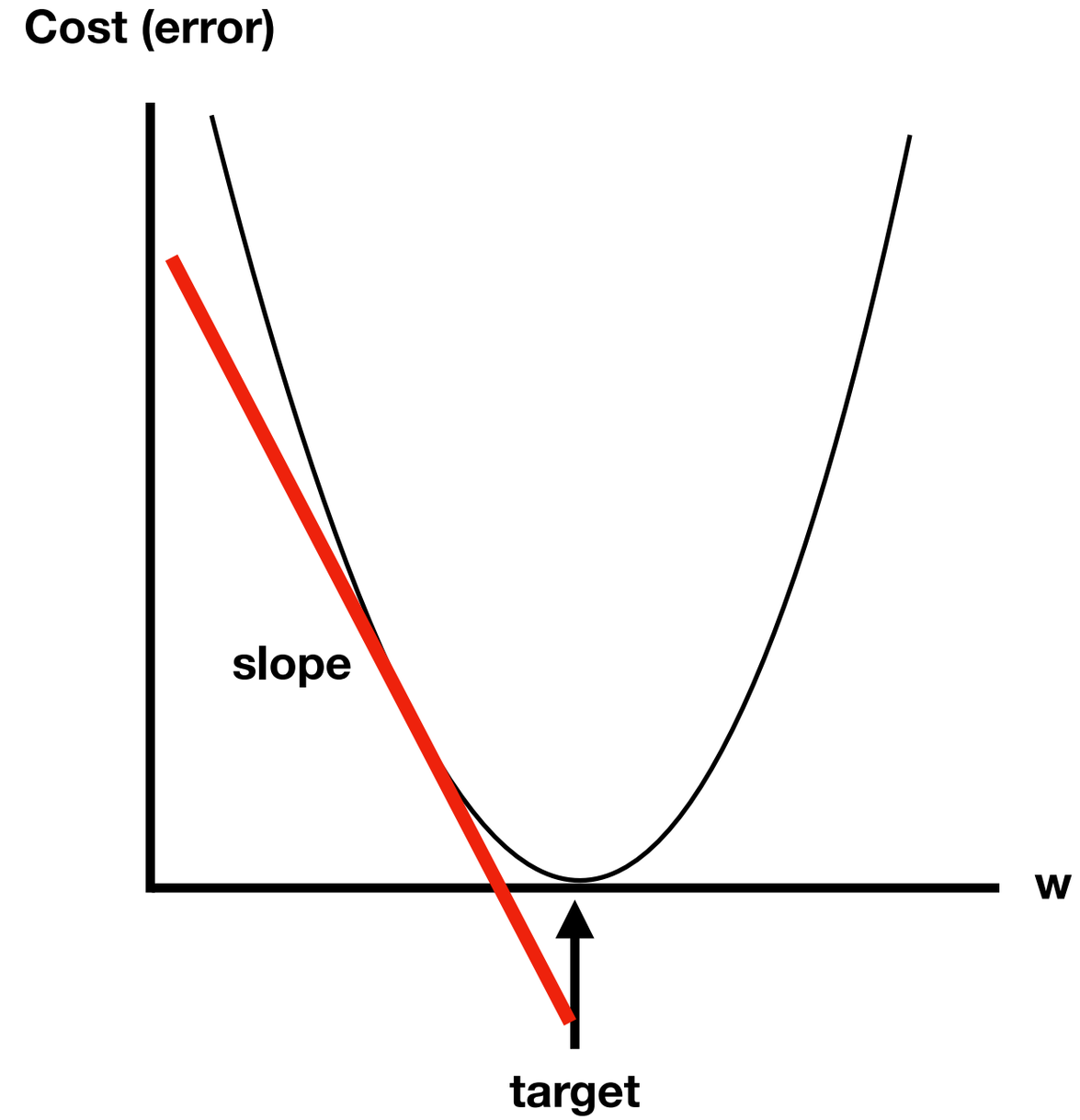


squared error = $(\text{prediction} - \text{target})^2$

Cost Function

How much to adjust weights/bias?

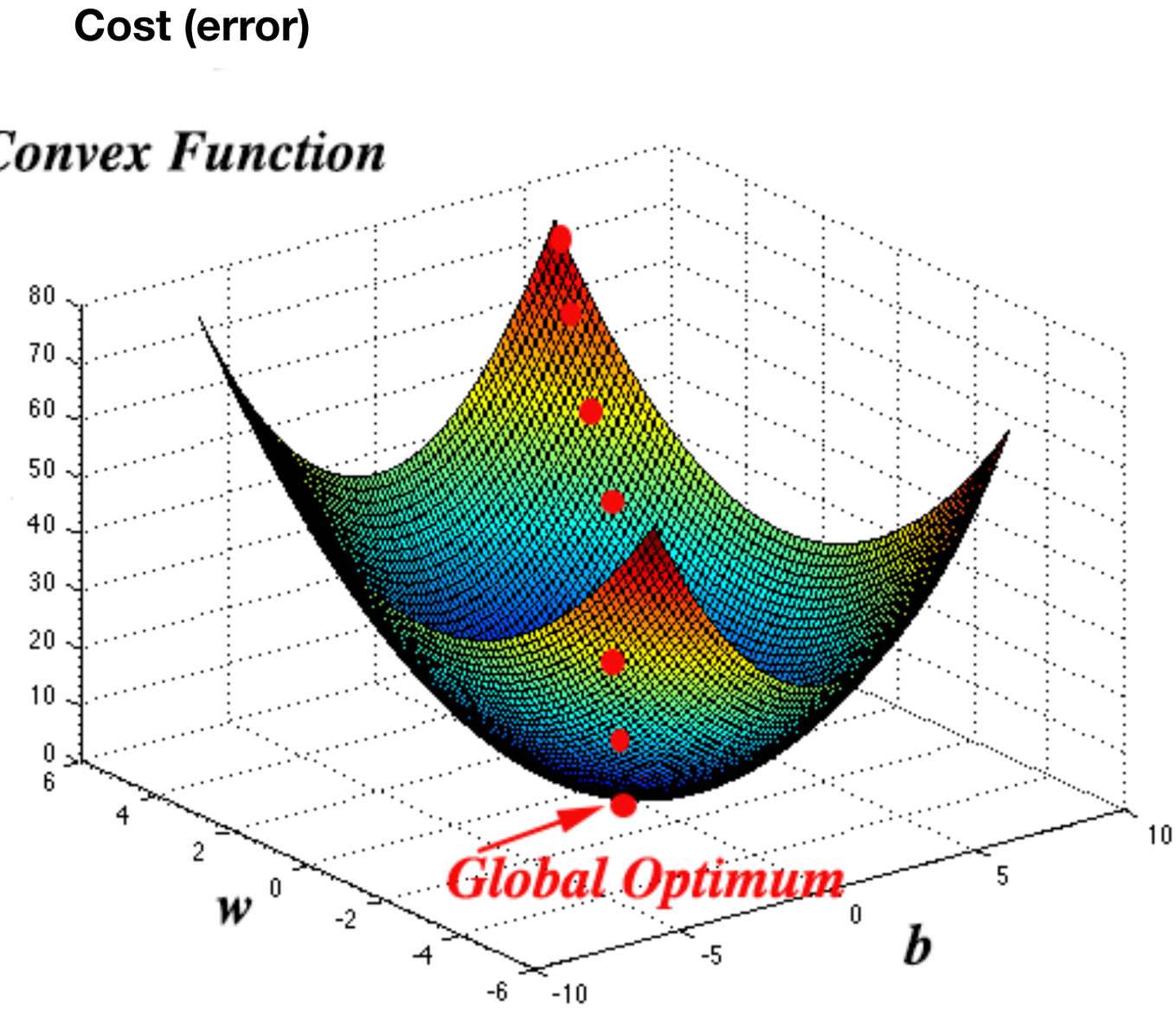
- Subtract the slope of the cost function
- Works if above or below target
- Change gets smaller as approach target



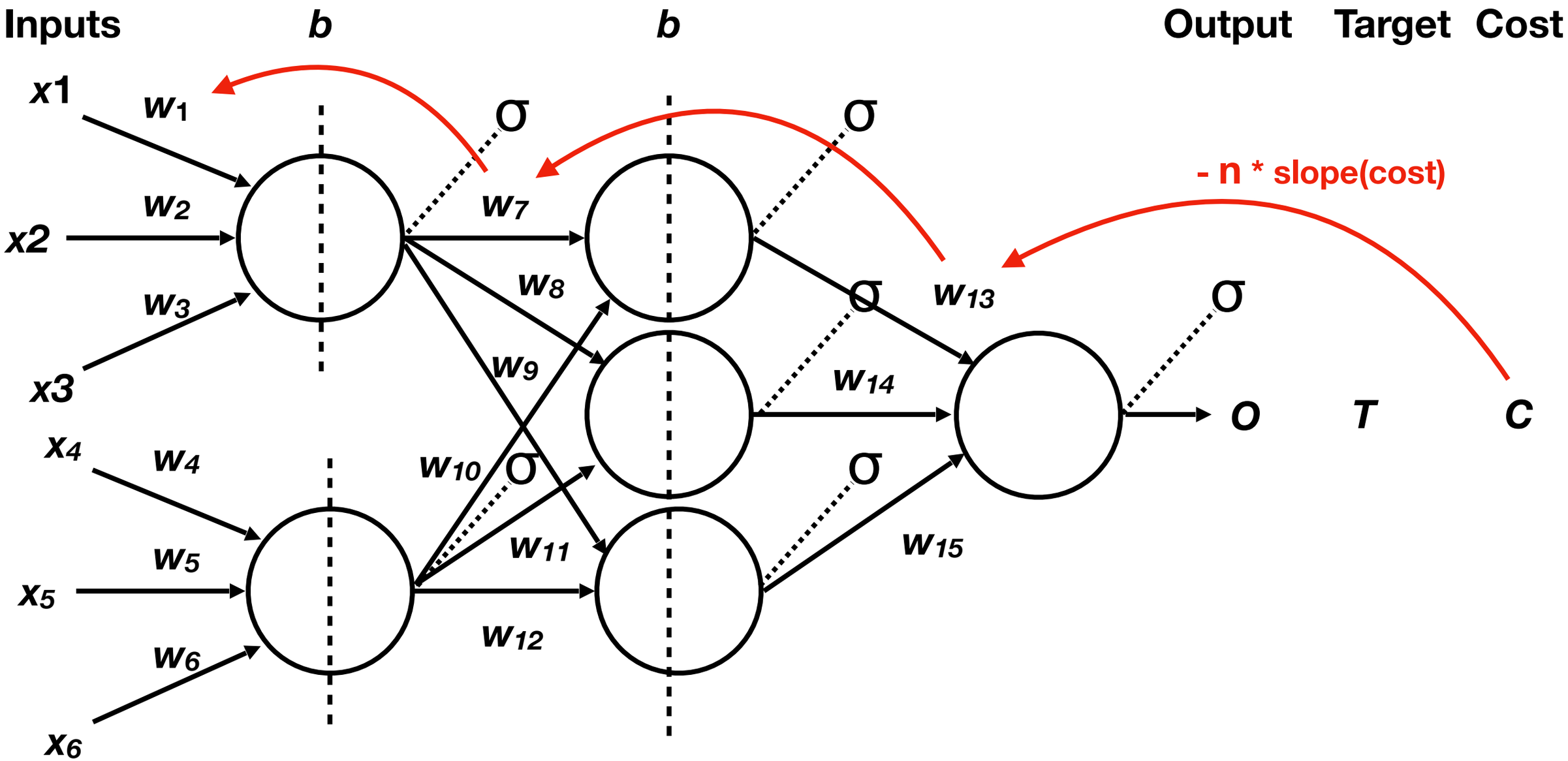
Cost Function

Where do we get the slop of the cost function?

- Approximation (rise/run)
- Algebraically
- Calculus



Back Propagation



$w_7 = 0.3$

$w_{7+1} = 0.3 - n*slope(costW_{13})$

$w_{13} = 0.8$

$w_{13+1} = 0.8 - n*slope(cost)$

0.6 1 $(0.4)^2$