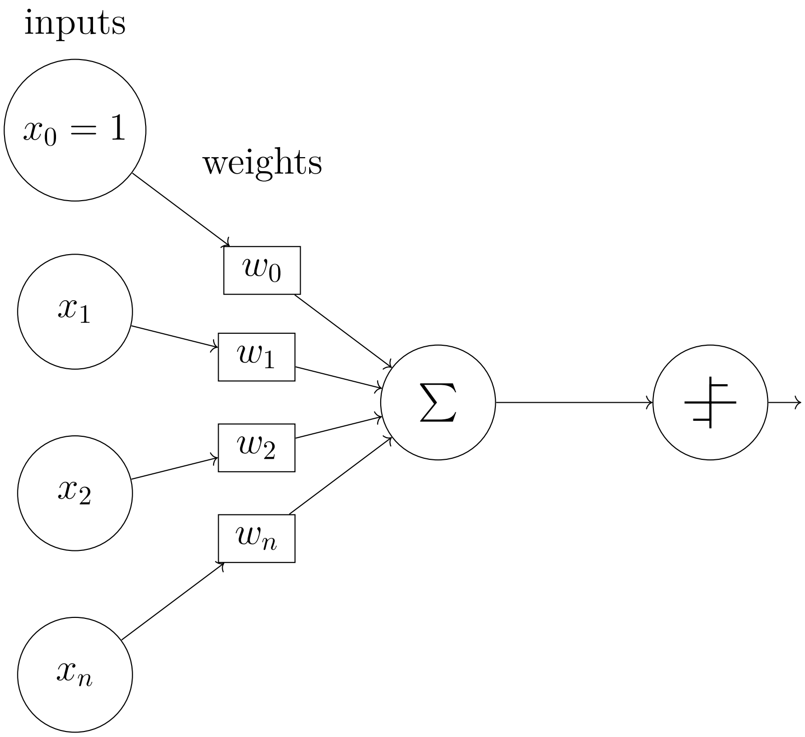
1 A perceptron

A perceptron:



Activation functions:

- Step function: $f(x) = 1$ if $x > 0$. $f(x) = 0$ if $x \leq 0$.

  The step function is simple to use. However, it is not differentiable. Many optimization algorithms such as gradient descent requires a function to be differentiable.

- Sigmoid function: $f(x) = \dfrac{1}{1 + e^{-x}}$.

  The sigmoid function can approximate the step function. A general version of the sigmoid function is $f(x) = \dfrac{1}{1 + e^{-kx}}$ where k is a constant parameter. As k increases, the sigmoid function becomes more steep and is more close to the step function. However, the sigmoid function is differentiable and works well with many optimization algorithms.

  Problem: Towards either end of the sigmoid function, the Y values tend to respond very less to changes in X. The gradient at that region is going to be small. It gives rise to a problem of "vanishing gradients". Gradient is small or has vanished. The network refuses to learn further or is drastically slow.

- Rectified linear unit (ReLu) $f(x) = max(0, x)$.

  Any function can be approximated with a combination of ReLus. Some neurons are firing and other ones are not. When $x < 0$, the gradient is 0 and the neuron will stop responding to changes. A fix: Leaky Relu: $y = 0.01x$ for $x < 0$.

2 Learning a multi-layer feed-forward neural network

- Multi-layered:
  - a layer of input units
  - One or more layers of hidden units
  - A layer of output units
- Feed-forward:
  - information flows from input layer, to hidden layer, to output layer
  - no loops: the outputs of a unit cannot influence its inputs. (Recurrent neural networks have loops.)
- Each unit uses some activation function.

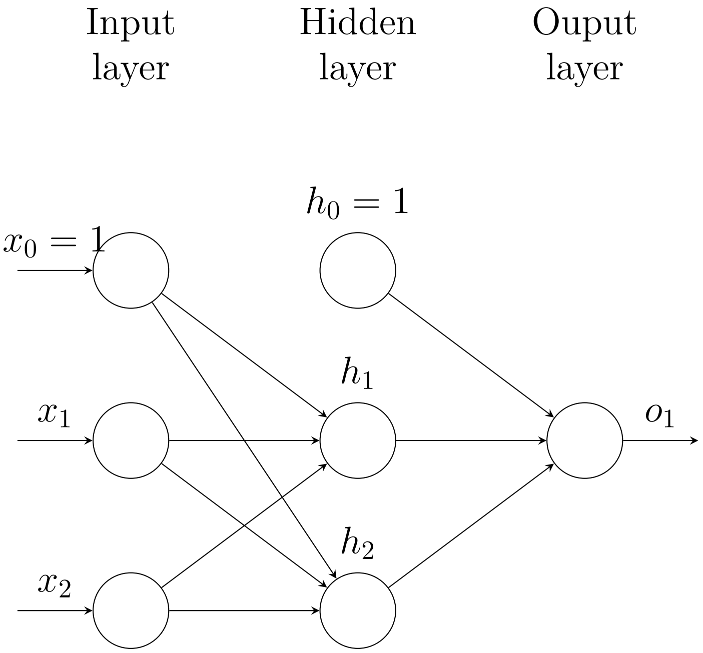
Representing the XOR function using a three-layered feed-forward network

The XOR function is defined by the following truth table.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

XOR can be modeled by using a neural network with one hidden layer.

- Two input units.
- Two hidden units
- One output unit
- The activation function is the step function.



$$h_1 = f(x_1 + x_2 - 0.5) \tag{1}$$
$$h_2 = f(-x_1 - x_2 + 1.5) \tag{2}$$
$$o_1 = f(h_1 + h_2 - 1.5) \tag{3}$$

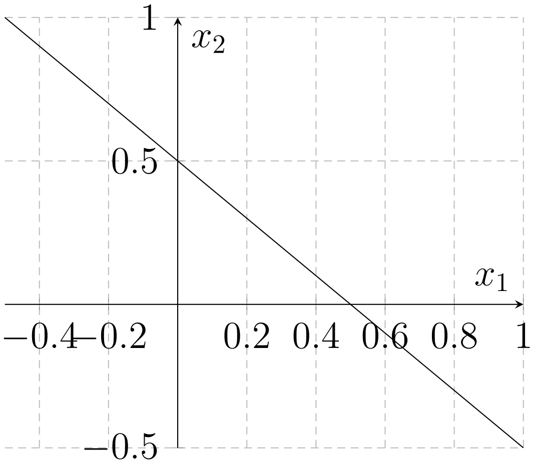
What do h_1, h_2 and o_1 compute?

By writing out the truth tables for h_1, h_2 and o_1, we can figure out the corresponding logical expressions.

x_1	x_2	h_1	h_2	o_1
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

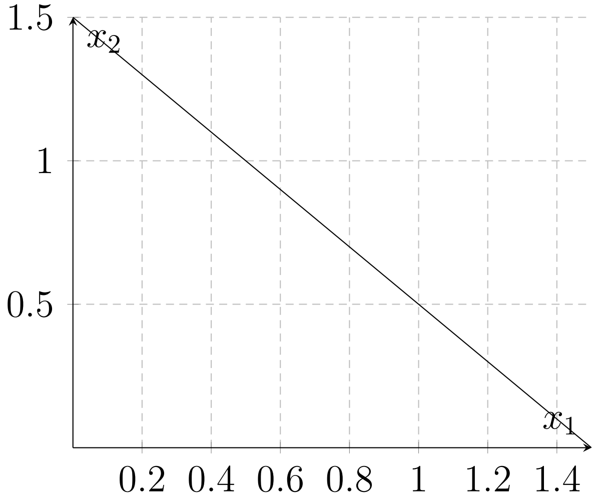
- h_1 is computing $(x_1 \vee x_2)$

  h_1 corresponds to the line $x_1 + x_2 - 0.5 = 0$ or $x_2 = -x_1 + 0.5$.

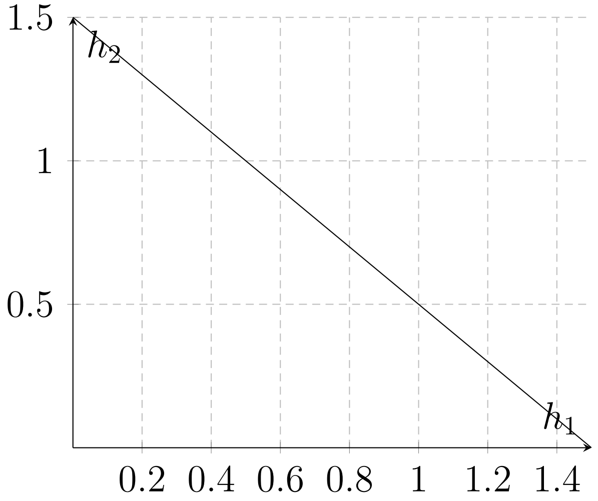


- h_2 is computing $(\neg(x_1 \wedge x_2))$

  h_2 corresponds to the line $x_1 + x_2 - 1.5 = 0$ or $x_2 = -x_1 + 1.5$.



- o_1 is computing $(h_1 \wedge h_2) \equiv ((x_1 \vee x_2) \wedge (\neg(x_1 \wedge x_2))) \equiv x_1 \oplus x_2$.



To describe the back-propagation algorithm, we first introduce some notation.

- A is the number of units in input layer.

  B is the number of units in hidden layer.

  C is the number of units in output layer.

- $x_i \in \{0, 1\}, i = 0, \ldots, A$ denote the values of the input units.

  $h_j \in \{0, 1\}, j = 0, \ldots, B$ denote the values of the hidden units.

  $o_k \in \{0, 1\}, k = 1, \ldots, C$ denote the values of the output units.

- $w1_{ij}$ is the weight on line between input unit x_i and hidden unit h_j.

  $w2_{jk}$ is the weight on line between hidden unit h_j and output unit o_k.

To measure the error between the desired output values and the actual output values, we will use the squared difference function.

$$\text{error} = \frac{1}{2} \sum_{k=1}^{C} (y_k - o_k)^2.$$

There are other possible error functions such as the absolute difference function. The advantage of the squared difference function is that it is differentiable and thus can be used in conjunction with optimization algorithms that require us to compute derivatives.

Gradient descent

The back-propagation algorithm uses the idea of gradient descent.

- A function f in many variables $x_1, \ldots, x_n$
- Goal is to minimize f.
- Start at a random point.
- Calculate the gradient — the derivative of f with respect to each x_i

  The gradient tells us: if I change x_i by 1, how does the value of f change (increase or decrease) and how much does the value of f change?
- In what direction should we change x_i? (Should we increase or decrease it?) We need to change x_i in the direction, which is opposite to the sign of the gradient.

  The gradient tells us how the function changes when we increase x_i.

  If f increases as x_i increases, the gradient is positive and we need to decrease x_i.

  If f decreases as x_i increases, the gradient is negative and we need to increase x_i.
- By what amount should we change x_i?

  The gradient tells us how fast f changes as we change x_i. We should change x_i in proportion to the gradient.
- In summary, we will change x_i in proportion to the negative of the gradient of f at the current point.

The back-propagation learning algorithm:

1. Initialize weights and thresholds to small random values.

   $w1_{ij} = \text{random}(-0.5, 0.5), i = 0, \ldots, A; j = 1, \ldots, B.$

   $w2_{jk} = \text{random}(-0.5, 0.5), j = 0, \ldots, B; k = 1, \ldots, C.$

2. Choose an input/output pair $(\bar{x}, \bar{y})$ from the training set where $\bar{x} = (x_1, \ldots, x_A)$ and $\bar{y} = (y_1, \ldots, y_C)$.

   Assign values to input units.

3. Determine the values of the hidden units.

$$h_j = f\left(\sum_{i=0}^{A} w1_{ij} \cdot x_i\right), j = 1, \ldots, B. \tag{4}$$

4. Determine the values of the output units.

$$o_k = f\left(\sum_{j=0}^{B} w2_{jk} \cdot h_j\right), k = 1, \ldots, C. \tag{5}$$

5. Determine how to adjust weights between hidden and output layer to reduce error for this training example. (Calculate the gradients with respect to the weights between the hidden and the output layers.)

$$\frac{\partial}{\partial w2_{jk}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 = -(y_k - o_k)\, o_k(1 - o_k)\, h_j$$

6. Determine how to adjust weights between input and hidden layer to reduce error for this training example.

$$\frac{\partial}{\partial w1_{ij}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 = -h_j(1-h_j)x_i \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'}$$

7. Adjust weights between hidden and output layer where α is the learning rate.

$$w2_{jk} \leftarrow w2_{jk} - \alpha \left(\frac{\partial}{\partial w2_{jk}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 \right) \tag{6}$$

$$w2_{jk} \leftarrow w2_{jk} - \alpha(-(y_k - o_k) o_k (1 - o_k) h_j) \tag{7}$$

$$w2_{jk} \leftarrow w2_{jk} + \alpha(y_k - o_k) o_k (1 - o_k) h_j \tag{8}$$

8. Adjust weights between input and hidden layer.

$$w1_{ij} \leftarrow w1_{ij} - \alpha \left(\frac{\partial}{\partial w1_{ij}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 \right) \tag{9}$$

$$w1_{ij} \leftarrow w1_{ij} - \alpha \left(-h_j(1-h_j)x_i \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \right) \tag{10}$$

$$w1_{ij} \leftarrow w1_{ij} + \alpha \left(h_j(1-h_j)x_i \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \right) \tag{11}$$

9. If the stopping criteria is not met, go to step 2 and repeat.

   Possible stop criteria:

   - Max number of epochs
     epoch = one time through the training set
   - Error is acceptably small.

How should we choose the learning rate?

- If the learning rate is small: move slowly, takes a long time, won't miss local minimum.
- If the learning rate is large: miss local minimum, learn quickly.
- A good range (0.05 to 0.35)
- Usually start out big and reduce value gradually.

The derivative of the sigmoid function:

$$\begin{aligned} f(x) &= \frac{p(x)}{q(x)} && (12) \\ f'(x) &= \frac{p'(x)q(x) - p(x)q'(x)}{q(x)^2} && (13) \end{aligned}$$

$$\begin{aligned} f(x) &= \frac{1}{1+e^{-x}} && (14) \\ f'(x) &= \frac{0 \cdot (1+e^{-x}) - (-e^{-x})}{(1+e^{-x})^2} && (15) \\ &= \frac{e^{-x}}{(1+e^{-x})^2} && (16) \\ &= \frac{1}{1+e^{-x}} \frac{e^{-x}}{1+e^{-x}} && (17) \\ &= \frac{1}{1+e^{-x}} \left(1 - \frac{1}{1+e^{-x}}\right) && (18) \\ &= f(x)(1-f(x)) && (19) \end{aligned}$$

The gradient for $w2_{jk}$:

$$\frac{\partial}{\partial w2_{jk}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 \tag{20}$$

$$= \frac{\partial}{\partial w2_{jk}} \frac{1}{2} (y_k - o_k)^2 \tag{21}$$

$$= -(y_k - o_k) \frac{\partial}{\partial w2_{jk}} o_k \tag{22}$$

$$= -(y_k - o_k) \frac{\partial}{\partial w2_{jk}} f \left(\sum_{j'=0}^{B} w2_{j'k} \cdot h_{j'} \right) \tag{23}$$

$$= -(y_k - o_k) f' \left(\sum_{j'=0}^{B} w2_{j'k} \cdot h_{j'} \right) \frac{\partial}{\partial w2_{jk}} \left(\sum_{j'=0}^{B} w2_{j'k} \cdot h_{j'} \right) \tag{24}$$

$$= -(y_k - o_k) o_k (1 - o_k) h_j \tag{25}$$

The gradient for $w2_{01}$:

$$\frac{\partial}{\partial w2_{01}} \frac{1}{2} (y - o_1)^2 \tag{26}$$

$$= -(y - o_1) \frac{\partial}{\partial w2_{01}} o_1 \tag{27}$$

$$= -(y - o_1) \frac{\partial}{\partial w2_{01}} f \left(\sum_{j'=0}^{B} w2_{j1} \cdot h_{j'} \right) \tag{28}$$

$$= -(y - o_1) f' \left(\sum_{j'=0}^{B} w2_{j1} \cdot h_{j'} \right) \frac{\partial}{\partial w2_{01}} \left(\sum_{j'=0}^{B} w2_{j1} \cdot h_{j'} \right) \tag{29}$$

$$= -(y - o_1) f'(o_1) \frac{\partial}{\partial w2_{01}} \left(\sum_{j'=0}^{B} w2_{j1} \cdot h_{j'} \right) \tag{30}$$

$$= -(y - o_1)\, o_1 (1 - o_1) \tag{31}$$

The gradient for $w1_{ij}$:

$$\frac{\partial}{\partial w1_{ij}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 \tag{32}$$

$$= \sum_{k'=1}^{C} \frac{\partial}{\partial w1_{ij}} \frac{1}{2} (y_{k'} - o_{k'})^2 \tag{33}$$

$$= - \sum_{k'=1}^{C} (y_{k'} - o_{k'}) \frac{\partial}{\partial w1_{ij}} o_{k'} \tag{34}$$

$$= - \sum_{k'=1}^{C} (y_{k'} - o_{k'}) \frac{\partial}{\partial w1_{ij}} f \left(\sum_{j'=0}^{B} w2_{j'k} \cdot h_{j'} \right) \tag{35}$$

$$= - \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) \frac{\partial}{\partial w1_{ij}} \left(\sum_{j'=0}^{B} w2_{j'k'} \cdot h_{j'} \right) \tag{36}$$

$$= - \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \frac{\partial}{\partial w1_{ij}} h_j \tag{37}$$

$$= - \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \frac{\partial}{\partial w1_{ij}} f \left(\sum_{i'=0}^{A} w1_{i'j} \cdot x_{i'} \right) \tag{38}$$

$$= - \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} h_j (1 - h_j) x_i \tag{39}$$

$$= -h_j (1 - h_j) x_i \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \tag{40}$$

The gradient for $w1_{0j}$:

$$\frac{\partial}{\partial w1_{0j}} \frac{1}{2} \sum_{k'=1}^{C} (y_{k'} - o_{k'})^2 \tag{41}$$

$$= \sum_{k'=1}^{C} \frac{\partial}{\partial w1_{0j}} \frac{1}{2} (y_{k'} - o_{k'})^2 \tag{42}$$

$$= -\sum_{k'=1}^{C} (y_{k'} - o_{k'}) \frac{\partial}{\partial w1_{0j}} o_{k'} \tag{43}$$

$$= -\sum_{k'=1}^{C} (y_{k'} - o_{k'}) \frac{\partial}{\partial w1_{0j}} f\left(\sum_{j'=0}^{B} w2_{j'k} \cdot h_{j'} \right) \tag{44}$$

$$= -\sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) \frac{\partial}{\partial w1_{0j}} \left(\sum_{j'=0}^{B} w2_{j'k'} \cdot h_{j'} \right) \tag{45}$$

$$= -\sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \frac{\partial}{\partial w1_{0j}} h_j \tag{46}$$

$$= -\sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \frac{\partial}{\partial w1_{0j}} f\left(\sum_{i'=0}^{A} w1_{i'j} \cdot x_{i'} \right) \tag{47}$$

$$= -\sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} h_j (1 - h_j) \tag{48}$$

$$= -h_j (1 - h_j) \sum_{k'=1}^{C} (y_{k'} - o_{k'}) o_{k'} (1 - o_{k'}) w2_{jk'} \tag{49}$$