Deep learning and its mathematical nature

Nadia Sukhorukova

Swinburne University of Technology

nsukhorukova@swin.edu.au

Vera Roshchina, Vinesha Peiris, Julien Ugon and Reinier Diaz-Millan

August 8, 2025

Overview

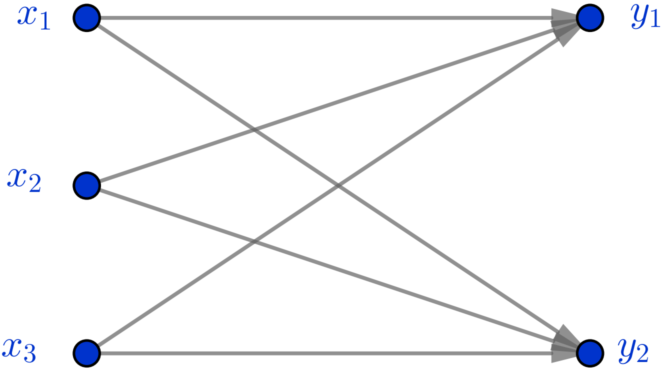
1. Deep learning

2. Kolmogorov-Arnold theorem and deep learning

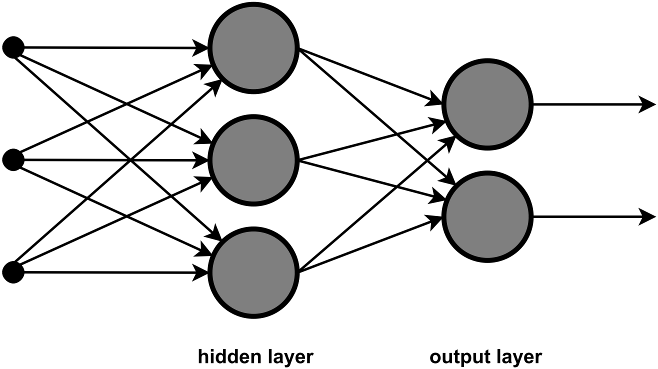
3. Application with no hidden layers: input to output

4. Artificial confidence

Neural Network (no hidden layer)



Neural Network (Wikipedia picture: creative commons)

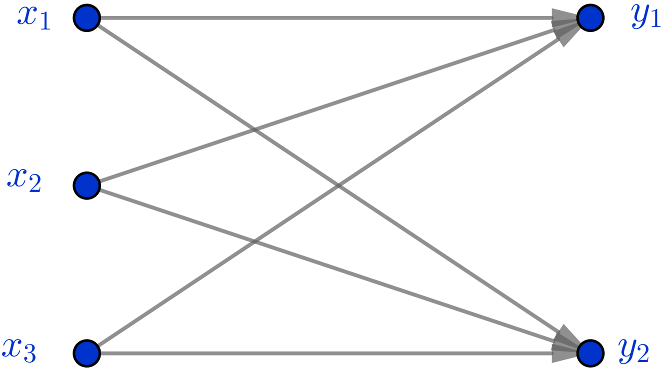


Neural Network: activation function and weights

$$\varphi(W, x) = \sigma\left(\sum_{j=1}^{n} w_j x_j + w_0\right), \tag{1}$$

- σ is called *activation function*. This is a chosen function, not subject to optimisation.
- W are weights (subject to optimisation).
- If one or more hidden layer is present, φ becomes a composition, where affine transformations are alternating with (different) activation functions.

Neural Network (no hidden layer)



David Hilbert, Andrey Kolmogorov, Vladimir Arnold

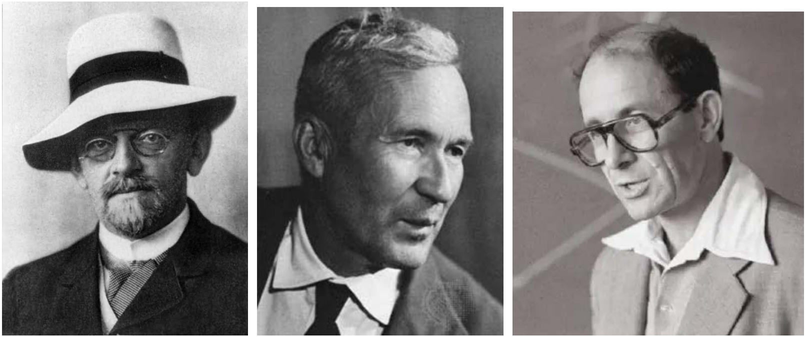


Figure: D. Hilbert: unknown author; A. Kolmogorov: Novosti Press; Vladimir Arnold: RIA Novosti

Hilbert’s thirteenth problem

Solve 7th-degree equation using algebraic (variant: continuous) functions of two parameters.

This theorem solved a more constrained, yet more general form of Hilbert’s thirteenth problem (continuous part). The algebraic part is still unresolved.

Kolmogorov-Arnold theorem

Kolmogorov-Arnold theorem states that every multivariate continuous function can be written as a finite composition of univariate functions and binary operation of addition:

$$f(\mathrm{x}) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right).$$

Kolmogorov-Arnold theorem: variants

1. 1962, George Lorentz: the outer functions Φ_q can be replaced by a single function Φ:
2. 1965, David Sprecher replaced the inner functions $\phi_{q,p}$ and, with some restrictions,

$$f(\mathrm{x}) = \sum_{q=0}^{2n} \Phi \left(\sum_{p=1}^{n} \lambda_p \phi(x_p + \eta q) + q \right).$$

There are still some limitations:

1. The theorem does not hold in general for complex multi-variate functions.
2. The non-smoothness of the inner functions has limited the practical use of the representation (we will come back to it).

Neural Network and universal approximators: Universal Approximation Theorem by Cybenko, 1989.

Theorem

Fix a continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ *(activation function, sigmoildal) and positive integers* d, D*, then for every continuous function* $f : \mathbb{R}^d \to \mathbb{R}^D$ *and every* $\epsilon > 0$ *there exists a continuous function* $f_\epsilon : \mathbb{R}^d \to \mathbb{R}^D$ *(the layer output) with representation*
$f_\epsilon = W_2 \circ \sigma \circ W_1$*, where* W_2, W_1 *are affine maps and* $\circ$ *denotes component-wise composition, such that the approximation bound*

$$\sup_{x \in K} \|f(x) - f_\epsilon(x)\| < \varepsilon$$

arbitrarily small (distance from f *to* f_ϵ *can be infinitely small).*

Universal Approximation Theorem: activation function choice

1. Hornik also showed in 1991 that it is not the specific choice of the activation function but the multilayer architecture itself (number of hidden layers, number of nodes in each layer) that gives neural networks the potential of being universal approximators.
2. Allan Pinkus in 1999 showed that the universal approximation property is equivalent to having a nonpolynomial activation function.

Discussion around Kolmogorov-Arnold theorem and its relevance to deep learning

1. F. Girosi and T. Poggio, "Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant," in Neural Computation, vol. 1, no. 4, pp. 465-469, Dec. 1989, doi: 10.1162/neco.1989.1.4.465.
2. Věra Kůrková . "Kolmogorov's Theorem Is Relevant", Neural Computation, vol. 3, no. 4, pp. 617-622, Dec. 1989https://doi.org/10.1162/neco.1991.3.4.617

Prof. Věra Kůrková

Prof. Věra Kůrková received Ph.D. in topology from Charles University, Prague in 1980 and DrSc (equiv. Prof.) in 1999. Since 1990, she was working for the Institute of Computer Science, Czech Academy of Sciences. From 2002 to 2008, she was the Head of the Department of Theoretical Computer Science. Her research interests include machine learning, mathematical theory of neurocomputing, inverse problems, nonlinear approximation, and optimization.
Homepage: https://www.cs.cas.cz/staff/kurkova/en
Cite from her homepage (by E. W. Dijkstra)
"The question of whether Machines Can Think... is about as relevant as the question of whether Submarines Can Swim."
"The problems of the real world are primarily those you are left with when you refuse to apply their effective solutions."

Universal Approximation Theorem: Cybenko-Hornik-Pinkus

Theorem

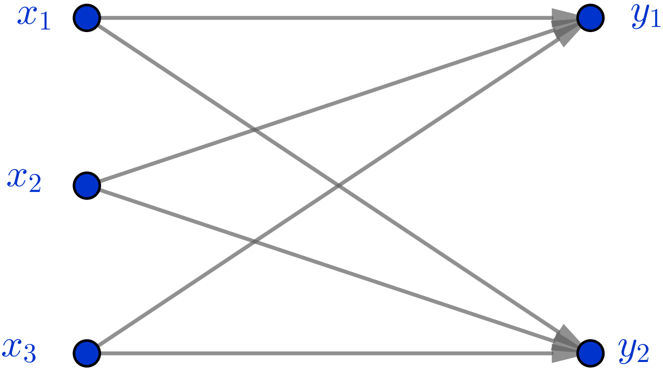
Fix a continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ (activation function) and positive integers d, D. The function σ is not a polynomial if and only if, for every continuous function $f : \mathbb{R}^d \to \mathbb{R}^D$ and every $\epsilon > 0$ there exists a continuous function $f_\epsilon : \mathbb{R}^d \to \mathbb{R}^D$ (the layer output) with representation $f_\epsilon = W_2 \circ \sigma \circ W_1$, where W_2, W_1 are affine maps and $\circ$ denotes component-wise composition, such that the approximation bound

$$\sup_{x \in K} \|f(x) - f_\epsilon(x)\| < \varepsilon$$

arbitrarily small (distance from f to f_ϵ can be infinitely small).

If σ is monotone then $\sigma \circ W_1$ is quasiaffine and f_ϵ the sum of quasiaffine (which may or may not be quasiconvex or quasiaffine itself).

Neural Network (no hidden layer)



Quasiconvex and quasiaffine functions

Definition

Let D be a convex subset of $\mathbb{R}^n$. A function $f : D \to \mathbb{R}$ is *quasiconvex* if and only if its sublevel set

$$S_\alpha = \{x \in D \,|\, f(x) \leq \alpha\}$$

is convex for any $\alpha \in \mathbb{R}$.

Definition

A function $f : D \to \mathbb{R}$, where D is a convex subset of $\mathbb{R}^n$, is called quasiconcave if $-f$ is quasiconvex.

Definition

Functions that are both quasiconvex and quasiconcave are called quasiaffine.

Important observations.

- Quasiconvex functions do not need to be continuous.
- In the case of univariate functions, quasiaffine functions are monotone functions.
- Sublevel sets of quasiaffine functions are half-spaces.
- If a function is quasiaffine on $\mathbb{R}^n$ (unconstrained problems) then, from the definition, its level sets must be half-spaces, and it is clear that the hyperplanes defining these half-spaces need to be parallel. In the presence of constraints, this observation is not valid.

Example where the hyperplanes-boundaries of sublevel sets are not parallel

Consider $f(x,y)=\frac{x}{y}$, where $y>0$. The sublevel sets are

$$S_\alpha=\{(x,y):\frac{x}{y}\leq\alpha, y>0\},$$

where α is a given real number. Then S_α can be described as

$$\{x-\alpha y\leq 0,\quad y>0\}.$$

Each sublevel set is still a half-space, but the corresponding hyperplanes (in this example they correspond to level sets) are not parallel. These hyperplanes intersect at $(0,0)$, but this point is excluded from the domain due to the requirement for the denominator y to be strictly positive.

Approximation by a quasiaffine function

Then the problem is as follows:

$$\text{minimise } \tilde{z} \tag{2}$$

subject to

$$f(x) - g(A, x) \leq \tilde{z},\ x \in X, \tag{3}$$

$$g(A, x) - f(x) \leq \tilde{z},\ x \in X, \tag{4}$$

where $g(A, x)$ is a quasiaffine function with respect to A. Since the sublevel sets of quasiaffine functions are half-spaces, the constraint set (3)-(4) is a polytope, since X is a finite grid.

A finite number of linear constraints may be added to (2)-(4), while the constraint set remains a polytope.

One possible method to solve is Bisection.

Bisection method: auxiliary problem

$$\text{minimise } \tilde{u} \tag{5}$$

subject to

$$f(x) - g(A, x) \le z + \tilde{u},\ \ x \in X, \tag{6}$$

$$f(x) - g(A, x) \le z + \tilde{u},\ \ x \in X, \tag{7}$$

where $z = \frac{1}{2}(u + l)$ is the current bisection point (bisecting the possible values of maximal deviation).
If an optimal solution $\tilde{u} \le 0$, the corresponding sublevel set of the maximal deviation function is not empty (update the upper bound), otherwise the set is empty (update the lower bound). If X is a finite grid, then (5)-(7) is a linear programming problem and can be solved efficiently at each step of the bisection method.

Ratios of two linear forms

Ratios of two linear forms (linear combinations of known functions, also called basis functions) are quasilinear as functions of their coefficients:

$$\frac{A^T G(x)}{B^T H(x)},$$

where $A, B \in \mathbb{R}^n$ are the variables,
$G(x) = (g_1(x), \ldots, g_n(x))$ and $H(x) = (h_1(x), \ldots, h_n(x))$. The denominator is positive.

Rational networks

These functions are very powerful approximation tools and one of the possibilities is just to approximate our function by rational (basis functions are monomials) or generalise rational (continuous basis functions) functions.
In this case, the structure (activation function applied to an affine mapping) is not relevant anymore.

Network approximation (quasiconvex model), joint work with Dr. V Roschina and Dr. V. Peiris, under review

$$\min_{w\in\mathbb{R}^{n+1}} \max_{i\in 1:N} \left| y^i - \sigma\left(\sum_{j=1}^{n} w_j x_j^i + w_0\right)\right|.$$

σ is the activation function, we work with leaky ReLU, the corresponding optimisation problem is quasiconvex (uniform approximation).

Network approximation (quasiconvex model): numerical experiments (1000 points for training and 370 for the test)

Method	Test set classification accuracy	Confusion matrix	
MATLAB toolbox	89.7%	108	25
		13	224
Uniform approximation	60.54%	77	56
		90	147

Table: Original dataset: classification results

Why our procedure is not efficient?

Network approximation (quasiconvex model): comments (1000 points for training and 370 for the test)

Uniform approximation, due to its nature, treats under-represented groups as valid points, while least squares approximation tends to “average” and therefore under-represented groups tend to be “ignored”. This is a great advantage when the under-represented groups are outliers, but in many cases these points are valid data.

On the other hand, the presence of ouliers may decrease may decrease the accuracy in the case of uniform approximation.

Why uniform approximation?

Therefore, our hypothesis is that uniform approximation approach is preferable in the following cases.

1. Absence (or small number) of outliers.
2. Presence of under-represented groups of valid data or uneven distribution of data between the classes (that is, one class is significantly larger than others).
3. Limited size of the available data, where most datapoints are valid and accurate.

Reduced dataset: 5+35 and 35+5

Method	Test accuracy	Confusion matrix	
MATLAB toolbox	64.3%	296	66
		291	347
Uniform approximation	69.5%	193	169
		136	502

Table: Reduced dataset: classification results for uneven number of points from each class in the training set: 5 ponts in Class 1 and 35 points in Class 2.

Method	Test accuracy	Confusion matrix	
MATLAB toolbox	74.6%	116	246
		8	630
Uniform approximation	66.5%	128	234
		101	537

Table: Reduced dataset: classification results for uneven number of points from each class in the training set: 35 points in Class 1 and 5 points in Class 2

Random training set size is 100

We start with the case where the training set contains 100 points in total.

Table: Reduced dataset: classification results for randomly generated training set of 100 points: repeated 10 times, the accuracy is averaged.

Method	Test set classification accuracy
MSE	56.94%
Uniform approximation	73.19%

The classification accuracy is higher for uniform approximation.

Random training set size is 50

The training set contains 50 points in total.

Table: Reduced dataset: classification results for randomly generated training set of 50 points: repeated 10 times, the accuracy is averaged.

Method	Test set classification accuracy
MSE	57.08%
Uniform approximation	71.58%

The classification accuracy is higher for uniform approximation, but the gap is slightly decreasing compared to the experiment with 100 points.

Random training set size is 20

The training set contains 20 points in total.

Table: Reduced dataset: classification results for randomly generated training set of 20 points: repeated 10 times, the accuracy is averaged.

Method	Test set classification accuracy
MSE	56.57%
Uniform approximation	69.21%

The classification accuracy is higher for uniform approximation. The gap between uniform approximation and MATLAB is quite significant.

Random training set size is 10

The training set contains 10 points in total.

Table: Reduced dataset: classification results for randomly generated training set of 10 points: repeated 10 times, the accuracy is averaged.

Method	Test set classification accuracy
MSE	77.00%
Uniform approximation	79.40%

The classification accuracy is higher for uniform approximation.

My co-authors

Title: Artificial Neural Networks with uniform norm based loss functions
In press, accepted by Advances in Computational Mathematics (Feb. 2023)

Figure: Vera Roshchina (UNSW) and Vinesha Peiris (Curtin)

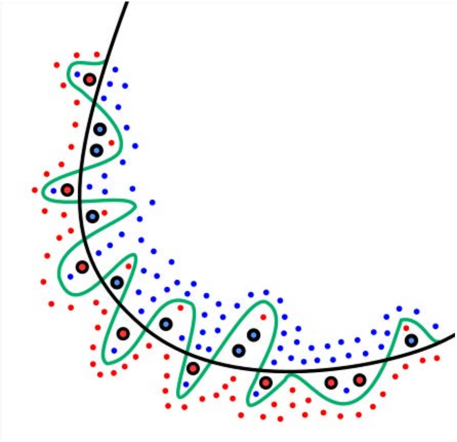
Gary Markus, Scientific American, October 2022. Co-author with Ernest Davis, Rebooting AI.

”For now, we are trapped in a “local minimum” in which companies pursue benchmarks, rather than foundational ideas, eking out small improvements with the technologies they already have rather than pausing to ask more fundamental questions.”

Alternatives to Deep learning

- KANs
- Rational approximation (sometimes rational networks)

Overfitting (Wiki)



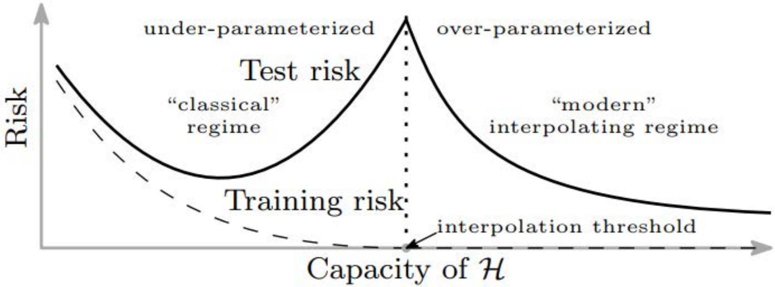
Belkin, M. (2021)

Belkin, M. (2021). Fit without fear: Remarkable mathematical phenomena of deep learning through the prism of interpolation. Acta Numerica, 30, 203-248. doi:10.1017/S0962492921000039

From this paper (also available Arxiv):

"...the best practice of modern deep learning is arguably much closer to interpolation than to the classical regimes (when training and testing losses match). For example in his 2017 tutorial on deep learning Ruslan Salakhutdinov stated that “The best way to solve the problem from practical standpoint is you build a very big system ... basically you want to make sure you hit the zero training error".”

Modern and classical regime (Belkin 2021 and conference publications)



Why overparametrisation does NOT lead to overfitting?

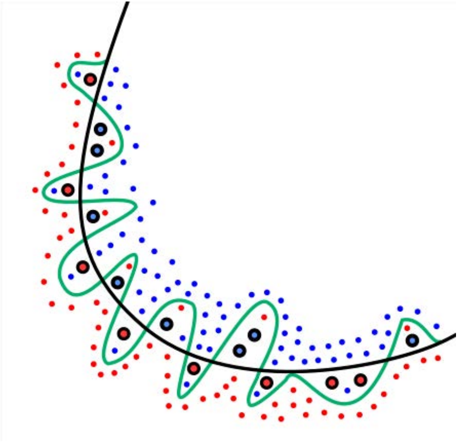
1. The models they use.
2. The optimisation method they use (stochastic gradient descent).

Belkin 2021:

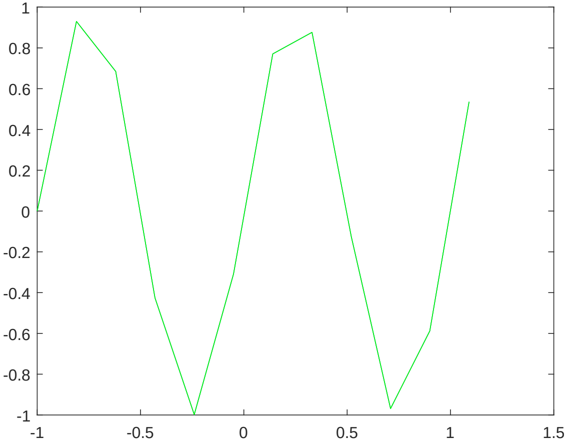
"... the smoothest function is "simplest"... the "maximum smoothness" guiding principle can be formulated as:

Select the smoothest function, according to some notion of functional smoothness, among those that fit the data perfectly".

Overfitting, again, Wiki



What is smoothness?



1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
1.5