WIKIPEDIA

Artificial neuron

An **artificial neuron** is a mathematical function conceived as a model of biological neurons, a neural network. Artificial neurons are elementary units in an artificial neural network.[1] The artificial neuron receives one or more inputs (representing excitatory postsynaptic potentials and inhibitory postsynaptic potentials at neural dendrites) and sums them to produce an output (or activation, representing a neuron's action potential which is transmitted along its axon). Usually each input is separately weighted, and the sum is passed through a non-linear function known as an activation function or transfer function. The transfer functions usually have a sigmoid shape, but they may also take the form of other non-linear functions, piecewise linear functions, or step functions. They are also often monotonically increasing, continuous, differentiable and bounded. Non-monotonic, unbounded and oscillating activation functions with multiple zeros that outperform sigmoidal and ReLU like activation functions on many tasks have also been recently explored. The thresholding function has inspired building logic gates referred to as threshold logic; applicable to building logic circuits resembling brain processing. For example, new devices such as memristors have been extensively used to develop such logic in recent times.[2]

The artificial neuron transfer function should not be confused with a linear system's transfer function.

Contents

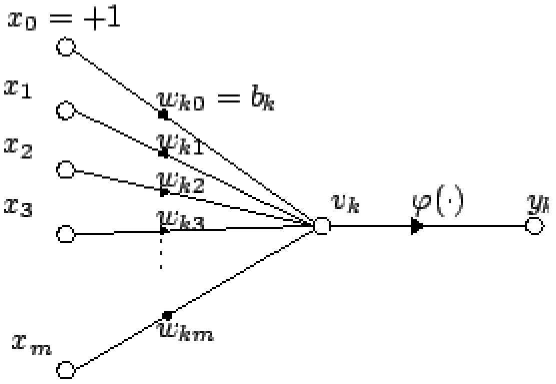
Basic structure

For a given artificial neuron k, let there be $m + 1$ inputs with signals x_0 through x_m and weights w_{k0} through w_{km}. Usually, the x_0 input is assigned the value +1, which makes it a *bias* input with $w_{k0} = b_k$. This leaves only m actual inputs to the neuron: from x_1 to x_m.

The output of the *k*th neuron is:

$$y_k = \varphi\left(\sum_{j=0}^{m} w_{kj} x_j\right)$$

Where φ (phi) is the transfer function (commonly a threshold function).



The output is analogous to the axon of a biological neuron, and its value propagates to the input of the next layer, through a synapse. It may also exit the system, possibly as part of an output vector.

It has no learning process as such. Its transfer function weights are calculated and threshold value are predetermined.

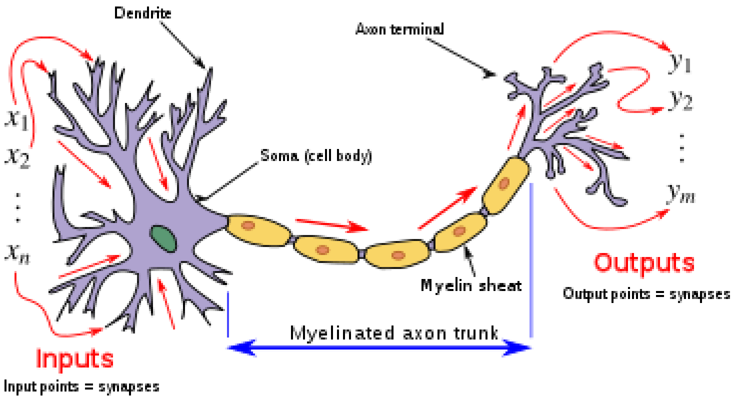
Types

Depending on the specific model used they may be called a **semi-linear unit**, **Nv neuron**, **binary neuron**, **linear threshold function**, or **McCulloch–Pitts** (**MCP**) **neuron**.

Simple artificial neurons, such as the McCulloch–Pitts model, are sometimes described as "caricature models", since they are intended to reflect one or more neurophysiological observations, but without regard to realism.[3]

Biological models

Artificial neurons are designed to mimic aspects of their biological counterparts. However a significant performance gap exists between biological and artificial neural networks. In particular single biological neurons in the human brain with oscillating activation function capable of learning the XOR function have been discovered.[4] However single artificial neurons with popular sigmoidal and ReLU like activation functions cannot learn the XOR function.[5]



Neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals

- *Dendrites* – In a biological neuron, the dendrites act as the input vector. These dendrites allow the cell to receive signals from a large (>1000) number of neighboring neurons. As in the above mathematical treatment, each dendrite is able to perform "multiplication" by that dendrite's "weight value." The multiplication is accomplished by increasing or decreasing the ratio of synaptic neurotransmitters to signal chemicals introduced into the dendrite in response to the synaptic neurotransmitter. A negative multiplication effect can be achieved by transmitting signal inhibitors (i.e. oppositely charged ions) along the dendrite in response to the reception of synaptic neurotransmitters.
- *Soma* – In a biological neuron, the soma acts as the summation function, seen in the above mathematical description. As positive and negative signals (exciting and inhibiting, respectively) arrive in the soma from the dendrites, the positive and negative ions are effectively added in summation, by simple virtue of being mixed together in the solution inside the cell's body.
- *Axon* – The axon gets its signal from the summation behavior which occurs inside the soma. The opening to the axon essentially samples the electrical potential of the solution inside the soma. Once the soma reaches a certain potential, the axon will transmit an all-in signal pulse down its length. In this regard, the axon behaves as the ability for us to connect our artificial neuron to other artificial neurons.

Unlike most artificial neurons, however, biological neurons fire in discrete pulses. Each time the electrical potential inside the soma reaches a certain threshold, a pulse is transmitted down the axon. This pulsing can be translated into continuous values. The rate (activations per second, etc.) at which an axon fires converts directly into the rate at which neighboring cells get signal ions introduced into them. The faster a biological neuron fires, the faster nearby neurons accumulate electrical potential (or lose electrical potential, depending on the "weighting" of the dendrite that connects to the neuron that fired). It is this conversion that allows computer scientists and mathematicians to simulate biological neural networks using artificial neurons which can output distinct values (often from −1 to 1).

Encoding

Research has shown that unary coding is used in the neural circuits responsible for birdsong production.[6][7] The use of unary in biological networks is presumably due to the inherent simplicity of the coding. Another contributing factor could be that unary coding provides a certain degree of error correction.[8]

History

The first artificial neuron was the Threshold Logic Unit (TLU), or Linear Threshold Unit,[9] first proposed by Warren McCulloch and Walter Pitts in 1943. The model was specifically targeted as a computational model of the "nerve net" in the brain.[10] As a transfer function, it employed a threshold, equivalent to using the Heaviside step function. Initially, only a simple model was considered, with binary inputs and outputs, some restrictions on the possible weights, and a more flexible threshold value. Since the beginning it was already noticed that any boolean function could be implemented by networks of such devices, what is easily seen from the fact that one can implement the AND and OR functions, and use them in the disjunctive or the conjunctive normal form. Researchers also soon realized that cyclic networks, with feedbacks through neurons, could define dynamical systems with memory, but most of the research concentrated (and still does) on strictly feed-forward networks because of the smaller difficulty they present.

One important and pioneering artificial neural network that used the linear threshold function was the perceptron, developed by Frank Rosenblatt. This model already considered more flexible weight values in the neurons, and was used in machines with adaptive capabilities. The representation of the threshold values as a bias term was introduced by Bernard Widrow in 1960 – see ADALINE.

In the late 1980s, when research on neural networks regained strength, neurons with more continuous shapes started to be considered. The possibility of differentiating the activation function allows the direct use of the gradient descent and other optimization algorithms for the adjustment of the weights. Neural networks also started to be used as a general function approximation model. The best known training algorithm called backpropagation has been rediscovered several times but its first development goes back to the work of Paul Werbos.[11][12]

Types of transfer functions

The transfer function (activation function) of a neuron is chosen to have a number of properties which either enhance or simplify the network containing the neuron. Crucially, for instance, any multilayer perceptron using a *linear* transfer function has an equivalent single-layer network; a non-linear function is therefore necessary to gain the advantages of a multi-layer network.

Below, u refers in all cases to the weighted sum of all the inputs to the neuron, i.e. for n inputs,

$$u = \sum_{i=1}^{n} w_i x_i$$

where **w** is a vector of *synaptic weights* and **x** is a vector of inputs.

Step function

The output y of this transfer function is binary, depending on whether the input meets a specified threshold, θ. The "signal" is sent, i.e. the output is set to one, if the activation meets the threshold.

$$y = \begin{cases} 1 & \text{if } u \geq \theta \\ 0 & \text{if } u < \theta \end{cases}$$

This function is used in perceptrons and often shows up in many other models. It performs a division of the space of inputs by a hyperplane. It is specially useful in the last layer of a network intended to perform binary classification of the inputs. It can be approximated from other sigmoidal functions by assigning large values to the weights.

Linear combination

In this case, the output unit is simply the weighted sum of its inputs plus a *bias* term. A number of such linear neurons perform a linear transformation of the input vector. This is usually more useful in the first layers of a network. A number of analysis tools exist based on linear models, such as harmonic analysis, and they can all be used in neural networks with this linear neuron. The bias term allows us to make affine transformations to the data.

See: Linear transformation, Harmonic analysis, Linear filter, Wavelet, Principal component analysis, Independent component analysis, Deconvolution.

Sigmoid

A fairly simple non-linear function, the sigmoid function such as the logistic function also has an easily calculated derivative, which can be important when calculating the weight updates in the network. It thus makes the network more easily manipulable mathematically, and was attractive to early computer scientists who needed to minimize the computational load of their simulations. It was previously commonly seen in multilayer perceptrons. However, recent work has shown sigmoid neurons to be less effective than rectified linear neurons. The reason is that the gradients computed by the backpropagation algorithm tend to diminish towards zero as activations propagate through layers of sigmoidal neurons, making it difficult to optimize neural networks using multiple layers of sigmoidal neurons.

Rectifier

In the context of artificial neural networks, the **rectifier** is an activation function defined as the positive part of its argument:

$$f(x) = x^+ = \max(0, x),$$

where *x* is the input to a neuron. This is also known as a ramp function and is analogous to half-wave rectification in electrical engineering. This activation function was first introduced to a dynamical network by Hahnloser et al. in a 2000 paper in Nature[13] with strong biological motivations and mathematical justifications.[14] It has been demonstrated for the first time in 2011 to enable better training of deeper networks,[15] compared to the widely used activation functions prior to 2011, i.e., the logistic sigmoid (which is inspired by probability theory; see logistic regression) and its more practical[16] counterpart, the hyperbolic tangent.

Pseudocode algorithm

The following is a simple pseudocode implementation of a single TLU which takes boolean inputs (true or false), and returns a single boolean output when activated. An object-oriented model is used. No method of training is defined, since several exist. If a purely functional model were used, the class TLU below would be replaced with a function TLU with input parameters threshold, weights, and inputs that returned a boolean value.

```
class TLU defined as:
    data member threshold : number
    data member weights : list of numbers of size X

    function member fire(inputs : list of booleans of size X) : boolean defined as:
        variable T : number
        T ← 0
        for each i in 1 to X do
            if inputs(i) is true then
                T ← T + weights(i)
            end if
        end for each
        if T > threshold then
            return true
        else:
            return false
        end if
    end function
end class
```

See also

- Binding neuron
- Connectionism

References

1. "Neuromorphic Circuits With Neural Modulation Enhancing the Information Content of Neural Signaling | International Conference on Neuromorphic Systems 2020". doi:10.1145/3407197.3407204 (https://doi.org/10.1145%2F3407197.3407204). S2CID 220794387 (https://api.semanticscholar.org/CorpusID:220794387).
2. Maan, A. K.; Jayadevi, D. A.; James, A. P. (1 January 2016). "A Survey of Memristive Threshold Logic Circuits". *IEEE Transactions on Neural Networks and Learning Systems*. **PP** (99): 1734–1746. arXiv:1604.07121 (https://arxiv.org/abs/1604.07121). Bibcode:2016arXiv160407121M (https://ui.adsabs.harvard.edu/abs/2016arXiv160407121M). doi:10.1109/TNNLS.2016.2547842 (https://doi.org/10.1109%2FTNNLS.2016.2547842). ISSN 2162-237X (https://www.worldcat.org/issn/2162-237X). PMID 27164608 (https://pubmed.ncbi.nlm.nih.gov/27164608). S2CID 1798273 (https://api.semanticscholar.org/CorpusID:1798273).
3. F. C. Hoppensteadt and E. M. Izhikevich (1997). *Weakly connected neural networks*. Springer. p. 4. ISBN 978-0-387-94948-2.
4. Gidon, Albert; Zolnik, Timothy Adam; Fidzinski, Pawel; Bolduan, Felix; Papoutsi, Athanasia; Poirazi, Panayiota; Holtkamp, Martin; Vida, Imre; Larkum, Matthew Evan (2020-01-03). "Dendritic action potentials and computation in human layer 2/3 cortical neurons" (https://www.science.org/doi/10.1126/science.aax6239). *Science*. **367** (6473): 83–87. Bibcode:2020Sci...367...83G (https://ui.adsabs.harvard.edu/abs/2020Sci...367...83G). doi:10.1126/science.aax6239 (https://doi.org/10.1126%2Fscience.aax6239). PMID 31896716 (https://pubmed.ncbi.nlm.nih.gov/31896716). S2CID 209676937 (https://api.semanticscholar.org/CorpusID:209676937).
5. Noel, Mathew Mithra; L, Arunkumar; Trivedi, Advait; Dutta, Praneet (2021-09-04). "Growing Cosine Unit: A Novel Oscillatory Activation Function That Can Speedup Training and Reduce Parameters in Convolutional Neural Networks". arXiv:2108.12943 (https://arxiv.org/abs/2108.12943) [cs.LG (https://arxiv.org/archive/cs.LG)].
6. Squire, L.; Albright, T.; Bloom, F.; Gage, F.; Spitzer, N., eds. (October 2007). *Neural network models of birdsong production, learning, and coding* (https://web.archive.org/web/20150412190625/https://clm.utexas.edu/fietelab/Papers/birdsong_review_topost.pdf) (PDF). New Encyclopedia of Neuroscience: Elservier. Archived from the original (https://clm.utexas.edu/fietelab/Papers/birdsong_review_topost.pdf) (PDF) on 2015-04-12. Retrieved 12 April 2015.
7. Moore, J.M.; et al. (2011). "Motor pathway convergence predicts syllable repertoire size in oscine birds" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3182746). *Proc. Natl. Acad. Sci. USA*. **108** (39): 16440–16445. Bibcode:2011PNAS..10816440M (https://ui.adsabs.harvard.edu/abs/2011PNAS..10816440M). doi:10.1073/pnas.1102077108 (https://doi.org/10.1073%2Fpnas.1102077108). PMC 3182746 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3182746). PMID 21918109 (https://pubmed.ncbi.nlm.nih.gov/21918109).
8. Potluri, Pushpa Sree (26 November 2014). "Error Correction Capacity of Unary Coding". arXiv:1411.7406 (https://arxiv.org/abs/1411.7406) [cs.IT (https://arxiv.org/archive/cs.IT)].
9. Martin Anthony (January 2001). *Discrete Mathematics of Neural Networks: Selected Topics* (https://books.google.com/books?id=qOy4yLBqhFcC&pg=PA3). SIAM. pp. 3–. ISBN 978-0-89871-480-7.

10. Charu C. Aggarwal (25 July 2014). *Data Classification: Algorithms and Applications* (https://books.google.com/books?id=gJhBBAAAQBAJ&pg=PA209). CRC Press. pp. 209–. ISBN 978-1-4665-8674-1.
11. Paul Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University, 1974
12. Werbos, P.J. (1990). "Backpropagation through time: what it does and how to do it" (https://zenodo.org/record/1262035). *Proceedings of the IEEE*. **78** (10): 1550–1560. doi:10.1109/5.58337 (https://doi.org/10.1109%2F5.58337). ISSN 0018-9219 (https://www.worldcat.org/issn/0018-9219).
13. Hahnloser, Richard H. R.; Sarpeshkar, Rahul; Mahowald, Misha A.; Douglas, Rodney J.; Seung, H. Sebastian (2000). "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit". *Nature*. **405** (6789): 947–951. Bibcode:2000Natur.405..947H (https://ui.adsabs.harvard.edu/abs/2000Natur.405..947H). doi:10.1038/35016072 (https://doi.org/10.1038%2F35016072). ISSN 0028-0836 (https://www.worldcat.org/issn/0028-0836). PMID 10879535 (https://pubmed.ncbi.nlm.nih.gov/10879535). S2CID 4399014 (https://api.semanticscholar.org/CorpusID:4399014).
14. R Hahnloser, H.S. Seung (2001). *Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks*. NIPS 2001.
15. Xavier Glorot, Antoine Bordes and Yoshua Bengio (2011). *Deep sparse rectifier neural networks* (http://jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf) (PDF). AISTATS.
16. Yann LeCun, Leon Bottou, Genevieve B. Orr and Klaus-Robert Müller (1998). "Efficient BackProp" (http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf) (PDF). In G. Orr; K. Müller (eds.). *Neural Networks: Tricks of the Trade*. Springer.

Further reading

- McCulloch, Warren S.; Pitts, Walter (1943). "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics*. **5** (4): 115–133. doi:10.1007/bf02478259 (https://doi.org/10.1007%2Fbf02478259).
- Samardak, A.; Nogaret, A.; Janson, N. B.; Balanov, A. G.; Farrer, I.; Ritchie, D. A. (2009-06-05). "Noise-Controlled Signal Transmission in a Multithread Semiconductor Neuron" (https://dspace.lboro.ac.uk/2134/12736). *Physical Review Letters*. **102** (22): 226802. Bibcode:2009PhRvL.102v6802S (https://ui.adsabs.harvard.edu/abs/2009PhRvL.102v6802S). doi:10.1103/physrevlett.102.226802 (https://doi.org/10.1103%2Fphysrevlett.102.226802). PMID 19658886 (https://pubmed.ncbi.nlm.nih.gov/19658886).

External links

- Artifical [*sic*] neuron mimicks function of human cells (https://www.youtube.com/watch?v=NhTZnnJJP64)
- McCulloch-Pitts Neurons (Overview) (http://www.mind.ilstu.edu/curriculum/modOverview.php?modGUI=212)