ARTIFICIAL INTELLIGENCE WITH NEURAL NETWORKS NOBEL PRIZES IN PHYSICS AND CHEMISTRY 2024

INTELIGENCIA ARTIFICIAL CON REDES NEURONALES PREMIOS NOBEL DE FÍSICA Y QUÍMICA 2024

Jorge E. Mahecha-Gómez*

Universidad de Antioquia UdeA, Facultad de Ciencias Exactas y Naturales, Instituto de Física; Calle 70 No. 52-21, Medellín-Colombia.

Abstract

John Joseph Hopfield began his career studying excitons in condensed matter physics, but his most important contributions were in the physics of computation and information, including his 1982 work on neural networks. Geoffrey Hinton, known as the "godfather" of artificial intelligence, laid the foundations for deep neural networks and developed the "backpropagation" method in 1986. These advances, along with Hopfield networks and the "Boltzmann machine", constitute the beginning of artificial intelligence.

David Baker is a pioneer in the design and prediction of three-dimensional protein structures, while Demis Hassabis has applied artificial intelligence to neuroscience. John Michael Jumper has investigated the use of AI to simulate protein folding and dynamics.

Hopfield and Hinton received the 2024 Nobel Prize in Physics, and Baker, Hassabis and Jumper received the Nobel Prize in Chemistry, sparking debates on interdisciplinarity and academic degrees in the sciences.

Keywords: Hopfield network, synaptic weight adjustment, deep learning, computational protein design, AlphaFold.

* jorge.mahecha@udea.edu.co doi: https://doi.org/10.15446/mo.n70.118564

Resumen

John Joseph Hopfield inició su carrera estudiando excitones en física de la materia condensada, pero sus aportes más relevantes fueron en física de la computación e información, destacando su trabajo de 1982 sobre redes neuronales. Geoffrey Hinton, conocido como el "padrino" de la inteligencia artificial, sentó las bases de las redes neuronales profundas y desarrolló el método de "backpropagation" en 1986. Estos avances, junto con las redes de Hopfield y la "máquina de Boltzmann", constituyen el inicio de la inteligencia artificial.

David Baker es pionero en el diseño y predicción de estructuras tridimensionales de proteínas, mientras que Demis Hassabis ha aplicado la inteligencia artificial a la neurociencia. John Michael Jumper ha investigado el uso de la IA para simular el plegamiento y dinámica de proteínas.

Hopfield y Hinton recibieron el Nobel de Física 2024, y Baker, Hassabis y Jumper el de Química, generando debates sobre la interdisciplinariedad y los títulos académicos en las ciencias.

Palabras clave: red de Hopfield, ajuste de pesos sinápticos, aprendizaje profundo, diseño computacional de proteínas, AlphaFold.

1. Introduction

1.1. The linear interpolation problem

Let's consider that (x_i, y_i), $i = 1, 2, ...n$, are n pairs of data which are visually located in the vecinity of a straight line in the $x - y$ plane. The problem is to find the optimal straight line among the possible straight lines. The equation of the searched line is $y = wx + b$; then, **optimal values** of w and b must be found. For this purpose, the "cost function" is defined

$$Q = \sum_{i=1}^{n} [y_i - (wx_i + b)]^2 = \sum_{i=1}^{n} (y_i^2 + w^2 x_i^2 + b^2 - 2wx_i y_i - 2by_i + 2wbx_i)$$

and the parameters w and b are required to be such that Q is **minimum**,

$$\frac{\partial Q}{\partial w} = 0, \quad \frac{\partial Q}{\partial b} = 0.$$

If a two-dimensional vector $\boldsymbol{u}$ is defined with components w and b, $\boldsymbol{u}^T = \{u_0, u_1\} = \{b, w\}$, the minimization of the cost function can be written in terms of the **gradient** with respect to the "weight vector" $\boldsymbol{u}$,

$$\boldsymbol{\nabla}_{\boldsymbol{u}} Q = \boldsymbol{0}.$$

The arithmetic mean of a random variable q is defined by $\overline{q} = (1/n)\sum_{i=1}^{n} q_i$. Then, the cost function can be written in the form

$$Q = Q_0 - \boldsymbol{H}^T \boldsymbol{u} + \boldsymbol{u}^T \underline{I} \boldsymbol{u},$$

or in terms of components of vectors $\boldsymbol{u}$, $\boldsymbol{H}$, and symmetric matrix $\underline{I}$,

$$Q = Q_0 - \sum_{i=0}^{1} H_i u_i + \sum_{i=0}^{1}\sum_{j=0}^{1} I_{ij} u_i u_j, \tag{1}$$

where

$$Q_0 = n\overline{y^2}, \ \boldsymbol{u} = \begin{pmatrix} b \\ \\ w \end{pmatrix}, \ \boldsymbol{H} = \begin{pmatrix} 2n\overline{y} \\ \\ 2n\overline{x \cdot y} \end{pmatrix}, \ \underline{I} = \begin{pmatrix} n & n\overline{x} \\ \\ n\overline{x} & n\overline{x^2} \end{pmatrix}.$$

The H_i and I_{ij} are depending on the data, which are random, then they are **random quantities**, which can assumed obey Gaussian distribution. The H_i are quadratically dependent on the "x'' and "y'', and the I_{ij} is quadratically dependent on the "x''.
The condition for minimum is

$$\boldsymbol{\nabla}_{\boldsymbol{u}} Q = -\boldsymbol{H} + 2\underline{I}\boldsymbol{u} = \boldsymbol{0}.$$

The two equations, after dividing by $2n$, can be written as,

$$-\overline{y} + b + w\overline{x} = 0, \quad -\overline{x \cdot y} + b\overline{x} + w\overline{x^2} = 0.$$

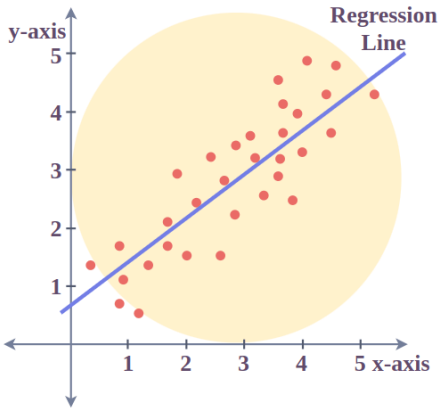

FIGURE 1. The regression line. Google image

n is the number of points on the graph, $(\overline{x}, \overline{y})$ is the coordinate of the "center of gravity" of the data set and $\overline{\ }$ denotes the arithmetic mean. The unknowns w and b, and their errors can be easily calculated.

If each red dot in Fig. 1 represents a mass of equal magnitude, the blue line were a rigid rod, each mass were attached to the blue rod with a massless perpendicular rod, and the gravity field acted perpendicular to the plane of the graph, then the system would remain static if the blue line is a least squares line, otherwise it would rotate.

1.2. The Ising model and the interpolation problem

The Ising model has analogy with the optimization problems previously discussed. It describes a system of interacting spins (typically is used $S_i = \pm 1$), with neighbor interactions J and an external magnetic field h, with the following Hamiltonian,

$$H = -J \sum_{i=1}^{N-1} S_i S_{i+1} - h \sum_{i=1}^{N} S_i.$$

Depending on J, spins tend to align (ferromagnetic, $J > 0$) or anti-align (antiferromagnetic, $J < 0$).

The external magnetic field favors alignment if $h > 0$ or antialignment if $h < 0$. Finding the ground state involves **minimizing the system's energy**, which becomes computationally challenging due to the exponential growth of configurations (as 2^N).

The model becomes more complex in disordered systems, where interactions and fields are site dependent, as in spin glasses. In this case the Hamiltonian is,

$$H = -\sum_{i=1}^{N}\sum_{j=1}^{N} J_{i,j} S_i S_j - \sum_{i=1}^{N} h_i S_i. \tag{2}$$

The Sherrington-Kirkpatrick (SK) of 1978 extension introduces random interactions and models disordered magnetic materials, as well as complex optimization problems and biological systems. The SK model's energy landscape features multiple local minima, making it difficult to find the global minimum and highlighting the concept of frustration.

There is a formal analogy between the SK model and multidimensional optimization problems in artificial intelligence. Both aim to minimize a cost function in environments with complex, disordered interactions. Methods inspired by statistical mechanics, such as simulated annealing or quantum annealing, are used to solve these problems. The SK model serves as a conceptual framework for understanding and addressing combinatorial optimization problems.

If we compare Eq.(1) and Eq.(2), we find a formal analogy. Obviously, in the former case $N = 2$, but it can easily generalized to the multi dimensional case when the the straight line model $y = wx + b$, in the 2-dimensional plane is changed to a hyperplane in an N-dimensional space,

$$w_1 x_1 + w_2 x_2 + \cdots + w_{N-1} x_{N-1} + b = x_N, \quad \mathbf{w}^T \mathbf{x} + b = x_N,$$

where $\mathbf{x^T} = \{x_1, x_2, \ldots x_{N-1}\}$, $\mathbf{w^T} = \{w_1, w_2, \ldots w_{N-1}\}$, and b is the bias term. The cost function formula is easly obtained from Eq.(1) by changing the upper limit of the sums.

The multidimensional optimization model with decision variables x and "weight" and "bias" parameters w_i, b_i is analogous to the SK model with spin variables S_i and random couplings J_{ij} and random external fields h_i. In both models, the objective is to minimize a cost function or energy, with the optimization landscape being shaped by frustration and disorder. The optimization problem in the multidimensional model can be approached using methods inspired by statistical mechanics, such as Simulated Annealing or Quantum Annealing, which are derived from the dynamics of spin glasses like the SK model. In essence, the SK model serves as a framework for understanding optimization problems with complex, **random interactions** and is used to develop algorithms for finding approximate solutions to difficult combinatorial problems in AI.

1.3. A simple model of biological neurons

Biological neurons have three main components: dendrites, soma, and axon. Dendrites receive signals from other neurons via synapses. The soma processes these inputs, summing excitatory and inhibitory signals. If the total exceeds an activation threshold, an action potential is generated and transmitted along the axon to other neurons.

The first neuron model, proposed by McCulloch and Pitts (1943) [1], described neurons as binary threshold units, forming a basis for neural networks. In 1949, Hebb introduced the Hebbian learning rule: Neurons that fire together, wire together. This strengthens synaptic connections when two neurons activate simultaneously, supporting learning and memory.

The Hebbian rule, $\Delta w = \eta \cdot x_i \cdot x_j$, adjusts synaptic weights based on neuron activations and has influenced modern theories of synaptic plasticity and deep learning. Key elements like activation functions were inherited by neural network models through the perceptron [2].

1.4. The Nobel prize of physics and chemistry 2024 [3, 4]

John Hopfield (`https://pni.princeton.edu/people/john-j-hopfield`): Developed Hopfield neural networks for

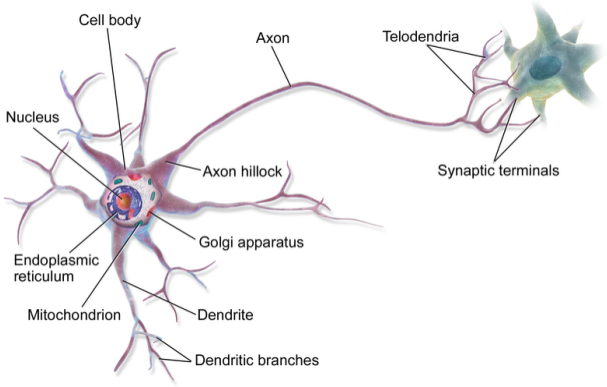

FIGURE 2. Neurons contain a nucleus and mitochondria, they also have dendrites and axons. Google image.

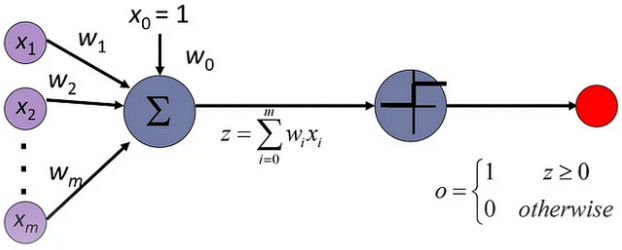

FIGURE 3. Model of perceptron with binary step activation function. Google image

pattern storage/retrieval, modeled associative memory, connected statistical physics with neural networks, advanced machine learning foundations, and proposed energy-based models for neural network optimization [2, 5].

Geoffrey Hinton (`http://www.cs.toronto.edu/~hinton/`): Pioneered deep learning, advanced backpropagation for training neural networks, developed Restricted Boltzmann Machines and Deep Belief Networks, and highlighted AI's societal impact [6, 7].

David Baker (`https://www.ipd.uw.edu/`): Innovated computational protein design, created the Rosetta algorithm for protein structure prediction, advanced biochemistry. He has successfully created entirely new types of proteins using amino

acids. This innovative approach allows for the design of proteins that do not exist in nature, potentially leading to significant advancements in medicine, vaccines, and nanotechnology. His work emphasizes the importance of understanding the relationship between a protein's amino acid sequence and its 3D structure, a concept discussed previously. By manipulating this relationship, Baker's designs can lead to proteins tailored for specific therapeutic roles or novel materials.

Demis Hassabis (`https://deepmind.google/`): Co-created AlphaFold for protein structure prediction, applied AI to molecular biology, connected biology with computer science, and drove scientific innovation in biomedicine.

John Michael Jumper (`https://scholar.google.com/citations?user=a5goOh8AAAAJ`): Developed precise AI models for protein structure prediction, collaborated on computational methods in biology, solved complex biological problems, and advanced biotechnology and medicine.

Hassabis and Jumper were recognized for their development of AlphaFold2, an artificial intelligence model that predicts the complex structures of proteins from their amino acid sequences.

2. Neural Networks and the Hopfield Model

2.1. Perceptron

The perceptron models a neuron. It receives D inputs, x_1, ... x_D, checks the result and produces an output. It performs a weighted sum of the inputs, checks the result and depending on it produces an output. It is used to classify data that are grouped into linearly separable classes (the simplest is binary classification). Each input has an associated weight w_1, ... w_D. The sum is

$$s = \sum_{i=1}^{D} w_i x_i.$$

The output, y, is determined by a certain **activation function**, $f(s)$, $y = f(s)$ (See Fig.4).

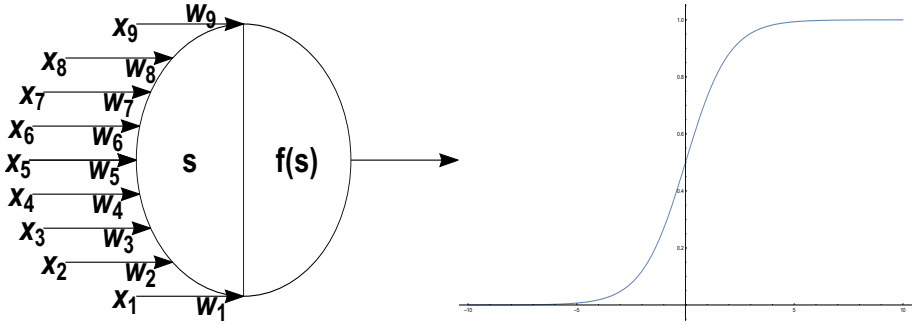

FIGURE 4. Single layer perceptron and sigmoid activation function.

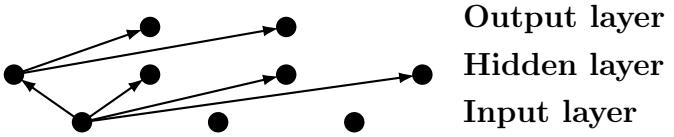

FIGURE 5. Multi layer perceptron.

Hopfield considered a SK-like Hamiltonian with $h = 0$ in Eq.(2) of the form

$$H_N = -\frac{1}{2}\sum_{i\neq j} J_{ij} S_i S_j,$$

where

$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu,$$

the $S_1, S_2, ...S_N$ are Ising spins $S_i = \pm 1$, and the ξ_i^μ are independent **random variables**. Initially, Hopfield used the model to implement an "associative memory" on a neural network [2].

The Hopfield network, a type of recurrent neural network (RNN), is an AI algorithm inspired by the SK model. It uses binary neurons (values $S_{ik} = \pm 1$) and a feedback loop to solve optimization problems, Given noisy inputs, the network recalls stored patterns by converging to the closest match.

A Hopfield Network, developed by John Hopfield in the early 1980s, is a type of recurrent neural network that serves as a content-addressable memory system. It consists of a single layer of interconnected neurons, where each neuron is connected to every other neuron except itself. The connections are bidirectional and symmetric, meaning the weight of the connection from neuron i to neuron j is the same as from j to i.

The Hopfield network operates similarly to spin glass systems in the SK model. Its energy function is:

$$H = -\sum_{i<j} J_{ij} S_i S_j,$$

where J_{ij} is the weight between neurons, and S_i is the state of the i-th neuron. The system minimizes this energy to find optimal solutions.

Hopfield networks are used to solve problems such as associative memory or constraint satisfaction problems. The Hopfield model can be framed as a spin glass problem, where the network of neurons interacts in a manner similar to the spins in the SK model.

2.2. Hopfield memory

Developed by John Hopfield in the 1980s, this network is a single-layer, fully interconnected system where connections are symmetric and bidirectional. Patterns are stored via a Hebbian learning rule.

2.3. Associative memory

Associative memory retrieves information based on similarity rather than specific addresses. It enables the recall of full patterns from partial or noisy inputs. Hopfield networks exemplify this capability but are just one implementation among broader systems that achieve retrieval based on content similarity.

3. Hinton

3.1. Recurrent neural networks

Recurrent Neural Networks (RNNs) are a class of neural networks designed for processing sequential data. They are particularly effective for tasks where context and order are important, such as time series analysis, natural language processing, and speech recognition.

In an RNN, the network maintains a hidden state that captures information about previous inputs in the sequence. The hidden state h_t at time step t is computed based on the current input x_t and the hidden state from the previous time step h_{t-1}.
The hidden state h_t at time step t can be calculated using the following formula:

$$h_t = \sigma\left(\underline{U}x_t + \underline{W}h_{t-1} + b\right),$$

where h_t is the current hidden state, x_t is the input at time step t, $\underline{U}$ is the weight matrix for the input layer. $\underline{W}$ is the weight matrix for the hidden layer (recurrent connections). b is the bias term, and σ is an activation function (commonly tanh or ReLU).

The output y_t at time step t is derived from the hidden state,

$$y_t = O(\underline{V}h_t + c),$$

where y_t is the output at time step t, $\underline{V}$ is the weight matrix for output connections, c is a bias term for the output layer, and O represents an activation function applied to produce the final output (e.g., softmax for classification tasks).

Training RNNs typically involves backpropagation through time (BPTT), where gradients are computed for each weight matrix based on a loss function L and chain rule for derivatives.

3.2. Generative models

Generative models are a key area of AI focused on learning data patterns to create new, similar data. Unlike discriminative models

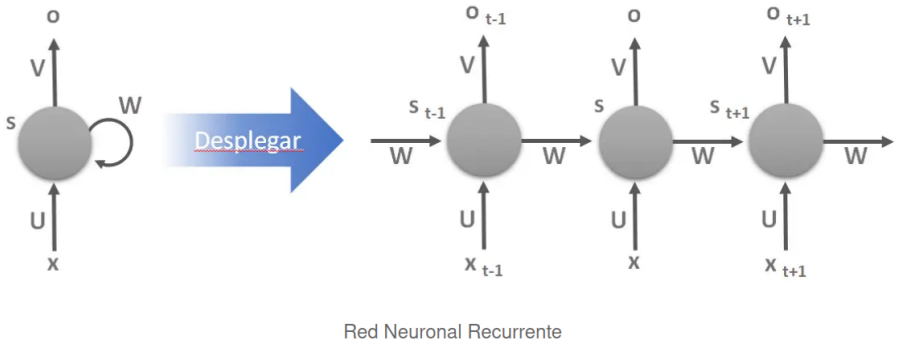

FIGURE 6. Recurrent neural network RNN. Google image.

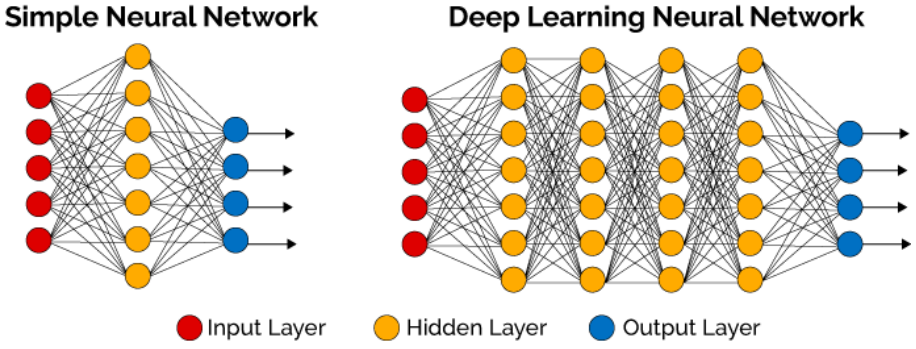

FIGURE 7. Deep neural networks. Google image.

that classify data, generative models aim to understand how data is generated.

They capture the data distribution $P(X, Y)$ (or $P(X)$ for unsupervised tasks), of the input data X and any associated labels Y, to produce realistic outputs. Learning relies on patterns in large datasets, often through unsupervised or semi-supervised techniques.

Applications include creating art, text, music, and synthetic datasets, enhancing creativity, automation, and AI interactions. Generative models play a crucial role in advancing AI systems and their potential impact continues to grow.

3.3. Neural network with hidden layers

3.3.1. Boltzmann machines

A Boltzmann Machine is a type of stochastic recurrent neural network that can learn a probability distribution over its set of inputs. It consists of visible units (which represent the input data) and hidden units (which capture the dependencies between the visible units). The energy of a configuration in a Boltzmann Machine is defined as:

$$E(\mathbf{s}) = -\sum_{i \in S} b_i s_i - \sum_{i,j \in S, j>i} w_{ij} s_i s_j,$$

where $E(\boldsymbol{s})$ is the energy of the state vector $\boldsymbol{s}$, b_i is the bias of the $i-th$ visible unit, w_{ij} is the weight between units i and j, s_i is the state of the $i-th$ unit (0 or 1).
A Restricted Boltzmann Machine (RBM) is a special case of a Boltzmann Machine where connections between hidden units are not allowed (See Fig.8). This restriction simplifies training and makes RBMs particularly useful for unsupervised learning tasks [6].

In an RBM, there are two layers: Visible layer which represents observed data, hidden Layer which captures features from the visible layer.

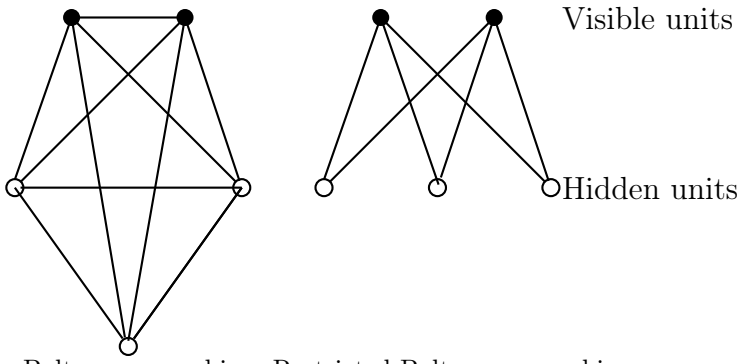

FIGURE 8. Boltzmann machine. Restricted Boltzmann machine.

3.3.2. Neural networks

Fig.7 shows neural networks. They have at least one input layer and one output layer. Data preparation for the input layer requires converting the problem features (measurements, words, sound, images, ...) into data. In a neural network, the embedding layer is responsible for converting discrete tokens (like words) into continuous vector representations. Word2Vec or GloVe for textual data and convolutional neural networks (CNN) for images are examples of tools used for embedding.
In NLP, tokenization is the process of breaking down a text into smaller units called tokens. These tokens can be words, phrases, or subwords, depending on the granularity required for the specific NLP task.

3.3.3. Convolutional neural networks

Convolutional neural networks (CNNs) are similar to regular neural networks but are designed for inputs with redundant or non-essential details. For example, images often contain information that can be compressed without losing much meaning. A 2D image can be represented with a number of bits equal to length×width×depth, where length and width are the number of pixels, and depth represents attributes like color or shades of gray. The convolution operation is fundamental to CNNs and is defined as (See Fig.9):

$$(f \circledast g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau.$$

3.3.4. Feedforward networks

A feedforward neural network is an artificial neural network where connections between nodes do not form cycles. Information moves in only one direction (from the input layer, through any hidden layers, and finally to the output layer). This structure differentiates FNNs from recurrent neural networks (RNNs), which allow for cycles and feedback loops.

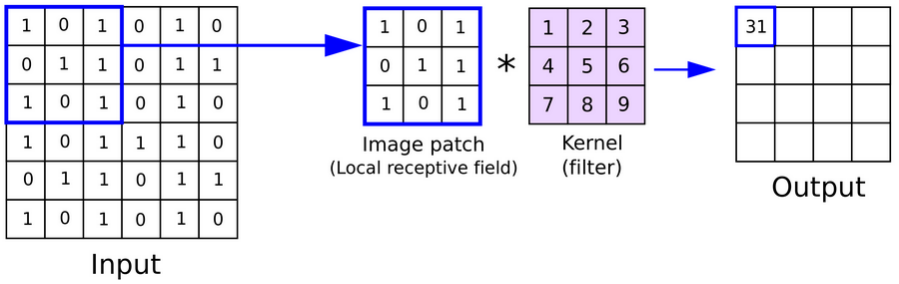

FIGURE 9. Convolutional neural network. Google image.

The input layer consists of neurons that receive the initial data. Each neuron in this layer corresponds to a feature of the input data. One or more hidden layers are present between the input and output layers. These layers perform computations and learn complex patterns in the data through weighted connections and activation functions. The output layer produces the final output of the network, which can represent classifications in a classification task or continuous values in a regression task.

In a feedforward network, each neuron computes a weighted sum of its inputs and applies an activation function to introduce non-linearity.

The operation of a neuron in a feedforward network can be mathematically represented as:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right),$$

where y is the output of the neuron, f is the activation function (e.g., ReLU, sigmoid), w_i are the weights associated with each input x_i, and b is the bias term.

Feedforward networks are typically trained using backpropagation, a method that calculates gradients of the loss function with respect to each weight by applying the chain rule. This allows for efficient weight updates to minimize prediction error.

Feedforward neural networks are widely used for various tasks, including: Classification (identifying categories for input data),

regression (predicting continuous outcomes), pattern recognition (recognizing patterns in data such as images or signals).

3.4. Attention

Neural networks without attention typically process input data in a sequential manner, where each layer of the network extracts features from the input and passes them to the next layer. In traditional architectures like feedforward neural networks or recurrent neural networks (RNNs), the entire input sequence is processed uniformly without differentiating the importance of different parts of the input [8].

Every part of the input is treated equally, which can lead to inefficiencies, especially in tasks involving long sequences where only certain parts may be relevant. In RNNs, for example, the context is limited to a fixed number of previous states, making it difficult to capture long-range dependencies effectively.

Neural networks with attention mechanisms enhance the model's ability to focus on specific parts of the input that are more relevant to the task at hand. Attention mechanisms dynamically assign weights to different elements of the input, allowing the model to prioritize certain inputs over others based on their relevance. some key characteristics of attention mechanism are: The model can "attend" to important parts of the input while ignoring less relevant information. This is particularly useful in tasks like natural language processing (NLP) and computer vision. Attention mechanisms allow models to maintain a broader context by considering relationships between all elements in the input sequence rather than just a fixed window. Thus, are appropriate in complex tasks involving long-range dependencies.

3.5. Transformers

Transformers are a specific type of neural network architecture that utilizes attention mechanisms extensively, without need of recurrence and convolutions. Introduced in the paper "Attention is All You Need" [8], transformers have revolutionized NLP and other fields by enabling parallel processing of input data.

Transformers use self-attention to compute attention scores for each element in relation to all other elements in the sequence. This allows for capturing dependencies regardless of their distance in the sequence. The self-attention mechanism can be mathematically expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where: Q (query), K (key), and V (value) are matrices derived from the input embeddings. d_k is the dimensionality of the keys.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

Since transformers do not inherently understand sequence order due to their parallel processing nature, positional encodings are added to input embeddings to provide information about token positions.

While traditional neural networks process inputs uniformly without differentiating their importance, attention mechanisms allow models to focus selectively on relevant parts of the data. Transformers take this a step further by employing self-attention and parallel processing, enabling them to capture complex relationships within data efficiently. This has led to significant advancements in various AI applications, particularly in natural language processing (NLP and LLM), computer vision, and protein structure.

4. The Nobel Prize of Chemistry 2024

Their work addresses fundamental challenges in understanding proteins, which are essential biological molecules that perform a vast array of functions in living organisms. This breakthrough addresses a long-standing challenge in molecular biology, enabling researchers to accurately determine how proteins fold into their functional forms.

4.1. Protein folding

The ability to predict protein structures accurately is crucial for understanding their biological functions and interactions, which can significantly impact drug development and disease treatment strategies.

The ability to design new proteins could result in novel therapies and vaccines tailored to combat diseases more effectively. For instance, engineered proteins could serve as targeted treatments for specific conditions or enhance the efficacy of existing medications. The insights gained from accurate protein structure predictions can facilitate the development of new materials with unique properties or functions, expanding the horizons of biotechnology and materials science. Both Baker's computational approaches and Hassabis and Jumper's AI-driven methods exemplify the interdisciplinary and collaborative nature of modern scientific research.

The relationship between a protein's amino acid sequence and its final folded structure is encapsulated in Anfinsen's dogma, which posits that the sequence contains all the necessary information for folding. This hypothesis suggests that under physiological conditions, proteins will fold into their most thermodynamically stable conformation.

One of the central issues in this field is Levinthal's paradox, which highlights that if a protein were to sample all possible conformations to find its native state, it would take an impractically long time (up to 10^{10} years for a typical protein). However, proteins fold in much shorter timescales (seconds to minutes), suggesting that they utilize efficient pathways or mechanisms to reach their native state.

The kinetics of protein folding can be modeled using various theories: Zipping and assembly mechanism describes how proteins form smaller loops and turns quickly before assembling into larger structures. All-or-none phase transition suggests that proteins fold through distinct states rather than continuously transitioning between them.

4.2. Components of proteine chains

A polypeptide chain is a linear sequence of amino acids linked by peptide bonds, formed through dehydration synthesis. Polypeptides serve as building blocks of proteins, which provide structural support, catalyze biochemical reactions, and regulate cellular processes. The sequence and length (usually >50 amino acids for proteins) determine biological functions. Ribosomes, molecular machines composed of rRNA and proteins, synthesize polypeptides during translation by decoding mRNA and linking amino acids via tRNA.
Amino acids, the building blocks of proteins, consist of an amino group (–NH_2), carboxyl group (–COOH), alpha carbon (Cα), and an R group that determines their properties. There are 20 standard amino acids, classified as essential (e.g., histidine, lysine), nonessential (e.g., alanine, serine), and conditionally essential (e.g., arginine, cysteine).

4.3. The machine learning problem

AlphaFold2 is a program for protein structure prediction. It has input given by amino acid sequence, multiple sequence alignments (MSAs), and evolutionary data. An Evoformer Module which processes MSAs to capture local and global residue relationships using attention mechanisms [8]. An Structure Module which generates 3D protein structures, refining predictions iteratively. And a Validation which achieved unprecedented accuracy (RMSD $\approx$0.8 Å) in CASP14.

The Transformer algorithm is a deep neural network architecture that has revolutionized the field of natural language processing, and AlphaFold uses it with great success to predict protein structure.
AlphaFold's Evoformer module is the heart of the system and is where the Transformer algorithm is applied. This algorithm is used to process amino acid sequence information from the target protein and from sequence and structure databases. The Transformer allows AlphaFold to analyze the complex relationships between amino acids in the protein sequence and how these relationships

are affected by evolutionary information contained in homologous protein sequences.
The Transformer allows AlphaFold to identify patterns and long-distance relationships between amino acids, which are crucial for determining the three-dimensional structure of the protein.
Unlike traditional recurrent neural networks, the Transformer can process the amino acid sequence in parallel, making it much more computationally efficient. This parallel processing capability is essential for AlphaFold, as protein sequences can be very long.
The Transformer does not replace biochemical knowledge, although it is a powerful tool, AlphaFold does not rely solely on machine learning. The program also incorporates biochemical and physical principles into its model, allowing it to make more accurate and realistic predictions. In short, the Transformer algorithm is a fundamental part of AlphaFold's success in predicting protein structure. This algorithm allows AlphaFold to analyze amino acid sequences efficiently and accurately, identifying complex patterns that determine the three-dimensional structure of the protein.

5. Concluding Remarks

Physics studies matter, energy, space, time, and their interactions, aiming to uncover the laws of the universe from the smallest particles to the largest cosmic scales. It includes fields like Classical Physics, Astrophysics, Biophysics, and more, often integrating interdisciplinary approaches. Chemistry, on the other hand, focuses on the composition, structure, and transformations of matter, covering areas like Organic Chemistry, Biochemistry, and Environmental Chemistry.
Both fields require specialized knowledge and training, which is why a software engineer, for instance, is not typically prepared to work as a physicist or chemist without additional education. This distinction has led to criticism of the 2024 Nobel Prize in Physics and Chemistry being awarded to individuals without formal backgrounds in these fields. Critics argue this decision overlooks the traditional boundaries of academic disciplines and the specialized training they require.

Philosophical ideas, such as those of Wittgenstein, remind us that concepts like "laws of nature" are constructs of language and thought, not inherent truths of the world. Reification (from the Latin *res*, thing) occurs when we treat these constructs as independently existing realities.

Advances in artificial intelligence, like AlphaFold and ChatGPT, demonstrate how physics and chemistry concepts and models can be applied in new ways. AlphaFold predicts protein structures using neural networks, while ChatGPT uses transformers to process and generate text (GPT stands for Generative Pre-trained Transformer). These tools highlight how interdisciplinary approaches, especially in AI, can transform our understanding of complex systems. However, they also blur traditional disciplinary lines, sparking debate about the roles of expertise and innovation in science.

References

[1] W. S. McCulloch and W. Pitts, Bulletin of Mathematical Biophysics **5**, 115 (1943).

[2] B. Tirozzi, *Modelli matematici di reti neurali*, Sistemi di elaborazione cognitiva (CEDAM, 1995).

[3] The Royal Swedish Academy of Sciences, Scientific Background to the Nobel Prize in Physics 2024 (2024).

[4] The Royal Swedish Academy of Sciences, Scientific Background to the Nobel Prize in Chemistry 2024 (2024).

[5] J. J. Hopfield, Proceedings of the National Academy of Sciences of the United States of America **79**, 2254 (1982).

[6] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, Cognitive science **9**, 147 (1985).

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature **323**, 533 (1986).

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, K. L., and I. Polosukhin, in *Advances in Neural Information Processing Systems* (2017).