Useful information about neural nets

- Neural nets are numerical classifiers with binary (0/1) output.
- The neuron is a primitive circuit element.
- Forward propagation computes the overall output of a neural net.
- Graphical intuition:
  inputs → First layer (draw lines) → Remaining layers (perform logic functions) → output (0/1)
- Backward propagation trains a neural net to imitate a given (unknown) function.
  - Threshold trick
  - Helper functions
  - Hill climbing (gradient ascent)
  - Quick formulas to replace the chain rule

Helper functions

$$Stairstep_T(x) = \begin{cases} 1 \text{ if } x \geq T \\ 0 \text{ if } x < T \end{cases}$$

$$Sigmoid_{S,M}(x) = \frac{1}{1 + e^{-S(x-M)}}$$

$$Performance = Accuracy\ (out^*, out) = -\ ½\ (out^* - out)^2$$

Quick formulas for backward propagation

$$w_{A \to B,new} = w_{A \to B,old} + \Delta w_{A \to B}$$

$$\Delta w_{A \to B} = r \times out_A \times \delta_B$$

$\delta_B = out_B\ (1 - out_B)\ (out^* - out)$ , if neuron B is in final layer

$\delta_B = out_B (1 - out_B) \sum_{outgoing\ C_i} w_{B \to C_i} \delta_{C_i}$ , if neuron B is <u>not</u> in final layer

A single step of backward propagation involves:

1. Compute output of each neuron using forward propagation and stairstep function
2. Compute $\delta_B$ for final layer
3. Compute $\delta_B$ for earlier layers
4. Compute updates for weights
5. Update all weights