PHÂN TÍCH
MÔ HÌNH HỌC SÂU

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

PGS.TS. PHÙNG TRUNG NGHĨA - TS. NGUYỄN THẾ VỊNH
(ĐỒNG CHỦ BIÊN)

PHÂN TÍCH MÔ HÌNH HỌC SÂU

NHÀ XUẤT BẢN KHOA HỌC - CÔNG NGHỆ - TRUYỀN THÔNG

MỤC LỤC

Từ viết tắt 8

Lời nói đầu 10

Chương 1

NÃO, LOGIC VÀ GIẤC MƠ MÔ HÌNH HÓA TRÍ TUỆ

1.1. Từ ước mơ hiểu trí tuệ đến não và tế bào 11

1.2. Logic hình thức như cây cầu mô hình hóa 15

1.3. McCulloch-Pitts và nơ-ron ngưỡng 19

1.4. AND, OR, NOT và giới hạn của mạch đã đấu dây 22

1.5. Từ ENIAC đến EDVAC: Khi nơ-ron logic gặp máy tính điện tử 25

CÂU HỎI ÔN TẬP VÀ BÀI TẬP 30

Chương 2

KẾT NỐI BIẾT HỌC

2.1. Hebb và kinh nghiệm: Từ mạch đặt sẵn đến dấu vết kết nối 31

2.2. Cell assembly: Trí nhớ như mẫu hoạt động phân tán 37

2.3. Máy tính số như phòng thí nghiệm mới 39

2.4. Trọng số: Con số hóa dấu vết kinh nghiệm 42

2.5. Bài học phân tích: Tham số là nơi kinh nghiệm nắn hình 46

CÂU HỎI ÔN TẬP VÀ BÀI TẬP 50

Chương 3

PERCEPTRON VÀ KHUÔN TUYẾN TÍNH

3.1. Rosenblatt và câu hỏi nhận dạng 51

3.2. Xác suất và kết nối ngẫu nhiên 54

3.3. Từ tín hiệu tới điểm số 56

3.4. Lớp hàm tuyến tính 59

3.5. Học trong khuôn tuyến tính 62
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 65

Chương 4

XOR, TẦNG ẨN VÀ VẤN ĐỀ BIỂU DIỄN

4.1. Khi lỗi nằm ở khuôn mô hình 66
4.2. XOR: Bốn điểm làm lộ giới hạn tuyến tính 70
4.3. Tầng ẩn như cách tạo điều kiện trung gian 74
4.4. Nhiều vùng quyết định từ nhiều nhát cắt 78
4.5. Tầng ẩn và vấn đề gán trách nhiệm 80
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 85

Chương 5

LAN TRUYỀN NGƯỢC VÀ HỌC NHIỀU TẦNG

5.1. Vấn đề gán trách nhiệm trong mạng nhiều tầng 86
5.2. Lượt tiến và lượt ngược 89
5.3. Quy tắc dây chuyền và ảnh hưởng gián tiếp 94
5.4. Kích hoạt mượt và gradient 97
5.5. Bài học từ bài Rumelhart-Hinton-Williams 100
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 104

Chương 6

HÀM LỖI, TỐI ƯU VÀ KHÁI QUÁT HÓA

6.1. Hàm lỗi: Biến mục tiêu thành đại lượng có thể sửa 105
6.2. Hạ dốc gradient: đi trong địa hình lỗi 112
6.3. Lô dữ liệu và bộ tối ưu 116
6.4. Huấn luyện, thẩm định và kiểm tra 119
6.5. Chính quy hóa và bài học phân tích 122
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 129

Chương 7

THỊ GIÁC, TÍCH CHẬP VÀ THIÊN KIẾN KHÔNG GIAN

7.1. Ảnh không phải chỉ là một vector dài 130

7.2. Tích chập: Kernel, stride, padding và kênh 133
7.3. Từ LeNet đến AlexNet: Dữ liệu, GPU và quy mô 136
7.4. Độ sâu, suy giảm và kết nối tắt 139
7.5. Bài học phân tích mô hình thị giác 141
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 149

Chương 8

CHUỖI, TRÍ NHỚ VÀ CƠ CHẾ CHÚ Ý

8.1. Dữ liệu chuỗi và nhu cầu bộ nhớ 150
8.2. RNN: Trạng thái ẩn và tham số dùng lại theo thời gian 153
8.3. LSTM: Cổng và dòng thông tin 157
8.4. Sequence-to-sequence và nút thắt vector ngữ cảnh 159
8.5. Bài học phân tích mô hình chuỗi..... 164
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 170

Chương 9

TRANSFORMER VÀ BƯỚC NGOẶT SONG SONG HÓA

9.1. Bế tắc của tính tuần tự..... 171
9.2. Self-attention: truy vấn, khóa và giá trị 174
9.3. Multi-head và tín hiệu vị trí 176
9.4. Một Transformer block được đọc như chuỗi phép tính..... 179
9.5. Bài học phân tích Transformer..... 185
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 189

Chương 10

TIỀN HUẤN LUYỆN VÀ MÔ HÌNH NỀN TẢNG

10.1. Embedding và câu hỏi biểu diễn 190
10.2. Word2Vec và học từ ngữ cảnh 193
10.3. BERT: Che token để học biểu diễn hai chiều..... 196
10.4. GPT: Dự đoán tiếp để học sinh chuỗi..... 199
10.5. Mô hình nền tảng và điều kiện sử dụng 203
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 210

Chương 11

MÔ HÌNH LỚN, QUY LUẬT TỈ LỆ VÀ CHI PHÍ HỆ THỐNG

11.1. Từ mô hình lớn đến quy luật tỉ lệ 211
11.2. Decoder-only LLM như một hệ tensor 215
11.3. Parameter count và memory footprint 218
11.4. Prefill, decode và KV cache 222
11.5. Triển khai: Latency, throughput và quantization 225
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 232

Chương 12

ĐỌC VÀ PHÂN TÍCH MÔ HÌNH ĐÃ CÔNG BỐ

12.1. Hồ sơ mô hình như một hiện vật kỹ thuật 233
12.2. Model card, license, task và cảnh báo 237
12.3. Config, tokenizer và checkpoint 239
12.4. Từ model id đến bảng phân tích 242
12.5. Mẫu báo cáo phân tích mô hình 244
CÂU HỎI ÔN TẬP VÀ BÀI TẬP 252
TÀI LIỆU THAM KHẢO 253

TỪ VIẾT TẮT

AI	Artificial Intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
BERT	Bidirectional Encoder Representations from Transformers	Biểu diễn mã hóa hai chiều dựa trên Transformer
BLEU	Bilingual Evaluation Understudy	Chỉ số đánh giá dịch máy
BN	Batch Normalization	Chuẩn hóa theo lô
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CPU	Central Processing Unit	Bộ xử lý trung tâm
CUDA	Compute Unified Device Architecture	Nền tảng tính toán song song
CV	Computer Vision	Thị giác máy tính
DL	Deep Learning	Học sâu
FFN	Feed-Forward Network	Mạng truyền thẳng
FLOPs	Floating Point Operations	Số phép toán dấu phẩy động
FP16	16-bit Floating Point	Số thực dấu phẩy động 16 bit
GAN	Generative Adversarial Network	Mạng đối nghịch tạo sinh
GLUE	General Language Understanding Evaluation	Bộ chuẩn đánh giá hiểu ngôn ngữ
GPT	Generative Pre-trained Transformer	Bộ chuyển đổi tạo sinh tiền huấn luyện
GPU	Graphics Processing Unit	Bộ xử lý đồ họa

HF	Hugging Face	Nền tảng lưu trữ và chia sẻ mô hình học máy
ILSVRC	ImageNet Large Scale Visual Recognition Challenge	Thử thách nhận dạng thị giác quy mô lớn ImageNet
KV Cache	Key-Value Cache	Bộ nhớ đệm khóa-giá trị trong kiến trúc Transformer
LLM	Large Language Model	Mô hình ngôn ngữ lớn
LSTM	Long Short-Term Memory	Bộ nhớ dài-ngắn hạn
MACs	Multiply-Accumulate Operations	Phép nhân-cộng tích lũy
ML	Machine Learning	Học máy
MLP	Multi-Layer Perceptron	Perceptron nhiều tầng
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
QKV	Query, Key, Value	Truy vấn, khóa và giá trị
ReLU	Rectified Linear Unit	Hàm kích hoạt tuyến tính chỉnh lưu
RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
Seq2Seq	Sequence-to-Sequence	Mô hình biến đổi chuỗi sang chuỗi
SGD	Stochastic Gradient Descent	Hạ gradient ngẫu nhiên
SOTA	State of the Art	Mức tiên tiến nhất tại thời điểm so sánh
TPU	Tensor Processing Unit	Bộ xử lý tensor
VAE	Variational Autoencoder	Bộ tự mã hóa biến phân
ViT	Vision Transformer -	Transformer cho thị giác máy tính
WMT	Workshop on Machine Translation	Bộ chuẩn và hội thảo dịch máy
XOR	Exclusive OR	Phép hoặc loại trừ

LỜI NÓI ĐẦU

Học sâu (Deep learning) thường được mô tả như một hộp đen: dữ liệu đi vào, dự đoán đi ra, ở giữa là một khối tham số khổng lồ khó hiểu. Cách nhìn đó có một phần đúng, nhưng nếu dừng ở đó thì người học sẽ mất khả năng đặt câu hỏi. Mô hình không tự nhiên xuất hiện. Mỗi kiến trúc, hàm lỗi, thuật toán tối ưu, cơ chế chú ý, chuẩn hóa, bộ nhớ đệm hay thủ thuật triển khai đều ra đời để trả lời một bế tắc cụ thể trong lịch sử.

Cuốn sách ***Phân tích mô hình học sâu*** được viết theo một nhịp khác với nhiều tài liệu học sâu thông thường. Thay vì bắt đầu bằng danh sách mô hình hiện đại, cuốn sách đi từ khởi nguyên: từ nỗ lực mô hình hóa nơ-ron bằng logic, đến ý tưởng học nằm trong kết nối, perceptron, giới hạn tuyến tính, vấn đề tầng ẩn, lan truyền ngược, tối ưu hóa, mạng tích chập, mô hình chuỗi, attention, Transformer, tiền huấn luyện, mô hình lớn và cuối cùng là cách đọc một mô hình đã công bố trong bài báo, báo cáo kỹ thuật, repository hoặc nền tảng chia sẻ mô hình.

Tinh thần xuyên suốt của cuốn sách là: mô hình cho khuôn, tham số cho hình cụ thể, dữ liệu nắn hình, hàm lỗi cho tiêu chuẩn, đạo hàm cho hướng sửa, tốc độ học cho bước đi và kiềm chế giữ hình không bị méo. Khi gặp một ý tưởng mới, cuốn sách luôn hỏi: trước đó người ta vướng gì, trực giác mới là gì, toán học sinh ra từ đâu, tensor đi qua mô hình như thế nào, các tham số nằm ở đâu, chi phí tính toán và bộ nhớ phát sinh ra sao, kết quả được đánh giá bằng chỉ số nào và giới hạn của kết luận là gì.

Mục tiêu cuối cùng không phải là học thuộc tên các kiến trúc. Sau khi đọc xong, người đọc cần có khả năng đọc một mô hình đã công bố từ nhiều nguồn khác nhau: bài báo, báo cáo kỹ thuật, model card, tài liệu framework, repository mã nguồn, file cấu hình, tokenizer/processor, checkpoint, chỉ số đánh giá và cảnh báo sử dụng. Hugging Face chỉ là một ví dụ thuận tiện cho thực hành, năng lực cốt lõi là đọc mô hình như một hiện vật khoa học và kỹ thuật.

Các tác giả

Chương 1

NÃO, LOGIC VÀ GIẤC MƠ MÔ HÌNH HÓA TRÍ TUỆ

Từ rất lâu, con người đã muốn hiểu trí tuệ của chính mình: ý nghĩ, ký ức, suy luận và lựa chọn dựa vào đâu trong thế giới vật chất? Khi câu hỏi ấy được đặt vào não, tế bào thần kinh và tín hiệu, một hướng đi mới xuất hiện: liệu hoạt động của não có thể được mô hình hóa bằng một hệ thống hình thức hay không? Chương này đi từ ước mơ hiểu trí tuệ đến logic hình thức, rồi đến mô hình nơ-ron bật/tắt đầu tiên. Qua đó, ta thấy một ý tưởng khoa học được sinh ra từ một câu hỏi rất cụ thể, được đơn giản hóa thành khuôn, tạo ra các phép tính logic, để lộ giới hạn của mạch cố định, rồi lan sang cách người ta nghĩ về máy tính điện tử sớm.

MỤC TIÊU HỌC TẬP:

- Trình bày được vì sao câu hỏi về trí tuệ cần một mô hình trung gian giữa não sinh học và suy luận hình thức.
- Giải thích được vai trò của mệnh đề, giá trị đúng/sai và các phép AND, OR, NOT trong việc mô hình hóa tín hiệu thần kinh.
- Mô tả được nơ-ron McCulloch-Pitts bằng đầu vào, ngưỡng, đầu ra, tín hiệu kích thích, tín hiệu ức chế và độ trễ.
- Biểu diễn được AND, OR, NOT bằng nơ-ron ngưỡng và chỉ ra giới hạn của một mạch có cấu trúc đặt sẵn.
- Giải thích được vì sao mô hình McCulloch-Pitts trở thành cầu nối giữa nghiên cứu não, logic toán học và máy tính điện tử sớm.

1.1. TỪ ƯỚC MƠ HIỂU TRÍ TUỆ ĐẾN NÃO VÀ TẾ BÀO

Từ rất lâu trước khi có những mô hình toán học về não, con người đã có một ước mơ dai dẳng: hiểu được trí tuệ của chính mình. Ta biết mình có thể nhìn, nhớ, tưởng

tượng, suy luận, lựa chọn và sai lầm. Ta biết trong ta có một đời sống tinh thần phong phú: ý thức, cảm giác, ý định, niềm tin, nỗi sợ, hy vọng. Nhưng chính vì quá gần với ta, trí tuệ lại trở thành một trong những điều khó hiểu nhất.

Một câu hỏi cổ xưa nằm ở trung tâm của ước mơ đó: Vật chất và ý thức liên hệ với nhau như thế nào? Nếu ý nghĩ, ký ức và quyết định là những hiện tượng tinh thần, chúng dựa vào cái gì trong thế giới vật chất? Khi khoa học thần kinh dần chỉ ra vai trò đặc biệt của não, câu hỏi không biến mất; nó chỉ trở nên cụ thể hơn. Làm sao một khối vật chất mềm, ẩm, gồm các tế bào sống lại có thể sinh ra tri giác, trí nhớ, suy luận và hành vi?

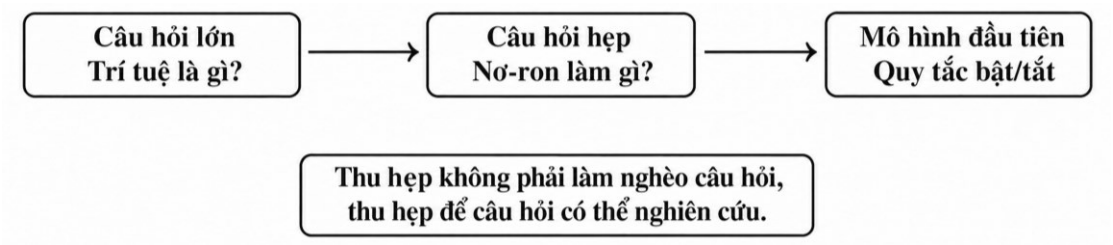
Cái khó của trí tuệ là nó không chỉ có một mặt. Khi nhìn, ta tiếp nhận thế giới. Khi nhớ, ta giữ lại dấu vết của điều đã qua. Khi suy luận, ta nối các điều kiện để đi đến kết luận. Khi lựa chọn, ta biến nhận biết thành hành động. Những năng lực này trong đời sống thường hòa vào nhau, nhưng để nghiên cứu, ta phải tách chúng ra. Nếu không tách, "trí tuệ" sẽ là một từ quá rộng; nếu tách quá vụn, ta lại đánh mất bức tranh chung. Bảng 1.1 gom bốn năng lực quen thuộc để cho thấy mỗi năng lực kéo theo một câu hỏi vật chất khác nhau.

Bảng 1.1. Một số năng lực tinh thần và câu hỏi vật chất tương ứng

Năng lực	Cách ta gặp trong đời sống	Câu hỏi khi nhìn từ não
Tri giác	Nhìn thấy hình, nghe âm thanh, nhận ra vật thể	Tín hiệu cảm giác được tổ chức thành nhận biết thế nào?
Trí nhớ	Gợi lại một người, một nơi, một sự kiện	Dấu vết của kinh nghiệm nằm ở đâu trong hệ thần kinh?
Suy luận	Từ điều kiện đi đến kết luận	Hoạt động vật chất nào có thể mang dạng của lập luận?
Hành động	Quyết định nói, viết, đi, dừng lại	Tín hiệu thần kinh chuyển thành phản ứng ra sao?

Một mô hình nơ-ron hình thức không sinh ra để giải thích ngay toàn bộ các năng lực trên. Nó sinh ra từ một lát cắt hẹp hơn: Nếu suy luận có thể viết thành quy tắc và nếu nơ-ron có thể lý tưởng hóa thành đơn vị bật/tắt, liệu có thể nối hai phía đó với nhau không? Mô hình đầu tiên không ôm hết trí tuệ, nó chọn một cửa hẹp để đi vào trí tuệ.

Trong khoa học, một câu hỏi tốt không nhất thiết là câu hỏi lớn nhất; thường nó là câu hỏi đủ nhỏ để có thể kiểm tra, nhưng đủ sâu để mở ra hướng hiểu mới. Nếu hỏi toàn bộ trí tuệ cùng lúc, ta không có chỗ bám. Nếu hỏi một đơn vị thần kinh có thể nhận tín hiệu, so với một điều kiện và phát tín hiệu hay không, ta có thể bắt đầu vẽ mô hình, viết ký hiệu và kiểm tra hệ quả. Hình 1.1 minh họa bước thu hẹp đó như một chuyển động từ câu hỏi rất lớn sang một khuôn có thể thao tác.

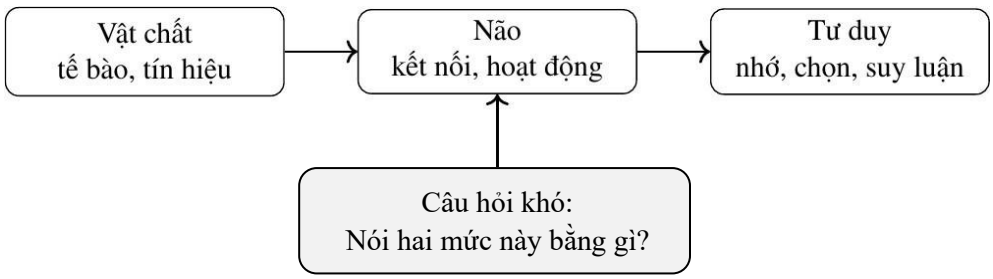


Hình 1.1. *Từ câu hỏi lớn đến một câu hỏi đủ hẹp để mô hình hóa*

Khi nhìn gần hơn vào não, ta không thấy ngay "ý nghĩ". Ta thấy tế bào thần kinh, sợi trục, synapse, tín hiệu điện hóa, những kết nối chằng chịt. Ở mức vật chất, đó là các tế bào và tín hiệu. Ở mức trải nghiệm, đó lại là cảm giác, quyết định và lập luận. Khoảng cách giữa hai mức này rất lớn. Chính khoảng cách đó tạo ra nhu cầu mô hình hóa: Cần một cây cầu đủ đơn giản để nối hoạt động vật chất của não với những thao tác tinh thần có thể phân tích.

Điều khó nằm ở chỗ não không chỉ là một vật thể, mà còn là nơi ta quy chiếu kinh nghiệm chủ quan của chính mình. Nếu nhìn não như một khối vật chất, ta có thể cân, cắt lát, nhuộm màu, quan sát tế bào. Nếu nhìn đời sống tinh thần, ta gặp những thứ không dễ đặt lên bàn cân: cảm giác màu đỏ, ký ức về một khuôn mặt, ý định giơ tay, hoặc một chuỗi suy luận thầm lặng. Hai phía này đều thật theo cách của chúng. Khoa học cần một con đường để nói từ phía vật chất sang phía hoạt động tinh thần mà không chỉ dựa vào ẩn dụ.

Một ví dụ cụ thể giúp thấy khoảng cách đó. Khi ta chạm tay vào vật nóng và rụt tay lại, ở mức trải nghiệm ta nói: "Tôi thấy nóng nên rút tay". Ở mức sinh học, có tín hiệu từ thụ thể, đường dẫn thần kinh, hoạt động của tủy sống và cơ. Câu chuyện thứ nhất gần với cảm nhận chủ quan; câu chuyện thứ hai gần với vật chất và cơ chế. Nếu muốn hiểu trí tuệ bằng khoa học, ta phải học cách nối hai câu chuyện này: không phủ nhận trải nghiệm, nhưng cũng không dừng ở trải nghiệm. Hình 1.2 đặt khoảng cách ấy thành một sơ đồ ba điểm: vật chất, não và tư duy.



Hình 1.2. *Từ vật chất đến tư duy: khoảng cách cần một mô hình trung gian*

Từ sơ đồ đó có thể tách ba mức đọc. Ở mức vật chất, ta quan sát tế bào, sợi trục, synapse và tín hiệu điện hóa. Ở mức hoạt động thần kinh, ta quan tâm cách tín hiệu truyền qua mạng kết nối. Ở mức đời sống tinh thần, ta gặp tri giác, trí nhớ, suy luận và lựa chọn. Mô hình hình thức phải đi giữa ba mức này: đủ gần vật chất để không bay khỏi não, nhưng đủ trừu tượng để có thể phân tích suy luận.

Có hai việc khác nhau trong nghiên cứu não. Việc thứ nhất là mô tả não thật càng đầy đủ càng tốt. Việc thứ hai là chọn một lát cắt đủ đơn giản để có thể lập luận. Nếu cố giữ mọi chi tiết sinh học ngay từ đầu, câu hỏi sẽ trở nên quá rộng: mỗi nơ-ron có hình dạng, hóa học, thời gian đáp ứng và môi trường sinh học riêng. Nhưng nếu bỏ quá nhiều, mô hình sẽ không còn nói được gì đáng kể về não. Vấn đề không phải là chọn giữa "đúng hoàn toàn" và "sai hoàn toàn", mà là chọn mức đơn giản hóa phù hợp với câu hỏi.

Trước khi có một mô hình nơ-ron hình thức, nhu cầu sâu hơn đã xuất hiện: không chỉ vẽ lại tế bào thần kinh, mà còn đặt hoạt động của tế bào thần kinh vào một ngôn ngữ có luật rõ. Nếu làm được điều đó, một phần của trí tuệ có thể chuyển từ vùng mơ hồ của ẩn dụ sang vùng có thể phân tích bằng ký hiệu, quy tắc và mạch.

Việc thu hẹp này không làm câu hỏi nghèo đi, nó làm câu hỏi có điểm tựa. Nếu hỏi trực tiếp "ý thức là gì", ta dễ rơi vào tranh luận quá rộng. Nếu hỏi "một nơ-ron có thể được xem như một đơn vị nhận tín hiệu và phát tín hiệu không", câu hỏi đã nhỏ hơn nhiều. Nó chưa giải thích toàn bộ trí tuệ, nhưng nó cho phép ta đặt viên gạch đầu tiên: một đơn vị vật chất có thể được mô tả bằng một quy tắc hình thức.

Ba cách nói thường đi cạnh nhau nhưng không thể thay thế cho nhau: ẩn dụ, mô tả sinh học và mô hình hình thức. Ẩn dụ giúp ta hình dung nhanh, chẳng hạn nói não như một mạng tín hiệu, nhưng dễ làm ta tưởng hai sự vật giống nhau nhiều hơn thực tế. Mô tả sinh học giữ sự phong phú của mô não thật, nhưng quá nhiều chi tiết sẽ làm câu hỏi hình thức mất điểm tựa. Mô hình hình thức làm một việc khác: chọn một số thuộc tính đủ rõ để viết thành ký hiệu và quy tắc.

Nhu cầu lúc này đã chín dần: Phải có một khuôn không chỉ ví von rằng nơ-ron "giống" một công tắc, mà còn cho phép thao tác bằng ký hiệu. Khuôn ấy phải nói rõ đầu vào được hiểu thế nào, đầu ra được hiểu thế nào và điều kiện nào làm đơn vị phát tín hiệu. Chỉ khi đó, câu hỏi về nơ-ron mới rời khỏi ẩn dụ để trở thành câu hỏi có thể phân tích.

Tế bào thần kinh trở thành một điểm gặp tự nhiên. Nó đủ vật chất để thuộc về sinh học: có hình dạng, có tín hiệu, có kết nối. Nhưng nó cũng đủ giống một đơn vị xử lý để gợi ra câu hỏi hình thức: Nếu có tín hiệu vào và tín hiệu ra, liệu có thể viết quy tắc biến tín hiệu vào thành tín hiệu ra hay không? Câu hỏi này dẫn chương sách sang logic hình thức, nơi suy luận được tách khỏi lời kể tự nhiên và viết thành dạng có luật.

1.2. LOGIC HÌNH THỨC NHƯ CÂY CẦU MÔ HÌNH HÓA

Ở nửa đầu thế kỷ XX, một hướng trả lời hấp dẫn xuất hiện từ logic hình thức. Logic cho phép biến suy luận thành ký hiệu và quy tắc. Nếu một mệnh đề có thể đúng hoặc sai, nếu các mệnh đề có thể kết hợp bằng AND, OR, NOT, thì có thể hỏi tiếp: Liệu nơ-ron cũng có thể được lý tưởng hóa như một phần tử tạo ra tín hiệu có/không? Đó không phải là tuyên bố rằng nơ-ron sinh học thật chỉ là công tắc, mà là một chiến lược mô hình hóa: bỏ bớt chi tiết sinh học để giữ lại một khuôn tối thiểu có thể tính toán.

Trong ngôn ngữ hằng ngày, khi nói "lập luận có logic", ta thường muốn nói lập luận nghe hợp lý. Logic hình thức hẹp hơn và chặt hơn. Nó không quan tâm trước hết đến nội dung cụ thể của câu, mà quan tâm đến dạng của câu và quy tắc biến đổi giữa các câu. Câu "trời đang mưa" có thể đúng hoặc sai. Câu "cảm biến A bật" cũng có thể đúng hoặc sai. Khi bỏ nội dung riêng của từng câu, ta còn lại một dạng trừu tượng: một mệnh đề P có giá trị đúng hoặc sai. Chính sự trừu tượng này làm logic trở thành ngôn ngữ rất mạnh cho mô hình hóa.

Logic hình thức biến suy nghĩ thành một trò chơi có luật rõ. Ta có ký hiệu, chẳng hạn P, Q. Ta có phép nối, chẳng hạn $P \wedge Q$ nghĩa là P và Q cùng đúng. Ta có quy tắc suy ra, chẳng hạn nếu P đúng và $P \Rightarrow Q$ đúng, thì Q đúng. Khi mọi thứ được viết bằng ký hiệu và luật biến đổi, ta có thể kiểm tra lập luận mà không cần tranh luận bằng cảm giác. Sức hấp dẫn của nó nằm ở khả năng biến một phần của suy nghĩ thành hệ quy tắc.

Mệnh đề là gì?

Trong logic hình thức, mệnh đề là một câu có thể hỏi "đúng hay sai". "Hôm nay trời mưa" là mệnh đề. "Cảm biến số 1 đang bật" cũng là mệnh đề. "Màu xanh đẹp hơn màu đỏ" không thuận tiện làm mệnh đề trong logic cơ bản vì nó phụ thuộc cảm nhận. Logic hình thức ưu tiên những câu có thể đưa về đúng/sai rõ ràng, vì khi đó nội dung có thể được thay bằng ký hiệu như P hoặc Q.

Khi một câu đã trở thành mệnh đề, ta có thể làm điều rất mạnh: thay nội dung cụ thể bằng ký hiệu. "Công tắc đang bật" có thể gọi là P. "Đèn sáng" có thể gọi là Q. Khi đó câu "nếu công tắc đang bật thì đèn sáng" trở thành $P \Rightarrow Q$. Việc thay bằng ký hiệu không phải để làm câu văn bí hiểm hơn. Nó giúp ta nhìn thấy dạng lập luận nằm bên dưới nội dung. Những câu như "công tắc đang bật" hoặc "tín hiệu đi vào nơ-ron" rất thuận tiện cho logic cơ bản vì có thể kiểm tra trạng thái đúng/sai; những câu phụ thuộc cảm nhận, như "âm thanh này hay", khó đi vào khuôn ấy hơn.

Bốn khái niệm sẽ đi theo toàn bộ đoạn này được tóm lại trong Bảng 1.2.

Bảng 1.2. Các khái niệm nền của logic hình thức

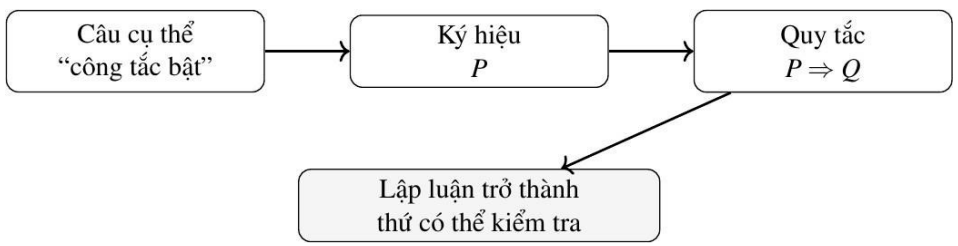
Khái niệm	Cách hiểu trực giác	Vai trò trong mô hình hóa nơ-ron
Mệnh đề	Một phát biểu có thể nhận giá trị đúng hoặc sai	Gợi ý cách xem tín hiệu nơ-ron như có hoặc không
Giá trị chân lý	Đúng/sai, thường ký hiệu bằng 1/0	Tạo cầu nối với trạng thái bật/tắt
Phép logic	Cách kết hợp mệnh đề như AND, OR, NOT	Cho phép ghép nơ-ron thành mạch tính toán
Hệ hình thức	Ký hiệu và quy tắc thao tác rõ ràng	Biến suy luận thành thứ có thể phân tích

Logic hình thức không làm cho trí tuệ trở nên đơn giản, nó làm cho một phần của trí tuệ trở nên có thể thao tác. Khi một suy luận được đưa về mệnh đề đúng/sai, ta có thể bỏ qua sắc thái, cảm xúc, ký ức cá nhân và bối cảnh xã hội. Việc bỏ qua đó có vẻ thô, nhưng nó tạo ra một khuôn đủ rõ để xây mô hình. Cái giá của khuôn này cũng xuất hiện ngay từ đầu: những gì không đưa được về đúng/sai sẽ bị đặt ra ngoài mô hình.

Chữ "hình thức" ở đây không nói về hình thức bề ngoài. Nó nói về dạng và luật. Ví dụ, hai câu "nếu trời mưa thì đường ướt" và "nếu tín hiệu vào đủ mạnh thì nơ-ron

bật" có nội dung khác nhau, nhưng cùng có dạng "nếu P thì Q". Khi chuyển sang dạng, nhiều tình huống khác nhau có thể được phân tích bằng cùng một bộ luật. Đó là sức mạnh của trừu tượng hóa.

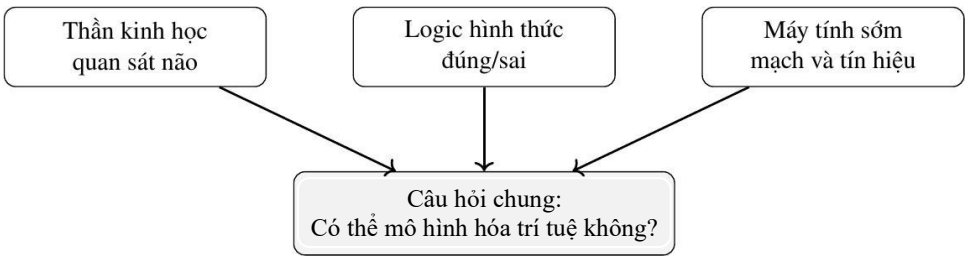
Sự phân biệt giữa "dạng" và "ý nghĩa" cũng rất quan trọng. Dạng cho ta biết câu được cấu trúc như thế nào. Ý nghĩa cho ta biết câu nói về điều gì trong thế giới. Logic hình thức đặc biệt mạnh khi ta tạm đặt ý nghĩa riêng sang một bên để xét dạng. Nếu dạng lập luận sai, nội dung có vẻ thuyết phục cũng không cứu được nó. Nếu dạng lập luận đúng, ta có một khung kiểm tra rất chắc. Vì vậy, logic có thể làm cây cầu giữa suy luận của con người và mạch tín hiệu: cả hai đều có thể được lý tưởng hóa thành các trạng thái và quy tắc biến đổi. Hình 1.3 biểu diễn quá trình chuyển từ câu cụ thể sang ký hiệu và quy tắc.



Hình 1.3. *Từ câu cụ thể đến ký hiệu và quy tắc hình thức*

Hai câu "nếu trời mưa thì đường ướt" và "nếu tín hiệu đủ mạnh thì nơ-ron bật" khác nhau về nội dung, nhưng cùng có dạng điều kiện. Khi đã chuyển thành $P \Rightarrow Q$, câu chuyện thời tiết và câu chuyện tín hiệu cùng nằm trong một khuôn kiểm tra. Đây là lý do logic hình thức có thể đi từ suy luận bằng lời sang mạch tín hiệu mà không cần giữ nguyên toàn bộ nội dung ban đầu.

Ba dòng chảy dẫn tới câu hỏi chung của chương được nối lại trong Hình 1.4.

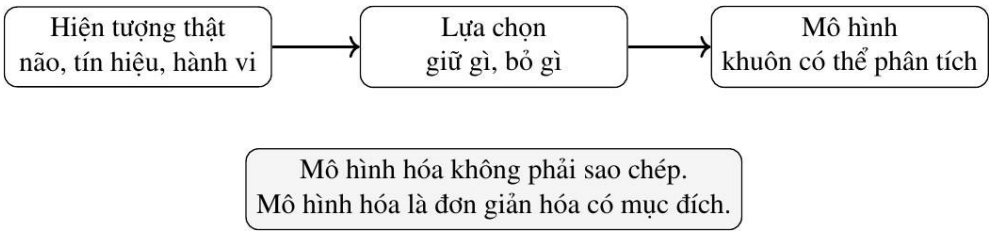


Hình 1.4. *Ba dòng chảy gặp nhau trước mô hình nơ-ron hình thức*

Trước năm 1943, mỗi dòng câu hỏi đã có một phần câu trả lời nhưng vẫn thiếu một mảnh nối. Sinh học thần kinh đã mô tả tế bào, tín hiệu và kết nối, nhưng chưa có

ngôn ngữ hình thức để nói nơ-ron có thể tính gì. Logic toán học đã có quy tắc cho mệnh đề đúng/sai, nhưng chưa được nối rõ với vật chất thần kinh. Máy tính và mạch đã gợi ra trạng thái rời rạc, nhưng chưa trở thành mô hình cho nhận thức. Mô hình nơ-ron hình thức xuất hiện ở chính khoảng giao nhau đó.

Nếu chọn lát cắt rời rạc, điều được giữ lại là tính có/không của tín hiệu và khả năng kết hợp tín hiệu bằng luật logic. Điều bị đặt sang một bên là độ mạnh yếu liên tục của tín hiệu, cơ chế biến đổi synapse và lịch sử kinh nghiệm của hệ. Hình 1.5 diễn đạt lựa chọn này như một bước đi từ hiện tượng thật sang khuôn có thể phân tích.



Hình 1.5. *Mô hình hóa như một hành động lựa chọn có mục đích*

Khuôn cơ bản của nơ-ron ngưỡng được minh họa trong Hình 1.6.

Mỗi khuôn như vậy quyết định trước loại câu trả lời mà mô hình có thể tạo ra. Một khuôn chỉ có đúng/sai sẽ không biểu diễn được mọi sắc thái mơ hồ. Một khuôn chỉ có mạch đã đấu dây sẽ không tự thay đổi qua kinh nghiệm. Sức mạnh của nó nằm ở sự rõ ràng, giới hạn của nó cũng nằm ở chính sự rõ ràng ấy.

Mô hình hóa khác mô tả đầy đủ thế nào?

Mô tả đầy đủ cố gắng kể càng nhiều chi tiết càng tốt. Mô hình hóa chọn một số chi tiết cần cho câu hỏi đang xét. Bản đồ tàu điện không vẽ đúng mọi con phố, nhưng giúp ta đi từ ga này sang ga khác. Một mô hình nơ-ron bật/tắt không mô tả đầy đủ nơ-ron sinh học, nhưng giúp ta hỏi nơ-ron có thể đóng vai trò phần tử tính toán hay không. Vì vậy, giá trị của mô hình nằm ở việc giữ đúng những yếu tố cần cho câu hỏi của nó.

Đến đây, câu hỏi không còn chỉ là câu hỏi triết học. Nó cần một người đứng ở phía não và một người đủ mạnh về logic. Warren McCulloch xuất phát từ y học, tâm thần học và thần kinh học, nhưng mối bận tâm của ông rộng hơn việc mô tả một tế bào riêng lẻ: ông muốn hiểu hoạt động thần kinh có thể mang hình thức của tư duy như thế nào. Có thể nhìn nơ-ron như một đơn vị nhận tín hiệu, chịu kích thích hoặc ức chế, rồi phát

hoặc không phát tín hiệu hay không? Trực giác ấy rất mạnh, nhưng nếu chỉ nói bằng lời thì vẫn chưa thành mô hình.

Walter Pitts đem đến phần còn thiếu đó. Pitts là một người trẻ đặc biệt say mê logic và toán học. Các tài liệu lịch sử ghi lại rằng, năm 1942 Pitts gặp McCulloch khi McCulloch đang tìm cách mô tả chức năng nơ-ron bằng logic nhị phân [1, 2]. Điểm gặp giữa họ là một bài toán cụ thể: Nếu hoạt động thần kinh có tính có/không, liệu quan hệ giữa các sự kiện thần kinh có thể được viết thành logic mệnh đề?

1.3. MCCULLOCH-PITTS VÀ NƠ-RON NGƯỠNG

Kết quả của cuộc gặp ấy là bài *A Logical Calculus of the Ideas Immanent in Nervous Activity*, công bố năm 1943 [3]. Tiêu đề của bài đã nói rõ tham vọng: xây một phép tính logic cho những "ý tưởng" nằm trong hoạt động thần kinh. Cách làm của họ bắt đầu từ tính bật/tắt của hoạt động thần kinh: nơ-ron được lý tưởng hóa thành phần tử có hai trạng thái, hoặc phát xung, hoặc không phát xung. Tín hiệu đầu vào cũng được xem như có hoặc không. Nếu tổng ảnh hưởng đầu vào đạt ngưỡng, nơ-ron phát tín hiệu ở thời điểm tiếp theo.

Từ đây, lát cắt rời rạc ở cuối mục trước có tên cụ thể: nơ-ron McCulloch-Pitts. Nó không phải một bức tranh đầy đủ về tế bào thần kinh thật, mà là một cách đặt câu hỏi: Nếu nơ-ron được xem như phần tử bật/tắt có ngưỡng, mạng gồm nhiều phần tử như vậy có thể biểu diễn những phép suy luận nào?

Ở mức phân tích mô hình, nơ-ron McCulloch-Pitts có ba thành phần: tập đầu vào, ngưỡng và quy tắc bật/tắt. Đó là một khuôn rất nghèo nếu so với nơ-ron sinh học, nhưng lại đủ mạnh để nối nơ-ron với logic. Khi đầu vào là 0 hoặc 1, đầu ra cũng là 0 hoặc 1, ta có thể hỏi: Cấu hình ngưỡng nào tạo ra AND? Cấu hình nào tạo ra OR? Làm sao biểu diễn NOT bằng một kết nối ức chế?

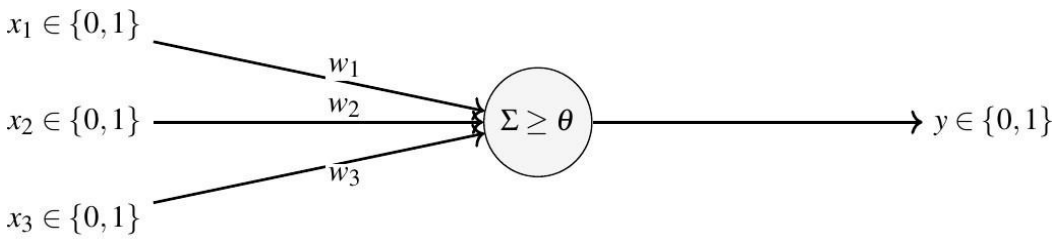
Trong nhiều hệ thống quen thuộc, ngưỡng là mức tối thiểu để một hành động xảy ra. Nước chưa đủ nóng thì chưa sôi. Âm thanh chưa đủ lớn thì ta chưa nghe thấy. Bằng chứng chưa đủ mạnh thì ta chưa kết luận. Trong nơ-ron ngưỡng, ý tưởng cũng tương tự: mỗi tín hiệu đầu vào góp một phần kích thích; khi tổng kích thích vượt qua một mức nhất định, đầu ra bật. Ngưỡng biến nhiều tín hiệu nhỏ thành một quyết định rời rạc.

Ngưỡng vì vậy không chỉ là một số trong công thức. Nó là tiêu chuẩn ra quyết định của mô hình. Nếu ngưỡng thấp, nơ-ron dễ bật. Nếu ngưỡng cao, nơ-ron khó bật.

Khi ta thay đổi ngưỡng, ta không chỉ thay đổi phép tính; ta thay đổi tính nhạy của đơn vị. Trong khuôn ban đầu này, ngưỡng được hiểu ở dạng đơn giản nhất: một mức đủ để biến tổng tín hiệu thành quyết định bật hoặc tắt.

Vì sao cần ngưỡng?

Không có ngưỡng, ta chỉ có tổng tín hiệu nhưng chưa có quyết định. Ngưỡng trả lời câu hỏi: "bao nhiêu là đủ?". Một ví dụ gần gũi là điểm đỗ: bài thi có nhiều câu, mỗi câu cho một phần điểm, nhưng phải đạt một mức nào đó mới được xem là đỗ. Nơ-ron ngưỡng cũng vậy: các tín hiệu đầu vào góp vào tổng, nhưng đầu ra chỉ bật khi tổng đó đủ lớn.



Nếu tổng tín hiệu đạt ngưỡng, nơ-ron bật.
Nếu không đạt ngưỡng, nơ-ron tắt.

Hình 1.6. *Nơ-ron logic kiểu McCulloch-Pitts ở mức khái quát*

Bảng 1.3 đặt nơ-ron sinh học cạnh nơ-ron lý tưởng hóa để thấy rõ phần nào được giữ lại và phần nào bị lược bỏ.

Bảng 1.3. Nơ-ron sinh học và nơ-ron lý tưởng hóa

Khía cạnh	Nơ-ron sinh học	Nơ-ron McCulloch-Pitts
Tín hiệu	Điện hóa, biến thiên theo thời gian	Nhị phân: 0 hoặc 1
Kết nối	Synapse có cơ chế sinh học phức tạp	Cạnh kích thích hoặc ức chế trong mạng logic
Quyết định phát xung	Phụ thuộc nhiều yếu tố sinh lý	Phụ thuộc ngưỡng đơn giản
Thời gian	Liên tục và giàu động lực	Rời rạc theo bước thời gian
Biến đổi theo kinh nghiệm	Có thể thay đổi ở nhiều mức sinh học	Chưa có cơ chế tự biến đổi trong khuôn cơ bản

Một cách viết hiện đại có thể ký hiệu nơ-ron ngưỡng như sau:

$$y = 1, \text{nếu } \sum_i w_i x_i \geq \theta$$

$$y = 0, \text{nếu } \sum_i w_i x_i < \theta$$

Công thức này chỉ là cách viết gọn trực giác trung tâm: mỗi đầu vào góp một phần tín hiệu, nơ-ron chỉ bật khi mức kích thích đủ lớn. Ở thời điểm năm 1943, bước then chốt là biến nơ-ron thành một phần tử tính toán rõ ràng.

Công thức ấy có thể mở ra thành bốn ý cụ thể. Có một số tín hiệu đi vào. Mỗi tín hiệu có mức ảnh hưởng riêng. Các ảnh hưởng đó được cộng lại để tạo thành một mức kích thích chung. Nếu mức kích thích chung đủ lớn, đơn vị phát tín hiệu; nếu chưa đủ, nó im lặng. Như vậy, công thức là mô tả nén của một quyết định bật/tắt.

Quyết định bật/tắt không nằm ở từng tín hiệu riêng lẻ, mà nằm ở tổ hợp của chúng. Một tín hiệu đơn có thể chưa đủ. Hai tín hiệu cùng xuất hiện có thể đủ. Một tín hiệu ức chế có thể làm mất hiệu lực của những tín hiệu kích thích. Ngay cả với một đơn vị rất đơn giản, kết quả của hệ đã phụ thuộc không chỉ vào từng phần tử, mà còn vào cách các phần tử được ghép lại.

Bốn thành phần tối thiểu của nơ-ron ngưỡng là đầu vào, mức đóng góp, ngưỡng và đầu ra. Đầu vào mang tín hiệu có hoặc không từ các đơn vị trước. Mức đóng góp làm mỗi tín hiệu có ảnh hưởng khác nhau trong phép cộng. Ngưỡng đặt tiêu chuẩn tối thiểu để phát tín hiệu. Đầu ra mang quyết định bật/tắt sang phần tiếp theo của mạch. Khi bốn thành phần này được đặt cạnh nhau, mô hình bắt đầu hiện ra như một cơ chế ra quyết định.

Khi tổng tín hiệu vượt ngưỡng và $y = 1$, công thức có thể đọc thành một câu văn: "khi bằng chứng tích lũy đủ mạnh, đơn vị này phát tín hiệu". Công thức khi đó là một mô tả nén, hiểu công thức là mở được mô tả ấy ra.

Một chi tiết quan trọng trong cách nghĩ của McCulloch-Pitts là sự phân biệt giữa tín hiệu kích thích và tín hiệu ức chế. Tín hiệu kích thích góp phần làm nơ-ron dễ bật hơn. Tín hiệu ức chế làm điều ngược lại: nó có thể ngăn nơ-ron phát tín hiệu dù các tín hiệu kích thích đang có mặt. Nhờ vậy, mô hình không chỉ biểu diễn những điều kiện kiểu "cùng đúng", mà còn biểu diễn những điều kiện kiểu "trừ khi".

Ảnh hưởng của ngưỡng lên cùng một cặp đầu vào được minh họa bằng các dòng số trong Bảng 1.4.

Ví dụ, một mạch có thể phát tín hiệu khi có ánh sáng và có chuyển động, nhưng tắt nếu xuất hiện một tín hiệu ức chế. Viết bằng ngôn ngữ tự nhiên, quy tắc đó có dạng: "bật khi hai điều kiện cùng có mặt, trừ khi có điều kiện ngăn lại". Ngay cả cấu trúc "trừ khi" cũng có thể được đưa vào một mạch gồm các đơn vị rất đơn giản. Vì thế, mô hình ngưỡng không chỉ là một phép cộng khô; nó là một khuôn nhỏ để ghép điều kiện.

Bảng 1.4. Ví dụ số: cùng đầu vào, ngưỡng khác nhau cho quyết định khác nhau

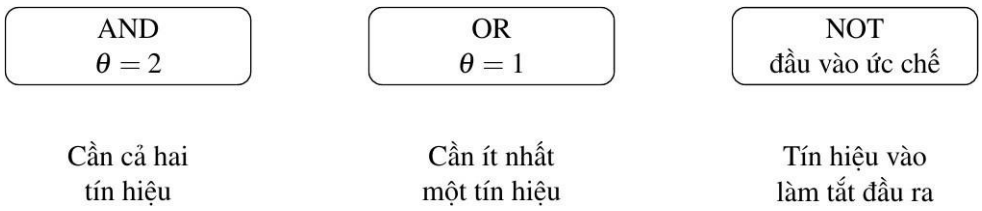
x_1	x_2	Tổng $x_1 + x_2$	Ngưỡng θ	Kết quả
1	0	1	1	Bật: chỉ cần một tín hiệu là đủ.
1	0	1	2	Tắt: một tín hiệu chưa đủ.
1	1	2	1	Bật: vượt ngưỡng thấp.
1	1	2	2	Bật: vừa đủ ngưỡng cao.

Ví dụ nhỏ này cho thấy cùng một đầu vào có thể tạo ra quyết định khác nếu tiêu chuẩn quyết định khác. Nếu ngưỡng bằng 1, hệ đang nói: "một bằng chứng là đủ". Nếu ngưỡng bằng 2, hệ đang nói: "phải có hai bằng chứng cùng lúc". Một thay đổi nhỏ ở quy tắc có thể làm thay đổi hoàn toàn hành vi của mạch. Ngưỡng thấp làm đơn vị nhạy hơn, ngưỡng cao làm đơn vị thận trọng hơn.

Từ đây có thể thấy tại sao lý tưởng hóa lại hữu ích. Nơ-ron thật phức tạp hơn rất nhiều, nhưng nơ-ron ngưỡng cho phép ta tách ra một câu hỏi hẹp: nếu một đơn vị có thể cộng tín hiệu và so với ngưỡng, nó có thể đóng vai trò gì trong một mạch suy luận? Câu hỏi đó đủ hẹp để phân tích, nhưng vẫn đủ gần với trực giác thần kinh học để không trở thành trò chơi ký hiệu thuần túy.

1.4. AND, OR, NOT VÀ GIỚI HẠN CỦA MẠCH ĐÃ ĐẤU DÂY

Khi nơ-ron đã được lý tưởng hóa thành phần tử bật/tắt, các phép logic cơ bản trở nên rất gần. Với hai đầu vào $x_1, x_2 \in \{0, 1\}$, nếu đặt $w_1 = w_2 = 1$ và ngưỡng $\theta = 2$, nơ-ron chỉ bật khi cả hai đầu vào đều bật. Đó là AND. Nếu vẫn đặt $w_1 = w_2 = 1$ nhưng ngưỡng $\theta = 1$, chỉ cần một đầu vào bật là nơ-ron bật. Đó là OR. Hình 1.7 đặt AND, OR và NOT vào cùng một mặt phẳng trực giác.



Hình 1.7. *Các phép logic cơ bản nhìn qua nơ-ron ngưỡng*

Giá trị của AND, OR, NOT không nằm ở việc ghi nhớ tên phép logic, mà ở việc thấy logic hình thức tách dạng khỏi nội dung. AND không quan tâm x_1 là ánh sáng, âm thanh, pixel hay tín hiệu thần kinh. AND chỉ nói rằng hai điều kiện phải cùng đúng. OR không quan tâm điều kiện cụ thể là gì. OR chỉ nói rằng ít nhất một điều kiện đúng là đủ. Khi nội dung bị bỏ đi, ta có một dạng tính toán có thể dùng lại ở nhiều nơi. Vì thế, các phép logic trở thành cầu nối tự nhiên giữa suy luận, mạch điện và nơ-ron hình thức.

AND và OR bằng lời

AND nghĩa là "phải cùng xảy ra". OR nghĩa là "chỉ cần một trong các điều kiện xảy ra". Nếu nói "mở cửa khi có thẻ hợp lệ AND nhập đúng mã", cả hai điều kiện đều cần. Nếu nói "báo động khi có khói OR nhiệt độ quá cao", một điều kiện đã đủ. Bảng chân trị là cách viết chính xác hóa trực giác đó.

Bảng chân trị trong Bảng 1.5 cho thấy cùng một tổng đầu vào có thể được đọc khác nhau tùy ngưỡng.

Bảng 1.5. Bảng chân trị và cách đọc bằng nơ-ron ngưỡng

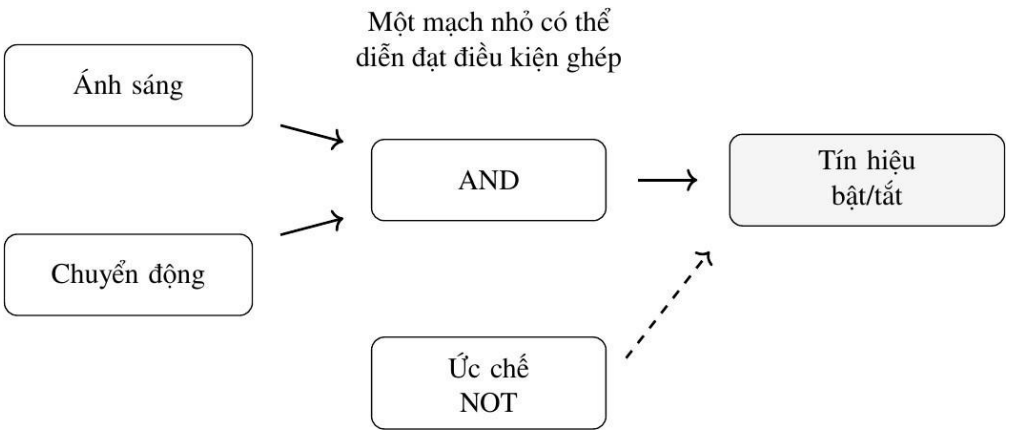
x_1	x_2	AND	OR	Cách đọc bằng ngưỡng
0	0	0	0	Tổng tín hiệu bằng 0, không đạt ngưỡng nào.
0	1	0	1	Tổng bằng 1, đủ cho OR nhưng chưa đủ cho AND.
1	0	0	1	Tổng bằng 1, đủ cho OR nhưng chưa đủ cho AND.
1	1	1	1	Tổng bằng 2, đủ cho cả AND và OR.

Sức mạnh của mô hình này không nằm ở việc nó giống não đến từng chi tiết. Sức mạnh nằm ở việc nó cho phép ta nói: một mạng gồm các phần tử rất đơn giản có thể biểu diễn các phép tính logic. Từ đây, trí tuệ không còn chỉ là một bí ẩn sinh học. Ít nhất một phần của nó có thể được đặt vào ngôn ngữ của mô hình, ký hiệu và cấu trúc. Cùng

một dạng logic có thể mang nhiều nội dung khác nhau: $P \wedge Q$ có thể là "có ánh sáng và có chuyển động", cũng có thể là "tín hiệu x_1 bật và tín hiệu x_2 bật".

Đọc theo cách này, ta thấy vì sao logic hấp dẫn với mô hình hóa não. Nếu một hành vi phức tạp có thể tách thành nhiều điều kiện con và nếu các điều kiện con có thể kết hợp theo quy tắc rõ ràng, thì có thể tưởng tượng một mạng gồm nhiều đơn vị đơn giản cùng tạo ra hành vi phức tạp hơn. Điều này chưa chứng minh não thật hoạt động như mạch logic. Nhưng nó mở một lối suy nghĩ mới: sự phức tạp có thể sinh ra từ tổ hợp của các phần tử đơn giản.

Một đơn vị AND hoặc OR riêng lẻ rất nghèo. Nó chỉ trả lời một câu hỏi nhỏ. Nhưng khi nhiều đơn vị như vậy được ghép lại, đầu ra của đơn vị này có thể trở thành đầu vào của đơn vị khác. Lúc đó, mạch không chỉ hỏi một điều kiện đơn, mà có thể hỏi một chuỗi điều kiện có cấu trúc. Ví dụ: tín hiệu chỉ bật nếu vừa có ánh sáng, vừa có chuyển động, nhưng không có tín hiệu ức chế. Dạng câu này nghe rất gần với suy luận bằng lời: điều kiện này và điều kiện kia cùng đúng, trừ khi có điều kiện thứ ba xuất hiện. Hình 1.8 biểu diễn một mạch ghép điều kiện như vậy.

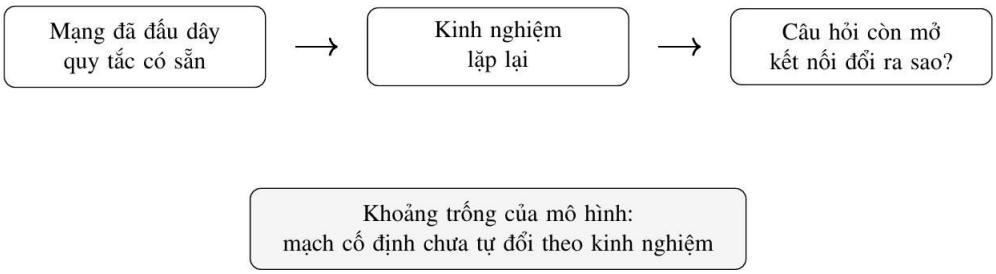


Hình 1.8. *Tổ hợp các cổng logic tạo ra một quy tắc phức tạp hơn từng cổng riêng lẻ*

Chính ví dụ này cũng cho thấy giới hạn. Quy tắc ghép không tự xuất hiện, nó được đặt vào mạch từ trước. Khi ta làm nơ-ron thành công tắc, ta có được tính rõ ràng, nhưng mất đi mức độ, cường độ, thời gian liên tục, nhiễu, thích nghi và khả năng tự biến đổi. Một công tắc có thể được đấu để tạo phép tính; bản thân việc đấu dây từ đâu mà có, thay đổi ra sao và kinh nghiệm làm nó khác đi thế nào thì mô hình này chưa trả lời đầy đủ.

Điểm giới hạn ấy có thể nhìn như một khe hở giữa mạch cố định và hệ có thể tự đổi cấu trúc. Tính toán logic cho biết hệ sẽ làm gì khi cấu trúc đã được đặt sẵn. Còn kinh

nghiệm muốn để lại dấu vết thì phải có cách làm thay đổi phản ứng của hệ. Hình 1.9 đặt khoảng trống này giữa mạng đã đấu dây và câu hỏi về kết nối thay đổi.



Hình 1.9. *Từ mạng đã đấu dây đến câu hỏi về kết nối thay đổi*

Đến đây, giá trị và giới hạn của ý tưởng đã hiện ra cùng lúc. McCulloch-Pitts cho ta một khuôn để nói nơ-ron như phần tử logic, khuôn ấy chưa giải thích cách kinh nghiệm làm thay đổi cấu trúc phản ứng của hệ. Nhưng khi nơ-ron đã được viết như phần tử bật/tắt có quy tắc, ý tưởng này bắt đầu vượt ra ngoài câu hỏi về não. Nó trở thành một cách nghĩ rộng hơn về hệ tính toán: nhiều phần tử đơn giản, ghép theo quy tắc rõ, có thể tạo ra hành vi có cấu trúc. Ở giữa thập niên 1940, cách nghĩ đó gặp một câu chuyện khác đang diễn ra rất nhanh: máy tính điện tử.

1.5. TỪ ENIAC ĐẾN EDVAC: KHI NƠ-RON LOGIC GẶP MÁY TÍNH ĐIỆN TỬ

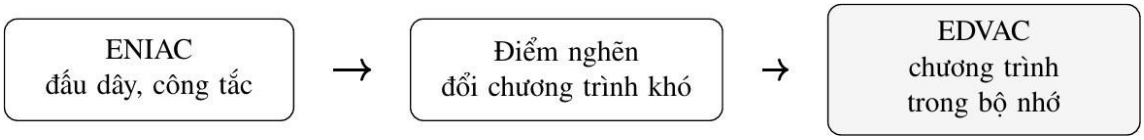
Ở giữa thập niên 1940, máy tính điện tử cũng đứng trước một bài toán gần như vậy: Làm sao để sức mạnh của các phần tử bật/tắt không bị khóa trong một cấu hình cứng. Máy có thể rất nhanh, nhưng nếu muốn đổi bài toán mà phải đấu lại dây, chuyển công tắc và cấu hình lại bảng cắm, thì tốc độ điện tử vẫn bị trói vào một kiến trúc khó thay đổi.

ENIAC là hình ảnh rõ nhất của điểm căng này. John Mauchly và J. Presper Eckert ở Moore School, University of Pennsylvania, phát triển ENIAC để phục vụ các tính toán đạn đạo thời chiến. Theo tư liệu của American Philosophical Society, máy được công bố năm 1946, nặng hơn 30 tấn, dài khoảng 150 feet[1] và dùng hơn 19.000 bóng đèn chân không [4]. Nhìn từ tốc độ, ENIAC là một bước nhảy lớn. Nhìn từ cách sử dụng, nó vẫn mang dấu vết của một mạch được cấu hình cho từng nhiệm vụ. Một bài toán mới đòi hỏi một cách đấu nối mới.

[1] 1 feet = 0,3048 m. 150 feet = 45,72 m (BT).

Bài toán kỹ thuật vì vậy đổi hướng. Câu hỏi không chỉ còn là "làm sao tính nhanh hơn" mà trở thành "làm sao để một máy điện tử có thể đổi nhiệm vụ mà không phải dựng lại mạch cho từng bài toán". Câu trả lời đang hình thành ở EDVAC là đưa chương trình vào bộ nhớ. Khi lệnh và dữ liệu cùng có thể được lưu, đọc và xử lý trong máy, việc đổi chương trình không còn đồng nghĩa với việc đấu lại toàn bộ phần cứng.

Hình 1.10 tóm lại bước chuyển ấy: phần cứng điện tử tạo ra tốc độ, nhưng kiến trúc chương trình lưu trong bộ nhớ mới làm máy tính trở thành một công cụ có thể đổi nhiệm vụ.



Hình 1.10. *Từ máy tính đấu dây sang ý tưởng chương trình lưu trong bộ nhớ*

John von Neumann bước vào câu chuyện này năm 1944. Theo American Philosophical Society, ông biết tới dự án ENIAC trong một cuộc gặp tình cờ với Herman Goldstine khi đang chờ tàu, rồi nhanh chóng nhận ra giá trị của máy tính điện tử đối với các tính toán lớn [4]. Đến năm 1945, ông viết *First Draft of a Report on the EDVAC*, một bản mô tả có ảnh hưởng rất lớn về máy tính chương trình lưu trong bộ nhớ [5]. EDVAC là kết quả của một mạng làm việc gồm Eckert, Mauchly, Goldstine, von Neumann và nhiều người khác. Vai trò đặc biệt của bản *First Draft* là làm cho kiến trúc mới có một ngôn ngữ trình bày rõ ràng: Máy được mô tả như một hệ gồm bộ nhớ, bộ điều khiển, bộ tính toán và các phần tử logic truyền tín hiệu.

Chính ở ngôn ngữ ấy, McCulloch-Pitts đi vào câu chuyện máy tính. Bài viết năm 1943 đã cho một cách nhìn rất gọn: một phần tử nhận tín hiệu, so với ngưỡng, rồi phát hoặc không phát tín hiệu ở bước tiếp theo. Nếu nhiều phần tử như vậy được ghép lại, ta có thể tạo ra mạch logic. Von Neumann không cần lấy toàn bộ sinh học của nơ-ron. Điều ông cần ở đây là một kiểu phần tử trừu tượng: có trạng thái rõ, nhận tín hiệu rõ, phát tín hiệu rõ và có thể ghép thành hệ lớn. Các phân tích lịch sử chỉ ra rằng cách biểu diễn bằng phần tử logic trong bản EDVAC có liên hệ trực tiếp với công thức hóa của McCulloch và Pitts [1]. Một nguồn tóm lược lịch sử về bài năm 1943 cũng nhấn mạnh

rằng trong mô tả kỹ thuật EDVAC, von Neumann xem máy như tập hợp các mô-đun logic, gần với cách McCulloch-Pitts biểu diễn mạng nơ-ron bằng sơ đồ [6].

Chi tiết thời gian làm mối nối này sâu hơn. Trong mô hình McCulloch-Pitts, nơ-ron không phát tín hiệu ngay tại khoảnh khắc nhận đầu vào. Mạng được đặt trên các bước rời rạc: tín hiệu đến, qua một độ trễ synapse, rồi đầu ra xuất hiện ở bước sau. Bài năm 1943 xem độ trễ synapse là độ trễ đáng kể của hệ thần kinh và mô tả hoạt động nơ-ron theo số lần độ trễ tính từ gốc thời gian [3]. Trong bản EDVAC, von Neumann cũng mô tả các phần tử logic có độ trễ cố định. Nghiên cứu lịch sử của Ohta cho rằng von Neumann đã đưa giả định độ trễ synapse cố định của McCulloch-Pitts vào mô hình mạch logic [7]. Từ phía máy tính, độ trễ ấy trở thành một vấn đề tổ chức nhịp: khi nào tín hiệu được xem là đã ổn định, khi nào phần tử tiếp theo được phép dùng tín hiệu đó và làm sao cả hệ không rơi vào hỗn loạn thời gian.

Mối quan hệ giữa ba mảnh này được tóm lại trong Bảng 1.6. ENIAC đặt vấn đề bằng một cỗ máy nhanh nhưng cứng. EDVAC trả lời bằng kiến trúc chương trình lưu trong bộ nhớ. McCulloch-Pitts góp vào tầng sâu hơn: một cách mô tả hệ tính toán bằng các phần tử bật/tắt có độ trễ rời rạc.

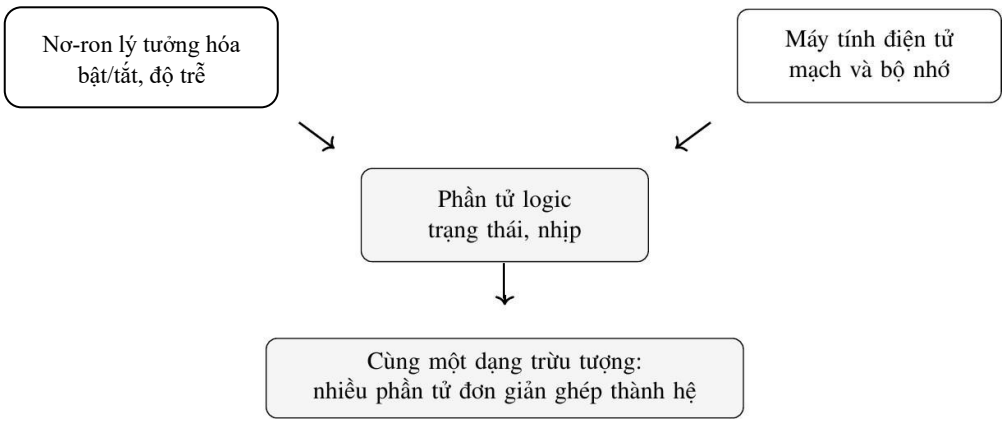
Bảng 1.6. Ba mảnh ghép lịch sử quanh máy tính điện tử sớm

Mốc	Điểm chính	Ý nghĩa đối với câu chuyện của Chương
ENIAC	Máy điện tử dùng bóng đèn chân không, rất nhanh nhưng đổi chương trình bằng cấu hình dây và công tắc	Làm lộ giới hạn của một hệ nhanh nhưng cứng
EDVAC	Kiến trúc chương trình lưu trong bộ nhớ, nơi lệnh và dữ liệu cùng có thể được lưu và xử lý	Chuyển trọng tâm từ mạch riêng cho từng bài toán sang một máy có thể đổi nhiệm vụ
McCulloch-Pitts	Nơ-ron và mạng nơ-ron được lý tưởng hóa thành phần tử bật/tắt có độ trễ rời rạc	Cung cấp ngôn ngữ phần tử, trạng thái và nhịp để mô tả hệ tính toán

Mối liên hệ này có ranh giới rõ. Máy tính điện tử không sinh ra từ một bài báo về nơ-ron. Nó còn đến từ lý thuyết tính toán, kỹ thuật mạch, nhu cầu quân sự, thiết kế bộ

nhớ và kinh nghiệm của các kỹ sư ở Moore School. Nhưng bài năm 1943 xuất hiện đúng lúc nhiều dòng suy nghĩ đang cần một ngôn ngữ chung: tín hiệu có/không, mệnh đề đúng/sai, phần tử logic, độ trễ, mạch ghép từ các phần tử nhỏ. Từ một giả thuyết về nơ-ron, mô hình McCulloch-Pitts bắt đầu có đời sống thứ hai: nó trở thành một cách nói về hệ tính toán nói chung.

Hình 1.11 biểu diễn quan hệ này như một cây cầu hai chiều. Từ phía não, nơ-ron được lý tưởng hóa thành phần tử bật/tắt có ngưỡng và độ trễ. Từ phía máy, mạch điện tử được tổ chức thành các phần tử logic có tín hiệu vào, tín hiệu ra và nhịp truyền. Ở giữa là câu hỏi lớn hơn: nếu nhiều phần tử đơn giản được ghép lại theo quy tắc, hệ có thể tạo ra hành vi phức tạp đến đâu?



Hình 1.11. *Cầu nối giữa nơ-ron logic và máy tính điện tử sớm*

Câu chuyện này làm rõ giá trị của mô hình khởi nguyên. McCulloch và Pitts không mô tả đầy đủ nơ-ron sinh học, họ chưng cất nơ-ron thành một phần tử logic có ngưỡng và độ trễ. ENIAC không phải máy tính mềm dẻo theo nghĩa chương trình lưu trong bộ nhớ, nó cho thấy tốc độ điện tử vẫn có thể bị khóa trong cấu hình dây. EDVAC chưa thể quy về một cá nhân; nó là nơi nhiều ý tưởng kỹ thuật được viết lại thành kiến trúc. Đặt ba mảnh này cạnh nhau, ta thấy một thời điểm đặc biệt: sinh học thần kinh, logic toán học và kỹ thuật máy tính cùng học cách nói bằng phần tử rời rạc, trạng thái bật/tắt, độ trễ và quy tắc ghép nối.

Nhìn lại từ cuối chương, câu hỏi không còn là "nơ-ron có giống máy tính không". Câu hỏi sâu hơn là: Khi một hiện tượng phức tạp được tách thành các phần tử đơn giản và quy tắc ghép nối, ta có thể hiểu thêm điều gì và ta đã bỏ lại điều gì ở ngoài mô hình? McCulloch-Pitts cho ta một ngôn ngữ logic để nói về nơ-ron. ENIAC cho thấy tốc độ

điện tử chưa đủ nếu kiến trúc còn cứng. EDVAC làm nổi bật vai trò của chương trình, bộ nhớ và tổ chức tín hiệu theo nhịp. Ba mảnh ấy khép lại chương bằng một ý thống nhất: mô hình là một khuôn. Khuôn ấy có thể rất nghèo so với thế giới thật, nhưng khi được đặt đúng câu hỏi, nó có thể làm nhiều lĩnh vực bắt đầu nói chuyện với nhau.

Bài tập phân tích ý tưởng

Hãy chọn một phép logic hai đầu vào, ví dụ AND hoặc OR. Viết lại phép đó theo bốn tầng: mô hình cho khuôn, ngưỡng cho tiêu chuẩn bật/tắt, tín hiệu đầu vào kích hoạt mô hình, đầu ra cho biết mô hình đang tính gì. Sau đó viết một đoạn ngắn giải thích vì sao ví dụ này vẫn là một mạch có cấu trúc đặt sẵn.

Bài tập đọc nguồn

Đọc phần tóm tắt hoặc phần mở đầu của bài McCulloch-Pitts (1943). Ghi lại ba điều: bài đang hỏi câu hỏi gì, bài lý tưởng hóa nơ-ron như thế nào và điểm nào trong bài cho thấy mô hình vẫn là một mạng logic có cấu trúc đặt sẵn. Không cần tóm tắt toàn bộ bài, chỉ cần trả lời đúng ba câu hỏi trên.

TÓM TẮT CHƯƠNG

Chương này bắt đầu từ câu hỏi rất cũ về quan hệ giữa vật chất và tư duy, rồi thu hẹp dần vào một câu hỏi có thể mô hình hóa: nơ-ron có thể được xem như một phần tử nhận tín hiệu, so với ngưỡng và phát tín hiệu hay không. Logic hình thức cung cấp ngôn ngữ đúng/sai, AND, OR, NOT; McCulloch và Pitts dùng ngôn ngữ ấy để lý tưởng hóa nơ-ron thành phần tử bật/tắt có ngưỡng và độ trễ. Mô hình này cho thấy các phép suy luận đơn giản có thể được biểu diễn bằng mạng phần tử logic, đồng thời làm rõ giới hạn của một cấu trúc đặt sẵn. Khi đặt cạnh ENIAC, EDVAC và bản mô tả của von Neumann, nơ-ron logic không chỉ thuộc về câu chuyện não, mà còn tham gia vào cách người ta hình dung hệ tính toán điện tử sớm.

CÂU HỎI ÔN TẬP VÀ BÀI TẬP

1. Nêu bốn năng lực tinh thần được chương dùng để mở đầu câu hỏi về trí tuệ.
2. Liệt kê ba phép logic cơ bản được dùng để minh họa nơ-ron bật/tắt.
3. Kể tên các thành phần chính trong mô hình nơ-ron McCulloch-Pitts.
4. Vì sao câu hỏi "trí tuệ dựa vào đâu trong thế giới vật chất" cần được thu hẹp trước khi có thể mô hình hóa?
5. Logic hình thức giúp nối suy luận bằng lời với tín hiệu bật/tắt của nơ-ron ở điểm nào?
6. Giải thích vì sao mô hình McCulloch-Pitts vừa có giá trị mở đường vừa có giới hạn sinh học rõ ràng.
7. Hãy biểu diễn AND và OR bằng một nơ-ron ngưỡng hai đầu vào, sau đó giải thích vì sao chỉ thay ngưỡng đã làm thay đổi hành vi của mạch.
8. Chọn một quy tắc đời thường có dạng điều kiện và viết lại nó bằng mệnh đề logic.
9. Vẽ một mạch nhỏ kết hợp AND và NOT để phát hiện một tình huống gồm hai điều kiện.
10. Phân tích vì sao một mạng logic có cấu trúc đặt sẵn vẫn chưa trả lời câu hỏi kinh nghiệm làm thay đổi phản ứng của hệ như thế nào.
11. Đánh giá mối liên hệ giữa McCulloch-Pitts, ENIAC, EDVAC và von Neumann qua khái niệm phần tử logic, độ trễ và tổ chức tín hiệu.
12. Đề xuất một cách mở rộng mô hình nơ-ron bật/tắt để mô tả tốt hơn cường độ tín hiệu hoặc khả năng thay đổi theo kinh nghiệm.