

SỞ GIÁO DỤC & ĐÀO TẠO THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG THPT CHUYÊN TRẦN ĐẠI NGHĨA

TIỂU LUẬN:

SỰ QUÁ KHỚP (OVERFITTING) VÀ CHƯA KHỚP (UNDERFITTING) TRONG HỌC MÁY

GIÁO VIÊN HƯỚNG DẪN : Th.S Hồ Ngọc Lâm

HỌC SINH THỰC HIỆN :

01 – Nguyễn Phạm Quốc An

19 – Nguyễn Nhật Phi

24 – Choi Won Seok

26– Dương Ngọc Thịnh

MÔN: TIN CHUYÊN

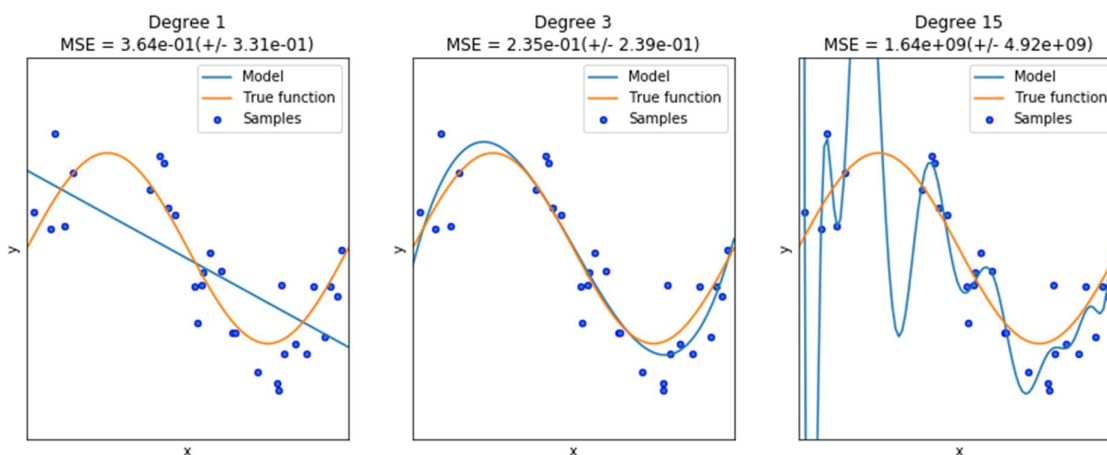
Thành phố Hồ Chí Minh, năm 2023

MỤC LỤC

1. GIỚI THIỆU	1
2. KIẾN THỨC LIÊN QUAN.....	1
2.1 Bias và Variance	1
2.2 Chưa khớp (Underfitting)	2
2.3 Quá khớp (Overfitting)	3
2.4 Vừa khớp (Good fitting)	3
2.5 Ví dụ	3
3. ĐÁNH GIÁ LỖI.....	5
4. KHẮC PHỤC CHƯA KHỚP VÀ QUÁ KHỚP	6
3.1 Các giải pháp khắc phục quá khớp.....	6
3.2 Các giải pháp khắc phục chưa khớp.....	10
5. TÀI LIỆU THAM KHẢO.....	12

1. GIỚI THIỆU

Khi xây dựng một mô hình học máy, mô hình sau khi huấn luyện có thể đạt hiệu quả không tốt khi dự đoán với tập dữ liệu mới. Chuyện này xảy ra là do mô hình của ta chưa tổng quát hoá được với toàn bộ tập dữ liệu. Nguyên nhân cũng khá dễ hiểu khi mà tập huấn luyện của ta chỉ là một tập nhỏ chưa thể đại diện cho toàn thể dữ liệu được và hơn nữa có thể nó còn bị nhiễu nữa. Người ta chia nguyên nhân ra làm 2 loại chính là *chưa khớp* (**Underfit**) hoặc *quá khớp* (**Overfit**).



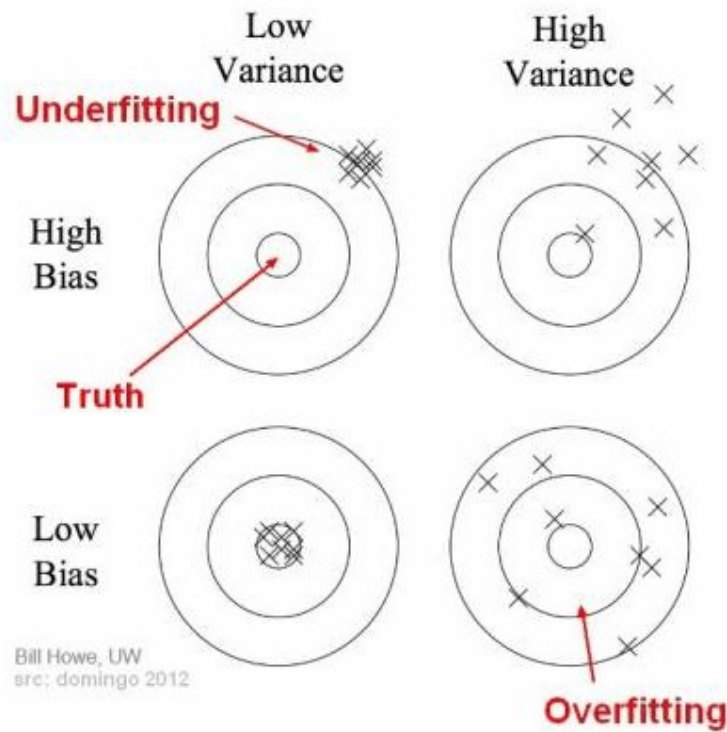
Hình 1: $y=\sin(2\pi x)$ model. Underfit: degree 1 (left); Goodfit: degree 3 (center); Overfit: degree 15 (right)

2. KIẾN THỨC LIÊN QUAN

2.1 Bias và Variance

Trước hết để hiểu rõ Underfitting và Overfitting trong machine learning là gì thì ta cần phải hiểu Bias và Variance là gì.

- **Bias:** là độ lệch, biểu thị sự chênh lệch giữa giá trị trung bình mà mô hình dự đoán và giá trị thực tế của dữ liệu. Xảy ra do các giả định đơn giản trong việc khớp dữ liệu khi làm mô hình.
- **Variance:** là phương sai, biểu thị độ phân tán của các giá trị mà mô hình dự đoán so với giá trị thực tế. Xảy ra do mô hình phức tạp cố khớp dữ liệu đơn giản.



Hình 2: Giá trị thật dữ liệu (ground truth) ở giữa tâm các đường tròn. Các dấu X là các giá trị dự đoán. Ta thấy nếu độ lệch lớn (high bias) thì giá trị dự đoán rất xa tâm. Tuy nhiên nếu phương sai lớn (high variance) thì các giá trị dự đoán phân tán rộng so với giá trị thật.

2.2 Chưa khớp (Underfitting)

Mô hình được coi là chưa khớp nếu nó chưa được phù hợp với tập dữ liệu huấn luyện (**Training set**) và cả các dữ liệu mới (**Test data**) khi dự đoán. Nguyên nhân có thể là do mô hình chưa đủ độ phức tạp cần thiết để bao quát được tập dữ liệu.

Ví dụ, ở Hình 1 ở bên trái, tập dữ liệu huấn luyện cho có dạng hàm hàm số $y = \sin(2\pi x)$, tuy nhiên mô hình dự đoán chỉ là một đường thẳng. Rõ ràng như vậy thì nó không những không thể ước lượng được giá trị mới mà còn không hiệu quả với cả tập dữ liệu (x, y) có sẵn.

Vậy, một mô hình “Chưa khớp” có thể nhận biết qua việc nó có độ phân tán các giá trị nhỏ (Low Variance), nhưng thuật toán còn sai nên giá trị dự đoán khi so với giá trị thực tế có độ lệch cao (High Bias).

2.3 Quá khớp (Overfitting)

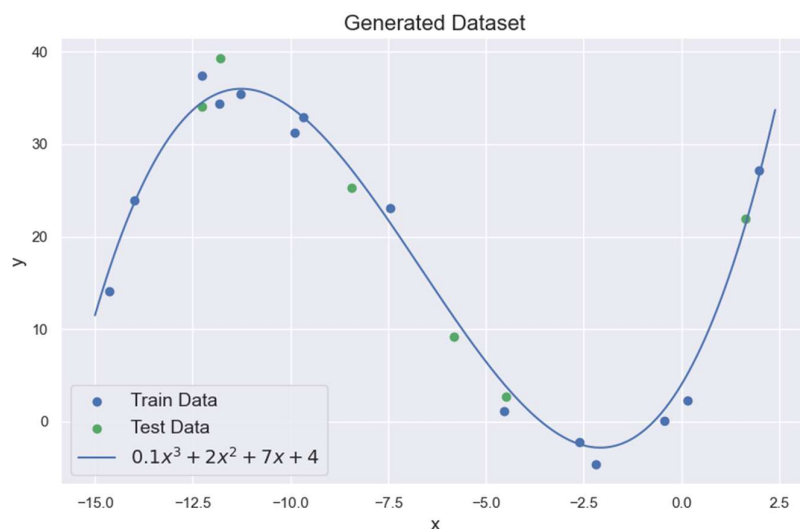
Mô hình rất hợp lý, rất khớp với tập huấn luyện nhưng khi đem ra dự đoán với dữ liệu mới thì lại không phù hợp. Nguyên nhân có thể do ta chưa đủ dữ liệu để đánh giá hoặc do mô hình của ta quá phức tạp. Mô hình bị quá phức tạp khi mà mô hình của ta sử dụng cả những giá trị nhiễu lớn trong tập dữ liệu để học, dẫn tới mất tính tổng quát của mô hình. Ví dụ như ở hình 1 phía bên phải ở trên. Mô hình của ta gần như mong muốn bao được hết tất cả các điểm làm cho biên độ dao động của nó lớn quá mức. Do vậy, trong thực tế, giá trị được dự đoán sẽ lệch đi rất nhiều so với giá trị thật.

Vậy, một mô hình “Quá khớp” được nhận biết khi mô hình dự đoán khi so với tập huấn luyện có độ lệch thấp (Low Bias), nhưng không thể dự đoán chính xác trong thực tế và chỉ cần thay đổi đầu vào một ít thì đầu ra sẽ thay đổi rất nhiều (High Variance).

2.4 Vừa khớp (Good fitting)

Mô hình này là được xem như là mô hình lý tưởng, nằm giữa 2 mô hình chưa khớp và quá khớp cho ra kết quả hợp lý với cả tập dữ liệu huấn luyện và các giá trị mới, tức là nó mang được tính tổng quát như hình 1 ở giữa phía trên. Tuy nhiên trên thực tế được mô hình như vậy rất hiếm.

2.5 Ví dụ

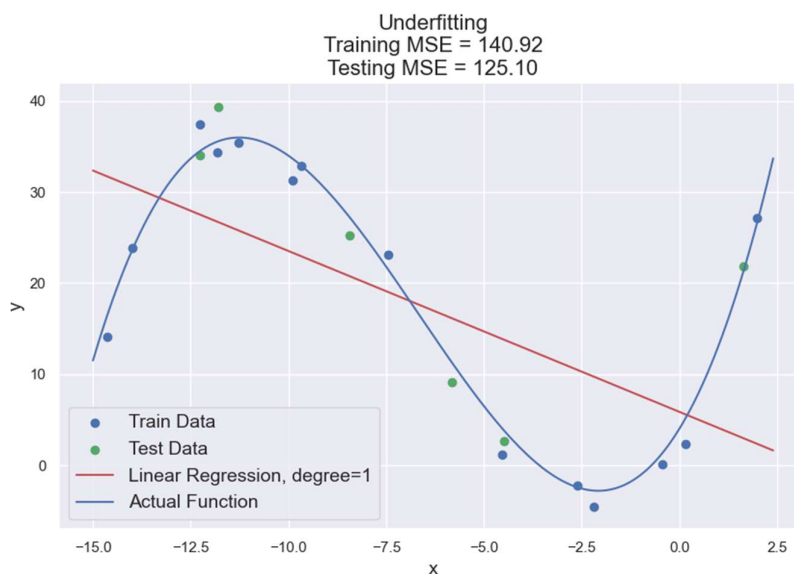


Đây là tập dữ liệu được tạo ngẫu nhiên từ $f(x) = 0.1x^3 + 2x^2 + 7x + 4$ và có cộng thêm ngẫu nhiên số liệu nhiễu, có 20 cặp (x,y) được chia thành tập huấn luyện (70%) và tập thử nghiệm (30%) để đào tạo mô hình áp dụng Polynomial Regression.

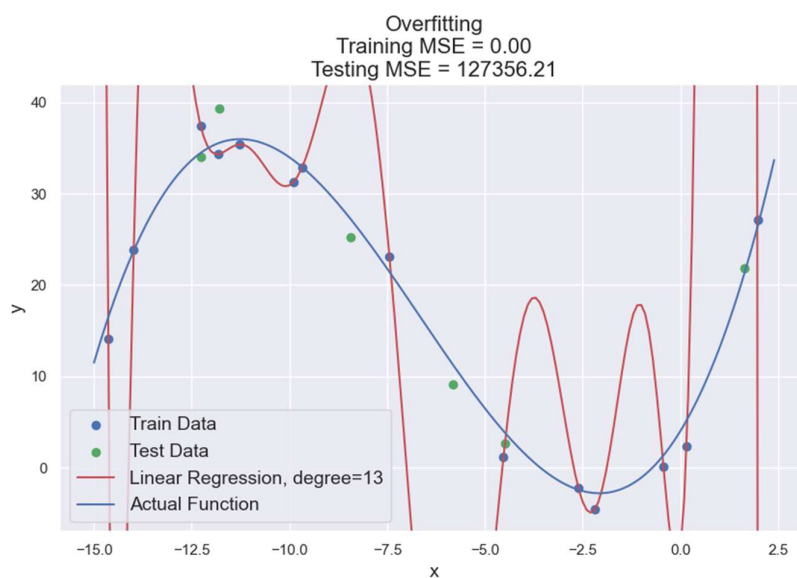
Để đánh giá chất lượng mô hình trên tập huấn luyện và tập thực tế, ta sử dụng hàm mất mát Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

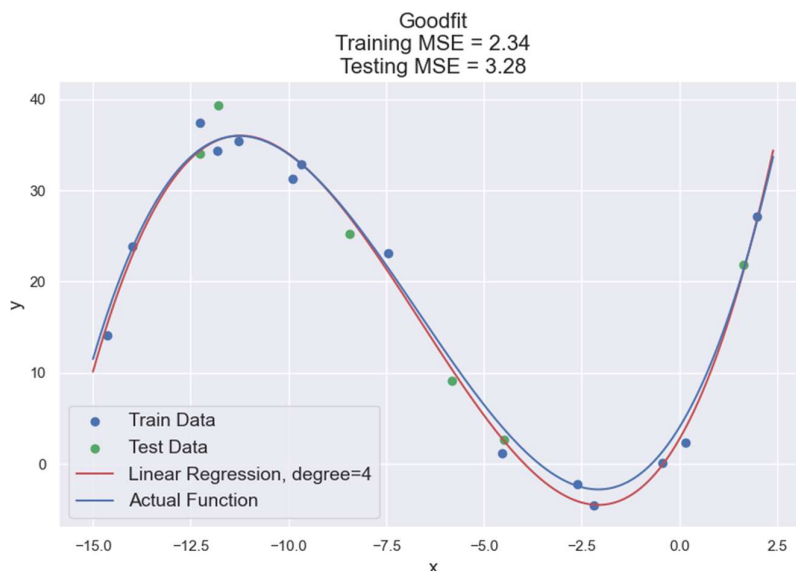
Với Y_i là giá trị thực và \hat{Y}_i là giá trị dự đoán.



Với hàm số dự đoán là bậc 1, ta thấy rằng mô hình này có độ chênh lệch cao (High Bias) so với tập huấn luyện và giá trị mất mát thực nghiệm (Testing MSE) cao nhưng kết quả dự đoán không phân tán và theo khuôn mẫu (Low Variance). Đây là trường hợp của **Underfitting**.



Với hàm số dự đoán là bậc 13, ta thấy được rằng mô hình này dự đoán quá khớp với dữ liệu trong tập huấn luyện (Low Bias) và là do mô hình đang cố gắng mô tả nhiều hơn là dữ liệu. Vì thế nên khi cho vào tập thử nghiệm thì dự đoán cho ra rất khác so với dữ liệu thực (High Variance). Đây là trường hợp của **Overfitting**.



Với hàm số dự đoán là bậc 4, ta được mô hình dự đoán khá giống với mô hình thực có độ lệch thấp (Low Bias) và dự đoán không phân tán khắp nơi như bậc 13 (Low Variance) nên đây là một mô hình tốt - Goodfit.

3. ĐÁNH GIÁ LỖI

Gọi trung bình lỗi của toàn bộ tập dữ liệu được đánh giá là:

$$E(\theta) = \frac{1}{m} \sum_{i=1}^m \text{err}(\hat{y}_i, y_i)$$

Trong đó, $E_X(\theta)$ là lỗi ứng với tham số θ ước lượng được của tập dữ liệu gồm có m mẫu. $\text{err}(\hat{y}, y)$ thể hiện cho sự khác biệt giữa giá trị dự đoán \hat{y} và giá trị thực tế y . Đương nhiên nếu $\hat{y} = y$ thì $\text{err}(\hat{y}_i, y_i) = 0$. Thường người ta lấy $\text{err}(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$ như là các hàm mất mát (loss function) của mô hình. Khi đó hàm mất mát của ta được gọi là **lỗi trung bình bình phương** (MSE - Mean Squared Error):

$$E(\theta) = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

Thông thường, dữ liệu của ta sẽ được phân chia làm 3 phần là **tập huấn luyện** (training set) 60%, **tập kiểm chứng** (cross validation set) 20% và **tập kiểm tra** (test set) 20%. Ứng với mỗi phần ta sẽ đưa ra thông số lỗi tương ứng:

- **Tập huấn luyện:** $E_{train} = \frac{1}{m_{train}} \sum_{i=1}^{m_{train}} (\hat{y}_{i_{train}} - y_{i_{train}})^2$
- **Tập kiểm chứng:** $E_{CV} = \frac{1}{m_{cv}} \sum_{i=1}^{m_{cv}} (\hat{y}_{i_{cv}} - y_{i_{cv}})^2$
- **Tập kiểm tra:** $E_{test} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} (\hat{y}_{i_{test}} - y_{i_{test}})^2$

4. KHẮC PHỤC CHƯA KHỚP VÀ QUÁ KHỚP

4.1 Phòng tránh quá khớp trong các mô hình machine learning truyền thống

Nội dung của những phương pháp này được áp dụng trên những lớp mô hình machine learning truyền thống không bao gồm các mạng thần kinh nơ ron (neural network).

4.1.1 Giảm số lượng biến và sử dụng mô hình ít phức tạp

Như ví dụ đã được phân tích trên về các hàm số có các bậc 1, 4, 15 thì hiện tượng quá khớp thường xảy ra với hàm số có bậc cao, mặc dù nó khớp rất tốt các điểm trên tập huấn luyện. Đây là một minh chứng cho thấy chúng ta thường đối mặt với hiện tượng quá khớp khi xây dựng mô hình trên những bộ dữ liệu có kích thước nhỏ nhưng sử dụng những mô hình có độ phức tạp cao. Do đó một cách đơn giản để tránh quá khớp là giảm nhẹ độ phức tạp của mô hình bằng cách giảm bớt số lượng tham số, số lượng biến đầu vào và chuyển sang sử dụng những mô hình ít phức tạp hơn.

4.1.2 Kỹ thuật điều chuẩn (Regularization)

Trong phương pháp điều chuẩn chúng ta tìm cách cộng thêm vào giá trị của hàm mất mát (loss function) một thành phần kiểm soát để làm cho mô hình có xu hướng học được một kết quả khái quát hơn trên bộ dữ liệu huấn luyện.

$$J(\theta) = E_X(\theta) + \lambda E_\theta(\theta)$$

Trong đó, $E_X(\theta)$ là hàm mất mát ban đầu và $\lambda E_\theta(\theta)$ là số hạng chính quy hoá đóng vai trò như một biện pháp kiểm soát kết quả.

Lấy ví dụ trong phương trình hồi qui tuyến tính giữa x và y chúng ta sẽ tìm cách tối ưu hàm mất mát dạng MSE như sau:

$$MSE(x, y) = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - w^T x)^2$$

Theo phương pháp điều chuẩn, chúng ta sẽ cùng cộng thêm một phân tử gọi là thành phần điều chuẩn (*regularization term*) vào hàm MSE.

$$MSE(x, y) = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - w^T x)^2 + \underbrace{\theta R(w)}_{\text{regularization term}}$$

Thành phần điều chuẩn được cộng thêm chủ yếu là một hàm norm chuẩn L_1 hoặc L_2 của véc tơ trọng số W của mô hình và $\theta > 0$ là hệ số điều chuẩn.

Hàm norm dùng để đo khoảng cách giữa hai điểm dữ liệu nhiều chiều, tức hai vector. Trong đó, norm L_1 được định nghĩa là tổng giá trị tuyệt đối của các phần tử trong vector, norm L_2 là căn bậc 2 của tổng bình phương của các phần tử trong vector:

$$\|x\|_1 = \sum |x_i|$$

$$\|x\|_2 = \sqrt{\sum x_i^2}$$

Tổng quát hoá, norm L_{pth} là căn bậc P của tổng các phần tử trong vector mũ P:

$$\|x\|_p = \sqrt[p]{\sum |x_i|^p}$$

4.2 Phòng tránh quá khớp trong mạng nơ ron

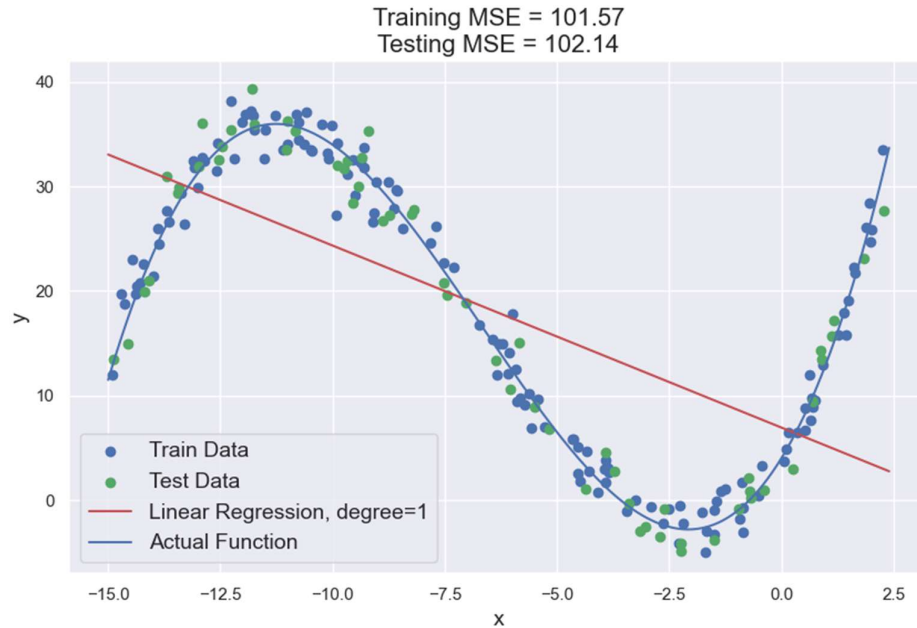
Quá khớp là hiện tượng thường gặp khi huấn luyện các mạng thần kinh nơ ron. Một phần nguyên nhân là bởi số lượng tham số của một mạng nơ-ron có thể lớn tùy ý nên hàm biểu diễn của mạng nơ-ron có khả năng biểu diễn lớn và có độ phức tạp cao. Vì vậy, Trong nhiều trường hợp bộ dữ liệu có kích thước quá nhỏ, sử dụng mạng nơ ron sẽ dẫn tới hiện tượng quá khớp.

4.2.1 Sử dụng tập kiểm chứng (Cross-validation set)

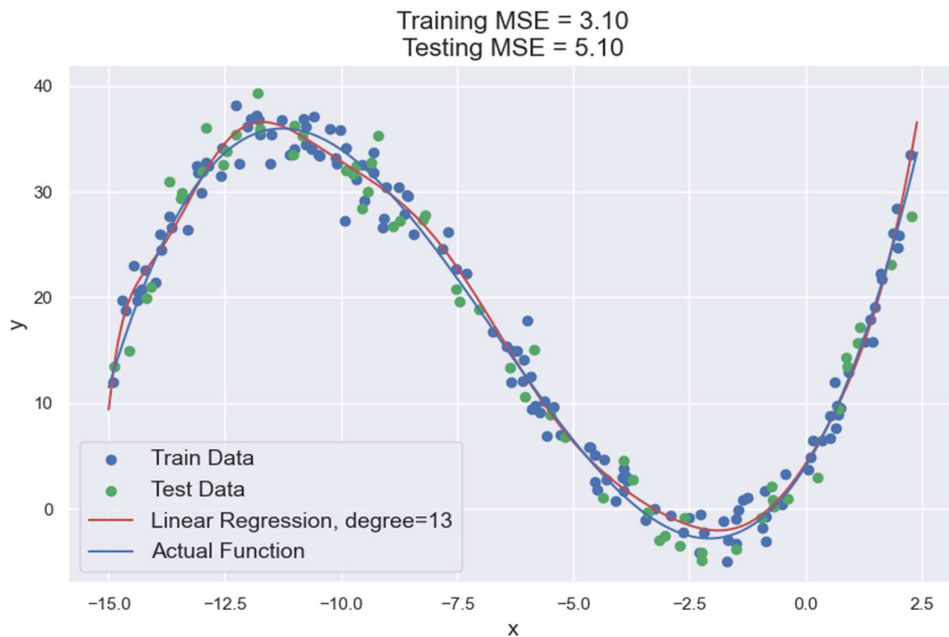
Tập kiểm chứng được tạo ra bằng cách chia đều tập huấn luyện ra thành k tập con có kích thước gần bằng nhau. Mô hình sẽ chạy k lần, mỗi lần lấy 1 trong k tập con khác với lần trước để làm tập Test và tập huấn trên $k - 1$ tập con còn lại, và mỗi lần chạy sẽ lưu lại kết quả của Test error và Train Error để mô hình cuối được xác định dựa trên trung bình các Test error và Train error. Cách làm này có tên gọi là ***k-fold cross validation***.

4.2.2 Thêm dữ liệu

Để tránh hiện tượng quá khớp, ta có thể lấy thêm dữ liệu để tập huấn mô hình. Tuy nhiên, việc làm này sẽ có trường hợp không hiệu quả. Để làm ví dụ, tạo thêm dữ liệu giống với mục 2.5 nhưng gấp 10 lần để tập huấn mô hình:



Qua đồ thị trên, thấy được rằng mô hình đơn giản ($d = 1$) không thay đổi nhiều và vẫn còn hiện tượng Underfitting.



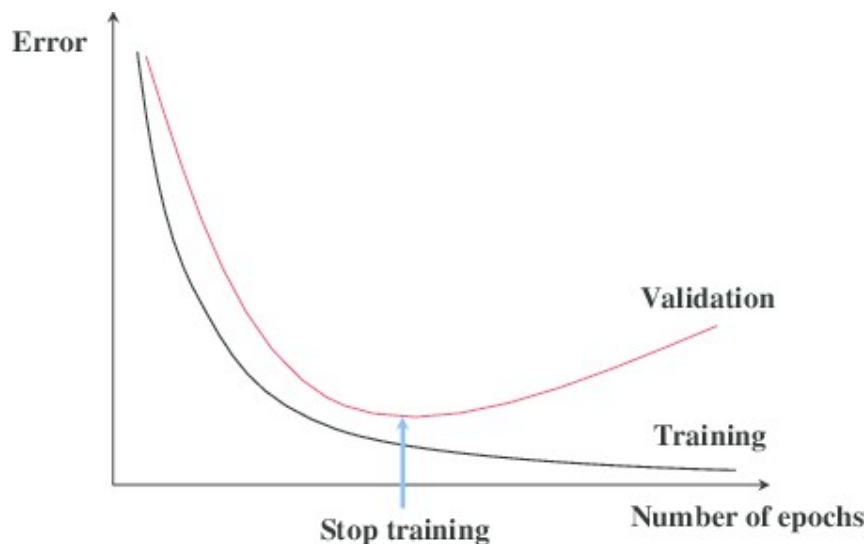
Còn mô hình phức tạp ($d = 13$) đã có nhiều thay đổi và tốt hơn hẳn trước. Điều này xảy ra vì với dữ liệu trước, mô hình được huấn luyện trên 14 điểm dữ liệu (70% của 20), mà đa thức bậc 13 có thể khớp hết được 14 điểm này nên xảy ra Overfitting vì mô hình rắng mô tả nhiều hơn dữ liệu. Với 140 điểm dữ liệu thì mô hình có thể hiểu rõ hơn về quan hệ giữa các dữ liệu thay vì nhiễu nên nó đã tăng được chất lượng mô hình. Tuy nhiên, nếu ban đầu mà ta đã tập huấn mô hình cực kỳ phức tạp như $d = 150$, thì lượng dữ liệu này vẫn là không đủ để ảnh hưởng mô hình.

→ Lấy thêm dữ liệu có thể giúp tăng chất lượng mô hình và giúp sửa hiện tượng **quá khớp** nếu mô hình không quá phức tạp nhưng sẽ không giúp sửa hiện tượng **chưa khớp**.

4.2.3 Phương pháp dừng sớm (Early Stopping)

Khi huấn luyện càng lâu thì giá trị **hàm mất mát** của mô hình trên tập huấn luyện càng nhỏ và mô hình có xu hướng khớp tốt dữ liệu trên tập huấn luyện hơn. Mặc dù sai số trên tập huấn luyện có xu hướng giảm dần theo thời gian nhưng trên tập kiểm định (validation data) điều này sẽ chưa chắc là đúng.

Phương pháp dừng sớm sẽ xác định đâu là epoch được lựa chọn làm điểm dừng phù hợp căn cứ vào sai số trên tập kiểm định. Đó là thời điểm mà sai số trên tập kiểm định bắt đầu có xu hướng tăng lên. Khi đó một quyết định dừng sớm quá trình huấn luyện sẽ giúp tránh hiện tượng quá khớp.



Điều kiện dừng được thiết lập là ngưỡng gia tăng hàm mất mát của epoch sau so với epoch trước lớn hơn $\Delta error$. Trong quá trình huấn luyện chúng ta cũng cần liên tục lưu lại các checkpoint cho mô hình sau mỗi epoch cho tới khi đạt được điểm dừng.

4.3 Các giải pháp khắc phục chưa khớp









4.3.1 Tăng độ phức tạp của mô hình

Một trong những nguyên nhân chính dẫn đến underfitting là sự đơn giản của mô hình. Nếu mô hình của bạn quá đơn giản để mô tả mối quan hệ trong dữ liệu, nó không thể khớp được dữ liệu huấn luyện. Trong trường hợp này, hãy xem xét việc sử dụng một mô hình phức tạp hơn như Decision Trees, Random Forests, hoặc Neural Networks. Các mô hình này có khả năng học được mối quan hệ phức tạp hơn trong dữ liệu và có thể giải quyết chưa khớp.

4.3.2 Tăng cường dữ liệu (Augmentation)

Tăng cường dữ liệu là một nguyên tắc bổ sung dữ liệu với một chi phí rẻ. Theo phương pháp này, từ một điểm dữ liệu chúng ta sẽ nhân bản thành nhiều điểm dữ liệu bằng cách áp dụng các kỹ thuật biến đổi mà giá trị sau biến đổi của chúng có thể mô phỏng lại một cách tương đối chính xác và đa dạng các trường hợp thực tế.

Sau đây là ví dụ về tăng cường dữ liệu trong NLP và Computer Vision:

Original	Flip	Rotation	Random crop
			
<ul style="list-style-type: none">• Image without any modification	<ul style="list-style-type: none">• Flipped with respect to an axis for which the meaning of the image is preserved	<ul style="list-style-type: none">• Rotation with a slight angle• Simulates incorrect horizon calibration	<ul style="list-style-type: none">• Random focus on one part of the image• Several random crops can be done in a row
Color shift	Noise addition	Information loss	Contrast change
			
<ul style="list-style-type: none">• Nuances of RGB is slightly changed• Captures noise that can occur with light exposure	<ul style="list-style-type: none">• Addition of noise• More tolerance to quality variation of inputs	<ul style="list-style-type: none">• Parts of image ignored• Mimics potential loss of parts of image	<ul style="list-style-type: none">• Luminosity changes• Controls difference in exposition due to time of day

Hình 6: Các hình ảnh minh họa Random Augmentation. Source: Augmentation

Image - Stanford Edu

Random Augmentation: Là việc áp dụng tập hợp các phương pháp biến đổi hình ảnh như Shift (dịch chuyển ảnh), Rotation (xoay ảnh), Bright Contrast (tạo tương phản màu sắc),... một cách ngẫu nhiên. Phương pháp này sẽ tạo ra thay đổi trên ảnh đầu vào mà không thay đổi nhãn của ảnh.

Trong NLP chúng ta có thể tăng cường dữ liệu bằng cách thay thế một số từ trong câu tại các vị trí ngẫu nhiên bằng những từ đồng nghĩa. Đối với bài toán phân loại văn bản thì đảo lộn vị trí các câu trong đoạn văn để tạo thành đoạn văn mới có cùng nhãn. Việc tận dụng các mô hình dịch máy cũng có thể giúp tạo ra một phương pháp tăng cường hiệu quả. Theo phương pháp này, từ một câu gốc A Tiếng Việt chúng ta có thể dịch sang câu B Tiếng Anh và sau đó dịch ngược trở lại từ câu B Tiếng Anh sang câu A' Tiếng Việt là một biến thể có nội dung tương tự như câu gốc A .

5. TÀI LIỆU THAM KHẢO

- 1 Overfitting - Wikipedia. (2017, August 1). Retrieved October 9, 2023, from <https://en.wikipedia.org/wiki/Overfitting>
- 2 Hai, D. M. (2017, December 24). [ML] Mô hình quá khớp (Overfitting). Retrieved October 10, 2023, from <https://dominhhai.github.io/vi/2017/12/ml-overfitting/>
- 3 tin tức bất động sản HOT & Nhanh | Meey Land, C. N., & Land, M. (n.d.). Data splitting là gì? Tỷ lệ Train/Dev/Test phù hợp nhất. Retrieved October 11, 2023, from <https://meeyland.com/chuyen-doi-so/data-splitting-la-gi-ty-le-traindevtest-phu-hop-nhat/>
- 4 Bias and Variance in Machine Learning - GeeksforGeeks. (2020, February 6). Retrieved October 12, 2023, from <https://www.geeksforgeeks.org/bias-vs-variance-in-machine-learning/>
- 5 Mean squared error - Wikipedia. (2021, April 1). Retrieved October 13, 2023, from https://en.wikipedia.org/wiki/Mean_squared_error
- 6 Agrawal, R. (2021, July 9). Master Polynomial Regression With Easy-to-Follow Tutorials. Retrieved October 15, 2023, from <https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/>
- 7 Vu, T. (2017, March 4). Bài 15: Overfitting. Retrieved October 7, 2023, from <https://machinelearningcoban.com/2017/03/04/overfitting/>
- 8 Deep AI KhanhBlog. (n.d.). Retrieved October 14, 2023, from https://phamdinhhkhanh.github.io/deepai-book/ch_ml/OvfAndUdf.html#tai-lieu-tham-khao