Artificial Neural Networks

Historical description

Victor G. Lopez
Artificial Neural Networks (ANN)

- An artificial neural network is a computational model that attempts to emulate the functions of the brain.
Characteristics of ANNs

- Modern ANNs are complex arrangements of processing units able to adapt their parameters using learning techniques.

- Their plasticity, nonlinearity, robustness and highly distributed framework have attracted a lot of attention from many areas of research.

- Several applications have been studied using ANNs: classification, pattern recognition, clustering, function approximation, optimization, forecasting and prediction, among others.

- To date, ANN models are the artificial intelligence methods that imitate human intelligence more closely.
1890s: A neuron model

- Santiago Ramón y Cajal proposes the brain works in a parallel and distributed manner, with neurons as basic processing units.

- He described the first complete biological model of the neuron.
Neural synapses

- A neural synapse is the region where the axon of a neuron interacts with another neuron.

- A neuron usually receives information by means of its dendrites, but this is not always the case.

- Neurons share information using electrochemical signals.
Action Potential

- The signal sent by a single neuron is usually weak, but a neuron receives many inputs from many other neurons.

- The inputs from all the neurons are integrated. If a threshold is reached, the neuron sends a powerful signal through its axon, called an action potential.
Neural Pathways

- The action potential is an all-or-none signal. It doesn’t matter if the threshold is barely reached or vastly surpassed, the resulting action potential is the same.

- This means that the action potential alone does not carry much information. All cerebral processes, like memory or learning, depend on neural pathways.

- There are over $10^{11}$ neurons in the human brain, forming around $10^{15}$ synapses. They form the basis of human intelligence and consciousness.
1943: McCulloch and Pitts

- Warren McCulloch (neurophysiologist) and Walter Pitts (matematician) wrote a paper describing a logical calculus of neural networks.

- Their model can, in principle, approximate any computable function.

- This is considered the birth of artificial intelligence.

_Bulletin of Mathematical Biophysics_  
_Volume 5, 1943_

_A Logical Calculus of the Ideas Immanent in Nervous Activity_  
_Warren S. McCulloch and Walter Pitts_  

_Because of the “all-or-none” character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions there can be constructed a net which it describes._
1958: The Perceptron

- Frank Rosenblatt (psychologist) proposes the Perceptron with a novel method of supervised learning.

- This is the oldest neural network still in use today.
Single-neuron Perceptron

- Here, the activation function $f$ was selected as a saturation function. This simulates the all-or-none property of the action potential.

- The single-neuron Perceptron can solve classification problems of two linearly separable groups.
Using several neurons, the Perceptron can classify objects into many categories, as long as they are linearly separable.

The number of total categories is $2^S$, with $S$ the number of neurons.
Training algorithm per neuron

1. Initialize the weights $W_0$.

2. Compute the output of the network for input $p_k$. If the output is correct, set
   \[ W_{k+1} = W_k \]

3. If the output is incorrect, set
   \[ W_{k+1} = W_k - \eta p_k, \quad \text{if } W_k^T p_k \geq 0 \]
   \[ W_{k+1} = W_k + \eta p_k, \quad \text{if } W_k^T p_k < 0 \]

- Here, $0 < \eta \leq 1$ is the learning rate.
Logical gates AND and OR

- Separation of the outputs of the logical gates AND and OR are simple examples of problems solvable by the single-layer Perceptron.

- In contrast, the outputs of the XOR gate are not linearly separable.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**AND**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**OR**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**XOR**
1959: ADALINE and MADALINE

- Bernard Widrow and Marcian Hoff developed models called ADALINE (adaptive linear elements) and MADALINE (Multiple ADALINE).

- The main difference with respect to the Perceptron is the absence of the threshold activation function.

- Training of these networks is performed using derivatives.
1970s: First Winter in ANNs research

- After the successful introduction and development of ANNs during the 1960s, interest in their applications decayed for almost two decades.

- The limitations of the single-layer Perceptron narrowed its possible practical implementations.

- Theoretical research showed that a multilayer Perceptron would drastically improve its performance, but there was no training algorithm for it.
1986: Multilayer Perceptron

\[ a^1 = f^1 (W^1 p + b^1) \]
\[ a^2 = f^2 (W^2 a^1 + b^2) \]
\[ a^3 = f^3 (W^3 f^2 (W^2 f^1 (W^1 p + b^1) + b^2) + b^2) + b^3 \]
In his 1974 PhD thesis, Paul Werbos proposed to use the backpropagation algorithm as a solution to the multilayer Perceptron training problem. His suggestion, however, remained ignored for more than a decade.

In 1986, the backpropagation method is finally popularized in a paper by Rumelhart, Hinton and Williams.

The multilayer Perceptron became the most powerful ANN model to date.

It is proven to solve nonlinearly separable classification problems, it can approximate any continuous function, it generalizes from particular samples, among many other applications.
Backpropagation Training

- This is a supervised learning. Then, we have a list of inputs and their corresponding target outputs, \((p_k, t_k)\).

- We can compute the output of the NN for each given input \(p_k\). This is called the forward propagation step. For a 3 layer network, this would be

\[
a_k = f^3(W^3 f^2(W^2 f^1(W^1 p_k + b^1) + b^2) + b^3)
\]

- Define the output error as \(e_k = t_k - a_k\).

- Now, it is our interest to minimize the squared error

\[
J = \frac{1}{2} e_k^2 = \frac{1}{2} (t_k - a_k)^2
\]

or the average sum of the squared error

\[
J = \frac{1}{2Q} \sum_{k=1}^{Q} e_k^2 = \frac{1}{2Q} \sum_{k=1}^{Q} (t_k - a_k)^2
\]
Backpropagation Training

- Use a gradient descent algorithm to update the weights $W_k$ while minimizing the error $e_k$

\[
W_{k+1} = W_k + \Delta W_k
\]

\[
\Delta W_k = -\eta \frac{\partial J}{\partial W_k}
\]

- The chain rule for derivatives can be used to obtain a clearer expression

\[
\frac{\partial J}{\partial W_k} = \frac{\partial J}{\partial e_k} \frac{\partial e_k}{\partial a_k} \frac{\partial a_k}{\partial W_k}
\]

- From the previous definitions we note that

\[
\frac{\partial J}{\partial e_k} = e_k, \quad \frac{\partial e_k}{\partial a_k} = -1
\]

and $\frac{\partial a_k}{\partial W_k}$ depends on the activation functions $f^i$. Notice that all activation functions must be differentiable.
Backpropagation Algorithm

1. Initialize the weights $W_0$.

2. By forward propagation, get $a_k$.

3. Calculate the error $e_k = t_k - a_k$.

4. Update the neural weights as

$$W_{k+1} = W_k + \eta \frac{\partial a_k}{\partial W_k} e_k$$
## Activation functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Input/Output Relation</th>
<th>Icon</th>
<th>MATLAB Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard Limit</td>
<td>(a = 0 \quad n &lt; 0) (a = 1 \quad n \geq 0)</td>
<td>![Hard Lim Icon]</td>
<td>hardlim</td>
</tr>
<tr>
<td>Symmetrical Hard Limit</td>
<td>(a = -1 \quad n &lt; 0) (a = +1 \quad n \geq 0)</td>
<td>![Symmetrical Hard Lim Icon]</td>
<td>hardlims</td>
</tr>
<tr>
<td>Linear</td>
<td>(a = n)</td>
<td>![Linear Icon]</td>
<td>purelin</td>
</tr>
<tr>
<td>Saturating Linear</td>
<td>(a = 0 \quad n &lt; 0) (a = n \quad 0 \leq n \leq 1) (a = 1 \quad n &gt; 1)</td>
<td>![Saturating Linear Icon]</td>
<td>satlin</td>
</tr>
<tr>
<td>Symmetric Saturating Linear</td>
<td>(a = -1 \quad n &lt; -1) (a = n \quad -1 \leq n \leq 1) (a = 1 \quad n &gt; 1)</td>
<td>![Symmetric Saturating Linear Icon]</td>
<td>satlin</td>
</tr>
<tr>
<td>Log-Sigmoid</td>
<td>(a = \frac{1}{1 + e^{-n}})</td>
<td>![Log-Sigmoid Icon]</td>
<td>logsig</td>
</tr>
<tr>
<td>Hyperbolic Tangent Sigmoid</td>
<td>(a = \frac{e^n - e^{-n}}{e^n + e^{-n}})</td>
<td>![Hyperbolic Tangent Sigmoid Icon]</td>
<td>tansig</td>
</tr>
<tr>
<td>Positive Linear</td>
<td>(a = 0 \quad n &lt; 0) (a = n \quad 0 \leq n)</td>
<td>![Positive Linear Icon]</td>
<td>poslin</td>
</tr>
<tr>
<td>Competitive</td>
<td>(a = 1 \quad ) neuron with max (n) (a = 0 \quad ) all other neurons</td>
<td>![Competitive Icon]</td>
<td>compet</td>
</tr>
</tbody>
</table>
In the 1990s, several applications of ANNs are studied and implemented. Areas as vision, pattern recognition, unsupervised learning and reinforcement learning take advantage of the ANNs adaptive characteristics.

Late in that decade, a new difficulty delays the advancement in the field. The basic backpropagation algorithm is not appropriate for several hidden layers, mainly because of limited computational capabilities.

Many researchers became pessimistic about ANNs.
In 2006, Hinton, Osindero and Teh published a fast learning algorithm for deep belief networks. This marks the dawn of deep learning.

The decade of 2010s has seen a boom in deep neural network applications. Companies as Microsoft, Google and Facebook have developed advanced deep learning ANNs. Optimism has returned to the field and human-level intelligence is expected to be achieved in a few decades.