The unreasonable effectiveness of mathematics, revisited
Big data and neuroscience

Jaime Gómez-Ramírez
Fundación Reina Sofía. Centre for Research in Neurodegenerative Diseases

April 11 2018
The effectiveness of mathematics

Einstein: The most incomprehensible thing about the world is that is comprehensible

Wigner: The unreasonable effectiveness of mathematics

Gelfand: The Unreasonable Ineffectiveness of Mathematics in biology
The effectiveness of mathematics

heat loss in coffee

\[
\frac{dQ}{dT} = A_s(T_{\text{coffee}} - T_{\text{room}})
\]
Wigner’s 1960 essay "the enormous usefulness of mathematics in natural science is something bordering on the mysterious"
Wigner’s 1960 essay "the enormous usefulness of mathematics in natural science is something bordering on the mysterious"

The typical interpretation of Wigner test is as follows:

**premise** math concepts arise from aesthetic impulse in humans
Wigner’s 1960 essay "the enormous usefulness of mathematics in natural science is something bordering on the mysterious"

The typical interpretation of Wigner test is as follows:

**premise** math concepts arise from aesthetic impulse in humans

**premise** is unreasonable to think that those same impulses are effective
Wigner’s 1960 essay "the enormous usefulness of mathematics in natural science is something bordering on the mysterious"

The typical interpretation of Wigner test is as follows:

- **premise** math concepts arise from aesthetic impulse in humans
- **premise** is unreasonable to think that those same impulses are effective
- **observation** nevertheless it so happens that they are effective
Wigner’s 1960 essay "the enormous usefulness of mathematics in natural science is something bordering on the mysterious"

The typical interpretation of Wigner test is as follows:

- **premise** math concepts arise from aesthetic impulse in humans
- **premise** is unreasonable to think that those same impulses are effective
- **observation** nevertheless it so happens that they are effective
- **consequence** it follows that math concepts are unreasonably effective (assuming that the aesthetic premise as valid)
The effectiveness of mathematics

- Wigner’s 1960 essay "the enormous usefulness of mathematics in natural science is something bordering on the mysterious"

- The typical interpretation of Wigner test is as follows:
  - premise math concepts arise from aesthetic impulse in humans
  - premise is unreasonable to think that those same impulses are effective
  - observation nevertheless it so happens that they are effective
  - consequence it follows that math concepts are unreasonably effective (assuming that the aesthetic premise as valid)

  e.g imaginary numbers, tensor. Math concepts appear and propagate
Wigner did seminal work on group theory applied to discover symmetry principles.
Wigner did seminal work on group theory applied to discover symmetry principles.

Group theory replaced previous methods of analysis in quantum mechanics, group pest, finding invariants instead of seeking for explicit solution by calculus.

The goal of science is not to explain nature (the black box) but to explain the regularities in the behavior of the object. "Not the things in themselves but the relationships between the things." (Poincaré)

The search for causal explanation in terms of mathematical principles necessitates the belief of the mathematical structure of the universe, the c-word.

Jaime Gómez-Ramírez

The effectiveness of mathematics
Wigner did seminal work on group theory applied to discover symmetry principles.

Group theory replaced previous methods of analysis in quantum mechanics, group pest, finding invariants instead of seeking for explicit solution by calculus.

The goal of science is not to explain nature (the black box) but to explain the regularities in the behavior of the object. "Not the things in themselves but the relationships between the things. (Poincaré)

The search for causal explanation in terms of mathematical principles necessitates the belief of the mathematical structure of the universe, the c-word.
We are "lucky" that regularities exist and that we can grasp them mathematically.

This is Newton's contribution and this is in essence why deep learning works.

Regularities are invariant with respect to space and time.

\[ T(A) \rightarrow T(X), T(B) \rightarrow T(Y) \]
We are "lucky" that regularities exist and that we can grasp them mathematically.

This is Newton’s contribution and this is in essence why deep learning works.
We are "lucky" that regularities exist and that we can grasp them mathematically.

- This is Newton’s contribution and this is in essence why deep learning works.

Regularities are invariant with respect to space and time. $A, B \rightarrow X, Y$ under $T(A), T(B) \rightarrow T(X), T(Y)$

Convolutional networks exploit image invariance to work (A cat is a cat is a cat)
What makes possible for us to discover regularities is the division between initial conditions and regularities.

Laws of nature are $IF$ initial conditions $THEN$ event.

That’s why causality is so hard, we need to include/exclude all possible combination of antecedents (initial conditions)

\[ t = \sqrt{\frac{2s}{g}} \]
Good doesn’t play dice eg. stochastic brownian motion
- Good doesn’t play dice eg. stochastic brownian motion
- Our knowledge of nature contains ’a strange hierarchy’ (Events we observed → Laws (regularities to discover) → Symmetry (invariance principles)
- Good doesn’t play dice eg. stochastic brownian motion
- Our knowledge of nature contains ’a strange hierarchy’ (Events we observed → Laws (regularities to discover) → Symmetry (invariance principles)
- The future is always uncertain but nevertheless there are correlations - laws- that we can discover
AI, Machine Learning, Deep Learning

- AI
- Machine Learning
- ANN are non linear mapping systems whose functioning principles are vaguely based on the nervous systems of mammals
- Deep learning

Data the most valuable asset and computation is a cheap commodity (information wants to be free)
Neurons with a binary threshold activation function analogous to first order logic sentences

By itself a neuron (or an ann) does very little but a sufficiently large network with appropriate structure and properly chosen weights can approximate with arbitrary accuracy any function.

"A Logical Calculus of Ideas Immanent in Nervous Activity McCulloch, Pitts, 1943"

’If it doesn’t rain ($x_1 w_1$) and homework done ($x_2 w_2$), go to the movies ($y$, output)’

$y = f\left(\sum_{k} w_x x_k\right)$

1

Jaime Gómez-Ramírez
A perceptron is any feedforward network of nodes with responses like equation (2).

\[ y = f \left( \sum_{k} w_x x_k \right) = f(z) \]
A perceptron is any feedforward network of nodes with responses like equation (2).

\[ y = f\left(\sum \limits_k w_k x_k\right) = f(z) \]  \hspace{1cm} (2)

In general, \( f \) is bounded nondecreasing nonlinear \textit{squeezing} function, eg. the sigmoid

\[ f(z) = \frac{1}{1 + e^{-z}}, \quad f'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} \]
Other choices are the tanh, step function and more recently the relu function.

\[ y = ReLU(z) = \max(0, z), \quad y' = 1, \quad z > 0 \]
Other choices are the tanh, step function and more recently the relu function.

\[ y = ReLU(z) = \max(0, z), y' = 1, z > 0 \]

ReLu works better, faster (gradient constant), \( y'(0) \) approximated
\[ y = \ln(1.0 + e^x) \]
Other choices are the tanh, step function and more recently the relu function.

\[ y = \text{ReLU}(z) = \max(0, z), y' = 1, z > 0 \]

- ReLu works better, faster (gradient constant), \( y'(0) \) approximated
  \[ y = \ln(1.0 + e^x) \]
- Reduced likelihood of gradient to vanish
Other choices are the tanh, step function and more recently the relu function.

\[ y = ReLU(z) = \max(0, z), y' = 1, z > 0 \]

ReLU works better, faster (gradient constant), \( y'(0) \) approximated

\[ y = \ln(1.0 + e^x) \]

Reduced likelihood of gradient to vanish

Sparsity produced when \( z \leq 0 \), sigmoids on the other hand tend to represent more dense representations
What can and can’t perceptrons do?

- (Single-layer) perceptrons can correctly classify only data sets that are linearly separable (they can be separated by a hyperplane)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>XOR(a, b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

![Diagram Xor Function](image.png)
What can and can’t perceptrons do?

- (Single-layer) perceptrons can correctly classify only data sets that are linearly separable (they can be separated by a hyperplane).
- The XOR function is famously non linearly separable and this is very important because many classification problems are not linearly separable.
What can and can’t perceptrons do?

- There are $2^{2^d}$ boolean functions of $d$ boolean input variables and only $O(2^{2^2})$ are linearly separable.
- For $d=2$, $14/16$ are linearly separable (XOR and its complement are the exceptions), but for $d = 4$, only $1882/65536$ are linearly separable.
What can and can’t perceptrons do?

- There are $2^{2^d}$ boolean functions of $d$ boolean input variables and only $O(2^{2^d})$ are linearly separable.
  - For $d=2$ $14/16$ are linearly separable (XOR and its complement are the exceptions), but for $d = 4$, only $1882/65536$ are linearly separable.

- Although at that time it was known that multilayer networks were more powerful than single layer ones, the learning algorithms for multilayer architectures were not known.
Deep networks

- ANN learn by example and use backpropagation
Deep networks

- ANN learn by example and use backpropagation
- If data are well-behaved it will learn not only the training examples but also the underlying relationships

Units within a layer are independent so they can be evaluated simultaneously. For example, a network with 2,000 nodes in two layers will produce a response in 2 time steps rather than in 2,000 steps if each neuron required to be processed serially (dependent)
Deep networks

- ANN learn by example and use backpropagation
- If data are well-behaved it will learn not only the training examples but also the underlying relationships
- ANN are adaptive and self-repairing, also has some fault tolerance due to its redundant parallel structure (dense connectivity makes it resilient to minor damage, graceful degradation)
Deep networks

- ANN learn by example and use backpropagation
- If data are well-behaved it will learn not only the training examples but also the underlying relationships
- ANN are adaptive and self-repairing, also has some fault tolerance due to its redundant parallel structure (dense connectivity makes it resilient to minor damage, graceful degradation)
- Units within a layer are independent so they can be evaluated simultaneously eg. network with 2,000 nodes in two layers will produce a response in 2 time steps rather than in 2,000 steps if each neuron required to be processes serially (dependent)
Deep networks

- ANN learn by example and use backpropagation
- If data are well-behaved it will learn not only the training examples but also the underlying relationships
- ANN are adaptive and self-repairing, also has some fault tolerance due to its redundant parallel structure (dense connectivity makes it resilient to minor damage, graceful degradation)
- Units within a layer are independent so they can be evaluated simultaneously eg. network with 2,000 nodes in two layers will produce a response in 2 time steps rather than in 2,000 steps if each neuron required to be processes serially (dependent)
- Until the advent of GPUs this advantage were not fully exploited by computers
Deep networks

Table: ANN versus real nervous system

<table>
<thead>
<tr>
<th>MLP</th>
<th>Nervous System</th>
</tr>
</thead>
<tbody>
<tr>
<td>feedforward</td>
<td>recurrent</td>
</tr>
<tr>
<td>dense(fullyconnected)</td>
<td>sparse(local)</td>
</tr>
<tr>
<td>$O(10^{2,3,4})$</td>
<td>$O(10^{10}), O(10^{15})$</td>
</tr>
<tr>
<td>static</td>
<td>dynamic: spike trains, synchronization, fatigue</td>
</tr>
</tbody>
</table>
A frame
Why MLP is better than one layer?

\[ y = mx \] is a system with one parameter, \( m \), what kind of datasets can separate? only the linearly separable ones.

\[ y = \sin(kx), \] also has one parameter, the frequency \( k \), but can separate any arbitrary distribution of points in the \( x \)-axis.
Any bounded function can be approximated with arbitrary accuracy if enough hidden units are available - multilayer perceptrons are universal approximators.
Universality of MLP

- Any bounded function can be approximated with arbitrary accuracy if enough hidden units are available - multilayer perceptrons are universal approximators
- How many layers do we need for this astounding property (universal approximators)? Kolmogorov showed that one hidden layer is sufficient
Universality of MLP

- Any bounded function can be approximated with arbitrary accuracy if enough hidden units are available - multilayer perceptrons are universal approximators
- How many layers do we need for this astounding property (universal approximators)? Kolmogorov showed that one hidden layer is sufficient
  - Any continuous function with n variables to a m-dimensional output can be implemented by a network with one hidden layer
Universality of MLP

- Any bounded function can be approximated with arbitrary accuracy if enough hidden units are available - multilayer perceptrons are universal approximators.

- How many layers do we need for this astounding property (universal approximators)? Kolmogorov showed that one hidden layer is sufficient.
  - Any continuous function with $n$ variables to an $m$-dimensional output can be implemented by a network with one hidden layer.

- Unfortunately the proof is not constructive, that is, it does not tell how the weights should be chosen to produce such a function.
Is it universal approximation a rare property? Not really, many other systems such as polynomials, trigonometric polynomials (eg Fourier series), wavelets, kernel regression systems (svm) have also universal properties
First layer detects the edges, and the second has the abstract concept of loop and straight lines, this is actually the hope of having a layer structure and it works because what Wigner already said
Cost $C(w)$, the gradient $\frac{dC(w)}{dw} = 0$ (huge column vector with $784 + 16 \times 16 + 16 \times 10 + 16 + 16 + 10$ dimensions).

The negative of the gradient which is the direction of the steepest increase gives the direction to take to decrease the error (cost) more quickly.
The method to calculate the gradient vector, which tells you which direction to take and how step the step is:

1. compute $\nabla C$
2. take step in $-\nabla C$ direction
3. repeat

Learning is finding the minimizing the weight function. Backprop is the algo used in gradient descent. Learning is 'just' finding the right weights and biases.
The cost of one training example is $C_0 = (a^L - y)^2$, the last activation is $a^L = \sigma(w_L a^{L-1} + b^L) = \sigma(z^L)$.
Curse of dimensionality refers to the apparent intractability of systematically searching through a high-dimensional space. As $n$ get bigger it gets harder and harder to sample all the boxes, with $n$ dimensions each allowing for $m$ states, we will have $m^n$ possible combinations.
Blessing of dimensionality

- In MLP approximation error decreases with the number of training samples error $O(1/\sqrt{N})$ and also with the number of hidden units error $O(1/M)$ and unlike other systems, eg polynomials this is independent of the input size and avoid the curse of dimensionality problem.

- From these results we can build bounds, for example

$$N > O(Mp/\epsilon)$$

where $N$ is the number of samples, $M$ the hidden nodes, $p$ input dimension ($Mp$ number of parameters) and $\epsilon$ the desired approximation error.

- More layers is better and do not harm
Bias–variance tradeoff is the problem of simultaneously minimizing two sources of error in the estimand. The bias-variance decomposition:

\[ MSE = E((\hat{\theta} - \theta)^2) = E(\hat{\theta} - \theta)^2 + Var(\hat{\theta}) = (Bias(\theta))^2 + Var(\theta) \]  

The bias/variance trade off in deep learning is not exactly a trade off it can be tackle algorithmically
Bias variance trade off

Table: Bias variance

<table>
<thead>
<tr>
<th>high var</th>
<th>high bias</th>
<th>high bias and var</th>
<th>low bias and var</th>
</tr>
</thead>
<tbody>
<tr>
<td>2%</td>
<td>15%</td>
<td>15%</td>
<td>0.5%</td>
</tr>
<tr>
<td>11%</td>
<td>15%</td>
<td>30%</td>
<td>1%</td>
</tr>
</tbody>
</table>

you don’t have the dialectical tension one thing or the other but in the table we have 4 cases rather than a trade off and luckily we can take action that fit every case.
• A bigger network will improve your fitting without hurting the variance problem, with the caveat that you regularize properly.

• Before we couldn’t make better one without hurting the other, now we can get both better.
Ensemble models

- Idea: you don’t want an organization with all the same (’good’) you may want to introduce variability
- Decision trees are grown by introducing a random element, eg at each node choose randomly the features to split the node
- Random forest (randomly constructed trees), each voting for a class, Bagging: boosting + aggregation.
- Great predictors but interpretability is obscured by the complexity of the model - accuracy generally requires more complex prediction methods.
Computational Topology

- Topology is concerned with the properties of space that are preserved under continuous deformations: stretching, crumpling and bending, but not tearing or gluing.

- Why to use topology over Big data?
  - It studies the invariants of continuous formations of the shape of data -resistant to threshold selection problem-
  - It allows measures of shape (clumps, holes and voids) which are invariant across scales.

Jaime Gómez-Ramírez
Persistent homology

Edges in a graph capture dyadic relationships.

Graphs can’t capture high order relationships but simplicial complex can

A simplicial complex is a generalized graph consisting on vertices, edges, triangles and simplices of higher dimension glue together.
Persistent homology

\[ C_0(X) = \langle v_1, v_2, v_3, v_4 \rangle, \]
\[ C_1(X) = \langle e_1, e_2, e_3, e_4, e_5 \rangle, \]
\[ C_2(X) = \sigma_1 \]

- boundary operator
  \[ \rho : C_1(X) \to C_0(X), \rho_2 : C_2(X) \to C_1(X) \]
  when applied to an edge it yields a difference of vertices, higher order operator to act on triangles (2-simplices),

- Loop is when we have
  \[ \rho_1(e_1 + e_2 + e_3) = 0 = \rho_1(e_1 + e_5 + e_4) \]
  both loops are in the kernel of \( \rho_1 \),

| Ker(\rho_1) = \{ x \in C_1(X), \rho_1(x) = 0 \} |
e₁ + e₂ + e₃ is obtained as the image of the triangle σ₁ under the map ρ₂, whereas is not the image of a triangle, in other words, $\text{Im}(\rho₂) = \{y \in C₁, \exists x \in C₂(X), ρ₂(x) = y\}$ then $e₁ + e₂ + e₃ \in \text{Im}(\rho₂)$ and $e₁ + e₅ + e₄ \notin \text{Im}(\rho₂)$.

The 1-D homology is the quotient space $H₁(X) = [\text{Ker}(\rho₁)/\text{Im}(\rho₂)]$

$$Hᵢ(X) = \frac{\text{Ker}(\rhoᵢ)}{\text{Im}(\rhoᵢ₊₁)}$$
Exploring the alpha desynchronization hypothesis in resting state networks with intracranial electroencephalography and wiring cost estimates

Jaime Gómez-Ramírez, Shelagh Freedman, Diego Mateos, José Luis Pérez Velázquez & Taufik A. Vallante

Figure 1 shows the results for group-level analysis of the covariance and the precision matrices for both groups using a threshold. The DMN nodes in MNI space are: Posterior Cingulate Cortex (B.16, 16), Left Temporoparietal junction (46, 46, 32), Right Temporoparietal junction (46, 46, 32) and Medial Prefrontal Cortex (1, 50, 5).

The connection between the mPFC and the PCC found in the control group in both covariance and precision matrices is not present in the converted group. Now, rather than using one threshold, we define the n-dimensional filtration of a simplicial complex as a nested sequence of simplicial complexes. Figure 2 shows the filtration of 4-simplices (the DMN) and filtration steps. Higher dimensional simplices appear. The persistent classes of a filtration of simplicial complexes can be visualized with barcodes.
Conclusions

- With enough imagination a classifier (regression) can be useful to solve a large number of problems.
- Deep learning works because there is structure in the world but we don’t know why because we don’t know anything about the initial conditions.

> laws of nature are precise beyond anything reasonable; we know virtually nothing about the initial conditions (Wigner)

- There are other ways to reduce complexity in big data while preserving maximal intrinsic information - computational topology.
- Occam’s dilemma (*lex parsimoniae*): accuracy generally requires more complex prediction methods, simple and interpretable functions do not make the most accurate predictions.
- The curse of dimensionality can be a blessing.
Thanks!