Neural Network and Curse of Dimensionality (CoD)

a survey

Longye Tian
longye.tian@anu.edu.au

Australian National University
School of Economics

May 2nd, 2025

Big picture

- CoD is prevalent in heterogenous agent macro
- Macro papers cite some NN paper and say “NN can deal with CoD”
- Why and when? (Function class)

Structure

- Background
  - What is CoD? Big $\mathcal{O}$ notation
  - What is NN?
- A survey on two important papers
  - Barron 1993
  - Poggio et al 2017

What is the curse of dimensionality?

Increase in the dimension of input space, the amount of data or computational power need to grow exponentially to maintain the same approximation accuracy.

Big O notation

Let $f(n)$ and $g(n)$ be functions from the set of natural numbers to the set of real numbers. We say that $f(n)$ is $O(g(n))$ if and only of there exists positive constants c and n_0 such that

$$f(n) \leq c \cdot g(n), \quad \text{for all } n \geq n_0$$

In other words, $f(n) \in O(g(n))$ means that $f(n)$ grows no faster than $g(n)$ up to a constant factor.

Example of Big O notation

$f(n)$ is $O(n^2)$, this means $f(n)$ can be $2n^2$ or $100n^2$ or $n^2 + 3n + 7$, as long as the leading factor is n^2

- $O(1)$: constant time
- $O(n)$: linear time
- $O(n^2)$: quadratic time
- $O(log\, n)$: logarithmic time
- $O(2^n)$: exponential time

Neuron

The basic unit of neural networks are called **neuron/nodes/perceptron/unit**.

- It is a simple function
$$g : \mathbb{R}^k \to \mathbb{R}$$
- take a vector $z \in \mathbb{R}^k$ as input
- transform it using an **affine mapping**
- parameterized by a vector of **weights** $\gamma \in \mathbb{R}^{k+1}$.
- applies a non-linear function (**activation function**) f to output a real number.

Neuron

The basic unit of neural networks are called **neuron/nodes/perceptron/unit**.

- It is a simple function

$$g : \mathbb{R}^k \to \mathbb{R}$$

Formally,

$$g(z;\gamma) = f\left(\underbrace{\gamma_0}_{\text{bias}} + \sum_{i=1}^{k} \gamma_i z_i\right)$$

Layer

A **layer** with p nodes is a vector function stacking p neurons, i.e.,

$$G : \mathbb{R}^k \to \mathbb{R}^p$$

or

$$G(z;\Gamma) = \begin{bmatrix} g(z;\gamma_1) \\ \cdots \\ g(z;\gamma_p) \end{bmatrix}, \quad \text{where } \Gamma = (\gamma_1, \gamma_2, \cdots, \gamma_p)$$

We call it a layer with **width** p.

Remark: In this presentation, we assume that all neurons in a layer share the same input and same activation function. It can be more sophisticated.

Neural network/neural net/multilayer perception

A **neural network** is a composition of several layers. Let $G_1, \cdots, G_\ell$ denote ℓ layers of conformable dimensions. A neural network is a function

$$N : \mathbb{R}^m \to \mathbb{R}^n$$

$$N(x) = (\underbrace{G_\ell}_{output} \circ \underbrace{G_{\ell-1} \circ \cdots \circ G_2}_{hidden} \circ \underbrace{G_1}_{input})(x)$$

- we call the number of layer ℓ the **length of neural net**
- we call the largest width of layers the **width of neural net**

Remark: Only the input and output layers are contrained by the dimensions of the variable of interest. Hidden layers are not

Deep Neural Nets

Deeper neural nets are more flexible and have the potential learn more complex function.

- More than three layers $\Longrightarrow$ deep
- Less than three layers $\Longrightarrow$ shallow

Universal Approximation Theorem

A neural network can approximate any continuous function from one-dimensional space to another with any desired non-zero amount of error, provided it is given sufficient depth or sufficient width.

Curse of dimensionality

For a given approximation error, the number of parameters required to approximate a function with a neural network increase **linearly with the dimension** of the input x, while it increases exponentially for most other classes of approximators.

Neural Network Approximation and CoD

- Barron 1993: Universal Approximation Bounds in Superpositions of a Sigmoidal Function
- Poggio et al 2017: Why and When can deep-but not shallow-network avoid the curse of dimensionality: A review

Barron 1993 - Key messages

For a specific function class $\Gamma_{C,B}$ (see later for detailed desciption)

- Neural network with one hidden layer of n nodes has error bound $\mathcal{O}(1/n)$ indenpendent of dimension d.
- Fixed basis method: error bound no better than $\mathcal{O}(1/n^{2/d})$

Sigmoidal functions

Sigmoidal: look like s (sigma)

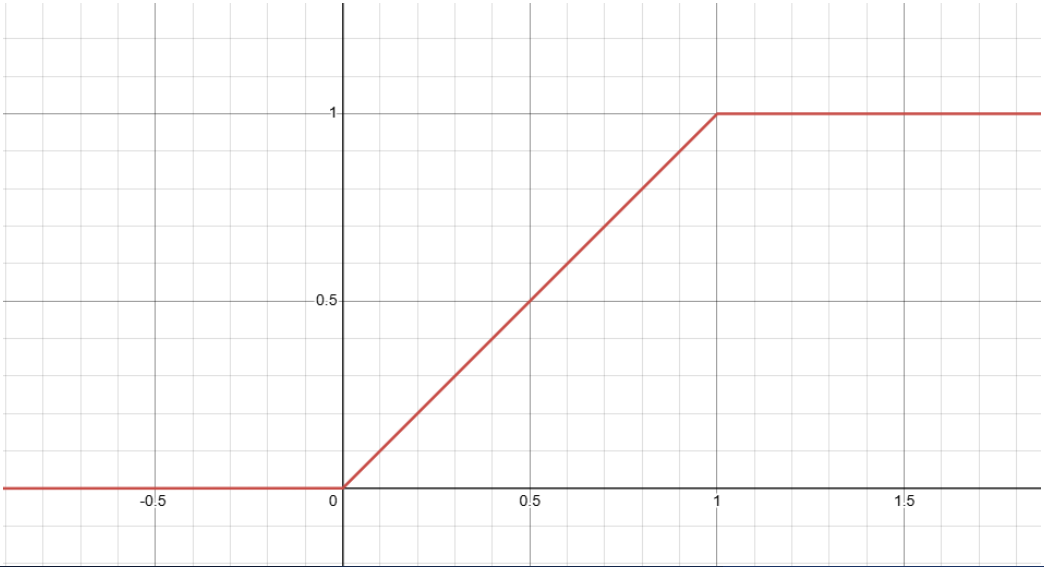
Definition 1

A sigmoidal function $\varphi(z)$ is a bounded measurable function on a real line satisfies

- $\varphi(z) \to 1$ as $z \to \infty$
- $\varphi(z) \to 0$ as $z \to -\infty$

Remark: the very popular ReLu $\varphi(z) = \max\{0, z\}$, we have $\varphi(z) - \varphi(z-1)$ is sigmoidal.

Sigmoidal function



Barron 1993 - function class

Let Γ_c be the set of functions f on $\mathbb{R}^d$ for which there exists a complex-valued Fourier transform measure $\hat{F}(d\omega)$ such that

$$f(x) = \int_{\mathbb{R}^d} e^{i\omega \cdot x} \hat{F}(d\omega)$$

with the first moment of the Fourier magnitude distribution bounded by

$$C_f = \int_{\mathbb{R}^d} |\omega||\hat{F}(d\omega)| \leq C$$

where $|\omega|$ represents the Euclidean norm of the frequency vector. For a bounded set $B \subset \mathbb{R}^d$ containing the origin $\Gamma_{C,B}$ is the set of functions f on B that can be represented by the above integrals for $x \in B$ with

$$C_{f,B} = \int \sup_{x \in B} |\omega \cdot x||\hat{F}(d\omega)| \leq C$$

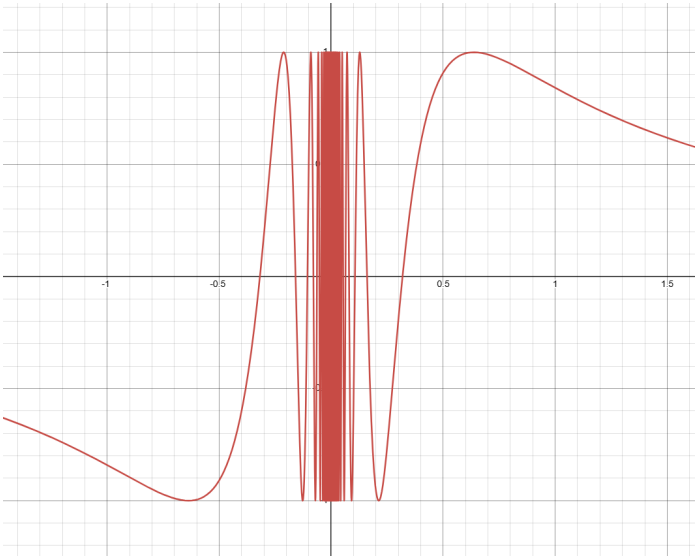
Interpretation of C_f and $C_{f,B}$

- It is an average of the norm of the frequency vector weighted by the Fourier distribution

$$C_f = \int_{\mathbb{R}^d} |\omega||\hat{F}(d\omega)| \leq C$$

- It measure the extend the function oscillates

Example that does not work - $\sin(1/x)$



Barron 1993 - Key theorem

Theorem 2

For every function f with finite C_f and every $n \geq 1$, there exists a neural network with one hidden layer of n-sigmoidal nodes:

$$f_n(x) = \sum_{k=1}^{n} c_k \phi(a_k \cdot x + b_k) + c_0$$

such that the approximation error with respect to any probability measure μ on the ball $B_r = \{x : |x| \leq r\}$ satisfies

$$\int (f(x) - f_n(x))^2 \, \mu(dx) \leq \frac{(2rC_f)^2}{n}$$

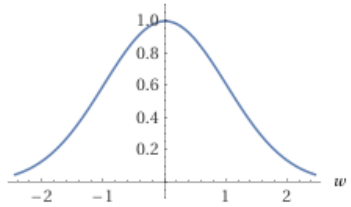
Remark: the dimension d doesn't matter with approximation error of $\mathcal{O}(1/n)$. One caveat: be careful with C_f that depends on d.

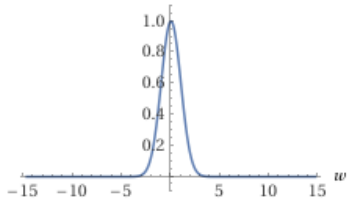
Barron 1993 - function class

What kind of functions are in this class?

- Gaussian function
$$f(x) = e^{-|x|^2/2}, \quad C_f \leq \sqrt{d}$$
- Positive definite functions $f(x)$ on $\mathbb{R}^d$
$$\sum_{i,j} \alpha_i \alpha_j f(x_i - x_j) \geq 0$$
for all choices of points and scalars.
  - Covariance functions;
  - Characteristic functions

Gaussian





Barron 1993 - function class

- Ridge functions

$$f(x) = g(a \cdot x), \quad a \in \mathbb{R}^d, |a| = 1, \quad g \text{ univariate}$$

  Remark: This type is particular good as C_f is independent d.
- Sigmoidal network with smooth activation function (has limitation as discussed below)
- Radial function

$$f(x) = g(|x|)$$

- Linear functions, quadratic forms, polynomials, step functions

Barron 1993 - functions

- Sufficiently smooth functions: $C^{\lfloor d/2 \rfloor + 2}$ $(\mathbb{R}^d)$
- Boolean functions: functions defined on domain $\{0,1\}^d$

Barron 1993 - Key message

- For Barron class functions, single hidden layer net achieve approximation error of order $\mathcal{O}(1/n)$ for any dimension d
- Using fixed basis functions (polynomial, etc), the error cannot be smaller than $\mathcal{O}(1/n^{2/d})$

Remark: $d = 1000$, for $\epsilon = 10^{-5}$, need

- NN: $N = 10^5$ nodes for any dimensions
- Fixed basis we need $N_{fb} = 10^{5\times 1000/2} = 10^{2500}$ basis functions

Poggio et al 2017

Main message: Deep convolutional net have the theoretical guarantee which shallow nets do not have: they can avoid the curse of dimensionality for compositional functions.

- The most interesting set of problems consists of compositional functions composed of a hierarchy of constituent functions that are local: An example is

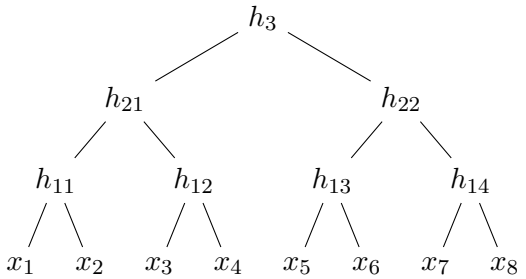
$$f(x_1, \cdots, x_8) = h_3(h_{21}(x_1, x_2), h_{12}(x_3, x_4), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8)))$$

- Key aspect of convolutional network that can given them an exponential advantage is not weight sharing but locality at each level of the hierarchy.

Poggio et al 2017 - Compositional functions with Binary tree structure

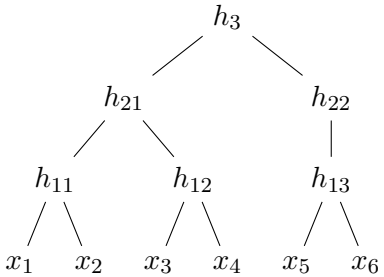
The function is composed of with functions in a binary tree structure (each take 2 inputs and 1 outputs)

$$f(x_1, \cdots, x_8) = h_3(h_{21}(x_1, x_2), h_{12}(x_3, x_4), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8)))$$



Poggio et al 2017 - General compositional function

For general function with directed acyclic graph, i.e.,



Poggio et al 2017 - function class

Since f is unknown, we need to make some assumptions on the class of functions from whih the unknown target function is chosen.

- $f \in W_m^d$: functions of n variables with continuous partial derivatives of order up to m
- $f \in W_m^{d,2}$: compositional functions with binary tree architecture where consituent functions are in W_m^2
- Argument: many real-world functions have this compositional structure where each constituent function depends on only a small number of variables

Poggio et al 2017 - complexity

Let

- N: total number of nodes in a network
- V_N: set of all networks given with total number of nodes N
- $V_N \subset V_{N+1}$
- The degree of approximation is defined by

$$dist(f, V_n) = \inf_{P \in V_N} \|f - P\|$$

- For example if $dist(f, V_n) = \mathcal{O}(N^{-\gamma})$ for some $\gamma > 0$, then a network with nodes $N = \mathcal{O}(\epsilon^{-1/\gamma})$ will be sufficient to gurantee an approximation with accuracy at least ϵ

Poggio et al 2017 - Key theoerm 1

Theorem 3 (Shallow network leads to CoD)

For general functions $f \in W_m^d$, shallow networks requires

$$N = \mathcal{O}(\epsilon^{-d/m})$$

nodes to achieve accuracy ϵ. This implies CoD.

Remark: Let $\epsilon = 10^{-5}$, i.e., for a d-dimensional C^m function, use shallow network to approximate with error 10^{-5} need nodes of order

$$(10^{-5})^{-d/m} = 10^{5d/m}$$

- For $d = 10, m = 1$, $(10^{-5})^{-10} = 10^{50}$
- For $d = 100, m = 1$, $(10^{-5})^{-100} = 10^{500}$

Poggio et al 2017 - key theorem 2

Theorem 4 (Deep network with Binary Tree structure)

For compositional functions $f \in W_m^{d,2}$, deep networks need only

$$N = \mathcal{O}((d-1)\epsilon^{-2/m})$$

nodes to achieve approximation error ϵ.

Remark: Note here we have linear dependency on input. Let $\epsilon = 10^{-5}$, i.e., for a d-dimensional C^m function with binary tree structure, we need

$$(d-1) \times (10^{-5})^{-2/m}) = (d-1) \times 10^{10/m}$$

- For $d = 10, m = 1$, $(10-1) \times (10^{-5})^{-2/m} = 9 \times 10^{10}$
- For $d = 100, m = 1$, $(100-1) \times (10^{-5})^{-2/m} = 99 \times 10^{10}$

Poggio et al 2017 - key theorem 3

Theorem 5 (General compositional functions)

For functions with general directed acyclic graph structure, deep network requires

$$N = \mathcal{O}(\sum_v \epsilon^{-d_v/m_v})$$

where d_v, m_v is the number of inputs and smoothness parameter to each constituent functions

Remark: If we have same smoothness level, then

$$N = \mathcal{O}(\sum_v \epsilon^{-d_v/m_v}) = \mathcal{O}(\epsilon^{-\max_v dv/mv})$$

Poggio et al 2017 - Key message

- For compositional functions, local dimensionality matters
- If the function is very smooth, CoD is less severe.

Are these applicable to macro?

- Is the policy function smooth?
- Does the policy function has finite mean in Fourier distribution
- Does the policy function have low local dimensions?