Introduction to some of the pure mathematics underpinning deep learning

Finnur Lárusson March 2024

Notes for three lectures given in *The Other Pure Seminar* at the University of Adelaide

---

Let $\sigma : \mathbb{R} \to \mathbb{R}$ be any continuous function, except not a polynomial. Let $n \in \mathbb{N}$. Consider all functions of the form $\sigma \circ \phi$, where $\phi : \mathbb{R}^n \to \mathbb{R}$ is affine, that is, $\phi(x) = a \cdot x + b$ with $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.

Theorem 1. Such functions span a dense linear subspace of $\mathscr{C}(\mathbb{R}^n, \mathbb{R})$.

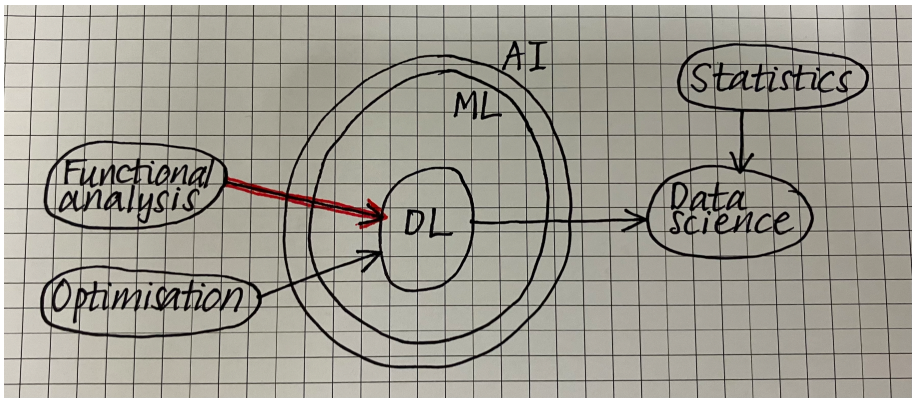
More explicitly, for every continuous function $g : \mathbb{R}^n \to \mathbb{R}$, every compact subset $K \subset \mathbb{R}^n$, and every $\epsilon > 0$, there are $a_1, \ldots, a_m \in \mathbb{R}^n$, $b_1 \ldots, b_m \in \mathbb{R}$, and $c_1 \ldots, c_m \in \mathbb{R}$ for some $m \in \mathbb{N}$ (depending on σ, g, K, and ϵ), such that setting $\phi_i(x) = a_i \cdot x + b_i$, we have

$$\Big| g - \sum_{i=1}^{m} c_i \, \sigma \circ \phi_i \Big| < \epsilon \quad \text{on} \quad K.$$

This is density with respect to the natural topology on $\mathscr{C}(\mathbb{R}^n, \mathbb{R})$.

It is easy to see that polynomials have to be excluded: If σ is a polynomial, then $\sigma \circ \phi$ is a polynomial with $\deg(\sigma \circ \phi) \leq \deg \sigma$, so the functions $\sigma \circ \phi$ span a finite-dimensional subspace of $\mathscr{C}(\mathbb{R}^n, \mathbb{R})$.

Later we will consider how to prove Theorem 1. First we will describe where it comes from and why it is interesting. It comes from the field of artificial intelligence, more specifically, from deep learning!



The figure and the following comments are meant to provide very quick and oversimplified background. The figure shows functional analysis and optimisation as the main areas of pure mathematics that underpin deep learning. In these talks, we will focus on the former, signified by the red arrow.

Artificial intelligence seeks to get computers to mimic human intelligence.

Machine learning seeks to get a computer to use data to improve its own program, that is, to "learn" from data. Today, artificial intelligence and machine learning are nearly synonyms: most current developments in artificial intelligence use machine learning.

Deep learning is an approach to machine learning, using deep artificial neural networks.

Deep learning and statistics both support data science. Statistics makes inferences about a population from samples, based on an assumed statistical model. Deep learning looks for patterns in huge data sets.

From the introduction to [1] (see the list of references at the end of the notes): *Deep learning has undoubtedly established itself as the outstanding machine learning technique of recent times ... through a series of overwhelming successes in widely different application areas. Perhaps the most famous application of deep learning and certainly one of the first where these techniques became state-of-the-art is image classification ... In this area, deep learning is nowadays the only method that is seriously considered ... A second famous application area is the training of deep-learning-based agents to play board games or computer games ... probably the most prominent achievement yet is the development of an algorithm that beat the best human player in the game of Go ... a feat that was previously unthinkable owing to the extreme complexity of this game ... deep learning has also led to impressive breakthroughs in the natural sciences. ... One of the most astounding recent breakthroughs in scientific applications is the development of a deep-learning-based predictor for the folding behavior of proteins ... This predictor is the first method to match the accuracy of lab-based methods. Finally, in the vast field of natural language processing ... impressive advances were made based on deep learning...*

A neural network is a scheme to compute maps $\mathbb{R}^n \to \mathbb{R}^m$ (or often $[0,1]^n \to \mathbb{R}^m$) or rather a family of maps depending on parameters. In deep learning, after a choice of neural network, training data are used to tune the parameters so as to approximate a "dream function" with desired properties. This is the learning process. Optimisation methods, the basic one being gradient descent, are used to reduce the error in matching the training data. Luckily, in this setting, the gradient can be effectively computed, even for huge networks. We will not consider the optimisation aspect any further.

Example. Training a network to recognise handwritten digits: see [17], Section 3.

So what exactly is a neural network and how is it a recipe for computing maps $\mathbb{R}^n \to \mathbb{R}^m$? The following "very general definition ... encompasses virtually all network architectures used in practice" [5].

A network is a finite directed acyclic graph with a set V of nodes, none of them isolated, and a set E of edges, such that:

- V is partitioned into the set I of *input nodes* with no incoming edges, the set O of *output nodes* with no outgoing edges, and the set H of *hidden nodes*. (The word "deep" refers to the presence of hidden nodes.)
- To every $v \in H$ is associated a continuous *activation function* $\sigma_v : \mathbb{R} \to \mathbb{R}$ and a *bias* $b_v \in \mathbb{R}$.

- To every $v \in O$ is associated a bias $b_v \in \mathbb{R}$.
- To every $e \in E$ is associated a *weight* $w_e \in \mathbb{R}$.

The weights and biases are the *trainable parameters* of the network.

The network computes a map $\mathbb{R}^n \to \mathbb{R}^m$ as follows, where n is the number of input nodes and m is the number of output nodes.

- Each $v \in I$ receives an input signal x_v and transmits it along each of its outgoing edges.
- Each $v \in H$ receives a signal x_u from the initial node u of each of its incoming edges e, computes
$$x_v = \sigma_v\left(b_v + \sum_{e=(u,v)\in E} w_e x_u\right),$$
and transmits x_v along each of its outgoing edges.
- Each $v \in O$ receives a signal x_u from the initial node u of each of its incoming edges e and computes the output
$$x_v = b_v + \sum_{e=(u,v)\in E} w_e x_u.$$

The resulting output map $\mathbb{R}^n \to \mathbb{R}^m$ of the network is a (typically complicated) composition of affine maps and activation functions.

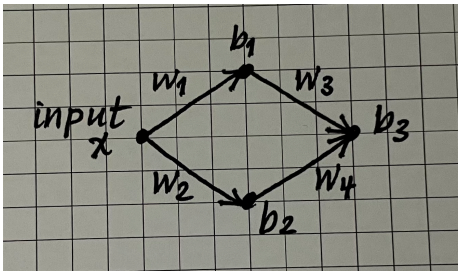
When there are no hidden nodes (so the network is not deep), the output function is affine, that is, of the form $f(x) = Ax + b$, where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$. "Deep" learning is then simply linear regression. Deep learning can be viewed as a nonlinear regression method that allows effective computations with huge data sets.

Activation functions are usually increasing and they are often the same function for every hidden node. *Sigmoidal functions*, such as arctan, tanh, and the logistic function $t \mapsto (1 + e^{-t})^{-1}$, were commonly used as activation functions, but the *rectified linear unit* ReLU $t \mapsto \max\{0, t\}$ is now more popular because it allows for quicker training (there is apparently some advantage to its lack of differentiability at "the corner").

How big are neural networks in practice? AlexNet won a famous image recognition competition in 2012. It has about 650,000 nodes and 60 million parameters. The paper describing it is one of the most influential papers in computer vision (Wikipedia). It has had about 150,000 citations (Google Scholar). AlexNet uses ReLU. GPT-4, released in March 2023, is rumoured to have 1.76 trillion parameters (Wikipedia).

Example. Consider the tiny network below with activation function $\sigma = \text{ReLU}$. The output function $f : \mathbb{R} \to \mathbb{R}$ of the network is given by the formula
$$f(x) = b_3 + w_3\sigma(b_1 + w_1 x) + w_4\sigma(b_2 + w_2 x),$$
depending on the weights w_1, w_2, w_3, w_4 and the biases b_1, b_2, b_3, which can take arbitrary real values. The output function is a linear combination of the constant function 1 and two functions of the form $\sigma \circ$ affine function. The possible output functions are any

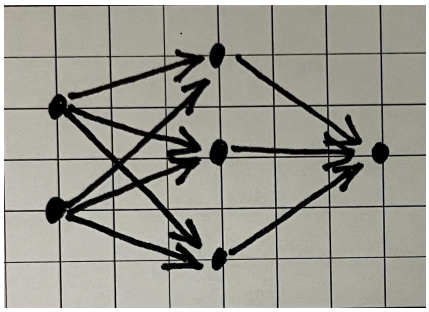
affine function, any piecewise affine function with one singularity, and some (not all) piecewise affine functions with two singularities.



We now come to the **fundamental question:** Is there a theoretical obstruction to deep learning achieving its goal? More formally, can any continuous map $\mathbb{R}^n \to \mathbb{R}^m$ be approximated arbitrarily closely by a neural network?

A variety of affirmative answers are provided by theorems dating from the late 1980s to as recently as last year. They are called *universal approximation theorems.*

We restrict ourselves to fully connected feed-forward networks with one hidden layer and a single output node (with zero bias). Here is an example:



Let the input be $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, where n is the number of input nodes, and let us work with a single activation function σ. The output function $f : \mathbb{R}^n \to \mathbb{R}$ is then of the form

$$f(x) = \sum_{i=1}^{m} c_i \, \sigma(a_i \cdot x + b_i).$$

The number of hidden nodes is m. The components of the vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ and the numbers $c_1, \ldots, c_m \in \mathbb{R}$ are the weights, and the numbers $b_1, \ldots, b_m \in \mathbb{R}$ are the biases. The network has $m(n+2)$ parameters. (In practice, the number of parameters tends to be roughly quadratic in the number of nodes.)

Theorem 1 is precisely the universal approximation theorem for fully connected feed-forward networks with a single output node, a single hidden layer with an arbitrary number of nodes, and the weakest possible assumption on the activation function σ [10]. Next we will present a proof of a weaker, earlier version the theorem [3] (see also [9]), followed by another version of the theorem with a very different proof [14].

Following [3], we will prove the universal approximation theorem for fully connected feed-forward networks with a single output node, a single hidden layer with an arbitrary number of nodes, and a continuous activation function $\sigma : \mathbb{R} \to \mathbb{R}$ such that

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

Such a function is obviously not a polynomial. For example, σ could be the logistic function $t \mapsto (1 + e^{-t})^{-1}$, but note that σ is not required to be increasing.

Let $I = [0, 1]$. Consider all functions of the form $\sigma \circ \phi$, where $\phi : \mathbb{R}^n \to \mathbb{R}$ is affine, that is, $\phi(x) = ax + b$ with $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.

Theorem 2. Such functions span a dense linear subspace of $\mathscr{C}(I^n, \mathbb{R})$.

We will use functional analysis to prove the theorem. Here is a very brief summary of the basic concepts and results we need.

- Functional analysis brings together linear algebra and topology. Its fundamental concepts are topological vector spaces X (here over $\mathbb{R}$) and continuous linear maps between them. Of particular interest are continuous linear functionals $X \to \mathbb{R}$. They form the dual space X^* of X. In finite dimensions there is nothing new here: functional analysis is simply linear algebra.

- If X is locally convex, for example $X = \mathscr{C}(I^n, \mathbb{R})$, then X^* is a large and useful space (Hahn-Banach theorem). Some other spaces, for example $L^p(I)$, $0 < p < 1$, have no nontrivial continuous linear functionals.

- Riesz representation theorems usefully describe the elements of X^* for some spaces X that are important in practice. For example, every continuous linear functional on $\mathscr{C}(I^n, \mathbb{R})$ is given by $f \mapsto \int_{I^n} f \, d\mu$ for a unique regular real Borel measure μ. This brings in measure theory, including advanced integration theory. We will need one of the important theorems that allow us to exchange limits and integration under suitable conditions.

- We will also need one of the several versions of the Fourier transform. Let $\mathscr{S}_n$ be the Schwarz space of rapidly decreasing smooth functions $\mathbb{R}^n \to \mathbb{R}$. The Fourier transform is an automorphism of $\mathscr{S}_n$ that converts the action of a differential operator $P(\partial/\partial x_1, \ldots, \partial/\partial x_n)$ into multiplication by the polynomial $P(x_1, \ldots, x_n)$. There is an induced Fourier transform with the same good properties on the dual space $\mathscr{S}_n^*$ of "tempered distributions". A measure μ as above is a tempered distribution.

We now proceed to the proof of the theorem. Suppose that μ is a regular real Borel measure on I^n such that

$$\int_{I^n} \sigma(ax + b) \, d\mu(x) = 0$$

for all $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. We need to show that $\mu = 0$. By Hahn-Banach and Riesz, this means that the linear subspace of $\mathscr{C}(I^n, \mathbb{R})$ spanned by all functions $\sigma \circ \phi$ is not contained in any hyperplane and is therefore dense.

We will show that $\mu = 0$ by showing that its Fourier transform $\hat{\mu}$ vanishes. The Fourier transform is defined as follows, for $h \in \mathscr{S}_n$:

$$\hat{\mu}(h) = \mu(\hat{h}) = \int_{I^n} \hat{h}\, d\mu = \int_{I^n} \int_{\mathbb{R}^n} h(t) e^{-itx}\, dt\, d\mu(x) = \int_{\mathbb{R}^n} \left(\int_{I^n} e^{-itx}\, d\mu(x) \right) h(t) dt.$$

To complete the proof, we need to go from

$$\int_{I^n} \sigma(ax + b)\, d\mu(x) = 0 \quad \text{for all } a \in \mathbb{R}^n, b \in \mathbb{R}$$

to

$$\int_{I^n} e^{itx}\, d\mu(x) = 0 \quad \text{for all } t \in \mathbb{R}^n.$$

Fix $a \in \mathbb{R}^n$, $a \neq 0$, and $b, c \in \mathbb{R}$, and let

$$\sigma_s(x) = \sigma(s(ax + b) + c).$$

As $s \to \infty$, the functions $\sigma_s : \mathbb{R}^n \to \mathbb{R}$ converge pointwise to the function γ defined by

$$\gamma(x) = \begin{cases} 1 & \text{if } ax + b > 0, \\ 0 & \text{if } ax + b < 0, \\ \sigma(c) & \text{if } ax + b = 0. \end{cases}$$

By Lebesgue's dominated convergence theorem, the limit is preserved by integration, so

$$0 = \lim_{s \to \infty} \int_{I^n} \sigma_s\, d\mu = \int_{I^n} \gamma\, d\mu = \sigma(c)\mu(P_{a,b}) + \mu(H_{a,b}),$$

where $P_{a,b}$ is the hyperplane in $\mathbb{R}^n$ defined by the equation $ax + b = 0$ and $H_{a,b}$ is the open halfspace defined by the equation $ax + b > 0$. It follows that $\mu(P_{a,b}) = \mu(H_{a,b}) = 0$ for all a, b. Also, $\mu(\mathbb{R}^n) = 0$.

Fix $a \in \mathbb{R}^n$. For a bounded measurable function f on $\mathbb{R}$, let

$$F(f) = \int_{I^n} f(ax)\, d\mu(x).$$

Then $F(1) = 0$. Also, taking f to be the characteristic function of the interval $[b, \infty)$, we get

$$F(f) = \mu(P_{a,-b}) + \mu(H_{a,-b}) = 0.$$

Similarly, $F(f) = 0$ if f is the characteristic function of the interval (b, ∞). By linearity, F vanishes for the characteristic function of any interval and hence for any step function (linear combination of characteristic functions of intervals).

The continuous function $e(y) = e^{iy} = \cos y + i \sin y$ on $\mathbb{R}$ can be uniformly approximated by step functions on the compact interval $a \cdot I^n$, so

$$0 = F(e) = \int_{I^n} e^{iax}\, d\mu(x),$$

and the proof of Theorem 2 is complete.

Example (from [3]). Use a neural network to approximately solve the decision problem for a closed subset $D \subset I^n$. Our "dream function" is the characteristic function f of D.

For $x \in I^n$, let $d(x) = \min\{\|x - y\| : y \in D\}$ be the distance from x to D. Take $\epsilon > 0$ and let

$$f_\epsilon(x) = \max\left\{0, 1 - \frac{d(x)}{\epsilon}\right\},$$

so $f_\epsilon : I^n \to [0,1]$ is continuous with $f_\epsilon(x) = 0$ if $d(x) \geq \epsilon$ and $f_\epsilon(x) = 1$ if $x \in D$.

By the universal approximation theorem, there is a neural network with output function $g : I^n \to \mathbb{R}$ such that $|f_\epsilon - g| < \frac{1}{2}$ on I^n. If $g(x) \geq \frac{1}{2}$, we guess that $x \in D$, and if $g(x) < \frac{1}{2}$, we guess that $x \notin D$. The guess is correct for all $x \in D$ and for all x at a distance at least ϵ from D. If $0 < d(x) < \epsilon$, then the answer for x depends on the choice of g and may or may not be correct.

We will now give a very different proof of another version of the universal approximation theorem. To simplify the notation we consider functions $\mathbb{R}^n \to \mathbb{R}$ with $n = 1$.

Theorem 3. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a *smooth* function, not a polynomial. All functions of the form $\sigma \circ \phi$, where $\phi : \mathbb{R} \to \mathbb{R}$ is affine, span a dense linear subspace of $\mathscr{C}(\mathbb{R}, \mathbb{R})$.

For the proof we need two ingredients. The first is the most classical of all approximation theorems.

Weierstrass approximation theorem (1885). Let $f : [a,b] \to \mathbb{R}$ be continuous. For every $\epsilon > 0$, there is a polynomial p such that $|f - p| < \epsilon$ on $[a,b]$.

Here is a simple proof (with straightforward details omitted) using convolution with the Gauss kernel. We extend f to a continuous function on $\mathbb{R}$ with compact support and set

$$f_\epsilon(z) = \frac{1}{\epsilon\sqrt{\pi}} \int_{\mathbb{R}} f(x) e^{-(x-z)^2/\epsilon^2}\, dx, \quad z \in \mathbb{C}.$$

On $\mathbb{R}$, $f_\epsilon \to f$ uniformly as $\epsilon \to 0$, while on $\mathbb{C}$, f_ϵ is holomorphic and hence uniformly approximated by its Taylor polynomials on compact subsets of $\mathbb{C}$, in particular on $[a,b]$.

The second ingredient is a curious old result from [2].

Proposition. If $\sigma : \mathbb{R} \to \mathbb{R}$ is smooth and not a polynomial, then there is $b \in \mathbb{R}$ with $\sigma^{(n)}(b) \neq 0$ for all $n \geq 0$.

Proof (with some gaps left for you to fill in). For each $n \geq 0$, let A_n be the closed set where $\sigma^{(n)}$ vanishes. We argue by contradiction and assume that $\bigcup A_n = \mathbb{R}$. By Baire's theorem, in every nonempty interval, some A_n has nonempty interior, so the open set $\bigcup A_n^\circ$ is dense in $\mathbb{R}$. It is a disjoint union of open intervals. Note that $A_0^\circ \subset A_1^\circ \subset \cdots$.

Next observe that if $I \subset J$ are nonempty open intervals with $I \subset A_m^\circ$ and $J \subset A_n^\circ$, $m < n$, then $J \subset A_m^\circ$. Hence each of the connected components I of $\bigcup A_k^\circ$ lies in A_n° for some n, so on I, σ agrees with a polynomial of degree at most $n-1$. It follows that $B = \mathbb{R} \setminus \bigcup A_n^\circ$ has no isolated points (think of joining the graphs of two polynomials at a point, with the resulting graph being smooth at the point). We will show that B is empty.

We apply Baire's theorem again, this time to $B = \bigcup(A_n \cap B)$, and conclude that the union of the interiors (relative to B) of the sets $A_n \cap B$ is dense in B. As a very particular consequence, assuming B is nonempty, one of these interiors is nonempty, that is, there is an open interval I such that $\varnothing \neq B \cap I \subset A_m$ for some m, so $\sigma^{(m)} = 0$ on $B \cap I$. Since B has no isolated points, it follows that $\sigma^{(k)} = 0$ on $B \cap I$ for all $k \geq m$ (think of difference quotients). The same is easily seen to hold on those connected components of $\bigcup A_n^\circ$ (if any) that have both endpoints in $B \cap I$. Hence there is an open subinterval of I that contains points of B and lies in A_m°, but this is absurd. □

We can now prove Theorem 3, following [14], Proposition 3.4. Let V be the linear subspace of $\mathscr{C}(\mathbb{R}, \mathbb{R})$ spanned by all functions of the form $\sigma \circ \phi$, where $\phi : \mathbb{R} \to \mathbb{R}$ is affine. By the proposition, there is $b \in \mathbb{R}$ such that $\sigma^{(n)}(b) \neq 0$ for all $n \geq 0$. Letting ϕ be the constant function b, we see that the constant function $\sigma(b)$ lies in V. For every $a \in \mathbb{R}$ and $h \neq 0$, the function

$$x \mapsto \big(\sigma((a+h)x + b) - \sigma(ax + b)\big)/h$$

lies in V. Letting $h \to 0$, we see that the function

$$x \mapsto \frac{d}{da}\sigma(ax + b)\bigg|_{a=0} = \sigma'(b)x$$

lies in $\overline{V}$. Similarly, the function

$$x \mapsto \frac{d^k}{da^k}\sigma(ax + b)\bigg|_{a=0} = \sigma^{(k)}(b)x^k$$

lies in $\overline{V}$ for every $k \geq 2$. Since $\sigma^{(k)}(b) \neq 0$ for all $k \geq 0$, we see that $\overline{V}$ contains all polynomials, so the Weierstrass approximation theorem implies that V is dense in $\mathscr{C}(\mathbb{R}, \mathbb{R})$. The proof of Theorem 3 is complete.

The proof shows that the theorem holds even if we restrict our networks to have weights in any sequence of nonzero numbers converging to 0. Also, by a trivial modification of the proof of the proposition, we only need to assume that the continuous activation function σ is smooth and not a polynomial on some nonempty open interval, not necessarily on all of $\mathbb{R}$.

Theorems 2 and 3 do not capture today's favourite activation function ReLU (it is neither smooth nor sigmoidal), but the proofs given here can be continued along similar lines so as to produce two different proofs of Theorem 1, which does, of course, cover ReLU (see [10] and [14]).

Finally, we will summarise interesting answers to the following questions for fully connected feed-forward networks.

- Does universal approximation hold for ReLU networks with bounded width but arbitrary depth?
- Do networks have to get bigger and bigger to approximate better and better?
- Is there a universal approximation theorem for complex-valued networks?

- If we only know a network's output function, can we figure out its innards?

There is a quick answer to the first question. By a simple trick, every output function $I^n \to \mathbb{R}$ of a ReLU network with a single hidden layer of width m can be computed by a ReLU network with m hidden layers of width $n+2$. For simplicity, take $n = 1$. Here is how we can compute any output function $f(x) = \sum_{i=1}^{m} c_i \sigma(\phi_i(x))$, where $\sigma = \text{ReLU}$ and $\phi_1, \ldots, \phi_m : \mathbb{R} \to \mathbb{R}$ are affine, by a network with m hidden layers of width 3 (modified from [7], Lemma 3). The first hidden layer computes the affine map

$$x \mapsto (x, \phi_1(x), T),$$

where T is a real number to be specified later. For $i = 2, \ldots, m$, the i^{th} hidden layer computes the affine map

$$(x, y, z) \mapsto (x, \phi_i(x), z + c_{i-1} y).$$

Finally, the output node computes the affine function

$$(x, y, z) \mapsto z + c_m y - T.$$

It is easily checked that if T is large enough, the resulting output function is the given function f.

The Kolmogorov–Arnold superposition theorem (1956–1957) solved Hilbert's 13th problem (1900) in the negative. The theorem says that continuous functions of several variables can be reduced to functions of a single variable via addition. For a readable survey, see [12]. Here is a special case of the theorem.

Theorem. There is $\lambda \in \mathbb{R}$ and continuous functions $\phi_1, \ldots, \phi_5 : I \to \mathbb{R}$ such that the following holds. For every continuous function $f : I^2 \to \mathbb{R}$, there is a continuous function $g : \mathbb{R} \to \mathbb{R}$ such that

$$f(x, y) = \sum_{k=1}^{5} g(\phi_k(x) + \lambda \phi_k(y)).$$

The following theorem of Maiorov and Pinkus ([11], Theorem 4) is a consequence.

Theorem. There is a real analytic, strictly increasing, sigmoidal function $\sigma : \mathbb{R} \to \mathbb{R}$ such that for each $n \geq 1$, universal approximation of continuous functions $I^n \to \mathbb{R}$ holds for networks with two hidden layers with $3n$ nodes in the first layer and $6n + 3$ nodes in the second layer and with activation function σ.

"We do not, for one moment, suggest that one try to construct and use the above mentioned σ. This σ is wonderfully smooth but unacceptably complex"([11], page 83).

Voigtlaender proved a universal approximation theorem for complex-valued networks [16]. He points out several applications for which complex-valued networks are better suited than real-valued ones.

Theorem. Complex-valued networks with a single hidden layer and continuous activation function $\sigma : \mathbb{C} \to \mathbb{C}$ have the universal approximation property for continuous functions $\mathbb{C}^n \to \mathbb{C}$ if and only if σ is not polyharmonic.

We say that σ is polyharmonic if $\Delta^k \sigma = 0$ for some $k \geq 1$. Here, Δ is the usual Laplacian $\partial^2/\partial x^2 + \partial^2/\partial y^2$ on $\mathbb{C} \cong \mathbb{R}^2$. For necessity, one observes that polyharmonicity is preserved by precomposition by affine maps $\mathbb{C} \to \mathbb{C}$, $z \mapsto az+b$, and that the space of polyharmonic functions of a given order is a closed (and obviously proper) subspace of the space of all continuous functions. For sufficiency, when σ is smooth, we can use the assumption that σ is not polyharmonic to show, along the lines of the proof of Theorem 3, that all monomials $z^j \bar{z}^k$ lie in the closure of the space of output functions.

To what extent is a network determined by its output function? Fefferman's answer in [6] is: For generic networks with a special activation function, completely!

We say that a network is *in standard order* if the biases in each hidden layer are positive and the nodes in the layer are ordered so that the biases form a strictly increasing sequence. We also say that a network is *generic* if it has no zero weights and if the ratio of the weights associated to two different edges with the same initial node is irrational.

Theorem. Consider two generic networks in standard order with the same number of input nodes and the same number of output nodes. If they have the same output function, then they are identical.

Fefferman uses the activation function $\tanh(x/2)$ and exploits its special properties. It extends to a meromorphic function on $\mathbb{C}$, whose poles form an arithmetic progression $(2k+1)i\pi$, $k \in \mathbb{Z}$. Reducing to the case of a single input node, Fefferman shows that there is a largest open subset U of $\mathbb{C}$ to which the output function can be analytically continued. He shows by 40 pages of very intricate analysis that the structure of the singular set $\mathbb{C} \setminus U$ determines the network.

As of 3 March 2024, [6] has only 9 citations in MathSciNet. Interestingly, all but one are from 2021–2023. After nearly three decades, Fefferman's work is being revisited and continued.

Brief guide to references not already cited. The survey [8] is an introduction to deep learning for applied mathematicians. The paper [15] is focused on applications in physics, but starts with a readable introduction to the basics of deep learning (Sections 2–4). The papers [4] and [17] describe very recent and possible future applications of deep learning in pure mathematics. There are pointers to further sources in [17]. The paper [13] uses algebraic topology to shed light on some fundamental questions about deep neural networks.

References

1. J. Berner, P. Grohs, G. Kutyniok, P. Petersen. *The modern mathematics of deep learning.* In: P. Grohs, G. Kutyniok, eds. *Mathematical Aspects of Deep Learning.* Cambridge University Press, 2022, 1–111.

2. E. Corominas, F. Sunyer i Balaguer. *Sur des conditions pour qu'une fonction infiniment dérivable soit un polynome.* Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences **238** (1954) 558–559.

3. G. Cybenko. *Approximation by superpositions of a sigmoidal function.* Mathematics of Control, Signals and Systems **2** (1989) 303–314.

4. A. Davies et al. *Advancing mathematics by guiding human intuition with AI.* Nature **600** (2021) 70–74.

5. R. DeVore, B. Hanin, G. Petrova. *Neural network approximation.* Acta Numerica **30** (2021) 327–444.

6. C. Fefferman. *Reconstructing a neural net from its output.* Revista Matemática Iberoamericana **10** (1994) 507–555.

7. B. Hanin. *Universal function approximation by deep neural nets with bounded width and ReLU activations.* Mathematics **2019**, 7, 992.

8. C. F. Higham, D. J. Higham. *Deep learning: an introduction for applied mathematicians.* SIAM Review **61** (2019) 860–891.

9. K. Hornik. *Approximation capabilities of multilayer feedforward networks.* Neural Networks **4** (1991) 251–257.

10. M. Leshno, V. Ya. Lin, A. Pinkus, S. Schocken. *Multilayered feedforward networks with a nonpolynomial activation function can approximate any function.* Neural Networks **6** (1993) 861–867.

11. V. Maiorov, A. Pinkus. *Lower bounds for approximation by MLP neural networks.* Neurocomputing **25** (1999) 81–91.

12. S. A. Morris. *Hilbert 13: Are there any genuine continuous multivariate real-valued functions?* Bulletin of the American Mathematical Society **58** (2021) 107–118.

13. G. Naitzat, A. Zhitnikov, L.-H. Lim. *Topology of deep neural networks.* Journal of Machine Learning Research **21** (2020) 1–40.

14. A. Pinkus. *Approximation theory of the MLP model in neural networks.* Acta Numerica **8** (1999) 143–195.

15. F. Ruehle. *Data science applications to string theory.* Physics Reports **839** (2020) 1–117.

16. F. Voigtlaender. *The universal approximation theorem for complex-valued neural networks.* Applied and Computational Harmonic Analysis **64** (2023) 33–61.

17. G. Williamson. *Is deep learning a useful tool for the pure mathematician?* Preprint, 2023, arXiv:2304.12602.