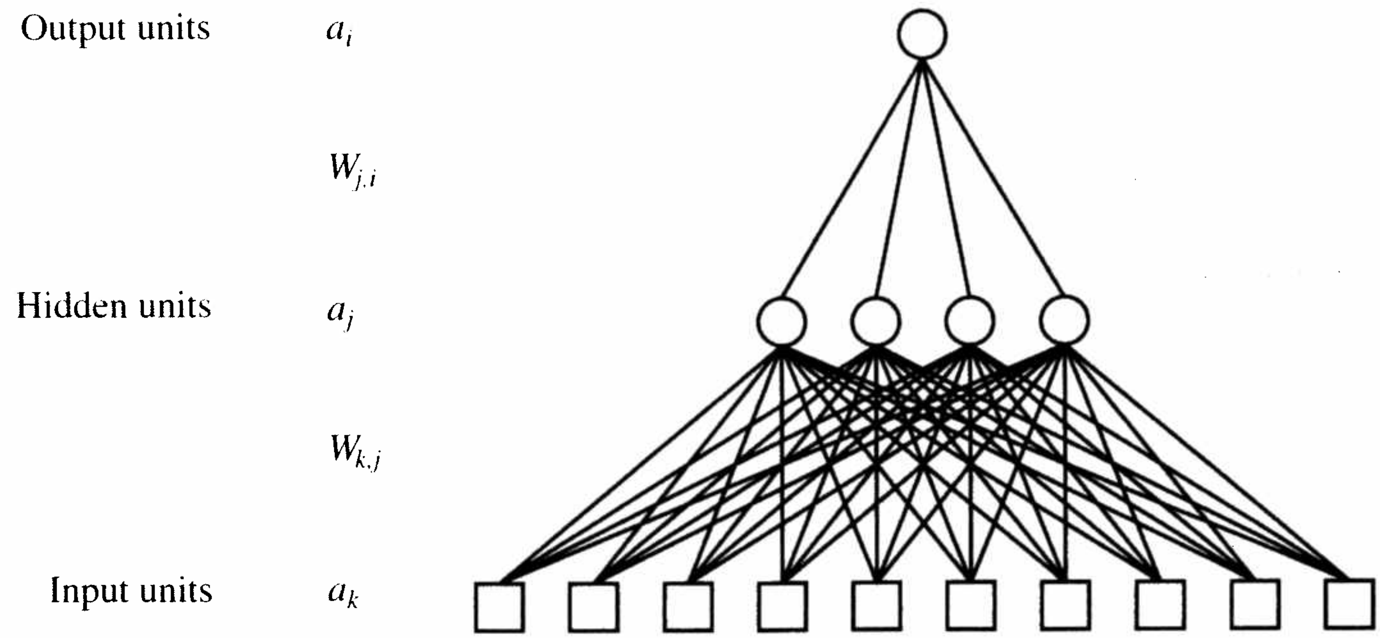


Figure 20.24 A multilayer neural network with one hidden layer and 10 inputs, suitable for the restaurant problem.

Back-propagation learning

Output layer: same as for single-layer perceptron,

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i$$

where $\Delta_i = Err_i \times g'(in_i)$

Hidden layer: **back-propagate** the error from the output layer:

$$\Delta_j = g'(in_j) \sum_i W_{j,i} \Delta_i \ .$$

Update rule for weights in hidden layer:

$$W_{k,j} \leftarrow W_{k,j} + \alpha \times a_k \times \Delta_j \ .$$

(Most neuroscientists deny that back-propagation occurs in the brain)

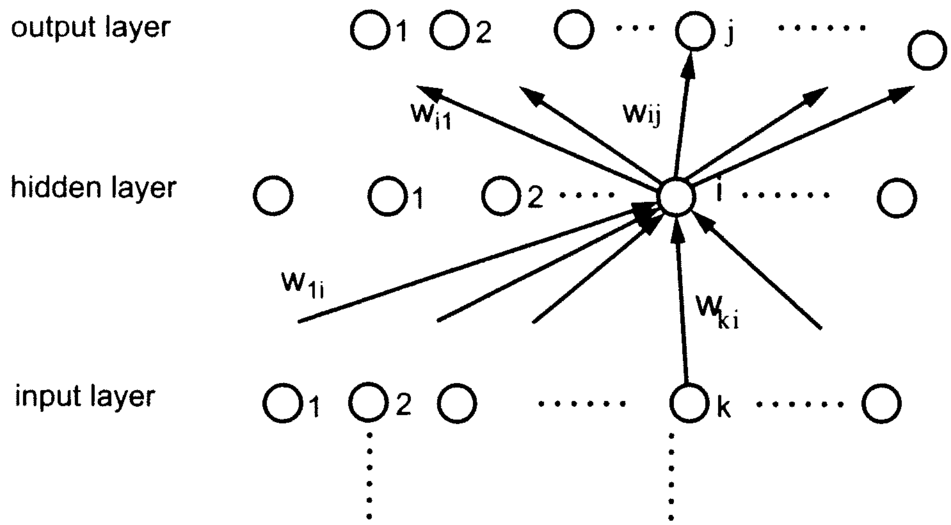


Figure 11.10 $\sum_j -\text{delta}_j * w_{ij}$ is the total contribution of node i to the error at the output. Our derivation gives the adjustment for w_{ki}.

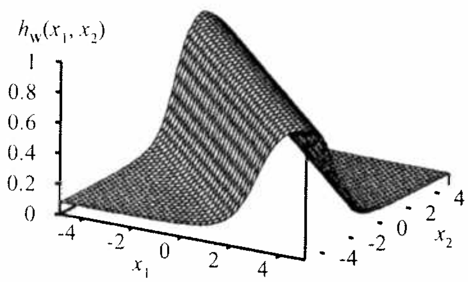
function BACK-PROP-LEARNING(*examples*, *network*) **returns** a neural network
  inputs: *examples*, a set of examples, each with input vector **x** and output vector **y**
    network, a multilayer network with L layers, weights $W_{j,i}$, activation function g

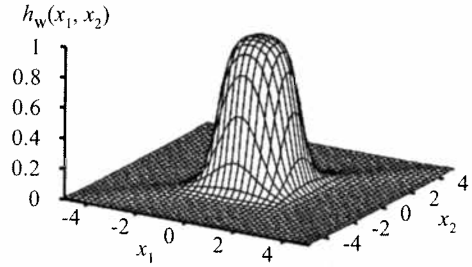
  repeat
    for each e **in** *examples* **do**
      for each node j in the input layer **do** $a_j \leftarrow x_j[e]$
      for $\ell = 2$ **to** M **do**
        $in_i \leftarrow \sum_j W_{j,i}\, a_j$
        $a_i \leftarrow g(in_i)$
      for each node i in the output layer **do**
        $\Delta_i \leftarrow g'(in_i) \times (y_i[e] - a_i)$
      for $\ell = M - 1$ **to** 1 **do**
        for each node j in layer ℓ **do**
          $\Delta_j \leftarrow g'(in_j) \sum_i W_{j,i}\, \Delta_i$
          for each node i in layer $\ell + 1$ **do**
            $W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i$
  until some stopping criterion is satisfied
  return NEURAL-NET-HYPOTHESIS(*network*)

Figure 20.25 The back-propagation algorithm for learning in multilayer networks.

Expressiveness of MLPs

All continuous functions w/ 2 layers, all functions w/ 3 layers





Combine two opposite-facing threshold functions to make a ridge

Combine two perpendicular ridges to make a bump

Add bumps of various sizes and locations to fit any surface

Proof requires exponentially many hidden units (cf DTL proof)

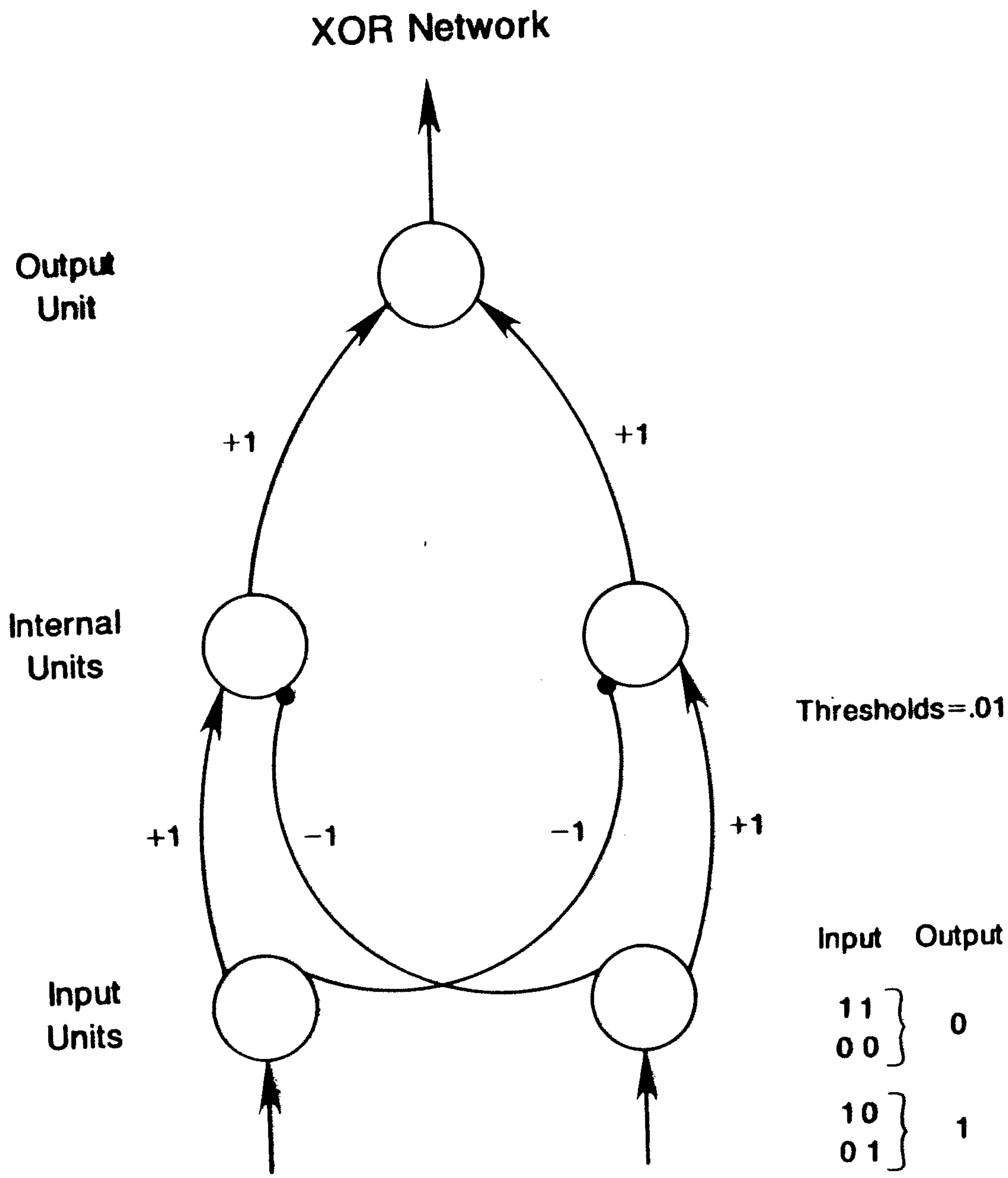


FIGURE 4. A network of linear threshold units capable of responding correctly on the XOR problem.

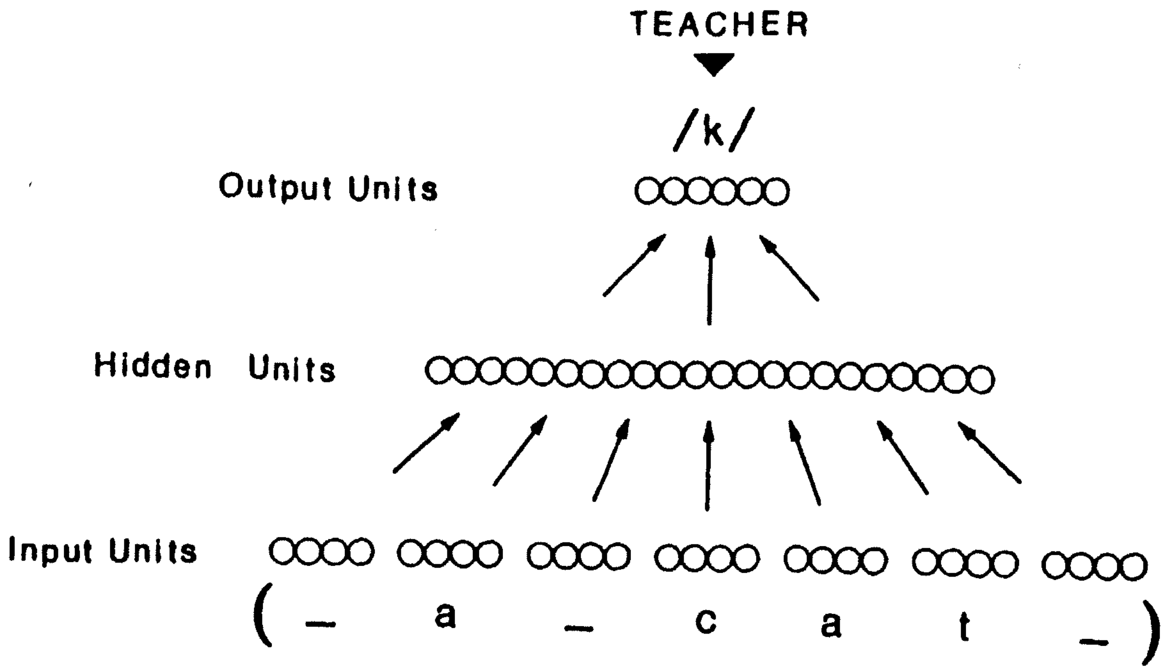


Figure 1: Schematic drawing of the NETtalk network architecture. A window of letters in an English text is fed to an array of 203 input units. Information from these units is transformed by an intermediate layer of 80 "hidden" units to produce patterns of activity in 26 output units. The connections in the network are specified by a total of 18629 weight parameters (including a variable threshold for each unit).

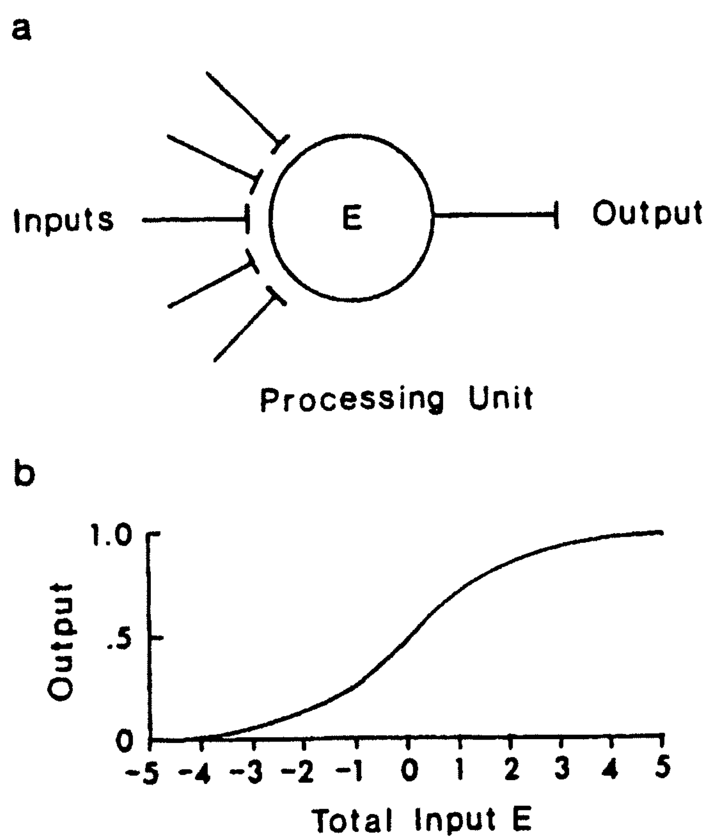


Figure 2: (a) Schematic form of a processing unit receiving inputs from other processing units. (b) The output $P(E)$ of a processing unit as a function of the sum E of its inputs.