**5.1 Generalization: The Goal of Machine Learning (Tổng quát hóa: Mục tiêu của học máy)**

* **Mục tiêu chính của học máy là tổng quát hóa**, tức là khả năng mô hình hoạt động tốt trên dữ liệu mới, chưa từng thấy, chứ không chỉ trên dữ liệu huấn luyện. Điều này khác biệt với việc ghi nhớ dữ liệu huấn luyện một cách máy móc.
* Tác giả giải thích rằng một mô hình học máy lý tưởng cần tìm ra các mẫu (patterns) trong dữ liệu huấn luyện để dự đoán chính xác trên dữ liệu kiểm tra, thay vì chỉ sao chép đầu ra từ dữ liệu đã học.
* Tổng quát hóa liên quan đến việc cân bằng giữa việc học đủ từ dữ liệu và tránh việc học quá mức những đặc điểm không quan trọng hoặc ngẫu nhiên, dẫn đến hiện tượng "quá khớp" (overfitting).
* Phần này nhấn mạnh rằng thành công của học máy không nằm ở việc đạt được độ chính xác cao trên tập huấn luyện, mà là ở khả năng áp dụng kiến thức đã học vào các tình huống thực tế mới.

**Underfitting and Overfitting (Thiếu khớp và Quá khớp)**

* **Thiếu khớp (underfitting)** xảy ra khi mô hình không học được đủ từ dữ liệu huấn luyện, dẫn đến hiệu suất kém trên cả dữ liệu huấn luyện và kiểm tra. Điều này thường do mô hình quá đơn giản hoặc thiếu dữ liệu.
* **Quá khớp (overfitting)** là khi mô hình học quá chi tiết, bao gồm cả nhiễu (noise) và các đặc điểm không đại diện trong dữ liệu huấn luyện, khiến nó không thể dự đoán tốt trên dữ liệu mới.
* Tác giả ví dụ: nếu một mô hình phân loại hình ảnh mèo và chó chỉ dựa vào màu nền thay vì đặc điểm của động vật, nó sẽ thất bại khi gặp dữ liệu mới với nền khác biệt.
* Việc tìm điểm cân bằng giữa thiếu khớp và quá khớp là thách thức cốt lõi, đòi hỏi người học máy phải điều chỉnh mô hình sao cho vừa đủ phức tạp để học các mẫu quan trọng mà không bị nhiễu làm sai lệch.

**The Nature of Generalization in Deep Learning (Bản chất của tổng quát hóa trong học sâu)**

* Trong học sâu, tổng quát hóa dựa trên giả thuyết manifold: dữ liệu thực tế thường nằm trên các cấu trúc thấp chiều (manifolds) trong không gian cao chiều, và mô hình học cách ánh xạ chúng.
* Tác giả giải thích rằng học sâu không chỉ học ánh xạ trực tiếp từ đầu vào đến đầu ra, mà còn học các biểu diễn trung gian (intermediate representations) giúp đơn giản hóa vấn đề.
* Một khía cạnh quan trọng là khả năng nội suy (interpolation): mô hình hoạt động tốt trên dữ liệu mới nếu nó nằm gần dữ liệu huấn luyện trong không gian biểu diễn, thay vì ngoại suy (extrapolation) ra xa.
* Điều này lý giải tại sao học sâu cần dữ liệu huấn luyện phong phú và đa dạng: nếu thiếu dữ liệu đại diện, mô hình không thể tổng quát hóa tốt, đặc biệt với các tình huống khác xa dữ liệu đã học.

**5.2 Evaluating Machine Learning Models (Đánh giá mô hình học máy)**

* **Việc đánh giá mô hình là bước quan trọng để kiểm tra khả năng tổng quát hóa**, vì hiệu suất trên dữ liệu huấn luyện không phản ánh chính xác khả năng thực tế của mô hình trên dữ liệu mới.
* Tác giả nhấn mạnh cần tách dữ liệu thành ba tập: huấn luyện (training), kiểm định (validation) và kiểm tra (test), để tránh việc mô hình "nhìn trộm" dữ liệu kiểm tra trong quá trình phát triển.
* Tập kiểm định giúp điều chỉnh siêu tham số (hyperparameters) và cấu trúc mô hình, trong khi tập kiểm tra chỉ được dùng một lần cuối cùng để đánh giá hiệu suất thực sự.
* Phần này cảnh báo về nguy cơ rò rỉ thông tin (information leakage) nếu không phân chia dữ liệu cẩn thận, dẫn đến kết quả đánh giá không trung thực.

**Training, Validation, and Test Sets (Tập huấn luyện, kiểm định và kiểm tra)**

* **Tập huấn luyện dùng để học các tham số của mô hình**, tập kiểm định dùng để tối ưu hóa mô hình (chọn siêu tham số), và tập kiểm tra đo lường hiệu suất cuối cùng trên dữ liệu độc lập.
* Tác giả đề xuất các phương pháp chia dữ liệu như **hold-out validation** (giữ một phần dữ liệu làm tập kiểm định), **K-fold validation** (chia dữ liệu thành K phần và luân phiên kiểm định), và **iterated K-fold validation** (lặp lại K-fold với xáo trộn dữ liệu).
* Hold-out đơn giản nhưng có thể không ổn định nếu dữ liệu nhỏ, trong khi K-fold và iterated K-fold phù hợp hơn với dữ liệu hạn chế, giúp giảm biến thiên trong kết quả đánh giá.
* Việc chọn phương pháp phụ thuộc vào kích thước dữ liệu và tính đại diện: dữ liệu càng ít hoặc không đồng nhất, càng cần phương pháp phức tạp hơn như K-fold để đảm bảo đánh giá đáng tin cậy.

**Beating a Common-Sense Baseline (Vượt qua đường cơ sở thông thường)**

* **Đường cơ sở thông thường (baseline)** là một mô hình đơn giản hoặc quy tắc dựa trên trực giác, dùng để so sánh và xác định xem mô hình học máy có thực sự mang lại giá trị hay không.
* Ví dụ, trong bài toán phân loại, baseline có thể là chọn nhãn phổ biến nhất; trong dự đoán giá nhà, có thể là giá trung bình của dữ liệu huấn luyện.
* Tác giả nhấn mạnh rằng mô hình học máy cần vượt qua baseline này để chứng minh tính hữu dụng, nếu không thì việc áp dụng học máy có thể không đáng giá thời gian và công sức.
* Baseline cũng giúp phát hiện vấn đề trong quy trình: nếu mô hình phức tạp không thể vượt qua một quy tắc đơn giản, có thể dữ liệu hoặc cách tiếp cận đang có lỗi.

**Things to Keep in Mind About Model Evaluation (Những điều cần lưu ý khi đánh giá mô hình)**

* **Đại diện dữ liệu (data representativeness)** là yếu tố then chốt: các tập dữ liệu phải phản ánh đúng tình huống thực tế, nếu không kết quả đánh giá sẽ không có ý nghĩa.
* Tác giả cảnh báo về **thiên lệch thời gian (temporal bias)**: trong dữ liệu chuỗi thời gian, việc xáo trộn ngẫu nhiên có thể phá hủy mối quan hệ thời gian, dẫn đến đánh giá sai lệch.
* **Rò rỉ dữ liệu (data leakage)** là một lỗi phổ biến cần tránh, ví dụ khi dữ liệu kiểm định bị lẫn vào quá trình huấn luyện hoặc tiền xử lý không được thực hiện độc lập.
* Cuối cùng, cần chọn đúng thước đo thành công (metric): ví dụ, trong bài toán không cân bằng (imbalanced data), accuracy có thể không đủ, cần dùng precision, recall hoặc AUC.

**5.3 Improving Model Fit (Cải thiện độ khớp của mô hình)**

* **Mục tiêu ở đây là khắc phục thiếu khớp**, tức là làm cho mô hình học tốt hơn từ dữ liệu huấn luyện trước khi lo lắng về tổng quát hóa lên dữ liệu mới.
* Tác giả lưu ý rằng nếu mô hình không thể đạt hiệu suất tốt trên tập huấn luyện, thì việc cải thiện tổng quát hóa là vô nghĩa, vì mô hình chưa đủ mạnh để học các mẫu cơ bản.
* Có ba cách chính để cải thiện: điều chỉnh tham số gradient descent (như learning rate), cải thiện kiến trúc mô hình (architecture priors), và tăng dung lượng mô hình (model capacity).
* Phần này nhấn mạnh tầm quan trọng của việc thử nghiệm: cần kiểm tra từng thay đổi trên tập kiểm định để đảm bảo mô hình tiến bộ mà không bị quá khớp.

**Tuning Key Gradient Descent Parameters (Điều chỉnh tham số gradient descent chính)**

* **Learning rate (tỷ lệ học)** là tham số quan trọng nhất trong gradient descent: quá cao có thể khiến mô hình không hội tụ, quá thấp làm chậm quá trình học.
* Tác giả khuyên dùng các bộ tối ưu hóa như RMSprop hoặc Adam, nhưng vẫn cần thử nghiệm để tìm learning rate phù hợp, thường bằng cách giảm dần từ giá trị lớn.
* Một tham số khác là **batch size**: batch nhỏ hơn có thể giúp mô hình học nhanh hơn nhưng dễ bị nhiễu, trong khi batch lớn hơn ổn định hơn nhưng chậm.
* Việc điều chỉnh đòi hỏi sự kiên nhẫn và kinh nghiệm, thường kết hợp với các công cụ như learning rate scheduler để tự động giảm tỷ lệ học theo thời gian.

**Leveraging Better Architecture Priors (Tận dụng kiến trúc tốt hơn)**

* **Kiến trúc mô hình (architecture priors)** là các giả định ban đầu về cách tổ chức mạng nơ-ron, ví dụ dùng CNN cho ảnh hoặc RNN cho chuỗi thời gian.
* Tác giả giải thích rằng chọn đúng kiến trúc giúp mô hình học hiệu quả hơn, vì nó phù hợp với cấu trúc của dữ liệu, thay vì dùng một mạng fully-connected chung chung.
* Ví dụ, trong bài toán hình ảnh, thêm lớp convolution sẽ tốt hơn là tăng số lớp dense, vì nó khai thác được tính cục bộ (locality) của dữ liệu ảnh.
* Việc cải thiện kiến trúc đòi hỏi hiểu biết về bài toán và kinh nghiệm thực tế, thường dựa trên các nghiên cứu hoặc mô hình đã được chứng minh trước đó.

**Increasing Model Capacity (Tăng dung lượng mô hình)**

* **Dung lượng mô hình (model capacity)** liên quan đến số lượng tham số, tức là độ phức tạp của mô hình: thêm lớp hoặc nơ-ron để tăng khả năng học.
* Tác giả lưu ý rằng mô hình quá đơn giản sẽ thiếu khớp, nhưng tăng dung lượng quá mức mà không kiểm soát có thể dẫn đến quá khớp, cần kết hợp với các kỹ thuật khác.
* Ví dụ, trong bài toán phân loại, tăng số đơn vị ẩn (hidden units) hoặc số lớp có thể giúp mô hình học các mẫu phức tạp hơn từ dữ liệu.
* Điều quan trọng là theo dõi hiệu suất trên tập kiểm định để đảm bảo rằng tăng dung lượng thực sự cải thiện khả năng học mà không làm mất khả năng tổng quát hóa.

**5.4 Improving Generalization (Cải thiện tổng quát hóa)**

* **Cải thiện tổng quát hóa nhằm giảm quá khớp**, đảm bảo mô hình hoạt động tốt trên dữ liệu mới bằng cách kiểm soát độ phức tạp và xử lý dữ liệu tốt hơn.
* Tác giả nhấn mạnh rằng đây là bước tiếp theo sau khi mô hình đã khớp tốt với dữ liệu huấn luyện, tập trung vào việc làm cho mô hình linh hoạt và ít nhạy cảm với nhiễu.
* Các kỹ thuật chính bao gồm quản lý dữ liệu (dataset curation), kỹ thuật đặc trưng (feature engineering), dừng sớm (early stopping), và điều chuẩn hóa (regularization).
* Phần này kết nối lại với mục tiêu cốt lõi của học máy: không chỉ học tốt mà còn áp dụng tốt, đặc biệt trong các tình huống thực tế phức tạp.

**Dataset Curation (Quản lý tập dữ liệu)**

* **Quản lý dữ liệu tốt là nền tảng của tổng quát hóa**: dữ liệu huấn luyện cần đa dạng, phong phú và đại diện cho bài toán thực tế để mô hình học được các mẫu có ý nghĩa.
* Tác giả khuyên nên thu thập thêm dữ liệu nếu có thể, hoặc dùng kỹ thuật tăng cường dữ liệu (data augmentation) như xoay ảnh, thêm nhiễu để mô phỏng các trường hợp mới.
* Ví dụ, trong phân loại ảnh, thêm các phiên bản ảnh bị biến dạng giúp mô hình học tốt hơn các đặc điểm cốt lõi thay vì phụ thuộc vào chi tiết cụ thể.
* Dữ liệu không đại diện (non-representative) sẽ khiến mô hình thất bại trong thực tế, dù có đạt độ chính xác cao trên tập huấn luyện.

**Feature Engineering (Kỹ thuật đặc trưng)**

* **Kỹ thuật đặc trưng là cách biến đổi dữ liệu đầu vào** để làm nổi bật các thông tin quan trọng, giúp mô hình học dễ dàng hơn mà không cần quá phức tạp.
* Tác giả giải thích rằng trong học sâu, việc này ít phổ biến hơn vì mạng nơ-ron tự học đặc trưng, nhưng vẫn hữu ích khi dữ liệu hạn chế hoặc cần giảm tải tính toán.
* Ví dụ, trong dự đoán giá nhà, thay vì dùng tọa độ thô, tính khoảng cách đến trung tâm thành phố có thể là một đặc trưng tốt hơn cho mô hình.
* Kỹ thuật này đòi hỏi hiểu biết sâu về bài toán và dữ liệu, nhưng có thể cải thiện đáng kể tổng quát hóa khi được áp dụng đúng.

**Using Early Stopping (Sử dụng dừng sớm)**

* **Dừng sớm (early stopping)** là kỹ thuật ngừng huấn luyện khi hiệu suất trên tập kiểm định bắt đầu giảm, tránh việc mô hình học quá mức dữ liệu huấn luyện.
* Tác giả gợi ý dùng callback EarlyStopping trong Keras, theo dõi loss trên tập kiểm định và dừng khi không còn cải thiện sau một số epoch nhất định.
* Phương pháp này đơn giản nhưng hiệu quả, giúp giữ lại mô hình ở trạng thái tối ưu trước khi quá khớp xảy ra, đặc biệt với mạng sâu có nhiều tham số.
* Tuy nhiên, cần chọn số epoch kiên nhẫn (patience) hợp lý để tránh dừng quá sớm, bỏ lỡ cơ hội cải thiện thêm.

**Regularizing Your Model (Điều chuẩn hóa mô hình)**

* **Điều chuẩn hóa (regularization)** là một tập hợp kỹ thuật nhằm kiểm soát độ phức tạp của mô hình, giúp ngăn chặn hiện tượng quá khớp (overfitting) khi mô hình học quá chi tiết dữ liệu huấn luyện. Mục tiêu là làm cho mô hình đơn giản hơn nhưng vẫn đủ khả năng tổng quát hóa tốt trên dữ liệu mới.
* **Giảm kích thước mạng (reducing network size)**: Một cách hiệu quả là giảm số lượng lớp (layers) hoặc số nơ-ron trong mỗi lớp, buộc mô hình phải tập trung vào các mẫu quan trọng nhất thay vì ghi nhớ toàn bộ dữ liệu. Ví dụ, thay vì dùng mạng với 10 lớp sâu, ta có thể thử mạng 3 lớp để xem hiệu suất thay đổi ra sao. Điều này dựa trên nguyên tắc Occam’s razor: mô hình đơn giản hơn thường tổng quát hóa tốt hơn nếu nó vẫn giải quyết được bài toán.
* **Thêm điều chuẩn trọng số (weight regularization)**: Kỹ thuật này áp dụng các hình phạt như L1 (Lasso) hoặc L2 (Ridge) lên trọng số của mô hình trong hàm mất mát (loss function). L1 khuyến khích một số trọng số bằng 0, tạo ra mô hình thưa thớt (sparse), trong khi L2 giữ trọng số nhỏ nhưng không triệt tiêu hoàn toàn, giúp giảm sự phụ thuộc vào bất kỳ đặc trưng riêng lẻ nào. Trong Keras, ta có thể thêm tham số kernel\_regularizer vào lớp Dense để áp dụng L2, ví dụ với giá trị 0.01, nhằm giữ trọng số ở mức vừa phải.
* **Thêm dropout**: Đây là kỹ thuật ngẫu nhiên "tắt" (drop) một tỷ lệ nơ-ron trong mỗi lần huấn luyện, thường từ 20-50%, để mô hình không quá phụ thuộc vào bất kỳ nơ-ron cụ thể nào. Dropout hoạt động như một cách mô phỏng việc huấn luyện nhiều mô hình con khác nhau, sau đó lấy trung bình kết quả, tăng tính mạnh mẽ (robustness). Trong thực tế, ta có thể thêm lớp Dropout(0.5) giữa các lớp Dense trong Keras, giúp mô hình học cách phân tán thông tin qua nhiều đường khác nhau. Kỹ thuật này đặc biệt hữu ích với mạng sâu, nơi nguy cơ quá khớp cao hơn do số lượng tham số lớn.