

Probability & Statistics – Xác Suất & Thống Kê

Nguyễn Quân Bá Hồng*

Ngày 26 tháng 10 năm 2024

Tóm tắt nội dung

This text is a part of the series *Some Topics in Advanced STEM & Beyond*:

URL: https://nqbh.github.io/advanced_STEM/.

Latest version:

- *Probability & Statistics – Xác Suất & Thống Kê*.

PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/probability_statistics/NQBH_probability_statistics.pdf.

TeX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/probability_statistics/NQBH_probability_statistics.tex.

Mục lục

1	Wikipedia	1
1.1	Wikipedia/universal approximation theorem	2
1.1.1	Setup	2
1.1.2	History	2
1.1.3	Arbitrary-width case	2
1.1.4	Arbitrary-depth case	3
1.1.5	Bounded depth & bounded width case	3
2	Probability – Xác Suất	4
3	Statistics – Thống Kê	4
4	Stochastic – Ngẫu Nhiên	4
5	Data Science (DS)	4
5.1	Wikipedia/Data Science	4
5.1.1	Foundations	4
5.1.2	Etymology	5
5.1.3	Data science & data analysis	5
5.1.4	Cloud computing for data science	5
5.1.5	Ethical consideration in data science	5
6	Deep Learning (DL)	5
7	Machine Learning (ML)	6
7.1	Wikipedia/Machine Learning	7
7.1.1	History	7
7.1.2	Relationships to other fields	7
7.1.3	Theory	8
7.1.4	Approaches	8
8	Artificial Intelligence (AI)	9
9	Miscellaneous	9
	Tài liệu	9

1 Wikipedia

Relationship among AL, ML, & DL: $DL \subset ML \subset AI$.

*A Scientist & Creative Artist Wannabe. E-mail: nguyenquanbahong@gmail.com. Bến Tre City, Việt Nam.

1.1 Wikipedia/universal approximation theorem

“In the mathematical theory of **artificial neural networks**, *universal approximation theorems* are theorems of the following form:

Theorem 1 (Universal approximation). *Given a family of neural networks, for each function f from a certain **function space**, there exists a sequence of neural networks ϕ_1, ϕ_2, \dots from the family, s.t. $\phi_n \rightarrow f$ as $n \rightarrow \infty$ according to some criterion, i.e., the family of neural networks is **dense** in the function space.*

The most popular version states that **feedforward networks** with non-polynomial **activation functions** are dense in the space of continuous functions between 2 **Euclidean spaces**, w.r.t. the **compact convergence** topology.

Universal approximation theorems are existence theorems: They simply state that there *exists* such a sequence $\phi_n \rightarrow f$, & do not provide any way to actually find such a sequence. They also do not guarantee any method, e.g., **backpropagation**, might actually find such a sequence. Any method for searching the space of neural networks, including backpropagation, might find a converging sequence, or not (i.e., the backpropagation might get stuck in a local optimum).

Universal approximation theorems are limit theorems: They simply state that for any f & a criteria of closeness $\epsilon > 0$, if there are *enough* neurons in a neural network, then there exists a neural network with that many neurons that does approximate f to within ϵ . There is no guarantee that any finite size, say, 10000 neurons, is enough.

1.1.1 Setup

Artificial neural networks are combinations of multiple simple mathematical functions that implement more complicated functions from (typically) real-valued vectors to real-valued vectors. The spaces of multivariate functions that can be implemented by a network are determined by the structure of the network, the set of simple functions, & its multiplicative parameters. A great deal of theoretical work has gone into characterizing these function spaces.

Most universal approximation theorems are in 1 of 2 classes. The 1st quantifies the approximation capabilities of neural networks with an arbitrary number of artificial neurons (“*arbitrary width*” case) & the 2nd focuses on the case with an arbitrary number of hidden layers, each containing a limited number of artificial neurons (“*arbitrary depth*” case). In addition to these 2 classes, there are also universal approximation theorems for neural networks with bounded number of hidden layers & a limited number of neurons in each layer (“*bounded depth & bounded width*” case).

1.1.2 History

Arbitrary width.

Arbitrary depth.

Bounded depth & bounded width.

Quantitative bounds.

Kolmogorov network.

Variants.

1.1.3 Arbitrary-width case

A spate of papers in the 1980s–1990s, from **George Cybenko** & Kurt Hornik etc., established several universal approximation theorems for arbitrary width & bounded depth. Most often quoted:

Theorem 2 (Universal approximation theorem). *Let $C(X, \mathbb{R}^m)$ denote the set of **continuous functions** from a subset X of a Euclidean \mathbb{R}^n space to a Euclidean space \mathbb{R}^m . Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ \mathbf{x})_i = \sigma(x_i)$, so $\sigma \circ \mathbf{x}$ denotes σ applied to each component of \mathbf{x} . Then σ is not polynomial iff $\forall m, n \in \mathbb{N}$, **compact** $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\epsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ s.t. $\sup_{\mathbf{x} \in K} \|f(\mathbf{x}) - g(\mathbf{x})\| < \epsilon$ where $g(\mathbf{x}) := C \cdot (\sigma \circ (A \cdot \mathbf{x} + b))$.*

Also, certain non-continuous activation functions can be used to approximate a sigmoid function, which then allows the above theorem to apply to those functions. E.g., the **step function** works. In particular, this shows that a **perceptron** network with a single infinitely wide hidden layer can approximate arbitrary functions.

Such an f can also be approximated by a network of greater depth by using the same construction for the 1st layer & approximating the identity function with later layers. **Proof sketch** has not specified how one might use a ramp function to approximate arbitrary functions in $C_0(\mathbb{R}^n, \mathbb{R})$. A sketch of the proof is that one can 1st construct flat bump functions, intersect them to obtain spherical bump functions that approximate the **Dirac delta function** δ , then use those to approximate arbitrary functions in $C_0(\mathbb{R}^n, \mathbb{R})$. The original proofs, e.g., the one by Cybenko, use methods from functional analysis, including the **Hahn–Banach** & **Riesz–Markov–Kakutani representation** theorems.

Notice also that the neural network is only required to approximate within a compact set K . The proof does not describe how the function would be extrapolated outside of the region.

The problem with polynomials may be removed by allowing the outputs of the hidden layers to be multiplied together (the “pi-sigma $\pi\sigma$ networks”), yielding the generalization:

Theorem 3 (Universal approximation theorem for pi-sigma $\pi\sigma$ networks). *With any nonconstant activation function, a 1-hidden-layer pi-sigma network is a universal approximator.*

1.1.4 Arbitrary-depth case

The “dual” versions of the theorem consider networks of bounded width & arbitrary depth. A variant of the universal approximation theorem was proved for the arbitrary depth case by Zhou Lu et al. in 2017. They showed that networks of width $n + 4$ with **ReLU** activation functions can approximate any **Lebesgue-integrable function** on n -dimensional input space w.r.t. **L^1 distance** if network depth is allowed to grow. It was also shown that if the width was $\leq n$, this general expressive power to approximate any Lebesgue integrable function was lost. ReLU networks with width $n + 1$ were sufficient to approximate any continuous function of n -dimensional input variables. The following refinement, specifies the optimal minimum width for which such an approximation is possible & is due to.

Theorem 4 (Universal approximation theorem (L_1 distance, ReLU activation, arbitrary depth, minimal width)). *For any **Bochner–Lebesgue p -integrable** function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, \mathcal{E} any $\varepsilon > 0$, there exists a **fully connected ReLU** network F of width exactly $d_m := \max\{n + 1, m\}$ satisfying $\int_{\mathbb{R}^n} \|f(\mathbf{x}) - F(\mathbf{x})\|^p d\mathbf{x} < \varepsilon$. Moreover, there exists a function $f \in L^p(\mathbb{R}^n, \mathbb{R}^m)$ \mathcal{E} some $\varepsilon > 0$, for which there is no **fully connected ReLU** network of width $< d_m := \max\{n + 1, m\}$ satisfying the above approximation bound.*

Remark 1. *If the activation is replaced by leaky-ReLU, \mathcal{E} the input is restricted in a compact domain, then the exact minimum width is $d_m := \max\{m, n, 2\}$.*

Quantitative refinement: In the case where $f : [0, 1]^n \rightarrow \mathbb{R}$, i.e., $m = 1$, & σ is the **ReLU activation function**, the exact depth & width for a ReLU network to achieve ε error is also known. If, moreover, the target function f is smooth, then the required number of layer & their width can be exponentially smaller. Even if f is not smooth, the curse of dimensionality can be broken if f admits additional “compositional structure”.

Together, the central result of

- KIDGER, PATRICK; LYONS, TERRY (July 2020). *Universal Approximation with Deep Narrow Networks*. Conference on Learning Theory. arXiv:1905.08539

yields the following universal approximation theorem for networks with bounded width:

Theorem 5 (Universal approximation theorem (uniform non-affine activation, arbitrary depth, constrained width)). *Let \mathcal{X} be a **compact subset** of \mathbb{R}^d . Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be any non-affine continuous function which is **continuously differentiable** at at least 1 point, with nonzero derivative at that point. Let $\mathcal{N}_{d,D;d+D+2}^\sigma$ denote the space of feed-forward neural networks with d input neurons, D output neurons, \mathcal{E} an arbitrary number of hidden layers each with $d + D + 2$ neurons, s.t. every hidden neuron has activation function σ \mathcal{E} every output neuron has the **identity** as its activation function, with input layer ϕ \mathcal{E} output layer ρ . Then given any $\varepsilon > 0$ \mathcal{E} any $f \in C(\mathcal{X}, \mathbb{R}^D)$, there exists $\hat{f} \in \mathcal{N}_{d,D;d+D+2}^\sigma$ s.t. $\sup_{\mathbf{x} \in \mathcal{X}} \|\hat{f}(\mathbf{x}) - f(\mathbf{x})\| < \varepsilon$, i.e., \mathcal{N} is **dense** in $C(\mathcal{X}; \mathbb{R}^D)$ w.r.t. the topology of **uniform convergence**.*

Quantitative refinement: The number of layers & width of each layer required to approximate f to ε precision known; moreover, the result hold true when \mathcal{X} & \mathbb{R}^D are replaced with any non-positively curved **Riemannian manifold**.

Certain necessary conditions for the bounded width, arbitrary depth case have been established, but there is still a gap between the known sufficient & necessary conditions.

1.1.5 Bounded depth & bounded width case

The 1st result on approximation capabilities of neural networks with bounded number of layers, each containing a limited number of artificial neurons was obtained by Maiorov & Pinkus. Their remarkable result revealed that such networks can be universal approximators & for achieving this property 2 hidden layers are enough.

Theorem 6 (Universal approximation theorem). *There exists an activation σ which is analytic, strictly increasing \mathcal{E} sigmoidal \mathcal{E} has the following property: For any $f \in C([0, 1]^d)$ \mathcal{E} $\varepsilon > 0$ there exist constants $d_i, c_{ij}, \theta_{ij}, \gamma_i$, \mathcal{E} vectors $\mathbf{w}^{ij} \in \mathbb{R}^d$ for which*

$$\left| f(\mathbf{x}) - \sum_{i=1}^{6d+3} d_{i\sigma} \left(\sum_{j=1}^{3d} c_{ij} \sigma(\mathbf{w}^{ij} \cdot \mathbf{x} - \theta_{ij}) - \gamma_i \right) \right| < \varepsilon, \quad \forall \mathbf{x} = (x_1, \dots, x_d) \in [0, 1]^d.$$

This is an existence result: activation functions providing universal approximation property for bounded depth bounded width networks exist. Using certain algorithmic & computer programming techniques, Guliyev & Ismailov efficiently constructed such activation functions depending on a numerical parameter. The developed algorithm allows one to compute the activation functions at any point of the real axis instantly. For the algorithm & the corresponding computer code see

- GULIYEV, NAMIG; ISMAILOV, VUGAR (November 2018). “Approximation capability of two hidden layer feedforward neural networks with fixed weights”. Neurocomputing. 316: 262–269. arXiv:2101.09181. doi:10.1016/j.neucom.2018.07.075. S2CID 52285996.

The theoretical result can be formulated as follows.

Theorem 7 (Universal approximation theorem). [...]

” – [Wikipedia/universal approximation theorem](#)

2 Probability – Xác Suất

Community – Cộng đồng. ANDREY NIKOLAEVICH KOLMOGOROV.

Resources – Tài nguyên.

1. SIMON J. D. PRINCE. *Computer Vision: Models, Learning, & Inference*.

3 Statistics – Thống Kê

Community – Cộng đồng.

Resources – Tài nguyên.

- 1.

4 Stochastic – Ngẫu Nhiên

Community – Cộng đồng. CAROLINE GEIERSBACH, MICHAEL HINTERMÜLLER.

Resources – Tài nguyên.

- 1.

5 Data Science (DS)

Community – Cộng đồng.

Resources – Tài nguyên.

- 1.

5.1 Wikipedia/Data Science

“*Data science* is an **interdisciplinary** academic field that uses **statistics**, **scientific computing**, **scientific methods**, processing, **scientific visualization**, **algorithms** & systems to extract or extrapolate **knowledge** & insights from potentially noisy, structured, or **unstructured data**.

Data science also integrates domain knowledge from the underlying application domain (e.g., natural sciences, information technology, & medicine). Data science is multifaceted & can be described as a science, a research paradigm, a research method, a discipline, a workflow, & a profession.

Data science is “a concept to unify statistics, **data analysis**, **informatics**, & their related **methods**” to “understand & analyze actual **phenomena**” with **data**. It uses techniques & theories drawn from many fields within the context of mathematics, statistics, **computer science**, **information science**, & **domain knowledge**. However, data science is different from **computer science** & **information science**. Turing Award winner **Jim Gray** imagined data science as a “4th paradigm” of science (**empirical**, **theoretical**, **computational**, & now data-driven) & asserted that “everything about science is changing because of the impact of **information technology**” & the **data deluge**.

A *data scientist* is a professional who creates programming code & combines it with statistical knowledge to create insights from data.

5.1.1 Foundations

Data science is an **interdisciplinary field** focused on **extracting knowledge** from typically **large data sets** & applying the knowledge & insights from that data to **solve problems** in a wide range of application domains. The field encompasses preparing data for analysis, formulating data science problems, **analyzing** data, developing data-driven solutions, & presenting findings to inform high-level decisions in a broad range of application domains. As such, it incorporates skills from computer science, statistics, information science, mathematics, **data visualization**, **information visualization**, **data sonification**, data **integration**, **graphic design**, **complex systems**, **communication** & **business**. Statistician **Nathan Yau**, drawing on **Ben Fry**, also links data science to **human-computer interaction**: users should be able to intuitively control & **explore** data. In 2015, the **American Statistical Association** identified **database management**, statistics & **machine learning**, & **distributed & parallel systems** as the 3 emerging foundational professional communities.

Relationship to statistics. Many statisticians, including **Nate Silver**, have argued that data science is not a new field, but rather another name for statistics. Others argue that data science is distinct from statistics because it focuses on problems & techniques unique to digital data. **Vasant Dhar** writes that statistics emphasizes quantitative data & description. In contrast, data science deals with quantitative & qualitative data (e.g., from images, text, sensors, transactions, customer information, etc.) & emphasizes prediction & action. **Andrew Gelman** of Columbia University has described statistics as a non-essential part of data science.

Stanford professor **David Donoho** writes that data science is not distinguished from statistics by the size of datasets or use of computing & that many graduate programs misleadingly advertise their analytics & statistics training as the essence of a data-science program. He describes data science as an applied field growing out of traditional statistics.

5.1.2 Etymology

5.1.3 Data science & data analysis

Data science & data analysis are both important disciplines in the field of **data management** & analysis, but they differ in several key ways. While both fields involve working with data, data science is more of an **interdisciplinary field** that involves the application of statistical, computational, & **machine learning** methods to extract insights from data & make predictions, while data analysis is more focused on the examination & interpretation of data to identify patterns & trends.

Data analysis typically involves working with smaller, structured datasets to answer specific questions or solve specific problems. This can involve tasks such as **data cleaning**, **data visualization**, & exploratory data analysis to gain insights into the data & develop hypotheses about relationships between **variables**. Data analysts typically use statistical methods to test these hypotheses & draw conclusions from the data. E.g., a **data analyst** might analyze sales data to identify trends in customer behavior & make recommendations for marketing strategies.

Data science, on the other hand, is a more complex & **iterative** process that involves working with larger, more complex datasets that often require advanced computational & statistical methods to analyze. Data scientists often work with **unstructured data** such as text or images & use machine learning algorithms to build predictive models & make data-driven decisions. In addition to **statistical analysis**, data science often involves tasks such as **data preprocessing**, **feature engineering**, & model selection. E.g., a data scientist might develop a recommendation system for an e-commerce platform by analyzing user behavior patterns & using **machine learning algorithms** to predict user preferences.

While data analysis focuses on extracting insights from existing data, data science goes beyond that by incorporating the development & implementation of predictive models to make informed decisions. Data scientists are often responsible for collecting & cleaning data, selecting appropriate analytical techniques, & deploying models in real-world scenarios. They work at the intersection of mathematics, computer science, & **domain expertise** to solve complex problems & uncover hidden patterns in large datasets.

Despite these differences, data science & data analysis are closely related fields & often require similar skills sets. Both fields require a solid foundation in statistics, **programming**, & **data visualization**, as well as the ability to communicate findings effectively to both technical & non-technical audiences. Both fields benefit from **critical thinking** & **domain knowledge**, as understanding the context & nuances of the data is essential for accurate analysis & modeling.

In summary, data analysis & data science are distinct yet interconnected disciplines within the broader field of **data management** & analysis. Data analysis focuses on extracting insights & drawing conclusions from **structured data**, while data science involves a more comprehensive approach that combines **statistical analysis**, computational methods, & machine learning to extract insights, build predictive models, & drive data-driven **decision-making**. Both fields use data to understand patterns, make informed decisions, & solve complex problems across various domains.

5.1.4 Cloud computing for data science

Cloud computing can offer access to large amounts of computational power & **storage**. In **big data**, where volumes of information are continually generated & processed, these platforms can be used to handle complex & resource-intensive analytical tasks.

Some distributed computing frameworks are designed to handle big data workloads. These frameworks can enable data scientists to process & analyze large datasets in parallel, which can reduce processing times.

5.1.5 Ethical consideration in data science

Data science involves collecting, processing, & analyzing data which often includes personal & sensitive information. Ethical concerns include potential privacy violations, bias perpetuation, & negative societal impacts.

Machine learning models can amplify existing biases present in training data, leading to discriminatory or unfair outcomes.”
– [Wikipedia/data science](#)

6 Deep Learning (DL)

Community – Cộng đồng.

Resources – Tài nguyên.

1. [\[LBH15\]](#). YANN LECUN, YOSHUA BENGIO, GEOFFREY HINTON. *Deep Learning*.

Những năm gần đây, sự phát triển của các hệ thống tính toán cùng lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn đã giúp machine learning tiến thêm 1 bước dài. 1 lĩnh vực mới được ra đời được gọi là *học sâu* (deep learning, DL). Deep learning đã giúp máy tính thực thi những việc vào 10 năm trước tưởng chừng là không thể: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói & chữ viết, giao tiếp với con người, chuyển đổi ngôn ngữ, hay thậm chí cả sáng tác văn thơ & âm nhạc.” – [\[Tiệ25, p. 15\]](#)

Φ: Start with simple things – Luôn bắt đầu từ những điều đơn giản. Khi bắt tay vào giải quyết 1 bài toán ML hay bất cứ bài toán nào, nên bắt đầu từ các thuật toán đơn giản. Không phải chỉ có các thuật toán phức tạp mới có thể giải quyết được vấn đề. Các thuật toán phức tạp thường có yêu cầu cao về khả năng tính toán & đôi khi nhạy cảm với cách chọn tham số. Ngược lại, các thuật toán đơn giản giúp ta nhanh chóng có 1 bộ khung cho mỗi bài toán. Kết quả của các thuật toán đơn giản cũng mang lại cái nhìn sơ bộ về sự phức tạp của mỗi bài toán. Việc cải thiện kết quả sẽ được thực hiện dần ở các bước sau.” – [Tiệ25, p. 17]

Approach. Để giải quyết mỗi bài toán ML, cần chọn 1 mô hình phù hợp. Mô hình này được mô tả bởi bộ các tham số ta cần đi tìm. Thông thường, lượng tham số có thể lên tới hàng triệu & được tìm bằng cách giải 1 bài toán tối ưu. Khi viết về các thuật toán ML, VKTiếp sẽ bắt đầu từ các ý tưởng trực quan. Các ý tưởng này được mô hình hóa dưới dạng 1 bài toán tối ưu. Các suy luận toán học & ví dụ mẫu trên Python sẽ giúp hiểu rõ hơn về nguồn gốc, ý nghĩa, & cách sử dụng mỗi thuật toán. Xen kẽ giữa các thuật toán ML, trình bày các kỹ thuật tối ưu cơ bản, với hy vọng giúp hiểu rõ hơn bản chất của vấn đề.

Audiences. Cuốn sách được thực hiện hướng tới nhiều nhóm độc giả khác nhau. Nếu không thực sự muốn đi sâu vào phần toán, vẫn có thể tham khảo mã nguồn & cách sử dụng các thư viện. Nhưng để sử dụng các thư viện 1 cách hiệu quả, cũng cần hiểu nguồn gốc của mô hình & ý nghĩa của các tham số. Còn nếu thực sự muốn tìm hiểu nguồn gốc, ý nghĩa của các thuật toán, có thể học được nhiều điều từ cách xây dựng & tối ưu các mô hình.

Python. Python là 1 ngôn ngữ lập trình miễn phí, có thể được cài đặt dễ dàng trên các nền tảng hệ điều hành khác nhau. Có rất nhiều thư viện hỗ trợ ML cũng như DL trên Python. Có 2 thư viện Python chính thường được sử dụng là **numpy**, **scikit-learn**.

- **numpy** www.numpy.org là 1 thư viện phổ biến giúp xử lý các phép toán liên quan đến các mảng nhiều chiều, hỗ trợ các hàm gần gũi với đại số tuyến tính. Cách xử lý các mảng nhiều chiều.
- **scikit-learn/sklearn** scikit-learn.org: 1 thư viện chứa đầy đủ các thuật toán ML cơ bản & rất dễ sử dụng. Tài liệu của scikit-learn cũng là 1 nguồn tham khảo chất lượng cho MLer. Scikit-learn được dùng để kiểm chứng các suy luận toán học & các mô hình được xây dựng thông qua **numpy**.

Inevitability of mathematics in ML. Có rất nhiều thư viện giúp tạo ra các sản phẩm ML/DL mà không yêu cầu nhiều kiến thức toán. Hướng tới việc giúp hiểu bản chất toán học đằng sau mỗi mô hình trước khi áp dụng các thư viện sẵn có. Việc sử dụng thư viện + yêu cầu kiến thức nhất định về việc lựa chọn mô hình & điều chỉnh các tham số.

7 Machine Learning (ML)

Resources – Tài nguyên.

1. ANDREW NG. *Machine Learning Course* on Coursera.
2. Machine Learning Mastery: Making Developers Awesome at Machine Learning: <https://machinelearningmastery.com>.
 - [Machine Learning Mastery/8 Inspirational Applications of Deep Learning](#).
3. Machine Learning cơ bản: <https://machinelearningcoban.com/>.
4. [Tiệ25]. VŨ HỮU TIỆP. *Machine Learning Cơ Bản*.
Mã nguồn cuốn ebook “Machine Learning Cơ Bản”: <https://github.com/tiepvupsu/ebookMLCB>.

Definition 1. “Machine learning (ML) is a field of study in AI concerned with the development & study of *statistical algorithms* that can learn from *data* & generalize to unseen data, & thus perform *tasks* without explicit *instructions*. Quick progress in the fields of *deep learning*, beginning in 2010s, allowed neural networks to surpass many previous approaches in performance.” – [Wikipedia/machine learning](#)

Định nghĩa 1. “Học máy (*machine learning, ML*) là 1 tập con của trí tuệ nhân tạo. Machine learning là 1 lĩnh vực nhỏ trong Khoa học Máy tính, có khả năng tự học hỏi dựa trên dữ liệu được đưa vào mà không cần phải được lập trình cụ thể: “Machine Learning is the subfield of computer science, that “gives computers the ability to learn without being explicitly programmed” – [Wikipedia](#).” – [Tiệ25, p. 15]

“ML finds application in many fields, including *natural language processing*, *computer vision*, *speech recognition*, *email filtering*, *agriculture*, & *medicine*. The application of ML to business problems is known as *predictive analysis*.”

Statistics & mathematical optimization/mathematical programming methods comprise the foundations of machine learning. *Data mining* is related field of study, focusing on *exploratory data analysis* (EDA) via *unsupervised learning*.

From a theoretical viewpoint, *probably approximately correct (PAC) learning* provides a framework for describing machine learning.” – [Wikipedia/machine learning](#)

Relationships of ML to AI. As a scientific endeavor, machine learning grew out of the quest for AI. In the early days of AI as an *academic discipline*, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed “*neural networks*”; these were mostly *perceptrons* & other models e.g. *ADALINE* that were later found to be reinventions of the *generalized linear models* of statistics. *Probabilistic reasoning* was also employed, especially in *automated medical diagnosis*. However, an increasing emphasis on the *logical, knowledge-based approach* caused a rift between AI & machine learning. Probabilistic systems were plagued by theoretical & practical problems of data acquisition & representation.

7.1 Wikipedia/Machine Learning

“Machine learning (ML) is a **field of study** in AI concerned with the development & study of **statistical algorithms** that can learn from **data** & **generalize** to unseen data, & thus perform **tasks** without explicit **instructions**. Quick progress in the field of **deep learning**, beginning in 2010s, allowed neural networks to surpass many previous approaches in performance.

ML finds application in many fields, including **natural language processing**, **computer vision**, **speech recognition**, **email filtering**, **agriculture**, & **medicine**. The application of ML to business problems is known as **predictive analytics**.

Statistics & **mathematical optimization/programming** methods comprise the foundations of machine learning. **Data mining** is a related field of study, focusing on **exploratory data analysis** (EDA) via **unsupervised learning**.

From a theoretical viewpoint, **probably approximately correct (PAC) learning** provides a framework for describing machine learning.

7.1.1 History

7.1.2 Relationships to other fields

AI. As a scientific endeavor, machine learning grew out of the quest for AI. In the early days of AI as an **academic discipline**, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed “**neural networks**”; these were mostly **perceptrons** & **other models** **ADALINE** that were later found to be reinventions of the **generalized linear models** of statistics. **Probabilistic reasoning** was also employed, especially in **automated medical diagnosis**.

However, an increasing emphasis on the **logical, knowledge-based approach** caused a rift between AI & machine learning. Probabilistic systems were plagued by theoretical & practical problems of data acquisition & representation. By 1980, **expert systems** had come to dominate AI, & statistics was out of favor. Work on symbolic/knowledge-based learning did continue within AI, leading to **inductive logic programming** (ILP), but the more statistical line of research was now outside the field of AI proper, in **pattern recognition** & **information retrieval**. Neural networks research had been abandoned by AI & computer science around the same time. This line, too, was continued outside the AI/CS field, as “**connectionism**”, by researchers from other disciplines including **John Hopfield**, **David Rumelhart**, & **Geoffrey Hinton**. Their main success came in the mid-1980s with the reinvention of **backpropagation**.

Machine learning (ML), reorganized & recognized as its own field, started to flourish in the 1990s. The field changed its goal from achieving AI to tackling solvable problems of a practical nature. It shifted focus away from the **symbolic approaches** it had inherited from AI, & toward methods & models borrowed from statistics, **fuzzy logic**, & **probability theory**.

Data compression. Main: **Wikipedia/data compression/ML**. There is a close connection between ML & compression. A system that predicts the **posterior probabilities** of a sequence given its entire history can be used for optimal data compression (by using **arithmetic coding** on the output distribution). Conversely, an optimal compressor can be used for prediction (by finding the symbol that compresses best, given the previous history). This equivalence has been used as a justification for using data compression as a benchmark for “general intelligence”.

An alternative view can show compression algorithms implicitly map strings into implicit **feature space vectors**, & compression-based similarity measures compute similarity within these feature spaces. For each compressor $C(\cdot)$ we define an associated vector space \mathfrak{N} s.t. $C(\cdot)$ maps an input string x , corresponding to the vector norm $\|\tilde{x}\|$. An exhaustive examination of the feature spaces underlying all compression algorithms is precluded by space; instead, feature vectors chooses to examine 3 representative lossless compression methods, LZW, LZ77, & PPM.

According to **AIXI** theory, a connection more directly explained in **Hutter Prize**, the best possible compression of x is the smallest possible software that generates x . E.g., in that model, a zip file’s compressed size includes both the zip file & the unzipping software, since you cannot unzip it without both, but there may be an even smaller combined form.

Examples of AI-powered audio/video compression software include **NVIDIA Maxine**, **AIVC**. Examples of software that can perform AI-powered image compression include **OpenCV**, **TensorFlow**, **MATLAB**’s Imaging Processing Toolbox (IPT) & High-Fidelity Generative Image Compression.

In **unsupervised machine learning**, **k-mean clustering** can be utilized to compress data by grouping similar data points into clusters. This technique simplifies handling extensive datasets that lack predefined labels & finds widespread use in fields such as **image compression**.

Data compression aims to reduce the size of data files, enhancing storage efficiency & speeding up data transmission. **k-means clustering**, an unsupervised machine learning algorithm, is employed to partition a dataset into a specified number of clusters, k , each represented by the **centroid** of its points. This process condenses extensive datasets into a more compact set of representative points. Particularly beneficial in **image** & **signal processing**, **k-means clustering** aids in data reduction by replacing groups of data points with their centroids, thereby preserving the core information of the original data while significantly decreasing the required storage space.

Large language models (LLMs) are also capable of lossless data compression, as demonstrated by **DeepMind**’s research with the Chinchilla 70B model. Developed by DeepMind, Chinchilla 70B effectively compressed data, outperforming conventional methods such as **Portable Network Graphics** (PNG) for images & **Free Lossless Audio Codec** (FLAC) for audio. It achieved compression of image & audio data to 43.4% & 16.4% of their original sizes, respectively.

Data mining. ML & **data mining** often employ the same methods & overlap significantly, but while ML focuses on prediction, based on *known* properties learned from the training data, data mining focuses on the **discovery** of (previously) *unknown* properties in the data (this is the analysis step of **knowledge discovery** in databases). Data mining uses many ML methods, but with different goals; on the other hand, ML also employs data mining methods as “**unsupervised learning**” or as a preprocessing step to improve learner accuracy. Much of the confusion between these 2 research communities (which do often have separate conferences & separate journals, **ECML PKDD** being a major exception) comes from the basic assumptions they work with: in ML, performance is usually evaluated w.r.t. the ability to *reproduce known* knowledge, while in knowledge discovery & data mining (KDD) the key task is the discovery of previously *unknown* knowledge. Evaluated w.r.t. known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task, supervised methods cannot be used due to the unavailability of training data.

ML also has intimate ties to optimization: Many learning problems are formulated as minimization of some **loss function** on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained & the actual problem instances (e.g., in classification, one wants to assign a **label** to instances, & models are trained to correctly predict the preassigned labels of a set of examples).

Generalization. Characterizing the generalization of various learning algorithms is an active topic of current research, especially for **deep learning** algorithms.

Statistics. ML & statistics are closely related fields in terms of methods, but distinct in their principal goal: statistics draws population **inferences** from a **sample**, while ML finds generalizable predictive patterns. According to **Michael I. Jordan**, the ideas of ML, from methodological principles to theoretical tools, have had a long pre-history in statistics. he also suggested the term data science as a placeholder to call the overall field.

Conventional statistical analyzes require the a priori selection of a model most suitable for the study data set. In addition, only significant or theoretically relevant variables based on previous experience are included for analysis. In contrast, ML is not built on a pre-structured model; rather, the data shape the model by detecting underlying patterns. The more variables (input) used to train the model, the more accurate the ultimate model will be.

Leo Breiman distinguished 2 statistical modeling paradigms: data model & algorithmic model, wherein “algorithm model” means more or less the ML algorithms like **Random Forest**.

Some statisticians have adopted methods from ML, leading to a combined field that they call *statistical learning*.

Statistical physics. Analytical & computational techniques derived from deep-rooted physics of disordered systems can be extended to large-scale problems, including ML, e.g., to analyze the weight space of **deep neural networks**. Statistical physics is thus finding applications in the area of **medical diagnostics**.

7.1.3 Theory

Main: **Wikipedia/computational learning theory** & **Wikipedia/statistical learning theory**. A core objective of a learner is to generalize from its experience. Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) & the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

The computational analysis of ML algorithms & their performance is a branch of **theoretical computer science** known as **computational learning theory** via the **Probably Approximately Correct Learning** (PAC) model. Because training sets are finite & the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The **bias-variance decomposition** is 1 way to quantify generalization **error**.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has under fitted the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to **overfitting** & generalization will be poorer.

In addition to performance bounds, learning theorists study the time complexity & feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in **polynomial time**. There are 2 kinds of **time complexity** results: Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

7.1.4 Approaches

ML approaches are traditionally divided into 3 broad categories, which corresponding to learning paradigms, depending on the nature of the “signal” or “feedback” available to the learning system:

- **Supervised learning:** The computer is presented with example inputs & their desired outputs, given by a “teacher”, & the goal to learn a general rule that maps inputs to outputs.
- **Unsupervised learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means toward an end (**feature learning**).

- **Reinforcement learning:** A computer program interacts with a dynamic environment in which it must perform a certain goal (e.g. **driving a vehicle** or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Although each algorithm has advantages & limitations, no single algorithm works for all problems.

Supervised learning.

Semi-supervised learning.

Reinforcement learning.

Dimensionality reduction.

Other types. ” – [Wikipedia/machine learning](#)

8 Artificial Intelligence (AI)

Resources – Tài nguyên.

1. [BV14]. LÊ HOÀI BẮC, TÔ HOÀI VIỆT. *Cơ Sở Trí Tuệ Nhân Tạo*.
2. [Aou14]. JOSEPH E. AOUN. *Robot-Proof: Higher Education in the Age of Artificial Intelligence*.
3. [Aou19]. JOSEPH E. AOUN. *Robot-Proof: Higher Education in the Age of Artificial Intelligence – Chạy Dua Với Robot: Học Tập Thời Trí Tuệ Nhân Tạo*.

9 Miscellaneous

Tài liệu

- [Aou14] Joseph E. Aoun. *Robot-Proof: Higher Education in the Age of Artificial Intelligence*. MIT Publisher, 2014, p. 187.
- [Aou19] Joseph E. Aoun. *Robot-Proof: Higher Education in the Age of Artificial Intelligence – Chạy Dua Với Robot: Học Tập Thời Trí Tuệ Nhân Tạo*. Trịnh Huy Nam dịch. Nhà Xuất Bản Thế Giới, 2019, p. 241.
- [BV14] Lê Hoài Bắc and Tô Hoài Việt. *Cơ Sở Trí Tuệ Nhân Tạo*. Nhà Xuất Bản Khoa Học & Kỹ Thuật, 2014, p. 229.
- [LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep Learning”. In: *Nature* 521 (2015), pp. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539). URL: <https://doi.org/10.1038/nature14539>.
- [Tiệ25] Vũ Khắc Tiệp. *Machine Learning Cơ Bản*. 2025, p. 422.