Multilayer Feedforward Networks Are Universal Approximators

Hornik, Stinchcombe and White

Figures borrowed from Michael Nielsen’s online book

Presented by – Aman Gupta
Background:
Simple 2-layer perceptron cannot model XOR

- The decision boundary is linear.
- Can only represent or approximate functions in a narrow range.
But..

• In the period between 1985 and 1989
  • researchers explored ability of multilayer feedforward networks to approximate continuous functions.

• Empirical evidence suggested that neural networks could approximate quite well nearly any function

• This paper tries to answer two fundamental questions
  • Are the successes observed indicative of some deep and important approximation capability of these networks?
  • Is the class of functions approximated by networks limited? How does it compare to the class of functions represented by the simple 2-layer perceptron?
Summary of results

The paper rigorously establishes that:

• Standard multilayer feedforward network architectures using arbitrary squashing functions can
  • approximate virtually any function of interest to any desired degree of accuracy, provided sufficiently many hidden units are available.
• Results establish multilayer feedforward networks as a class of universal approximators.
• “Failures in applications can be attributed to inadequate learning, inadequate numbers of hidden units, or the presence of a stochastic rather than a deterministic relation between input and target. “
• Results do not address the issue of how many units are needed to attain a given degree of approximation.
Previous Work

• Research before 1989 showed
  • Adequate approximations to an unknown function using monotone squashing functions can be achieved using two hidden layers. (*le Cun (1987) and Lapedes and Farber (1988)*).

• Another work presented representation result (perfect approximation) using one hidden layer, but with a continuum of hidden units. (*Irie and Miyake (1988)*).
  • Of little practical utility despite theoretical soundness.
Gallant and White (1988) showed that a particular single hidden layer feedforward network using the monotone "cosine squasher" is

- Capable of embedding as a special case a network which yields a fourier series approximation to a given function as its output.
- Such networks possess all the approximation properties of Fourier series representations.
- Still, results do not justify arbitrary multilayer feedforward networks as universal approximators, but only a particular class of single hidden layer networks.
“A neural network can compute any function” can have many meanings. Some assumptions of the model are:

• $f(x)$ is the function to be computed within some desired accuracy $\epsilon > 0$. Using enough hidden neurons we can always find a neural network whose output $g(x)$ satisfies $|g(x) - f(x)| < \epsilon$, for all inputs $x$.

• The class of functions approximated are the continuous functions.

• There is no limit on the number of nodes in the hidden layer.
Main Results - Definitions

• Here, \( x \) corresponds to network input, \( w \) corresponds to network weights from input to the intermediate layer, and \( b \) corresponds to a bias.

• \( \sum r (G) \) is the class of output functions for single hidden layer feedforward networks with
  • squashing at the hidden layer
  • no squashing at the output layer.

• Scalars \( \beta_i \) correspond to network weights from hidden to output layers.

**Definition 2.1**
For any \( r \in N = \{1, 2, \ldots \} \), \( A' \) is the set of all affine functions from \( \mathbb{R}' \) to \( \mathbb{R} \), that is, the set of all functions of the form \( A(x) = w \cdot x + b \) where \( w \) and \( x \) are vectors in \( \mathbb{R}' \), \( \cdot \) denotes the usual dot product of vectors, and \( b \in \mathbb{R} \) is a scalar.

**Definition 2.2**
For any (Borel) measurable function \( G(\cdot) \) mapping \( \mathbb{R} \) to \( \mathbb{R} \) and \( r \in N \) let \( \Sigma'(G) \) be the class of functions
\[
\{ f: \mathbb{R}^r \to \mathbb{R} : f(x) = \sum q \beta_i G(A_i(x)), x \in \mathbb{R}'^r, \beta_i \in \mathbb{R}, A_i \in \mathbb{A}'^r, q = 1, 2, \ldots \}. \]
Main Results - Definitions

- $\Psi$ represents any general squashing function

- $\Sigma \Pi$ represents the class of multilayer feedforward networks

**Definition 2.3**
A function $\Psi : R \rightarrow [0,1]$ is a squashing function if it is non-decreasing, $\lim_{\lambda \to -\infty} \Psi(\lambda) = 1$, and $\lim_{\lambda \to +\infty} \Psi(\lambda) = 0$.

**Definition 2.4**
For any measurable function $G(\cdot)$ mapping $R$ to $R$ and $r \in N$, let $\Sigma \Pi^r(G)$ be the class of functions

\[
\{ f : R' \to R : f(x) = \sum_{j=1}^{q} \beta_j \cdot \prod_{k=1}^{l_j} G(A_{jk}(x)), x \in R', \beta_j \in R, A_{jk} \in A', l_j, r_j \in N, \}
\]
Main Results - Definitions

• In general, both $\Sigma \Pi$ and $\Sigma$ are continuous functions.

• The closeness of 2 functions can be described using various definitions.

Definition 2.5
Let $C'$ be the set of continuous functions from $R'$ to $R$, and let $M'$ be the set of all Borel measurable functions from $R'$ to $R$. We denote the Borel $\sigma$-field of $R'$ as $B'$.

The classes $\Sigma'(G)$ and $\Sigma \Pi'(G)$ belong to $M'$ for any Borel measurable $G$. When $G$ is continuous, $\Sigma'(G)$ and $\Sigma \Pi'(G)$ belong to $C'$. The class $C'$ is a subset of $M'$, which in fact contains virtually all functions relevant in applications. Functions that are not Borel measurable exist (e.g., Billingsley, 1979, pp. 36–37) but they are pathological. Our first results concern approximating functions in $C'$; we then extend these results to approximating functions in $M'$.

Closeness of functions $f$ and $g$ belonging to $C'$ or $M'$ is measured by a metric, $\rho$. Closeness of one class of functions to another class is described by the concept of denseness.
Main Results - Definitions

- An element of S can approximate an element of T to any desired degree of accuracy.

  **Definition 2.6**

  A subset $S$ of a metric space $(X, \rho)$ is $\rho$-dense in a subset $T$ if for every $\varepsilon > 0$ and for every $t \in T$ there is an $s \in S$ such that $\rho(s, t) < \varepsilon$.

- A function can be approximated to any arbitrary precision provided the approximation meets certain criteria.

  **Definition 2.7**

  A subset $S$ of $C'$ is said to be uniformly dense on compacta in $C'$ if for every compact subset $K \subset R'$ $S$ is $\rho_K$-dense in $C'$, where for $f, g \in C'$ $\rho_K(f, g) = \sup_{x \in K} |f(x) - g(x)|$. A sequence of functions $\{f_n\}$ converges to a function $f$ uniformly on compacta if for all compact $K \subset R'$ $\rho_K(f_n, f) \to 0$ as $n \to \infty$. □
Main Results - Theorems

- Feedforward networks are capable of arbitrarily accurate approximation to any real-valued continuous function over a compact set
  - given a continuous non-constant function $G$
  - $G$ is not required to be a squashing function

- Multilayer feedforward networks can approximate any measurable function arbitrarily well, regardless of the continuous non-constant function $G$ used.

---

**Theorem 2.1**

Let $G$ be any continuous nonconstant function from $\mathbb{R}$ to $\mathbb{R}$. Then $\Sigma \Pi'(G)$ is uniformly dense on compacta in $C'$.

---

**Theorem 2.2**

For every continuous nonconstant function $G$, every $r$, and every probability measure $\mu$ on $(R', B')$, $\Sigma \Pi'(G)$ is $\rho_\mu$-dense in $M'$. 
Main Results - Theorems

- **Single hidden layer** feedforward networks can approximate any measurable function arbitrarily well for any squashing function \( \Psi \).

- **Multilayer feedforward** networks can approximate any measurable function arbitrarily well for any squashing function \( \Psi \).

---

**Theorem 2.3**

For every squashing function \( \Psi \), every \( r \), and every probability measure \( \mu \) on \((R',B')\), \( \Sigma\Pi'(\Psi) \) is uniformly dense on compacta in \( C' \) and \( \rho_\mu \)-dense in \( M' \).  

---

**Theorem 2.4**

For every squashing function \( \Psi \), every \( r \), and every probability measure \( \mu \) on \((R',B')\), \( \Sigma'(\Psi) \) is uniformly dense on compacta in \( C' \) and \( \rho_\mu \)-dense in \( M' \).
Main Results – Important Corollaries

**Corollary 2.5**

For every Boolean function $g$ and every $\varepsilon > 0$ there is an $f$ in $\Sigma'(\Psi)$ such that $\max_{x \in \{0, 1\}^n} |g(x) - f(x)| < \varepsilon$. 

\[ \square \]
Universality with one input and one output

• Let’s attempt to understand how a neural network can approximate a function with one input and scalar output (problem of universality)

• Idea easily extensible to functions from $\mathbb{R}^n$ to $\mathbb{R}^m$ where $n \in \mathbb{N}$, $m \in \mathbb{N}$
Single neuron with sigmoid activation

- Consider output of only one hidden neuron.
- Output can be modelled as $\sigma(w . x + b)$.
Effect of bias term

Output slides left on increasing bias
Effect of bias term

Output slides right on decreasing bias
Effect of weight term

Decreasing weight flattens out the output
Increasing weight makes the output similar to a step function
Logistic function as step function

- Increasing \( w \) makes the curve steep.
- If \( w \) is a large number, function behaves as a step function.
- The quantity \( s = \frac{-b}{w} \) decides the location of the step along the x-axis.
Combining output of 2 neurons

- Increasing $w$ makes the curve steep.
- The quantity plotted in the graph is $w_1 a_1 + w_2 a_2$.
- The output varies if you vary $s_1$ and $s_2$. 
• The quantity plotted in the graph is $w_1 a_1 + w_2 a_2$.
• If $w_1$ and $w_2$ sum to zero, the output (before activation in the last layer) is a bump.
• The output varies if you vary $s_1$ and $s_2$. 
Combining 2 bumps using 4 neurons

- 2 bumps can be generated using 2 pairs of neurons
Combining many neurons to get many bumps

- Many bumps can be generated by using many different pairs of neurons.
- The height of each bump is generated using weights.
Back to approximating original function

- $\sum_j w_j a_j$ is the output from the hidden neurons.
- But, output from network is $\sigma(\sum_j w_j a_j + b)$ where $b$ is the bias on the output neuron.
- Solution is to design a neural network whose hidden layer has a weighted output given by $\sigma^{-1} \circ f(x)$

$$f(x) = 0.2 + 0.4x^2 + 0.3\sin(15x) + 0.05\cos(50x)$$
Conclusions

• Standard multilayer feedforward networks are capable of approximating any measurable function to any desired degree of accuracy.

• These “mapping networks” are universal approximators.

• “Any lack of success in applications must arise from inadequate learning, insufficient numbers of hidden units or the lack of a deterministic relationship between input and target.”
Limitations

The paper does not discuss the following:

• The rate of improvement of approximation with respect to increase in the number of hidden units when the dimension $r$ of the input space is held fixed.
• The rate at which the number of hidden units needed to attain a given accuracy of approximation must grow as the dimension $r$ of the input space increases.
THANK YOU!