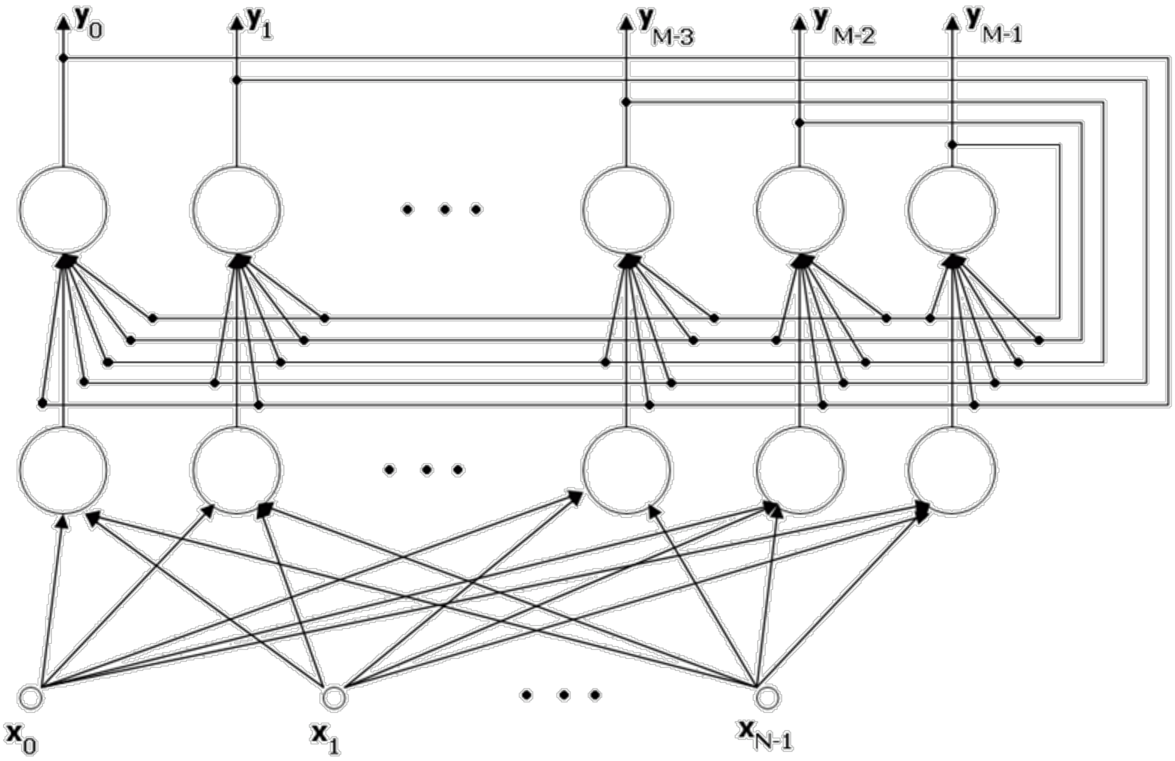
What kind of signals does the Hamming Network process?

Although the network works in an analog way, it processes binary signals – on the other hand, those signals can be "noisy", and have continuous values, not only zeros and ones.

What does the Hamming Network look like?



In the picture presented above we can see the Hamming Network. It can be divided into two basic sections:

- **input layer** – a layer built with neurons, all of those neurons are connected to all of the network inputs;
- **output layer** – which is called MaxNet layer; the output of each neuron of this layer is connected to input of each neuron of this layer, besides every neuron of this layer is connected to exactly one neuron of the input layer (as in the picture above).

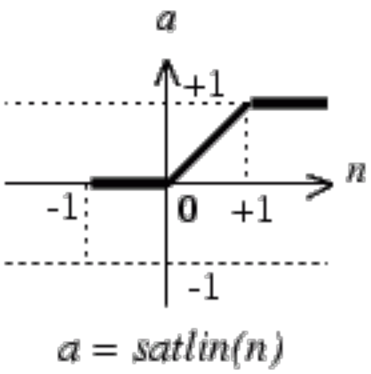
It's easy to see, that both layers have the same number of neurons.

How does the Hamming Network work?

Input layer neurons are programmed to identify a fixed number of patterns; the number of neurons in this layer matches the number of those patterns (M neurons – M patterns). Outputs of these neurons realise the function, which "measures" the similarity of an input signal to a given pattern.

The output layer is responsible for choosing the pattern, which is the most similar to testing signal. In this layer, the neuron with the strongest response stops the other neurons responses (it usually happens after few calculating cycles, and there must be a "0" on x(i) inputs during these cycles).

There should be the "1 of M" code used on the output, which points at the network answer (1 of M patterns is recognized). To achieve this, proper transfer function has to be used – in our case the best function was



How are the connections weights created?

Weights of the input layer neurons

Those weights are set to assure that the maximal responses of all neurons are equal, and that exactly one pattern causes specific neuron to response. There is no need to teach network to achieve that, it's sufficient to mathematically set weights.

$$w_{ij} = x_i^j \ ,$$

$$0 \leq i \leq N-1, \quad 0 \leq j \leq M-1$$

where:

- **w(i,j)** is the weight of the connection between neuron "j" and the input "i"
- **x(i,j)** is the value of signal "i" in pattern "j"

This equation becomes obvious, when we recall that the product of weights and impulses of a neuron can be interpreted as the cosinus of the angle between the weight vector and the impulse vector. When these vectors are equal, the neuron output will be "1", when vectors are different, the output value range will be from –1 to 1.

Input weights of the output layer neurons

At first, we need to determine the number of the output layer neurons inputs.

Each of those neurons is connected:

- to itself – weight=1
- to all of neurons of the output layer – weight of each connection is -1/M
- to appropriate neuron of the input layer – weight=1