

Chương 1

Giới thiệu

- 1.1 Ai nên đọc cuốn sách này?
- 1.2 Lịch sử các xu hướng trong Deep Learning
 - 1.2.1 Các tên gọi của DL
 - 1.2.2 Kích thước dữ liệu ngày càng tăng
 - 1.2.3 Kích thước mô hình ngày càng tăng
 - 1.2.4 Độ phức tạp, độ chính xác và mức độ ảnh hưởng thực tế ngày càng tăng

Từ xa xưa, các nhà phát minh đã khao khát tạo ra những cỗ máy có khả năng suy nghĩ. Khao khát này đã có từ thời Hy Lạp cổ đại qua những hình tượng thần thoại như Pygmalion, Daedalus, và Hephaestus; được coi là những nhà phát minh huyền thoại, và Galatea, Talos, Pandora; có thể xem như những dạng sống nhân tạo [Ovid và Martin, 2004; Sparkes, 1996; Tandy, 1997].

Ngay từ khi những chiếc máy tính đầu tiên có khả năng lập trình được tạo ra, người ta đã tự hỏi liệu máy móc có thể trở nên thông minh được hay không, và phải đến một thế kỷ sau, một cỗ máy như kỳ vọng mới ra đời [Lovelace, 1842]. Ngày nay, *trí tuệ nhân tạo* (*artificial intelligence* - AI) đã trở thành một lĩnh vực với vô vàn ứng dụng thực tiễn và trở thành chủ đề thu hút rất nhiều sự quan tâm của các nhà nghiên cứu trên toàn thế giới. ta sử dụng phần mềm thông minh để tự động hóa các công việc chân tay, nhận dạng hình ảnh, âm thanh, chẩn đoán y học và hỗ trợ nghiên cứu khoa học cơ bản.

Trong những ngày đầu của AI, ngành này đã nhanh chóng giải quyết những vấn đề được xem là rất phức tạp đối với con người, nhưng lại tương đối đơn giản đối với máy tính - những vấn đề có thể diễn tả được bằng một danh sách những quy tắc toán học hình thức. Thách thức thực sự đối với trí tuệ nhân tạo là giải quyết những tác vụ dễ thực hiện với con người nhưng lại khó diễn tả một cách tường minh (bằng ngôn ngữ toán học hình thức) - những vấn đề mà con người xử lý qua trực giác một cách rất tự nhiên, như nhận dạng lời nói hay khuôn mặt chẳng hạn.

Cuốn sách này đưa ra một giải pháp cho những vấn đề trực giác đó. Giải pháp này cho phép máy tính tự học từ những kinh nghiệm thu được và hiểu rõ hơn về thế giới thông qua một *hệ phân cấp khái niệm (a hierarchy of concepts)*, mà trong đó mỗi khái niệm lại được định nghĩa từ mối quan hệ của nó với những khái niệm đơn giản hơn. Bằng cách cho máy tính tự động thu thập tri thức từ kinh nghiệm, cách tiếp cận này giúp giảm bớt gánh nặng cho người vận hành trong việc mô tả tường minh tất cả tri thức cần thiết cho máy tính. Hệ phân cấp khái niệm cho phép máy tính tự học những khái niệm phức tạp từ những khái niệm đơn giản hơn. Nếu ta vẽ một đồ thị để mô tả hệ phân cấp khái niệm này thì đồ thị đó sẽ có rất nhiều lớp, và nó rất sâu (*deep*). Vì vậy, ta gọi cách tiếp cận AI này là *DL (deep learning - DL)*.

Rất nhiều AI thành công thế hệ đầu tiên ra đời trong môi trường thí nghiệm, và môi trường đó không yêu cầu máy tính phải có nhiều hiểu biết về thế giới bên ngoài. Ví dụ, hệ thống chơi cờ vua mang tên *Deep Blue* của công ty IBM đã đánh bại nhà vô địch thế giới Garry Kasparov vào năm 1997 [Hsu, 2002]. Cờ vua dĩ nhiên là một môi trường đơn giản, chỉ bao gồm 64 ô vuông và 32 quân cờ, trong đó mỗi quân chỉ có thể di chuyển theo những quy tắc xác định. Việc sáng tạo ra một chiến thuật chơi cờ hiệu quả là một thành tựu lớn, nhưng thách thức của hệ thống không nằm ở việc dạy cho máy tính hiểu về các quân cờ hay cách di chuyển của chúng. Luật chơi cờ vua được biểu diễn đầy đủ bằng một số quy tắc và có thể lập trình một cách dễ dàng.

Những tác vụ trừu tượng và logic khiến cho con người phải đau đầu lại là những tác vụ dễ dàng với máy tính. Máy tính từ lâu đã có khả năng đánh bại người chơi cờ vua giỏi nhất, nhưng gần đây mới theo kịp trình độ của người bình thường trong việc nhận dạng vật thể và lời nói. Cuộc sống hàng ngày của một con người đòi hỏi một lượng rất lớn tri thức về thế giới. Phần lớn những kiến thức này mang tính chủ quan và dựa nhiều vào trực giác của mỗi người, nên rất khó để có thể diễn đạt chúng một cách rõ ràng. Máy tính cần nắm bắt được cùng lượng tri thức như vậy về thế giới để có thể trở nên thông minh hơn. Một trong những thách thức chính của AI là cách đưa những tri thức không tường minh đó vào máy tính.

Một vài dự án AI cố gắng *gắn cứng (hard-code)* tri thức về thế giới thông qua *ngôn ngữ hình thức*. Máy tính có thể tư duy tự động về các *chỉ lệnh* thông qua các quy tắc suy luận logic. Cách tiếp cận AI này được gọi là tiếp cận thông qua *cơ sở tri thức (knowledge base)*. Tuy nhiên, không một dự án nào trong số đó đạt được thành công lớn. Một trong số những dự án nổi tiếng nhất là dự án Cyc

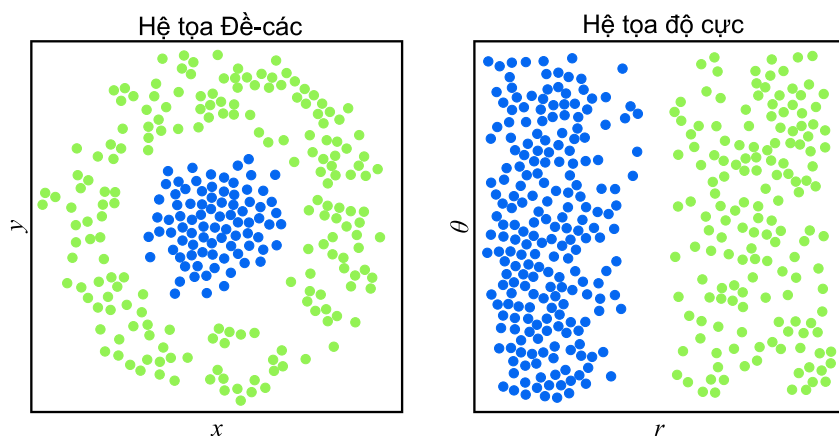
[Lenat & Guha, 1989]. Cyc là một công cụ suy luận và có một cơ sở dữ liệu gồm các chỉ lệnh được viết bằng ngôn ngữ CycL. Những chỉ lệnh này được đội ngũ nhân viên giám sát của Cyc đưa vào và đó là một quá trình đầy khó khăn. Người ta gặp vấn đề khi tạo ra các quy tắc khi cố gắng mô tả chính xác những môi trường phức tạp. Ví dụ, Cyc không hiểu được câu chuyện về một người tên Fred cạo râu vào buổi sáng [Linde, 1992]. Công cụ suy luận của Cyc đã phát hiện ra một sự mâu thuẫn trong câu chuyện: nó biết con người không có những bộ phận điện tử trên cơ thể, nhưng vì lúc đó Fred đang cầm một chiếc máy cạo râu, và Cyc cho rằng thực thể mang tên “FredWhileShaving” có chứa những bộ phận điện tử. Điều này dẫn đến việc Cyc đưa ra một mâu thuẫn trong suy luận: liệu Fred có còn là con người khi anh ta đang cạo râu hay không?

Những khó khăn mà các mô hình xây dựng dựa trên *tri thức gắn cứng* gặp phải cho ta thấy một hệ thống AI thực sự cần phải có khả năng tự thu thập tri thức riêng bằng cách trích xuất *pattern* từ dữ liệu thô. Khả năng này gọi là *học máy* (*machine learning* - ML). Sự xuất hiện của ML cho phép máy tính giải quyết các vấn đề liên quan đến tri thức về thế giới thực và đưa ra quyết định có vẻ chủ quan. Một thuật toán ML đơn giản như *logistic regression* có thể chẩn đoán và đưa ra lời khuyên có nên thực hiện phẫu thuật để hỗ trợ bà bầu khi sinh hay không [Mor-Yosef et al., 1990]. Một thuật toán khác, *naive Bayes*, có thể phân loại email hợp lệ và email rác.

Hiệu suất của các thuật toán ML đơn giản này phụ thuộc nhiều vào *cách biểu diễn* (*representation*) của dữ liệu đầu vào. Ví dụ, khi *logistic regression* chẩn đoán phẫu thuật khi sinh, hệ thống AI không kiểm tra bệnh nhân một cách trực tiếp. Thay vào đó, các bác sĩ đưa một số thông tin liên quan vào hệ thống, ví dụ như có tồn tại sẹo tử cung hay không.. Mỗi mẫu thông tin liên quan đến bệnh nhân được gọi là một *đặc trưng* (*feature*). Thuật toán Logistic Regression sẽ học mối liên hệ tương quan giữa mỗi đặc trưng với nhiều kết quả đầu ra. Tuy nhiên, thuật toán này không thể quyết định việc các đặc trưng được định nghĩa thế nào. Nếu Logistic Regression được cung cấp một bản chụp MRI của bệnh nhân, thay vì báo cáo chi tiết của bác sĩ, nó sẽ không thể đưa ra dự đoán chính xác. Những điểm ảnh đơn lẻ trong bản chụp MRI không có nhiều tương quan với các biến chứng có thể xảy ra trong khi thực hiện phẫu thuật.

Sự phụ thuộc vào cách biểu diễn thông tin là hiện tượng phổ biến trong khoa học máy tính và cuộc sống hàng ngày. Trong khoa học máy tính, những thao tác như tìm kiếm trong một tập dữ liệu có thể diễn ra nhanh hơn theo cấp số nhân nếu

tập hợp đó được tổ chức có cấu trúc và được lập chỉ mục (*index*) một cách thông minh. Người ta có thể dễ dàng thực hiện các phép toán số học trên hệ chữ số Ả Rập nhưng lại mất nhiều thời gian khi thực hiện trên hệ La Mã. Không có gì ngạc nhiên khi việc lựa chọn cách biểu diễn thông tin lại có ảnh hưởng lớn đến hiệu suất của các thuật toán ML. Hình 1.1 minh họa một ví dụ trực quan.



Hình 1.1: Ví dụ về sự khác biệt trong cách biểu diễn thông tin: giả sử ta muốn phân loại hai dạng dữ liệu bằng cách vẽ một đường thẳng phân tách giữa chúng trong một đồ thị đa điểm. Trong đồ thị bên trái, ta biểu diễn dữ liệu theo hệ tọa độ Đều-các, và đường thẳng như vậy không tồn tại. Trong đồ thị bên phải ta biểu thị dữ liệu dưới hệ tọa độ cực và ta chỉ cần vẽ một đường thẳng dọc là có thể phân đôi được hai tập điểm. (Hình vẽ có sự giúp đỡ của David Warde-Farley.)

Nhiều tác vụ AI có thể được giải quyết bằng cách đưa một bộ features phù hợp vào một thuật toán ML đơn giản. Ví dụ, một đặc trưng hữu ích cho việc nhận dạng một người qua giọng nói là ước lượng kích cỡ thanh quản của họ. Đặc trưng này cho phép ta nhận biết được người nói là đàn ông, đàn bà hay một đứa trẻ.

Tuy nhiên, với nhiều tác vụ, rất khó để biết nên trích xuất những features nào. Ví dụ, ta muốn viết một chương trình nhận dạng xe hơi trong ảnh. Ta biết xe hơi có bánh, nên sự xuất hiện của bánh xe là một feature. Tuy nhiên, rất khó để mô tả chính xác hình dạng của bánh xe thông qua giá trị của các điểm ảnh. Bánh xe có hình dạng vật lý khá đơn giản, nhưng ảnh chụp của nó trở nên phức tạp hơn do nhiều yếu tố khác: bóng của ánh nắng phủ lên bánh xe, ánh sáng lóa chói ở một

số bộ phận kim loại trên bánh xe, tấm chắn bùn của xe hoặc một vật thể nào đó che khuất vài bộ phận của bánh xe..

Một giải pháp cho vấn đề này là sử dụng ML sinh ra cách biểu diễn thông tin - chọn đặc trưng. Cách tiếp cận này được gọi là *học biểu diễn (representation learning)*. Những features học được thường hiệu quả hơn những features được chọn thủ công. Chúng cũng cho phép các hệ thống AI thích nghi nhanh hơn với các tác vụ mới và ít bị can thiệp bởi con người. Một thuật toán học biểu diễn có thể tìm ra bộ đặc trưng phù hợp cho những tác vụ đơn giản trong vài phút, và cho những tác vụ phức tạp trong thời gian vài giờ hoặc vài ngày. Các features thủ công yêu cầu nhiều thời gian và công sức từ con người; cả một cộng đồng các nhà nghiên cứu có thể mất hàng chục năm chỉ để làm được việc tương tự (chọn features).

Một ví dụ điển hình của thuật toán học biểu diễn là *bộ tự mã hóa (auto-encoder)*. Auto-encoder là sự kết hợp của một *hàm mã hóa (encoder)*, có chức năng chuyển đổi dữ liệu đầu vào thành một chiều biểu diễn thông tin khác, và một *hàm giải mã (decoder)*, có chức năng chuyển cách biểu diễn thông tin mới ngược trở về dạng ban đầu. Bộ *encoder* được huấn luyện để giữ lại những thông tin đặc trưng nhất của tập dữ liệu ban đầu. Các bộ *encoder* khác nhau có mục tiêu học ra những thuộc tính khác nhau từ dữ liệu.

Trong thiết kế các đặc trưng hay thuật toán học đặc trưng, mục tiêu của ta là tách rời các *biến tố (factors of variation)* đóng vai trò giải thích dữ liệu quan sát được. Chữ **tố** ở đây, viết gọn của **nhân tố (factor)** là chỉ đến những nguồn ảnh hưởng riêng lẻ. Những nhân tố như vậy thường là các đại lượng ta không quan sát được. Thay vào đó, chúng tồn tại dưới dạng những vật thể hay những lực không thể đo đạc trong thế giới vật chất, nhưng có tác động đến những đại lượng ta quan sát thấy. Chúng còn có thể tồn tại dưới dạng các *thành tố (construct)* khái niệm trong tâm trí con người, giúp đưa ra cách giải thích đơn giản hơn hay suy diễn những nguyên nhân đằng sau dữ liệu thu thập được. Ta có thể coi chúng như những khái niệm giúp giải thích sự đa dạng trong dữ liệu. Khi phân tích một đoạn ghi âm lời nói, những biến tố bao gồm tuổi, giới tính, chất giọng và những câu từ của người nói. Khi phân tích một bức ảnh chụp xe hơi, những biến tố bao gồm vị trí, màu sắc, góc quan sát chiếc xe và độ sáng của ánh nắng mặt trời.

Khó khăn chính của các ứng dụng AI là nhiều biến tố có thể gây ảnh hưởng tới dữ liệu. Những điểm ảnh trong bức ảnh chụp chiếc xe hơi màu đỏ có thể có màu gần với màu đen nếu ảnh được chụp vào ban đêm. Hình dạng cái bóng của chiếc xe tùy thuộc vào góc quan sát. Hầu hết các ứng dụng đòi hỏi ta phải tách rời các biến tố ra và loại bỏ những giá trị không cần thiết.

Dĩ nhiên, việc trích xuất các đặc trưng mức cao, trừu tượng từ dữ liệu thô có thể rất khó. Đa số những biến tố, chẳng hạn như chất giọng người nói, chỉ có thể được xác định khi AI có trình độ hiểu biết tinh tế, gần ngang mức con người. Khi việc học biểu diễn khó tương đương với việc giải quyết bài toán ban đầu, thì có vẻ như học biểu diễn cũng không giúp ta giải bài toán được bao nhiêu.

DL giải quyết vấn đề chính của phương pháp học biểu diễn bằng cách đưa ra cách biểu diễn mới dựa theo nhiều cách biểu diễn khác đơn giản hơn. DL cho phép máy tính xây dựng khái niệm phức tạp dựa trên khái niệm đơn giản. Hình 1.2 cho thấy một hệ thống DL có thể biểu diễn khái niệm về một bức ảnh chụp con người bằng việc kết hợp các khái niệm đơn giản hơn, như các góc và đường bao, các khái niệm này lại được xác định từ các đường biên.

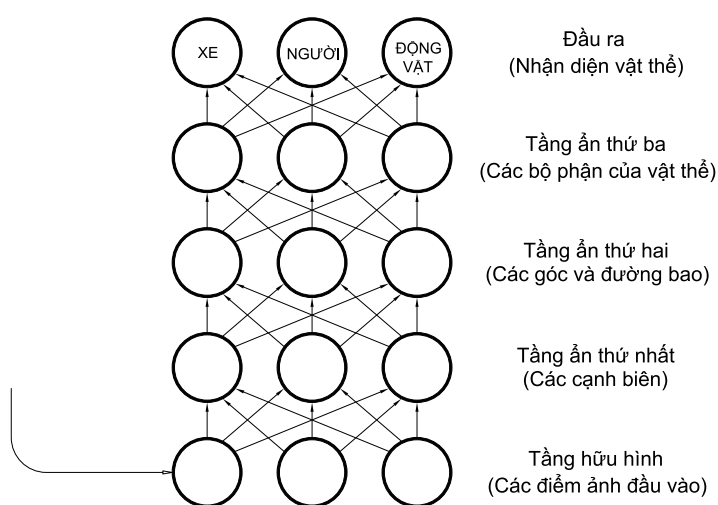


Figure 1.2: Minh họa một mô hình DL. Thật khó khăn để máy tính có thể hiểu được ý nghĩa của các dữ liệu đầu vào dạng thô, chẳng hạn như bức ảnh này, khi nó được biểu diễn bởi một tập các giá trị điểm ảnh. Hàm ánh xạ từ tập các điểm ảnh đến định dạng của vật thể cực kỳ phức tạp. Việc học và đánh giá hàm ánh xạ này có vẻ như là nhiệm vụ bất khả thi nếu muốn xử lý trực tiếp. DL giải quyết vấn đề này bằng cách chia nhỏ hàm ánh xạ cần tìm thành một chuỗi các hàm ánh xạ đơn giản lồng vào nhau, mỗi hàm ánh xạ được mô tả bởi một layer khác nhau của mô hình. Đầu vào

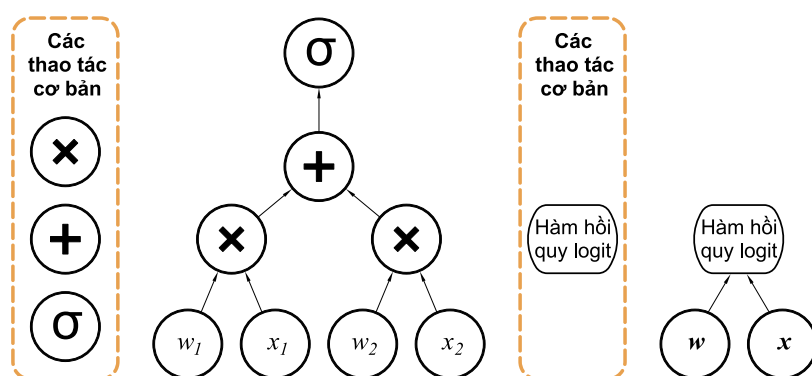
được mô tả bởi *visible layer*, ta đặt tên như thế là bởi các biến số ở layer này có thể quan sát được. Tiếp theo là một dãy các *hidden layer* phân giải đặc trưng theo hướng tăng dần mức độ trừu tượng tính từ *visible layer*. Chúng được gọi là *hidden layer* bởi vì giá trị ở những layers này không có sẵn trong dữ liệu; thay vào đó, mô hình phải tự xác định những khái niệm nào là hữu ích trong việc lý giải mối quan hệ giữa các dữ liệu quan sát được và tự tinh chỉnh giá trị cho mỗi tầng. Những hình ảnh bạn thấy ở đây là cách biểu diễn của các đặc trưng ở mỗi layer. Từ các điểm ảnh cho trước, lớp thứ nhất có thể dễ dàng nhận ra các cạnh biên bằng cách so sánh độ sáng giữa các điểm ảnh lân cận với nhau. Từ các cạnh biên đã được mô tả bởi layer thứ nhất, layer thứ 2 có thể dễ dàng tìm ra các góc và đường bao mở rộng, những đặc trưng này có thể được nhận diện bằng một tập các cạnh biên cơ bản. Từ các góc và đường bao được mô tả bởi layer thứ hai, layer thứ ba có thể phát hiện ra toàn bộ các phần chi tiết của vật thể, bằng cách tìm các tập đường bao và góc cụ thể. Cuối cùng, các mô tả dưới dạng các bộ phận của vật thể có thể được dùng để nhận diện vật thể trong ảnh. Những hình ảnh này được tái hiện với sự giúp đỡ của [Zeiler và Fergus, 2014]

Ví dụ điển hình cho mô hình DL là *mạng lan truyền thuận đa tầng* (*feed forward deep network*), còn gọi là *mạng perceptron đa tầng* (*multilayer perceptron - MLP*). Một MLP thực chất là một hàm toán học có chức năng ánh xạ dữ liệu đầu vào tới các giá trị đầu ra. Hàm này được hợp thành từ những hàm đơn giản. ta có thể coi mỗi lần áp dụng một hàm toán học khác nhau là một lần đưa ra cách biểu diễn mới cho dữ liệu đầu vào.

Ý tưởng của việc học cách biểu diễn đúng cho dữ liệu đã cho ta một góc nhìn về DL. Một góc nhìn khác về DL là độ sâu, cho phép máy tính học ra một chương trình nhiều công đoạn. Mỗi layer của cách biểu diễn có thể xem như là trạng thái của bộ nhớ máy tính sau khi thực hiện một loạt các *câu lệnh song song*. Các mạng với độ sâu lớn hơn có thể thực hiện nhiều lệnh tuần tự hơn. Những dãy lệnh tuần tự có công năng rất lớn, bởi những câu lệnh sau có thể sử dụng lại kết quả của những câu lệnh trước. Theo như cách nhìn này của DL, không phải tất cả thông tin trong các *hàm kích hoạt* của cùng một tầng đều mã hóa những biến tố giải thích dữ liệu đầu vào. Cách biểu diễn này còn lưu trữ thông tin trạng thái, giúp chương trình máy tính có thể giải thích dữ liệu đầu vào. Thông tin trạng thái này có thể tương tự như bộ đếm hay con trỏ trong một chương trình máy tính

truyền thống. Nó không liên quan tới nội dung đầu vào, nhưng giúp mô hình tổ chức quá trình xử lý hiệu quả hơn.

Có hai cách chính để đánh giá độ sâu của một mô hình. Cách thứ nhất dựa trên chuỗi các lệnh tuần tự phải thực thi trên toàn bộ kiến trúc. Ta có thể xem nó giống như độ dài của đường dài nhất trong lưu đồ mô tả cách tính đầu ra của mỗi mô hình ứng với với đầu vào tương ứng. Cũng như hai chương trình máy tính tương đương sẽ có độ dài khác nhau phụ thuộc vào ngôn ngữ được sử dụng để viết chúng, cùng một hàm có thể có *lưu đồ giải thuật (flowchart)* với độ dài khác nhau phụ thuộc vào những hàm được sử dụng trong từng bước đơn lẻ. Hình 1.3 minh họa việc lựa chọn ngôn ngữ có thể đưa ra độ sâu khác nhau cho cùng một kiến trúc như thế nào.



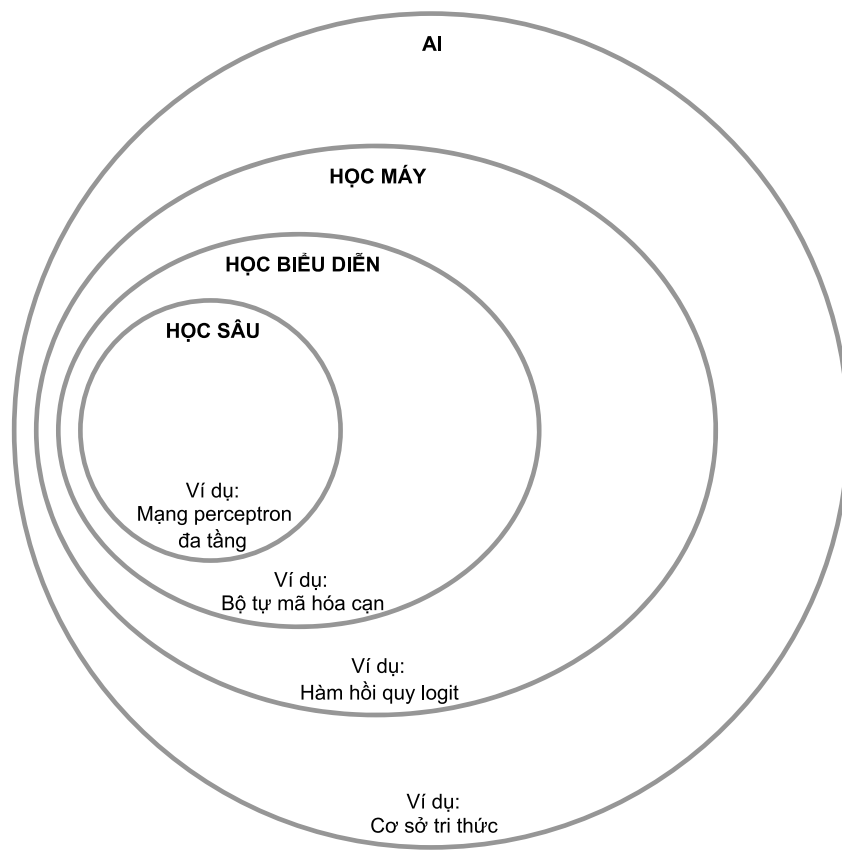
Hình 1.3: Minh họa một biểu đồ tính toán với đầu vào-ra, và mỗi node thực hiện một phép tính. Độ sâu là chiều dài của đường đi dài nhất từ đầu vào tới đầu ra. Tính toán trong những biểu đồ trên minh họa mô hình logistic regression, $\sigma(\mathbf{w}^T \mathbf{x})$, ở đây σ là hàm *logistic sigmoid*. Nếu ta sử dụng phép cộng, nhân là các thao tác cơ bản trong ngôn ngữ lập trình, thì mô hình này có độ sâu là 3. Nếu xem toàn bộ hàm logistic regression như là một thao tác cơ bản thì mô hình này chỉ có độ sâu là 1.

Một hướng tiếp cận khác, được sử dụng bởi các mô hình xác suất sâu, liên quan đến độ sâu của của một mô hình không phải là chiều sâu của biểu đồ tính toán nhưng độ sâu của biểu đồ mô tả cách các khái niệm có liên quan với nhau. Trong trường hợp này, độ sâu của biểu đồ tính toán cần được tính để tìm ra một biểu diễn của mỗi khái niệm có thể sâu hơn nhiều so với đồ thị của chính các khái niệm đó. Đó là vì cách hiểu của hệ thống về các khái niệm có thể được làm mịn hơn khi ta có thêm thông tin về các khái niệm phức tạp hơn. Ví dụ, một hệ thống AI quan sát một hình ảnh của khuôn mặt với một mắt trong bóng tối ban

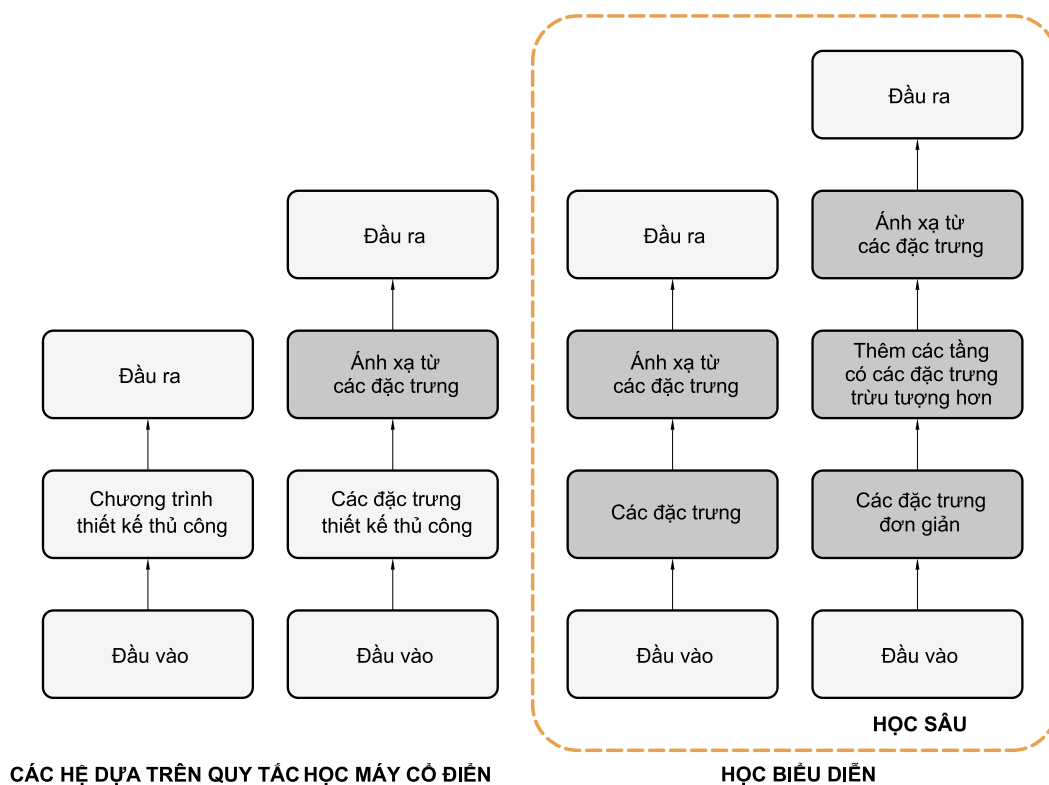
đầu sẽ chỉ nhìn thấy một mắt. Sau đó phát hiện rằng có một khuôn mặt, hệ thống có thể suy luận thêm là có thể tồn tại con mắt thứ hai. Trong trường hợp này, đồ thị các khái niệm chỉ bao gồm 2 tầng là mắt và khuôn mặt nhưng đồ thị các tính toán bao gồm $2n$ tầng nếu ta thực hiện n lần *tính chỉnh* ước lượng của ta về mỗi khái niệm khi biết các khái niệm khác.

Bởi vì rất khó xác định một trong hai cách - độ sâu của đồ thị tính toán hay độ sâu của mô hình đồ thị xác suất - cách nào phù hợp nhất, và bởi vì các cá nhân khác nhau chọn các thao tác cơ bản khác nhau để xây dựng đồ thị tính toán cho riêng mình, nên không có một giá trị độ sâu duy nhất cho một kiến trúc, tương tự với việc không có một giá trị duy nhất nào cho chiều sâu của một chương trình máy tính. Cũng như không có quy ước về độ sâu của một mô hình là bao nhiêu thì được xem là "sâu". Tuy nhiên, hoàn toàn có thể coi DL là lĩnh vực nghiên cứu những mô hình liên quan đến sự kết hợp một lượng lớn các hàm hoặc các khái niệm, lớn hơn các mô hình ML truyền thống.

Tóm lại, DL, chủ đề của cuốn sách, là một phương pháp tiếp cận AI. Cụ thể, nó là một loại ML, một kỹ thuật cho phép cải tiến hệ thống máy tính bằng tri thức và dữ liệu. Chúng tôi dám chắc rằng ML là một phương pháp tiếp cận khả thi duy nhất để xây dựng hệ thống AI có thể vận hành trong thế giới thực phức tạp. DL là một dạng cụ thể của ML, đạt được sức mạnh và sự mềm dẻo tuyệt vời thông qua việc biểu diễn thế giới như một hệ phân cấp các khái niệm, trong đó mỗi khái niệm được định nghĩa dựa trên mối liên hệ với những khái niệm đơn giản hơn, và nhiều cách biểu diễn trừu tượng có thể tính được từ những biểu diễn ít trừu tượng hơn. Hình 1.4 thể hiện mối liên hệ giữa những ngành AI khác nhau. Hình 1.5 mô tả sơ đồ ở mức cao về việc mỗi ngành hoạt động như thế nào.



Hình 1.4: Một biểu đồ Venn cho thấy DL là một dạng của học biểu diễn, và học biểu diễn lại là một dạng của ML, một trong những cách tiếp cận AI. Mỗi vùng của biểu đồ Venn bao gồm một ví dụ của một công nghệ AI.



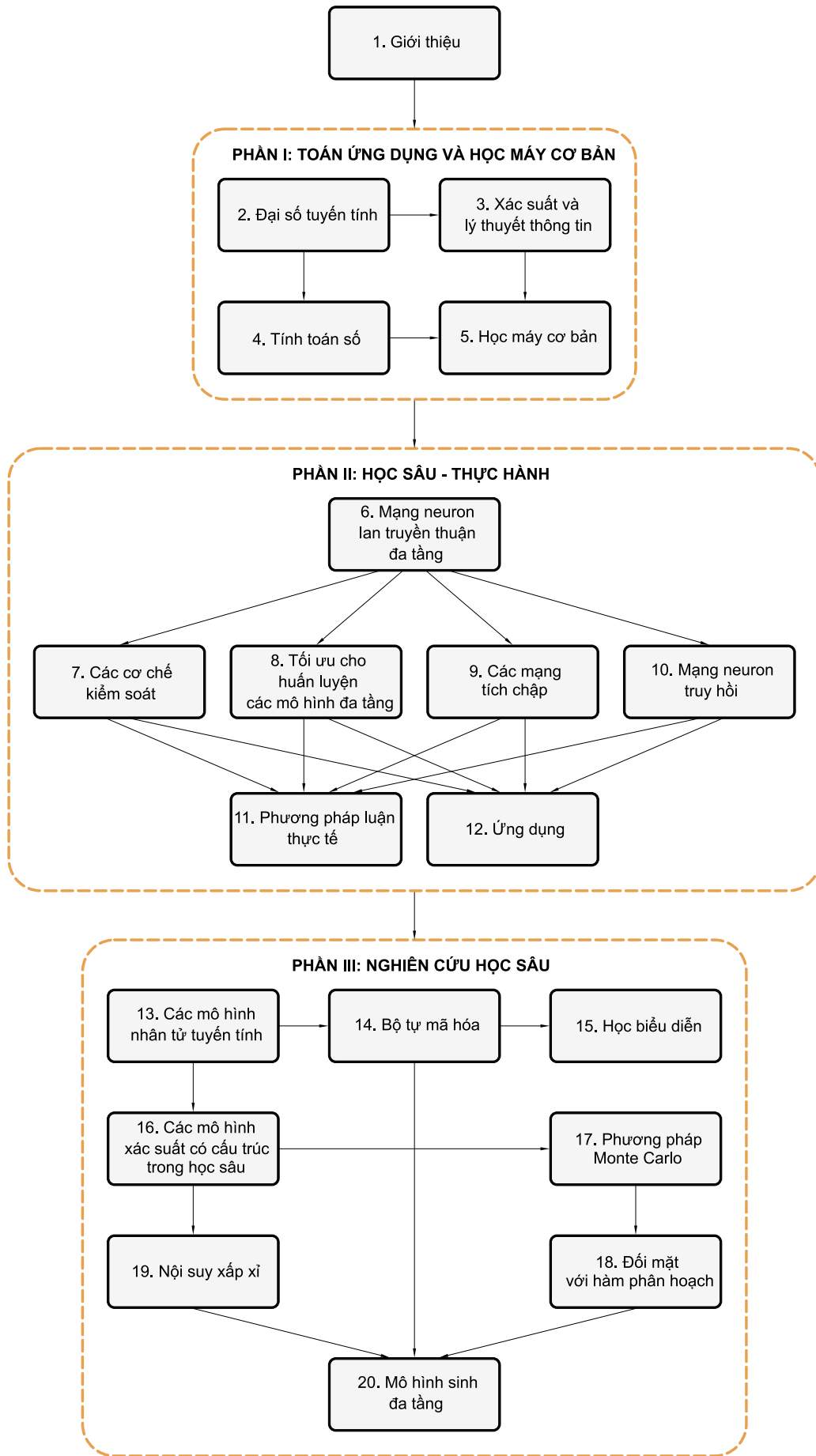
Hình 1.5: Biểu đồ cho biết các phần khác nhau của một hệ thống AI liên quan với nhau như thế nào bên trong mỗi ngành AI khác nhau. Hình vuông tô đậm là các thành phần ta có thể học được từ dữ liệu.

1.1 Ai nên đọc cuốn sách này?

Cuốn sách có thể hữu ích cho rất nhiều đối tượng khác nhau, nhưng chúng tôi hướng tới hai đối tượng chính. Thứ nhất là sinh viên các trường đại học hoặc sau đại học nghiên cứu về ML, bao gồm cả những người mới bắt đầu tìm hiểu về DL và AI. Thứ hai là những kĩ sư phần mềm, những người chưa có kiến thức nền tảng về ML, thống kê nhưng mong muốn nhanh chóng tiếp thu kiến thức về hai lĩnh vực này và bắt đầu sử dụng DL trong sản phẩm của mình. DL đã chứng minh được tính hữu ích của nó trong nhiều mảng, bao gồm computer vision, xử lí âm thanh và giọng nói, xử lí ngôn ngữ tự nhiên, robot, sinh-tin học và hóa học, trò chơi điện tử, máy tìm kiếm, quảng cáo trực tuyến và tài chính.

Cuốn sách này được chia thành ba phần để giúp cho nhiều đối tượng khác nhau một cách tốt nhất. Phần I giới thiệu công cụ toán học và những khái niệm ML cơ bản. Phần II mô tả những thuật toán DL phổ biến, là những công nghệ hữu dụng không thể thiếu. Phần III mô tả những ý tưởng mang tính suy đoán mà nhiều người tin rằng chúng có tầm quan trọng cho việc nghiên cứu DL trong tương lai.

Bạn có thể bỏ qua bất cứ phần nào trong sách mà bạn không quan tâm hoặc không phù hợp với kiến thức nền tảng của bạn. Ví dụ, bạn quen thuộc với đại số tuyến tính, xác suất, và khái niệm ML cơ bản có thể bỏ qua phần I, trong khi đó, những bạn chỉ muốn áp dụng ML vào sản phẩm sẽ không cần quan tâm tới phần III. Để giúp bạn lựa chọn chương cần đọc, chúng tôi cung cấp một sơ đồ tổng quan thể hiện bố cục nội dung trong sách.



Hình 1.6: Tổng quan bố cục nội dung cuốn sách. Mũi tên từ chương này tới chương khác có ý nghĩa cần phải đọc chương xuất phát của mũi tên mới có thể hiểu được chương mũi tên chỉ đến.

Chúng tôi tạm giả sử rằng tất cả các bạn đều có hiểu biết cơ bản về về khoa học máy tính như kiến thức lập trình, hiểu biết cơ bản về các vấn đề hiệu năng tính toán, lý thuyết độ phức tạp của thuật toán, các phương pháp tính cơ bản và một số thuật ngữ lý thuyết đồ thị.

1.2 Lịch sử các xu hướng trong DL

Cách dễ nhất để hiểu DL là tìm hiểu một vài bối cảnh lịch sử của nó. Thay vì liệt kê lịch sử của DL một cách chi tiết, chúng tôi đưa ra một vài điểm chính:

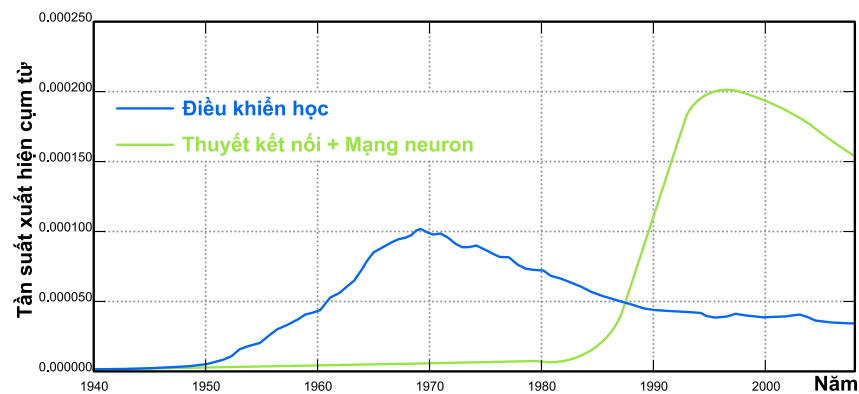
- DL có lịch sử lâu dài và phong phú, với nhiều tên gọi khác nhau, phản ánh những quan điểm triết học khác nhau, và đã trải qua những thời kì thịnh, suy.
- DL trở nên hữu ích hơn khi mà lượng dữ liệu huấn luyện ngày càng tăng.
- Kích thước mô hình DL ngày càng tăng do cơ sở hạ tầng máy tính (bao gồm phần cứng và phần mềm) ngày càng được cải thiện.
- DL đã và đang giải quyết những vấn đề phức tạp với độ chính xác ngày càng cao.

1.2.1 Các tên gọi của DL

Chúng tôi đoán rằng rất nhiều độc giả biết tới DL như là một công nghệ mới thú vị, và các bạn sẽ rất ngạc nhiên khi thấy cuốn sách đề cập tới "lịch sử" của một ngành mới. Thực tế, DL đã xuất hiện từ những năm 1940. DL có vẻ như là một đề tài mới, bởi vì nó ít được biết đến vài năm trước đây, trước khi nó trở nên nổi tiếng, và cũng vì có rất nhiều tên gọi khác nhau, chỉ gần đây thôi người ta mới gọi nó là Deep learning. Lĩnh vực này đã được đổi tên nhiều lần, phản ánh ảnh hưởng của nhiều nhà nghiên cứu khác nhau và nhiều góc nhìn khác nhau.

Việc mô tả đầy đủ, chi tiết về lịch sử của DL nằm ngoài phạm vi của cuốn sách này. Tuy nhiên, lướt qua vài dấu mốc lịch sử có thể hữu ích trong việc tìm hiểu DL. Nói chung, đã có ba làn sóng phát triển: DL được biết tới như là *điều khiển học* (cybernetics) những năm 1940-1960, sau đó DL được hiểu dưới góc nhìn

của *thuyết kết nối* (*connectionism*) những năm 1980-1990; và gần đây nổi lên dưới cái tên *Deep learning* từ năm 2006. Hình 1.7 sẽ minh họa rõ hơn một chút.



Hình 1.7: Hai làn sóng nghiên cứu mạng neuron nhân tạo, đo bằng tần suất xuất hiện từ "điều khiển học" và "thuyết kết nối" hay "mạng neuron", dựa trên Google Books (làn sóng thứ ba mới chỉ xuất hiện gần đây). Làn sóng đầu tiên bắt đầu bằng điều khiển học trong những năm 1940-1960, với sự phát triển của lý thuyết học tập sinh học [McCulloch and Pitts, 1943; Hebb, 1949] và các mô hình đầu tiên được thực thi như perceptron [Rosenblatt, 1958], cho phép huấn luyện một neuron. Làn sóng thứ hai bắt đầu với cách tiếp cận theo trường phái kết nối giai đoạn 1980-1995, đánh dấu sự ra đời của thuật toán lan truyền ngược [Rumelhart et al., 1986a] cho phép huấn luyện một mạng neuron có một hoặc hai hidden layer. Làn sóng thứ ba và cũng là làn sóng hiện tại, Deep learning, bắt đầu từ năm 2006 [Hinton et al., 2006; Bengio et al., 2007; Ranzato et al., 2007a] và mới chỉ xuất hiện trong sách vở, tính tới thời điểm năm 2016. Hai làn sóng đầu cũng đã xuất hiện trong sách vở rất lâu so với thời điểm các hoạt động nghiên cứu tương ứng bắt đầu.

Những thuật toán học sớm nhất mà ta biết được ngày nay mô phỏng các mô hình tính toán trong cơ chế học tập sinh học, những mô hình mô tả cách thức hoặc phỏng đoán về cách thức mà bộ não sử dụng để học tập. Kết quả là một trong những tên gọi mà DL đã từng có theo nghĩa này là *mạng neuron nhân tạo*. Góc nhìn tương ứng với tên gọi đó cho rằng những hệ thống DL được lấy cảm hứng từ bộ não sinh học. Mặc dù có những loại mạng neuron trong ML được sử dụng để tìm hiểu về chức năng bộ não [Hinton and Shallice, 1991], nhưng mạng neuron nói chung không được thiết kế theo mô hình chức năng sinh học thực thụ. Quan điểm neuron thần kinh trở thành một động lực cho DL xuất phát từ hai ý

tưởng chính. Một ý tưởng là não bộ bản thân nó chính là minh chứng cho việc tồn tại một hệ thống có hành vi thông minh, và một cách dễ nhận thấy để xây dựng một hệ thống thông minh là dịch ngược các nguyên lý tính toán bên trong não bộ tạo nên sự thông minh đó và sao chép chức năng của nó. Một ý tưởng khác, việc hiểu bộ não và các nguyên tắc làm nền tảng cho trí thông minh của con người là điều vô cùng thú vị, do vậy các mô hình ML ngoài khả năng giải quyết những vấn đề kỹ thuật còn có hữu ích trong việc giúp làm sáng tỏ nhiều câu hỏi khoa học cơ bản (về não bộ).

Thuật ngữ Deep learning hiện đại vượt xa quan điểm của khoa học thần kinh trong các mô hình ML hiện nay. Nó thiên về một nguyên tắc tổng quát hơn của việc học đó là học nhiều cấp độ nhỏ trong sự cấu thành của một sự vật hiện tượng. Nguyên tắc này có thể được áp dụng trong các lĩnh vực ML khác nhau, không nhất thiết chỉ cho các lĩnh vực lấy cảm hứng từ mạng neuron thần kinh.

Tiền thân của DL hiện đại là những mô hình tuyến tính đơn giản lấy cảm hứng từ khoa học thần kinh. Những mô hình đó được thiết kế để lấy một tập hợp giá trị đầu vào x_1, \dots, x_n và đi kèm với một giá trị đầu ra y . Chúng sẽ "học" ra một tập hợp *trọng số* (weight) w_1, \dots, w_n tính toán đầu ra của chúng $f(\mathbf{x}, \mathbf{w}) = x_1 w_1 + \dots + x_n w_n$. Lần sóng nghiên cứu mạng neuron đầu tiên này có tên là điều khiển học như minh họa trong hình 1.7.

Tế bào thần kinh McCulloch-Pitts [McCulloch và Pitts, 1943] là một mô hình chức năng não bộ xuất hiện từ sớm. Mô hình tuyến tính này đã có thể nhận biết hai danh mục dữ liệu khác nhau bằng cách kiểm tra xem $f(\mathbf{x}, \mathbf{w})$ là âm hay dương. Tất nhiên, để mô hình có thể xác định đúng danh mục của dữ liệu, ta cần thiết lập trọng số \mathbf{w} chính xác. Những trọng số có thể được con người thiết lập. Trong những năm 1950, thuật toán *perceptron* [Rosenblatt, 1958, 1962] đã trở thành mô hình đầu tiên có thể tự học những trọng số để xác định các danh mục từ dữ liệu đầu vào ứng với mỗi mục. Thuật toán *phần tử tuyến tính thích nghi* (*adaptive linear element* - *ADALINE*) ra đời cùng thời với thuật toán perceptron, trả về chính giá trị $f(\mathbf{x})$ để dự đoán một số thực [Widrow and Hoff, 1960] và có thể tự học để dự đoán những con số đó từ dữ liệu.

Những thuật toán học đơn giản này có ảnh hưởng rất lớn tới ML hiện đại. Thuật toán sử dụng để thay đổi trọng số trong ADALINE là một trường hợp đặc biệt, gọi là *trượt gradient ngẫu nhiên* (*stochastic gradient descent* - *SGD*). Các phiên

bản hơi khác của thuật toán SGD ngẫu nhiên vẫn là các thuật toán được sử dụng rộng rãi nhất trong các mô hình DL ngày nay.

Những mô hình dựa trên $f(\mathbf{x}, \mathbf{w})$ như đã được sử dụng bởi thuật toán perceptron và ADALINE được gọi chung là các mô hình tuyến tính. Những mô hình này vẫn được sử dụng rộng rãi trong các mô hình ML, qua nhiều trường hợp chúng được huấn luyện theo những cách khác nhau so với mô hình gốc.

Các mô hình tuyến tính có nhiều hạn chế. Hạn chế nổi tiếng nhất đó là chúng không thể học được hàm XOR, trong đó $f([0, 1], \mathbf{w}) = 1$ và $f([1, 0], \mathbf{w}) = 1$ nhưng $f([1, 1], \mathbf{w}) = 0$ và $f([0, 0], \mathbf{w}) = 0$. Một số nhà phê bình sau khi thấy được điểm yếu này đã phản ứng mãnh liệt chống lại các thuật toán học lấy cảm hứng từ sinh học [Minsky and Papert, 1969]. Đó là lần thoái trào đầu tiên trong quá trình phát triển của mạng neuron.

Ngày nay, *khoa học thần kinh (neuro-science)* vẫn là nguồn cảm hứng cho các nhà nghiên cứu về DL, nhưng nó không còn là ngọn hải đăng dẫn đường nữa.

Lí do chính cho việc sụt giảm vai trò của khoa học thần kinh là vì con người chưa có đủ thông tin về não bộ để có thể xem đó là kim chỉ nan. Muốn hiểu một cách sâu sắc các thuật toán mà não người sử dụng, ta cần quan sát hoạt động của ít nhất hàng ngàn tế bào thần kinh liên kết với nhau đồng thời - một nhiệm vụ bất khả thi. Do đó, ngay cả việc hiểu những thành phần đơn giản và đã được nghiên cứu kỹ lưỡng nhất trong não bộ cũng trở nên ngoài tầm với của khoa học hiện tại [Olshausen and Field, 2005].

Khoa học thần kinh thấp lè tè hi vọng rằng một thuật toán DL đơn lẻ có thể giải quyết nhiều nhiệm vụ khác nhau. Các nhà thần kinh học phát hiện ra rằng chồn có thể “nhìn” bằng vùng xử lý thính giác trong não bộ nếu não chúng được “đấu nối lại” để tín hiệu thị giác đi tới khu vực đó [Von Melchner et al., 2000]. Điều này cho thấy phần lớn não của các loài động vật có vú, có thể sử dụng một thuật toán đơn lẻ để giải quyết hầu hết những tác vụ nó cần thực hiện. Trước khi giả thuyết này được đưa ra, việc nghiên cứu ML bị phân hoá khá rõ rệt thành nhiều cộng đồng nghiên cứu: xử lý ngôn ngữ tự nhiên, thị giác máy tính, nhận dạng tiếng nói,.. Ngày nay, tuy những cộng đồng này vẫn còn tách biệt nhau, nhưng các nhóm nghiên cứu DL đã có thể làm việc với nhiều hoặc thậm chí là tất cả các lĩnh vực đó cùng lúc.

Ta có thể liệt kê ra một vài ý tưởng gợi ý từ khoa học thần kinh. Ý tưởng của việc nhiều đơn vị tính toán sẽ trở nên “thông minh” sau khi tương tác với nhau được

lấy cảm hứng từ bộ não. Dựa trên cấu trúc về hệ thống thị giác của động vật có vú, Fukushima đưa ra neocognitron [Fukushima 1980], một cấu trúc xử lý ảnh rất mạnh và sau này trở thành nền móng cho convolutional network hiện đại [LeCun et al., 1998b]; ta sẽ nói đến điều này trong phần 19.10. Hiện nay, hầu hết mạng neuron nhân tạo đều dựa trên mô hình neuron mang tên *bộ chỉnh lưu tuyến tính* (rectified linear unit – ReLU). Bản cognitron đầu tiên [Fukushima, 1975] là một bản phức tạp hơn và gần như hoàn toàn lấy cảm hứng từ các chức năng não bộ. Phiên bản hiện đại đơn giản hơn được phát triển dựa trên kết hợp nhiều ý tưởng từ các góc nhìn khác nhau: [Nair và Hilton, 2010] và [Glorot et al., 2011a] đều cho rằng khoa học thần kinh là một nguồn ảnh hưởng và [Jarrett et al., 2009] đưa ra các ảnh hưởng khác từ khía cạnh kỹ thuật. Dẫu cho khoa học thần kinh là một nguồn cảm hứng quan trọng, nhưng nó không phải thứ ta cần mô phỏng chính xác. Ta đã biết rằng các neuron thực sự của con người thực hiện việc tính toán rất khác so với các bộ chỉnh lưu tuyến tính hiện đại, nhưng các nỗ lực bắt chước các neuron nhiều hơn vẫn chưa mang đến sự cải thiện hiệu năng cho đến hiện tại. Mặc dù khoa học thần kinh đã thành công trong việc định hướng một vài cấu trúc mạng neuron, ta vẫn chưa đủ hiểu biết về *học tập sinh học* (biological learning) để áp dụng vào những *thuật toán tự học* (learning algorithms) khi huấn luyện các kiến trúc đó.

Giới truyền thông thường nhấn mạnh sự tương đồng giữa não bộ với DL. Thực tế là các nhà nghiên cứu DL lấy cảm hứng từ bộ não nhiều hơn so với những lĩnh vực ML khác như *mô hình hàm lõi* (*kernel machine*) hay thống kê Bayes, nhưng ta không nên coi DL như một phiên bản mô phỏng của não bộ. Các mô hình DL hiện đại hiện đang áp dụng ý tưởng từ nhiều lĩnh vực, đặc biệt là những nguyên tắc cơ bản của toán ứng dụng như đại số tuyến tính, xác suất, lý thuyết thông tin và toán tối ưu. Một số nhà nghiên cứu DL cho rằng khoa học thần kinh là nguồn cảm hứng quan trọng trong sự phát triển của DL, nhưng một số khác hoàn toàn không quan tâm đến khoa học thần kinh.

Cần lưu ý rằng, con người vẫn đang cố gắng hiểu cách não bộ làm việc ở cấp độ thuật toán. Đó là một lĩnh vực nghiên cứu riêng biệt với DL, thường được biết đến với tên gọi “khoa học thần kinh tính toán” (*computational neuroscience*). Chuyện các nhà nghiên cứu nhảy từ lĩnh vực này sang lĩnh vực kia và ngược lại giữa hai lĩnh vực này là chuyện rất dễ gặp. Ngành DL chủ yếu quan tâm tới làm thế nào để xây dựng được một hệ thống máy tính có khả năng giải thành công các tác vụ liên quan tới sự thông minh, trong khi ngành khoa học thần kinh tính

toán lại chủ yếu quan tâm tới xây dựng chính xác hơn các mô hình mô tả sự hoạt động của bộ não trong thực tế.

Trong thập niên 80, lần thứ hai làn sóng nghiên cứu mạng neuron trỗi dậy mạnh mẽ nhờ những trào lưu trong khoa học máy tính và công nghệ thông tin như *thuyết kết nối* (connectionism), hay *công nghệ xử lý phân tán song song* [Rumelhart et al., 1986c; McClelland et al., 1995]. Thuyết kết nối nổi lên trong bối cảnh khoa học nhận thức. Khoa học nhận thức là một cách tiếp cận liên ngành nghiên cứu tâm trí, kết hợp nhiều mức độ phân tích khác nhau. Đầu những năm 1980, hầu hết các nhà nghiên cứu khoa học nhận thức nghiên cứu các mô hình *suy luận ký hiệu* (symbolic reasoning). Dù rất phổ biến ở thời điểm bấy giờ, các mô hình ký hiệu cũng bế tắc trong việc giải thích cách hoạt động của não bộ sử dụng các neuron. Do đó, nhóm người theo thuyết kết nối đã bắt đầu nghiên cứu một mô hình nhận thức thực sự bắt nguồn từ các hoạt động thần kinh [Touretzky and Minton, 1985], qua đó khai quật nhiều ý tưởng của nhà tâm lý học Donald Hebb đưa ra trong những năm 1940 [Hebb, 1949].

Tư tưởng cốt lõi của *thuyết kết nối* là khi có một lượng lớn các đơn vị tính toán đơn giản được kết nối với nhau thành mạng lưới, chúng có thể trở nên thông minh. Khẳng định này đúng với cả các tế bào trong hệ thần kinh sinh học lẫn các *đơn vị ẩn* trong các mô hình tính toán.

Nhiều khái niệm chính xuất phát trong kỷ nguyên của thuyết kết nối vẫn đóng vai trò trọng tâm trong DL ngày nay. Một trong số đó là *biểu diễn phân tán* [Hinton et al., 1986]. Theo đó, mỗi điểm đầu vào của một hệ thống được biểu diễn bởi nhiều đặc trưng khác nhau, cũng như mỗi đặc trưng có thể tham gia biểu diễn nhiều điểm đầu vào. Một ví dụ đơn giản: giả sử ta có một hệ thống thị giác có thể nhận biết ô tô, xe tải và chim. Và những vật thể đó có thể mang màu đỏ, xanh lục hoặc xanh dương. Một cách để biểu diễn các đầu vào này là sử dụng một nơ-ron hoặc một đơn vị ẩn riêng biệt, mỗi nơ-ron hoặc đơn vị ẩn này sẽ kích hoạt một trong chín khả năng có thể xảy ra: ô tô đỏ, xe tải đỏ, chim đỏ, ô tô xanh, v.v. Như vậy, ta cần chín nơ-ron khác nhau để biểu diễn đầu vào, và mỗi nơ-ron phải học các sự kết hợp giữa màu sắc và vật thể một cách độc lập. Cách tốt hơn là sử dụng một biểu diễn phân tán, với ba neuron mô tả màu sắc và ba neuron mô tả đối tượng. Nghĩa là ta chỉ cần tổng cộng sáu neuron thay vì chín và neuron mô tả màu đỏ có thể học được màu đỏ từ ảnh ô tô, xe tải và ảnh chim, chứ không chỉ học từ ảnh của đúng một loại đối tượng. Khái niệm biểu diễn

phân tán chính là trọng tâm của cuốn sách này, và sẽ được mô tả chi tiết hơn trong chương 15.

Một thành tựu nổi bật khác của phong trào thuyết kết nối là việc áp dụng thành công thuật toán *lan truyền ngược* để huấn luyện các mạng neuron đa tầng cho phép tái biểu diễn bên trong mạng. Thuật toán này đã trải qua nhiều thăng trầm và tại thời điểm chúng tôi đang viết cuốn sách này (năm 2016), nó đang là phương pháp chủ đạo để huấn luyện các mô hình DL.

Đến những năm 1990, giới nghiên cứu đã có nhiều tiến bộ quan trọng trong việc mô hình hoá chuỗi bằng các mạng neuron. [Hochreiter, 1991; Bengio et al., 1994] đã chỉ ra những khó khăn về mặt toán học trong mô hình các chuỗi dài, như mô tả trong phần 10.7. [Hochreiter và Schmidhuber, 1997] đưa ra mạng LSTM (*Long short-term memory*) để giải quyết một vài khó khăn đó. Cho đến nay, LSTM đã được áp dụng rộng rãi vào nhiều tác vụ mô hình hoá chuỗi, bao gồm nhiều tác vụ xử lý ngôn ngữ tự nhiên tại Google.

Làn sóng nghiên cứu các mạng neuron kéo dài đến tận giữa 1990. Các dự án kêu gọi đầu tư mạo hiểm vào mạng neuron và các công nghệ AI khác bắt đầu có những phát biểu tham vọng không tưởng để tìm kiếm đầu tư. Khi các nghiên cứu AI không thể khóa lấp những kì vọng vô lý đó, các nhà đầu tư bắt đầu cảm thấy thất vọng. Giữa thời điểm đó, các ngành ML khác lại bắt đầu có những tiến bộ mới. *Mô hình hàm lõi (kernel machines)* [Boser et al., 1992; Cortes and Vapnik, 1995; Schölkopf et al., 1999] và các *mô hình đồ thị xác suất* [Jordan, 1998] đều đạt được kết quả tốt trên nhiều tác vụ quan trọng. Những yếu tố này làm cho quá trình phổ cập của mạng neuron đi vào thoái trào cho tới tận năm 2007.

Trong thời gian đó, mạng neuron tiếp tục đạt được hiệu suất ấn tượng trong nhiều tác vụ [LeCun et al., 1998b; Bengio et al., 2001]. Đóng góp rất lớn cho sự tồn tại và phát triển mạnh mẽ của mạng neuron là dự án mang tên *Tính toán Thần kinh và Nhận thức Thích nghi* (Neural Computation and Adaptive Perception - NCAP), được Viện Nghiên cứu Cao cấp Canada (CIFAR) khởi xướng. Chương trình này đã hợp nhất các nhóm nghiên cứu ML của Geoffrey Hinton tại Đại học Toronto, Yoshua Bengio tại Đại học Montreal, và Yann LeCun tại Đại học New York. CIFAR NCAP là một dự án nghiên cứu đa ngành, quy tụ nhiều nhà thần kinh học cũng như các chuyên gia hàng đầu trong lĩnh vực *thị giác máy tính*.

Lúc bấy giờ, huấn luyện các mạng DL vẫn được cho là quá khó. Hiện nay ta đều đã biết rằng các thuật toán ra đời từ những năm 1980 hoạt động khá tốt, nhưng phải đến tận những năm 2006 điều đó mới được chứng minh. Lý do đơn giản là vì các thuật toán này tốn nhiều tài nguyên tính toán so với nền tảng phần cứng bấy giờ.

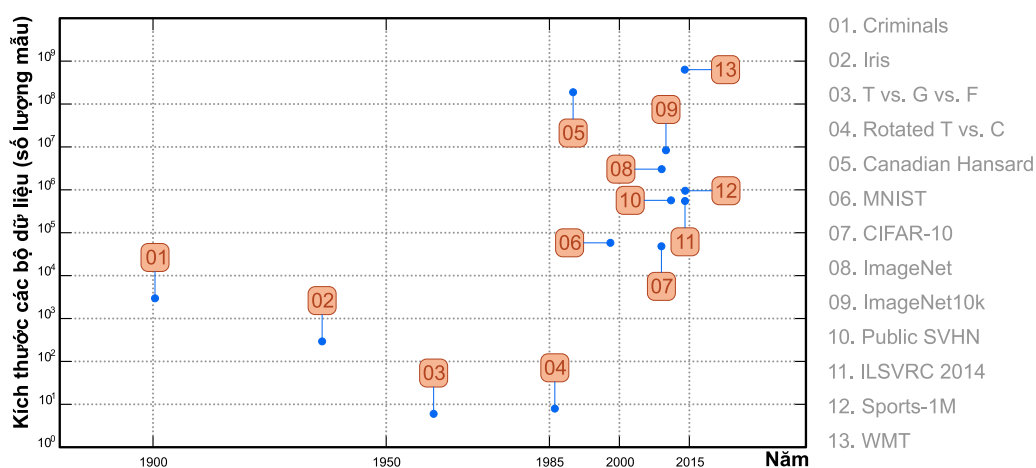
Làn sóng nghiên cứu mạng neuron lần thứ ba nổi lên với bước đột phá vào năm 2006. Geoffrey Hinton chứng minh rằng một loại mạng neuron có tên *mạng phân phối đa tầng* (deep belief network), một kiến trúc mạng neuron có thể được huấn luyện hiệu quả nhờ chiến lược mang tên *tiền huấn luyện tham lam theo lớp* (greedy layer-wise pre-training) [Hinton et al., 2006] mà chúng tôi sẽ mô tả chi tiết hơn trong phần 15.1. Sau đó, các nhóm nghiên cứu khác của CIFAR nhanh chóng chứng minh rằng có thể huấn luyện nhiều loại mạng đa tầng khác nhau với cùng một chiến lược [Bengio et al., 2007; Ranzato et al., 2007a], giúp cải thiện khả năng khái quát cho tập dữ liệu kiểm tra. Làn sóng này đã góp phần phổ cập việc sử dụng thuật ngữ DL để nhấn mạnh rằng các nhà nghiên cứu giờ đây có thể huấn luyện các mạng neuron sâu hơn rất nhiều so với trước đây, và tập trung sự chú ý vào tầm quan trọng về mặt lý thuyết của độ sâu trong mạng [Bengio và LeCun, 2007; Delalleau và Bengio, 2011; Pascanu et al., 2014a; Montufar et al., 2014]. Ở thời điểm đó, mạng neuron đa tầng tỏ ra vượt trội so với các hệ thống AI dựa trên những công nghệ ML khác cũng như các hệ chức năng được thiết kế thủ công. Làn sóng nghiên cứu các mạng neuron lần thứ ba vẫn đang tiếp diễn cho đến thời điểm chúng tôi viết nên cuốn sách này, dù trọng tâm của nghiên cứu DL đã thay đổi đáng kể trong suốt thời gian ấy. Làn sóng thứ ba ban đầu tập trung vào các kỹ thuật học không giám sát mới và tập trung vào nghiên cứu làm thế nào các mạng sâu có thể tổng quát hóa tốt trên các tập dữ liệu nhỏ, nhưng ngày nay giới nghiên cứu đang dành nhiều sự quan tâm hơn cho các thuật toán học có giám sát cổ điển và tiềm năng của mô hình DL khi tận dụng các bộ dữ liệu lớn gán nhãn sẵn.

1.2.2 Kích thước dữ liệu ngày càng tăng

Người ta có thể thắc mắc lý do tại sao DL chỉ mới được thừa nhận là một công nghệ đóng vai trò quan trọng trong thời gian gần đây, mặc dù các thử nghiệm đầu tiên với mạng neuron nhân tạo đã được thực hiện vào những năm 1950. DL đã được áp dụng thành công vào các ứng dụng thương mại từ thập niên 90, tuy

nhiên lúc bấy giờ nó thường được coi là một nghệ thuật hơn là một công nghệ, một thứ gì đó xa vời mà chỉ có những chuyên gia hàng đầu mới có thể sử dụng, cho tới gần đây. Trên thực tế thì đúng là cần có vài kỹ năng để đạt được kết quả khả quan khi sử dụng các thuật toán DL. May mắn thay, số kỹ năng cần thiết ấy đã giảm đáng kể khi lượng dữ liệu huấn luyện ngày càng tăng. Hiện nay, nhiều thuật toán đã có thể đạt được hiệu suất của con người trong các tác vụ phức tạp nhưng lại gần giống với các thuật toán đã phải vật lộn với những vấn đề trẻ con ở thập niên 80, mặc dù các mô hình mà ta huấn luyện bằng cùng thuật toán đó đã trải qua nhiều cải tiến, giúp đơn giản hoá việc huấn luyện các kiến trúc đa tầng.

Sự phát triển quan trọng nhất gần đây đó là ta đã có thể cung cấp cho các thuật toán này những nguồn lực mà chúng cần để thành công. Hình 1.8 cho thấy kích thước của các bộ dữ liệu tiêu chuẩn tăng nhanh đến thế nào qua thời gian. Xu hướng này được thúc đẩy bởi quá trình số hoá của xã hội ngày càng lớn. Con người ngày càng có nhiều hoạt động diễn ra trên máy tính, và càng nhiều thứ ta làm được ghi lại. Khi máy tính của ta càng ngày càng được kết nối với nhau, việc thu thập nhiều bản ghi dữ liệu và tập hợp chúng thành một tập dữ liệu phù hợp cho các ứng dụng ML càng trở nên đơn giản.



Hình 1.8: Sự tăng trưởng của dữ liệu qua thời gian. Trong đầu những năm 1900, các nhà thống kê đã nghiên cứu các tập dữ liệu sử dụng hàng trăm hay hàng nghìn phép đo được thực hiện thủ công [Garson, 1900; Gosset, 1908; Anderson, 1935; Fisher, 1936]. Trong thập niên 50 đến đầu thập niên 80, những người tiên phong trong lĩnh vực ML lấy cảm hứng từ sinh học thường làm việc với các bộ dữ liệu tổng hợp nhỏ, chẳng hạn như các ảnh bitmap độ phân giải thấp của các ký tự, được thiết kế để chịu chi phí tính

toán thấp và chứng minh rằng các mạng neuron có thể học các loại hàm cụ thể [Widrow and Hoff, 1960; Rumelhart et al., 1986b]. Trong những năm 1980 và 1990, ML mang ảnh hưởng của toán thống kê nhiều hơn và nhờ đó có thể tận dụng hiệu quả những tập dữ liệu lớn chứa hàng chục ngàn điểm dữ liệu, chẳng hạn như bộ dữ liệu chữ số viết tay MNIST (hình 1.9) [LeCun et al., 1998b]. Trong thập niên đầu của thế kỷ 21, nhiều bộ dữ liệu phức tạp hơn với cùng kích thước tiếp tục được tạo ra, chẳng hạn CIFAR-10 [Krizhevsky and Hinton, 2009]. Đến cuối thập niên đó và trong suốt những nửa đầu những năm 2010, kích thước các bộ dữ liệu đã lớn hơn đáng kể, chứa hàng trăm ngàn đến hàng chục triệu điểm dữ liệu, nhờ đó hoàn toàn thay đổi những gì DL có thể làm được. Trong đó bao gồm tập dữ liệu số nhà công khai bởi Google Street View [Netzer et al., 2011], nhiều phiên bản của bộ dữ liệu ImageNet [Deng et al., 2009, 2010a; Russakovsky et al., 2014a], và dữ liệu Sports-1M [Karpathy et al., 2014]. Ở phía trên cùng của biểu đồ, có thể thấy rằng các tập dữ liệu các câu văn đã được dịch, chẳng hạn như tập dữ liệu của IBM được xây dựng từ Canadian Hansard [Brown et al., 1990] và tập dữ liệu WMT 2014 từ tiếng Anh sang tiếng Pháp [Schwenk, 2014] thường vượt xa kích thước của các bộ dữ liệu khác.

8	9	0	1	2	3	4	7	8	9	0	1	2	3	4	5	6	7	8	6
4	2	6	4	7	5	5	4	7	8	9	2	9	3	9	3	8	2	0	5
0	1	0	4	2	6	5	3	5	3	8	0	0	3	4	1	5	3	0	8
3	0	6	2	7	1	1	8	1	7	1	3	8	9	7	6	7	4	1	6
7	5	1	7	1	9	8	0	6	9	4	9	9	3	7	1	9	2	2	5
3	7	8	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	0
1	2	3	4	5	6	7	8	9	8	1	0	5	5	1	9	0	4	1	9
3	8	4	7	7	8	5	0	6	5	5	3	3	3	9	8	1	4	0	6
1	0	0	6	2	1	1	3	2	8	8	7	8	4	6	0	2	0	3	6
8	7	1	5	9	9	3	2	4	9	4	6	5	3	2	5	5	9	4	1
6	5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5	6	7	8	9	6	4	2	6	4	7	5	5
4	7	8	9	2	9	3	9	3	8	2	0	9	8	0	5	6	0	1	0
4	2	6	5	5	5	4	3	4	1	5	3	0	8	3	0	6	2	7	1
1	8	1	7	1	3	8	5	4	2	0	9	7	6	7	4	1	6	8	4
7	5	1	2	6	7	1	9	8	0	6	9	4	9	9	6	2	3	7	1
9	2	2	5	3	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3
4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	2	1	2	1	3
9	9	8	5	3	7	0	7	7	5	7	9	9	4	7	0	3	4	1	4
4	7	5	8	1	4	8	4	1	8	6	6	4	6	3	5	7	2	5	9

Hình 1.9: Một số ví dụ đầu vào của bộ dữ liệu MNIST. Trong đó, "NIST" là viết tắt cho Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (National Institute of Standards and Technology), cơ quan đầu tiên thu thập bộ dữ liệu này, và "M" là viết tắt của từ "đã được điều chỉnh" (modified) bởi dữ liệu đã được tiền xử lý để phù hợp hơn với các thuật toán ML. Bộ dữ liệu MNIST bao gồm các bản quét (scan) của các chữ số viết tay được gán nhãn từ 0 đến 9 trong mỗi ảnh. Cho đến nay bài toán phân loại đơn giản này là một trong những phép kiểm tra được sử dụng rộng rãi nhất trong giới nghiên cứu DL, dẫn cho các kỹ thuật hiện đại ngày nay đã giải quyết nó một cách đơn giản. Geoffrey Hinton từng ví von MNIST là "ruồi giấm trong ML", có nghĩa là nó cho phép các nhà nghiên cứu ML dễ dàng thử nghiệm thuật toán của họ giống như cách mà các nhà sinh học thực hiện thí nghiệm trên ruồi giấm.

Kỷ nguyên của "dữ liệu lớn" là đòn bẩy đưa ML lên tầm cao mới, bởi gánh nặng chính của bài toán ước lượng thống kê - khái quát hoá dữ liệu mới sau khi chỉ quan sát một lượng dữ liệu nhỏ - đã nhẹ nhàng hơn rất nhiều. Tính đến năm 2016, có một nguyên tắc chung là các thuật toán DL có giám sát có thể đạt được hiệu suất chấp nhận được với khoảng 5,000 điểm dữ liệu cho mỗi nhãn, và sẽ chạm tới hoặc thậm chí vượt qua hiệu suất của con người khi được huấn luyện với bộ dữ liệu chứa ít nhất 10 triệu điểm dữ liệu đã gán nhãn. Tuy nhiên, làm sao để làm việc thành công với những bộ dữ liệu nhỏ hơn nhiều là một lĩnh vực nghiên cứu quan trọng, tập trung chủ yếu vào câu hỏi làm sao ta có thể tận dụng được một lượng lớn dữ liệu không có nhãn, sử dụng học không giám sát hoặc bán giám sát.

1.2.3 Kích thước mô hình ngày càng tăng

Một lý do quan trọng khác khiến mạng neuron cực kỳ thành công như ngày nay sau một khoảng thời gian nổi lên từ những năm 1980, thời gian mà mạng neuron chỉ có được những thành tựu khiêm tốn, là nhờ vào việc ta đã đạt được những thành công về phát triển phần cứng chạy tốt các mô hình lớn hơn rất nhiều. Một trong những bài học chính rút ra từ thuyết kết nối là động vật trở nên thông minh khi nhiều tế bào thần kinh của chúng hoạt động cùng nhau. Một tế bào thần kinh riêng lẻ hoặc một tập nhỏ các tế bào thần kinh không hữu ích lắm trong thực tế.

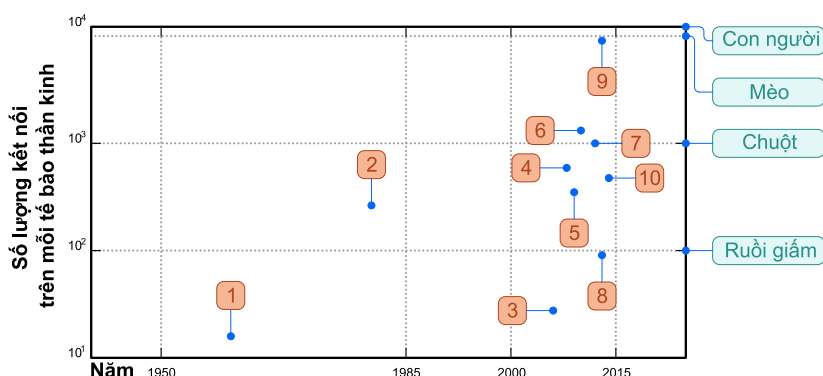
Các tế bào thần kinh sinh học được kết nối với mật độ không quá dày đặc. Như đã thấy trong hình 1.10, các mô hình ML của ta đã đạt được số lượng các kết nối trên một tế bào thần kinh ngang với não bộ của một số loài động vật có vú từ nhiều thập kỷ trước.

Như trong hình 1.11, ngày nay, nếu xét về số lượng các tế bào thần kinh, các mô hình mạng neuron trong quá khứ có số lượng tế bào ít đáng kinh ngạc và điều này chỉ thay đổi trong khoảng thời gian gần đây, khi kích thước mạng ngày càng tăng. Kể từ khi đưa ra khái niệm về các đơn vị ẩn, mạng neuron nhân tạo đã tăng gấp đôi kích thước sau mỗi 2,4 năm. Sự tăng trưởng này nhờ vào hai yếu tố: các hệ thống máy tính đang trở nên mạnh mẽ hơn với bộ nhớ lớn hơn và ta đang có các bộ dữ liệu lớn hơn. Các mạng có kích thước lớn hơn có thể đạt được độ chính xác cao hơn đối với các tác vụ phức tạp. Xu hướng này có thể sẽ còn tiếp tục trong nhiều thập kỷ tới. Các mạng neuron nhân tạo sẽ chưa thể đạt được số lượng tế bào thần kinh như não bộ của người ít nhất là cho đến năm 2050, trừ

khi các công nghệ mới xuất hiện cho phép sự tăng trưởng này diễn ra nhanh hơn. Các tế bào thần kinh sinh học có thể biểu diễn các chức năng phức tạp hơn các tế bào thần kinh nhân tạo hiện tại, vì vậy nhiều mạng neuron sinh học thậm chí có thể lớn hơn mạng trong hình 1.11.

Khi nhìn lại, ta sẽ thấy không có gì phải ngạc nhiên khi mạng neuron với ít tế bào thần kinh hơn so với một con đĩa sẽ không thể giải quyết được các vấn đề trí tuệ nhân tạo tinh vi. Ngay cả những mạng mới xuất hiện gần đây, nếu nhìn từ quan điểm của hệ thống tính toán thì chúng đã có số lượng tế bào thần kinh khá lớn, nhưng vẫn còn nhỏ hơn hệ thần kinh của các loài động vật có xương sống tương đối nguyên thủy như ếch.

Kích thước mô hình ngày càng tăng theo thời gian nhờ vào việc CPU trở nên nhanh hơn, sự ra đời của các GPU có thể sử dụng đa mục đích (sẽ được mô tả chi tiết trong phần 12.1.2), kết nối mạng nhanh hơn và cơ sở hạ tầng phần mềm tốt hơn cho các hệ thống máy tính phân tán, là một trong những xu hướng quan trọng nhất trong lịch sử của DL. Xu hướng này được dự kiến sẽ tiếp tục phát triển tốt trong tương lai.



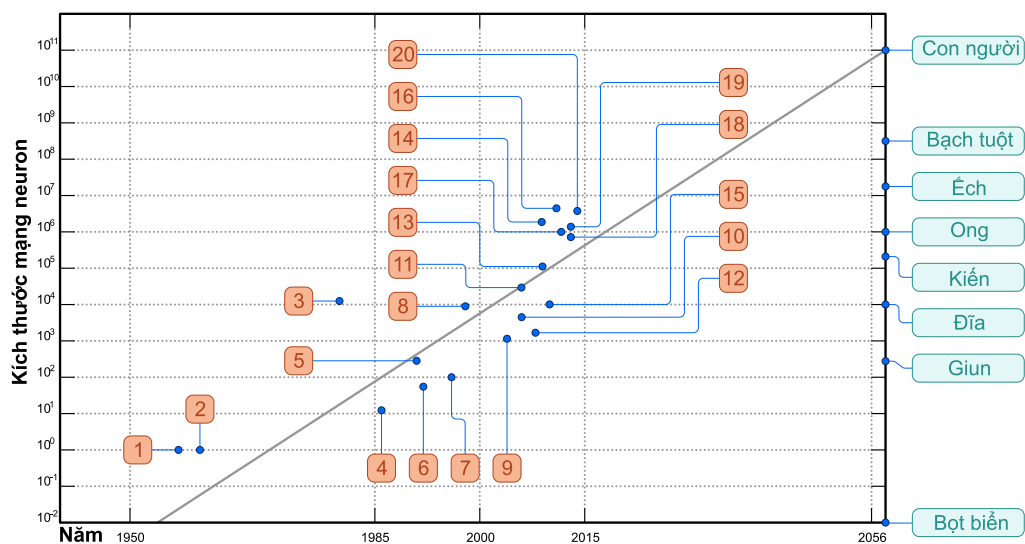
Hình 1.10: Số lượng các kết nối trên mỗi tế bào thần kinh theo thời gian. Ban đầu, số lượng kết nối giữa các tế bào thần kinh trong các mạng neural nhân tạo bị giới hạn bởi khả năng làm việc của phần cứng. Ngày nay, số lượng kết nối giữa các tế bào thần kinh hầu như được xem xét như một phần của việc thiết kế mô hình. Một số mạng neuron nhân tạo có số lượng kết nối trên mỗi neuron gần giống như một con mèo, phổ biến hơn là các mạng neuron có số lượng kết nối trên mỗi neuron tương đương như các động vật có vú nhỏ hơn như chuột. Ngay cả bộ não của con người cũng không có số lượng kết nối quá lớn cho mỗi tế bào thần kinh. Kích thước mạng neuron sinh học lấy từ Wikipedia [2015].

1. Phần tử tuyến tính thích nghi [Widrow and Hoff, 1960]

2. Neocognitron [Fukushima, 1980]
3. Mạng tích chập tăng tốc bởi GPU [Chellapilla et al., 2006]
4. Máy Boltzmann đa tầng [Salakhutdinov and Hinton, 2009a]
5. Mạng tích chập không giám sát [Jarrett et al., 2009]
6. Mạng perceptron đa tầng tăng tốc bởi GPU [Ciresan et al., 2010]
7. Bộ tự mã hóa phân tán [Le et al., 2012]
8. Mạng tích chập đa GPU [Krizhevsky et al., 2012]
9. Mạng tích chập không giám sát COTS HPC [Coates et al., 2013]
10. GoogLeNet [Szegedy et al., 2014a]

1.2.4 Độ phức tạp, độ chính xác và mức độ ảnh hưởng thực tế ngày càng tăng

Từ những năm thập niên 80, DL đã được cải thiện liên tục trong khả năng dự đoán và nhận dạng có độ chính xác cao. Hơn nữa, DL ngày càng được áp dụng rộng rãi hơn trong nhiều ứng dụng một cách rất thành công.



Hình 1.11: Kích thước mạng neural theo thời gian. Kể từ khi đưa ra khái niệm về các đơn vị ẩn, mạng neuron nhân tạo đã tăng gấp đôi kích thước, khoảng 2,4 năm một lần. Kích thước mạng neuron sinh học lấy từ Wikipedia [2015].

Các mô hình đa tầng đầu tiên được sử dụng để nhận biết các vật thể riêng lẻ trong hình ảnh đã được cắt để chỉ còn chứa những vật thể cần được nhận diện

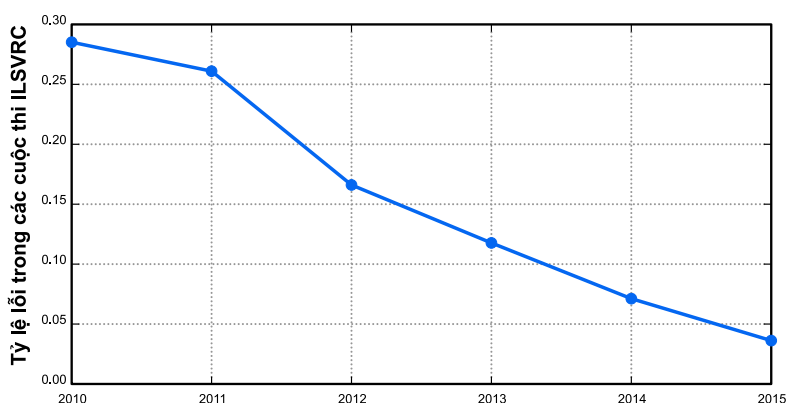
và chúng có kích thước rất nhỏ [Rumelhart et al. 1986]. Kể từ đó đã có sự tăng dần về kích thước ảnh mà các mạng neuron có thể xử lý được. Các mạng nhận diện vật thể hiện đại xử lý các bức ảnh có độ phân giải cao và không yêu cầu ảnh bị cắt gần đối tượng cần nhận dạng [Krizhevsky et al. 2012]. Tương tự như vậy, các mạng đầu tiên có thể nhận diện chỉ hai loại đối tượng (hoặc trong một số trường hợp, sự vắng mặt hoặc sự hiện diện của một loại đối tượng duy nhất), trong khi các mạng hiện đại ngày nay có khả năng nhận diện ít nhất 1.000 loại đối tượng khác nhau. *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) là cuộc thi lớn nhất về nhận dạng vật thể được tổ chức hàng năm. Một khoảnh khắc ấn tượng báo hiệu cho sự trỗi dậy của deep learning là khi mạng tích chập lần đầu tiên giành chiến thắng với một khoảng cách khá xa với mô hình xếp thứ hai, giảm tỷ lệ lỗi top-5 tốt nhất thời điểm đó từ 26.1% xuống 15.3% [Krizhevsky et al. 2012], nghĩa là mạng tích chập sẽ đưa ra một danh sách dự đoán nhãn của bức ảnh theo thứ tự độ tin cậy giảm dần nếu nhãn chính xác của bức ảnh nằm trong năm dự đoán đầu tiên thì được tính là một lần dự đoán chính xác và ngược lại được tính là một lần dự đoán lỗi, mô hình đã đạt kết quả độ lỗi 15.3% trên tập kiểm tra theo cách tính trên. Từ đó, các cuộc thi tiếp theo liên tục ghi nhận sự chiến thắng của mạng neuron tích chập đã đưa tỷ lệ lỗi top-5 xuống 3.6%, tính tới thời điểm cuốn sách này đang được viết, như mô tả trong hình 1.12.

DL cũng có tác động đáng kể đến bài toán nhận dạng giọng nói. Sau khi cải thiện trong suốt những năm 1990, tỷ lệ lỗi cho bài toán nhận dạng giọng nói gần như không có sự cải thiện đáng kể nào vào khoảng những năm 2000. Sự ra đời của DL [Dahl et al. 2010, Denget al. 2010, Seide et al. 2011, Hinton et al. 2012] giúp bài toán nhận dạng giọng nói giảm tỷ lệ lỗi một cách tuyệt vời, với một số tỷ lệ lỗi giảm một nửa. ta sẽ khám phá điều này chi tiết hơn trong phần 12.3.

Những mô hình mạng đa tầng cũng đã đạt được những thành công ngoạn mục trong bài toán phát hiện người đi bộ và phân vùng ảnh (image segmentation) [Sermanet et al. 2013, Farabet et al. 2013, Couprie et al. 2013] và đạt được độ chính xác hơn hẳn con người trong bài toán phân loại biển báo giao thông [Ciresanet al. 2012].

Mạng đa tầng ngày càng có khả năng giải quyết được nhiều tác vụ phức tạp hơn khi quy mô và độ chính xác của chúng tăng lên. Goodfellow [Goodfellow et al. 2014] đã chỉ ra rằng các mạng neuron có khả năng đưa ra một đoạn văn bản miêu tả nội dung một bức ảnh thay vì chỉ dừng lại ở việc xác định những đối

tượng nào xuất hiện trong bức ảnh đó. Trước đây, nhiều người tin rằng cách học như trên đòi hỏi một chuỗi dữ liệu phải có nhãn riêng lẻ cho từng phần tử của chúng [Gülçehre và Bengio, 2013]. Các mạng neuron hồi quy (*recurrent neural network*), như mô hình LSTM vừa nói đến ở phần trên, hiện đang được sử dụng để mô hình hóa các mối quan hệ giữa các chuỗi với nhau thay vì một đầu vào cố định. Thuật toán *học chuỗi-tới-chuỗi* (sequence-to-sequence learning) đang là một phương pháp học đang tạo ra cuộc cách mạng cho ứng dụng rất nổi tiếng hiện nay: dịch máy (machine translation) [Sutskever et al. 2014, Bahdanau et al. 2015].



Hình 1.12: Tỷ lệ lỗi giảm dần theo thời gian. Kể từ khi các mạng đa tầng đạt đến quy mô cần thiết để cạnh tranh trong cuộc thi ILSVRC, chúng đã liên tục giành được giải cao nhất mỗi năm với tỷ lệ lỗi ngày càng giảm dần. Dữ liệu lấy từ [Russakovsky et al., 2014, He et al., 2015].

Xu hướng tăng dần độ phức tạp này đã đưa đến sự ra đời của *máy Turing neuron* (Neural Turing machine) [Graves et al. 2014], một mô hình có thể học cách đọc và ghi nội dung tùy ý vào các tế bào bộ nhớ. Những mạng neuron như thế có thể tự học để tạo ra các chương trình đơn giản từ những ví dụ mẫu mô tả các hành vi mong muốn. Ví dụ, một mô hình có thể học các sắp xếp các chữ số từ những chuỗi đã được sắp xếp đã từ trước. Công nghệ tự lập trình này đang trong giai đoạn sơ khai, nhưng trong tương lai, về nguyên tắc chúng có thể được áp dụng cho hầu hết mọi công việc.

Một thành tựu vượt trội của DL là sự mở rộng của nó đến lĩnh vực *học củng cố* (Reinforcement Learning). Đối với học củng cố, một tác nhân tự chủ học cách thực hiện một nhiệm vụ bằng cách thử và sai (trial and error) mà không có sự

hướng dẫn nào từ con người. DeepMind đã công bố một hệ thống học củng cố dựa trên DL có khả năng học được cách chơi các trò trong các game của *Atari*, ngoài ra hệ thống này còn đạt được hiệu suất tương đương con người ở nhiều nhiệm vụ khác nhau [Mnih et al., 2015]. Trong lĩnh vực nghiên cứu về robot (robotics) DL đồng thời đã cải thiện đáng kể hiệu suất của học củng cố [Finn et al., 2015].

Nhiều ứng dụng của DL mang lại lợi nhuận rất cao. Hiện nay, DL được sử dụng bởi nhiều hãng công nghệ hàng đầu trên thế giới, gồm Google, Microsoft, Facebook, IBM, Baidu, Apple, Adobe, Netflix, NVIDIA và NEC.

Những tiến bộ trong mảng DL đã và đang phụ thuộc rất lớn vào những tiến bộ trong việc phát triển kiến trúc hạ tầng phần mềm. Các thư viện phần mềm như *Theano* [Bergstra et al. 2010, Bastien et al., 2012], *PyLearn2* [Goodfellow et al., 2013], *Torch* [Collobert et al., 2011], *DistBelief* [Dean et al., 2012], *Caffe* [Jia, 2013], *MXNet* [Chen et al., 2015], và *TensorFlow* [Abadi et al., 2015] đã hỗ trợ rất lớn cho các dự án nghiên cứu hoặc các sản phẩm thương mại.

DL cũng đã đóng góp cho nhiều thành tựu của các ngành khoa học khác. Mạng neural tích chập dùng trong bài toán nhận dạng vật thể đã cung cấp một mô hình xử lý hình ảnh mà các nhà thần kinh học có thể nghiên cứu và áp dụng [DiCarlo 2013]. DL cũng cung cấp các công cụ hữu ích để xử lý một lượng lớn dữ liệu và đưa ra các dự đoán hữu ích trong các lĩnh vực khoa học. Nó đã được áp dụng rộng rãi và thành công trong dự đoán các phân tử tương tác như thế nào để giúp các công ty dược phẩm điều chế các loại thuốc mới [Dahl et al., 2014], giúp tìm kiếm các hạt hạ nguyên tử [Baldi et al., 2014], xây dựng bản đồ 3-D về bộ não con người từ ảnh chụp của kính hiển vi [Knowles-Barley et al., 2014]. Chúng tôi hy vọng rằng trong tương lai DL sẽ xuất hiện trong ngày càng nhiều những lĩnh vực khoa học khác nhau.

Tóm lại, DL là một cách tiếp cận của ML, sử dụng rất nhiều kiến thức của ta về não bộ con người, về thống kê và toán ứng dụng để phát triển trong suốt những thập kỷ qua. Trong những năm gần đây, DL đã cho thấy sự tăng trưởng vượt bậc về mức độ phổ biến và sự hữu ích của nó. Các kết quả mà nó có được phần lớn là nhờ vào những yếu tố: phần cứng máy tính ngày càng mạnh mẽ hơn, các bộ dữ liệu lớn hơn và các kỹ thuật cao cấp để huấn luyện các mạng neuron nhiều tầng hơn. Những năm tiếp theo sẽ mở ra nhiều thách thức và cơ hội để ta cải thiện các mô hình DL, và đưa nó vượt qua những giới hạn mới.