Universal Approximation Theory

Based on

Approximation by Superpositions of a Sigmoidal Function

George Cybenko (1989)

Eslam Khalaf

July 10, 2026

Presentation Outline

Why Do We Need Universal Approximation?

Suppose we want a neural network to learn an unknown function.
Examples include:

- Image classification
- Speech recognition
- Predicting stock prices
- Autonomous driving
- Medical diagnosis

The natural question is:

Fundamental Question

Can a neural network represent **any** continuous function, or are there functions that it can never learn?

The Problem

Assume there exists an unknown function

$$f(x)$$

and we wish to approximate it using a neural network

$$G(x).$$

Instead of requiring

$$G(x) = f(x),$$

we only require

$$|f(x) - G(x)| < \varepsilon$$

for an arbitrarily small error ε.

George Cybenko's Contribution (1989)

In 1989, George Cybenko answered this question with a rigorous proof.

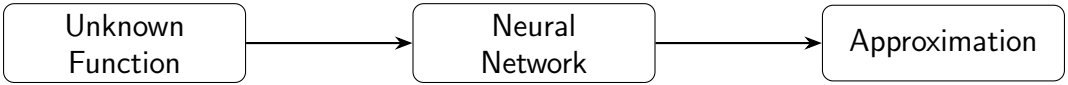
His paper showed that:

- A neural network with **one hidden layer** is sufficient.
- The activation function only needs to be continuous and sigmoidal.
- Such networks can approximate any continuous function on a compact domain.

This result became known as the

Universal Approximation Theorem.

The Big Picture



The entire paper asks one question:

Can this approximation be made arbitrarily accurate?

From Biological to Artificial Neurons

Artificial neural networks are inspired by the human brain.

Biological Neuron

- Receives signals
- Processes information
- Sends an output signal

(A typical biological neuron)

Artificial neurons imitate this behavior mathematically.

Artificial Neuron

Each neuron performs three simple steps:

1. Compute a weighted sum
2. Add a bias
3. Apply an activation function

Mathematically,

$$z = w^T x + b$$

followed by

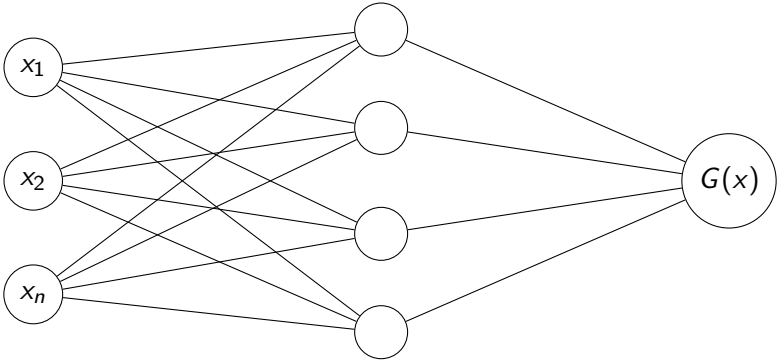
$$a = \sigma(z).$$

Here,

- x is the input,
- w is the weight vector,
- b is the bias

Single Hidden Layer Neural Network

Cybenko studies a network with one hidden layer.



Only one hidden layer is used throughout the paper.

Mathematical Model

The output of the network is

$$G(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^T x + b_i).$$

where

- N is the number of hidden neurons,
- w_i are the weights,
- b_i are the biases,
- α_i are the output weights,
- σ is the activation function.

This is the only model studied in Cybenko's paper.

The Sigmoidal Activation Function

A function is called **sigmoidal** if

$$\lim_{x \to -\infty} \sigma(x) = 0, \qquad \lim_{x \to +\infty} \sigma(x) = 1.$$

The most common example is the logistic function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Notice how the output smoothly transitions from 0 to 1.

What Does It Mean to Approximate a Function?

Suppose we have an unknown function

$$f(x).$$

Instead of finding an exact representation, we allow a small error.
We want to construct a neural network

$$G(x)$$

such that

$$|f(x) - G(x)| < \varepsilon,$$

where $\varepsilon > 0$ can be made arbitrarily small.

Key Idea

Approximation means getting as close as we want—not necessarily matching the function exactly.

Continuous Functions

Cybenko considers the space

$$C(I_n),$$

where

$$I_n = [0,1]^n$$

is the n-dimensional unit cube.

A continuous function has no jumps or breaks.

Examples include:

$$x^2, \qquad \sin x, \qquad e^x.$$

The theorem applies to every continuous function on this domain.

Measuring Approximation Error

To measure how close two functions are, we use the **supremum norm** (or infinity norm):

$$\|f - G\|_\infty = \max_{x \in I_n} |f(x) - G(x)|.$$

This measures the **largest error** anywhere in the domain.

Interpretation

If

$$\|f - G\|_\infty < \varepsilon,$$

then the neural network is never more than ε away from the true function.

The Goal of Universal Approximation

Let

$$S = \left\{ \sum_{i=1}^{N} \alpha_i \sigma(w_i^T x + b_i) \right\}$$

be the set of all single-hidden-layer neural networks.

The central question becomes:

$$\boxed{\overline{S} = C(I_n) \ ?}$$

where $\overline{S}$ denotes the closure of S.

In simple words:

Goal

Can neural networks approximate every continuous function, no matter how complicated it is?

The Main Result

After defining the approximation problem, Cybenko asks:

Main Question

Is the set of single-hidden-layer neural networks dense in the space of continuous functions?

The answer is given by the Universal Approximation Theorem.

Theorem 1: Universal Approximation Theorem

Theorem 1 (Cybenko, 1989)

Let σ be a continuous sigmoidal function.
Then finite sums of the form

$$G(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^T x + b_i)$$

are dense in

$$C(I_n).$$

What Does Theorem 1 Mean?

In simple words:

For any continuous function

$$f(x)$$

and any desired accuracy

$$\varepsilon > 0,$$

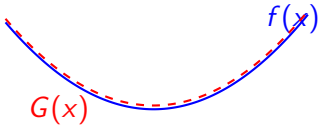
there exists a neural network $G(x)$ such that:

$$\|f - G\|_\infty < \varepsilon.$$

Meaning

A sufficiently large one-hidden-layer neural network can represent any continuous function.

Visual Intuition



A neural network can create increasingly accurate approximations by combining many simple functions.

How Can Simple Neurons Create Complex Functions?

Each neuron computes:

$$\sigma(w^T x + b)$$

which is a simple nonlinear function.

The network combines many neurons:

$$\alpha_1 \sigma(w_1^T x + b_1) + \alpha_2 \sigma(w_2^T x + b_2) + \cdots$$

Together, these simple components can form very complicated shapes.

Proof Idea: Contradiction

Cybenko proves the theorem by contradiction.

Assume:

$$\overline{S} \neq C(I_n)$$

meaning that some continuous functions cannot be approximated.

Then there exists a function separated from all neural networks.
This separation is obtained using the:

Hahn-Banach Theorem.

From Functions to Measures

The separating object can be represented as a measure:

$$L(f) = \int_{I_n} f(x) d\mu(x)$$

using the:

Riesz Representation Theorem.

The separation implies:

$$\int_{I_n} \sigma(w^T x + b) d\mu(x) = 0$$

for every possible neuron.

Where Theorem 2 Appears

The previous equation says:

$$\int \sigma(w^T x + b) d\mu(x) = 0$$

for every weight and bias.

In other words:

The measure μ is invisible to every neuron.

Theorem 2 proves this is impossible unless:

$$\mu = 0.$$

This contradiction proves Theorem 1.

Important Note

The Universal Approximation Theorem proves:

- A solution exists.
- One hidden layer is enough.
- Approximation can be arbitrarily accurate.

However, it does **not** prove:

- How to find the weights.
- How many neurons are needed.
- How fast training will be.

Why Do We Need Theorem 2?

Theorem 1 states:

$$\overline{S} = C(I_n)$$

meaning neural networks can approximate every continuous function.

However, the proof requires showing that sigmoid neurons are powerful enough to distinguish different functions.

This property is called:

Discriminatory Property

Theorem 2: Statement

Theorem 2 (Cybenko, 1989)

Let σ be a bounded measurable sigmoidal function.
Then the family

$$\sigma(w^T x + b)$$

is discriminatory.

This means sigmoid neurons can detect every non-zero difference between functions.

What Does Discriminatory Mean?

Suppose there exists a measure μ such that:

$$\int_{I_n} \sigma(w^T x + b) d\mu(x) = 0$$

for every possible neuron.

The discriminatory property says:

$$\boxed{\mu = 0}$$

must be true.

In simple words:

Intuition

If every sigmoid neuron sees nothing, then there is actually nothing to detect.

Why Are Sigmoid Functions Powerful?

A sigmoid can approximate a step function.

Consider:

$$\sigma(\lambda x)$$

as

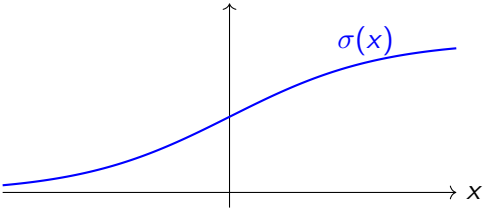
$$\lambda \to \infty.$$

The result becomes:

$$\begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases}$$

So sigmoid neurons can approximate decisions between regions.

Visual Intuition: Sigmoid to Step Function



As the slope increases, the sigmoid becomes closer to a step.

Connection to Theorem 1

Theorem 2 tells us:

Sigmoid neurons can distinguish functions

Therefore:

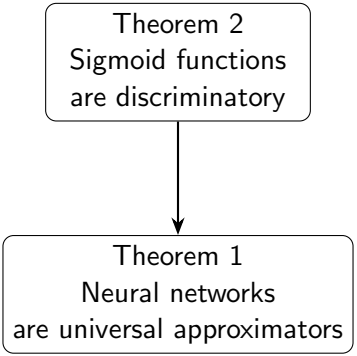
No non-zero error can hide from all neurons

This eliminates the contradiction in the proof of Theorem 1.

Final Connection

Theorem 2 provides the mathematical foundation that makes universal approximation possible.

The Two Theorems Together



Theorem 2 is the key ingredient needed to prove Theorem 1.

Summary of Cybenko's Results

The paper establishes two important results:

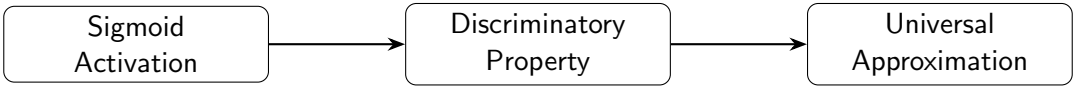
1. **Theorem 2**
   Continuous sigmoidal functions are discriminatory.

$$\int \sigma(w^T x + b) d\mu(x) = 0 \Rightarrow \mu = 0$$

2. **Theorem 1**
   Single-hidden-layer neural networks can approximate every continuous function.

$$\overline{S} = C(I_n)$$

The Big Picture



Simple nonlinear components can combine to represent extremely complex functions.

What the Theorem Does and Does Not Say

It proves

- Representation is possible
- One hidden layer is enough
- Error can become arbitrarily small

It does not prove

- How to train the network
- How many neurons are needed
- Fast convergence

Impact on Modern Deep Learning

Cybenko's work provided the theoretical foundation for:

- Modern neural network architectures
- Deep learning research
- Function approximation with machine learning models

Although modern networks are much deeper and use different activations, the central idea remains:

Neural networks are powerful function approximators.

Final Takeaway

Main Idea

A neural network does not need to explicitly know the formula of a function.
By combining enough simple nonlinear units, it can approximate any continuous function.

Questions?