**Bài toán và phân tích:** Dự báo có mưa hay không có mưa sau 2-3 giờ tính từ thời điểm dự đoán.

* **Input**: có **6** input đầu vào

+ **Tốc độ gió** , **hướng gió** : Căn cứ vào kiếm thức địa lý sgk lớp 10

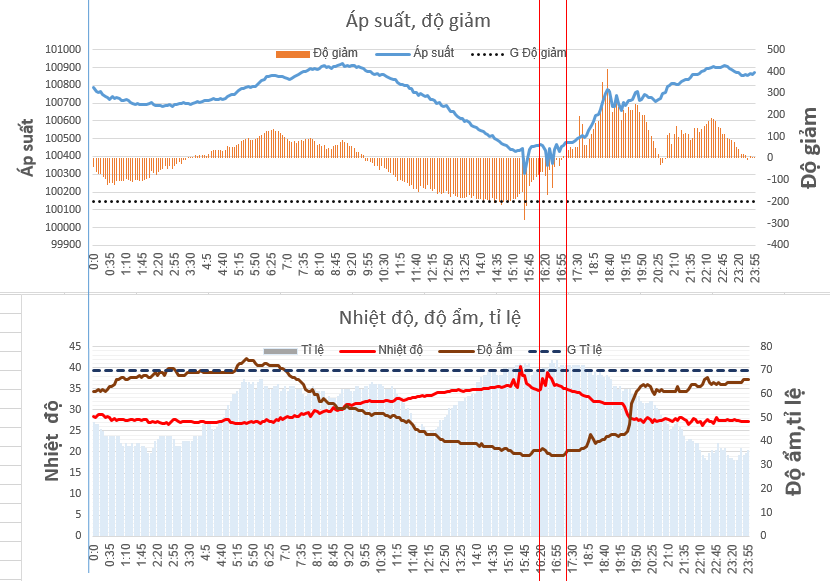
+ **Giờ trong ngày :** môi trường ở thời điểm khác nhau là khác nhau

+ **Nhiệt độ avg trong 1 giờ**

**+ Độ ẩm avg trong 1 giờ**

**+ Tổng độ giảm áp suất trong 2 giờ:**

*Căn cứ vào ý kiến chuyên gia trong tài liệu Giáo trình Khí tượng Nông nghiệp (Đoàn Văn Điếm chủ biên) của Đại học Nông nghiệp 1, tại chương XI phần 1.3b.*



*Ta thấy rằng khi áp suất giảm cao với nhiệt độ thấp và độ ẩm cao thì sẽ là dấu hiệu cho khả năng mưa*

* **Ouput**: ở đây đề xuất 2 đề xuất:

(1) **1 output** : nhận giá trị là 0 và 1.

+ Chia tập data ra thành 2 class : 1 có mưa và 1 không có mưa.

(2) **4 output** : nhận giá trị là 0 và 1.

+ Chia thành 4 class : ko mưa , mưa nhỏ, mưa vừa, mưa to.

+ Giá trị tại output nào = 1 thì thuộc class đó.

**Chon phương pháp:**

Ta đặt

X = [1, x1,x2,x3,x4,x5,x6] là vector intput.

W = [w0,w1,w2,w3,w4,w5,w6] là vector hệ số cần tối ưu.

Y = [y1] hoặc Y = [y1,y2,y3,y4] là vector kết quả dự đoán.

Như phân tích ở trên ta xem bài toán trên thuộc loại **Classification** và dữ liệu **unlinearly separble.**

1. **Linear Regression**

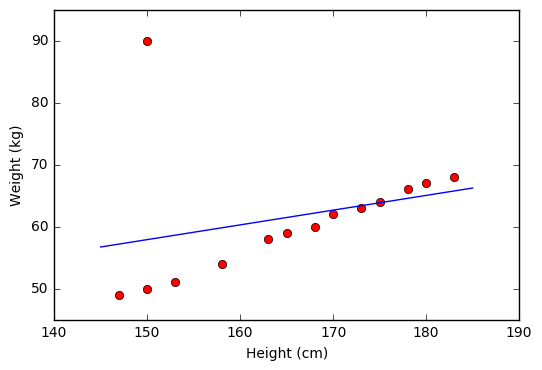
* Đây là 1 thuật toán binary regression có thể giải quyết được đề xuất 1.

. *Nhưng thuật toán này có 1 yêu cầu là dữ liệu phải* ***linearly separable.***

*. Có hạn chế là không biểu diễn được các mô hình phức tạp.*

*. Nhạy cảm với nhiệu.*

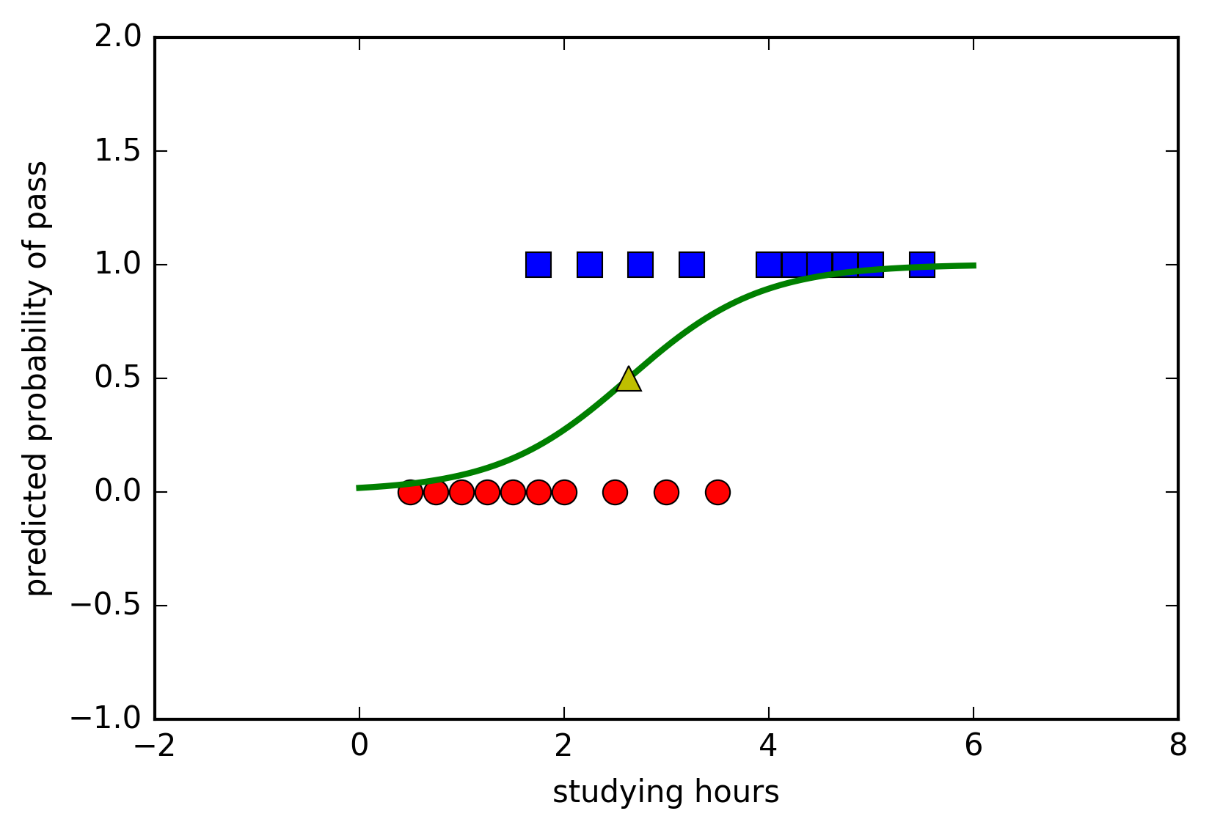
*. Boundnary là linear.*

**

* Thuật toán này không thích hợp với bài toán.

1. **Logistic Regression**

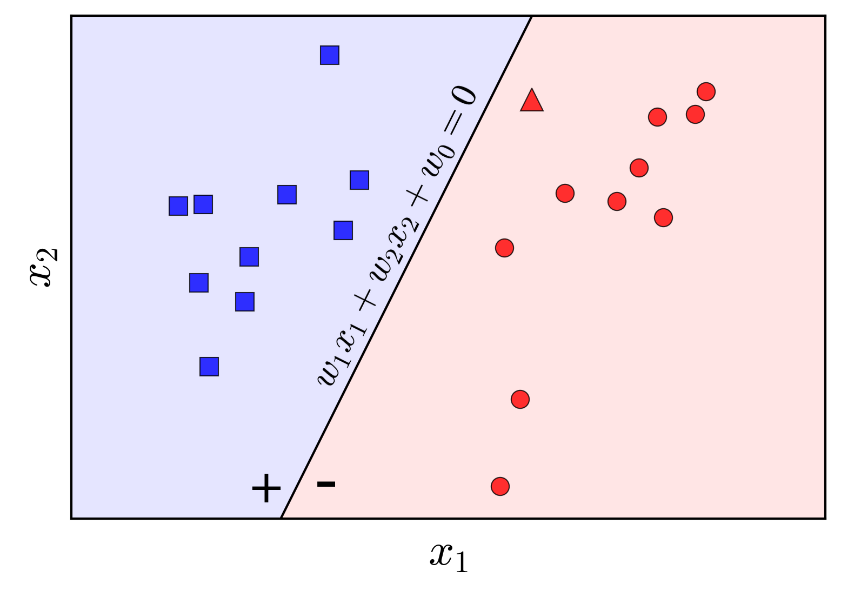
* Thuật toán này nhược điểm cũng tương tự như **Linear Reg** và **PLA** nhưng nó có vượt hơn ở chổ có thể xử lý khi dữ liệu có những điểm tiến xâu vào lãnh thổ của class khác ( gần linearly), áp dụng xác suất để tăng độ tin cậy.
* Nên có thể áp dụng để giải đề xuất 1 nhưng không khuyên khích do với dữ liệu intput như trên nhiễu rất nhiều và độ lệch giữa 2 class rất cao dễ dẫn đến dự đoán sai.

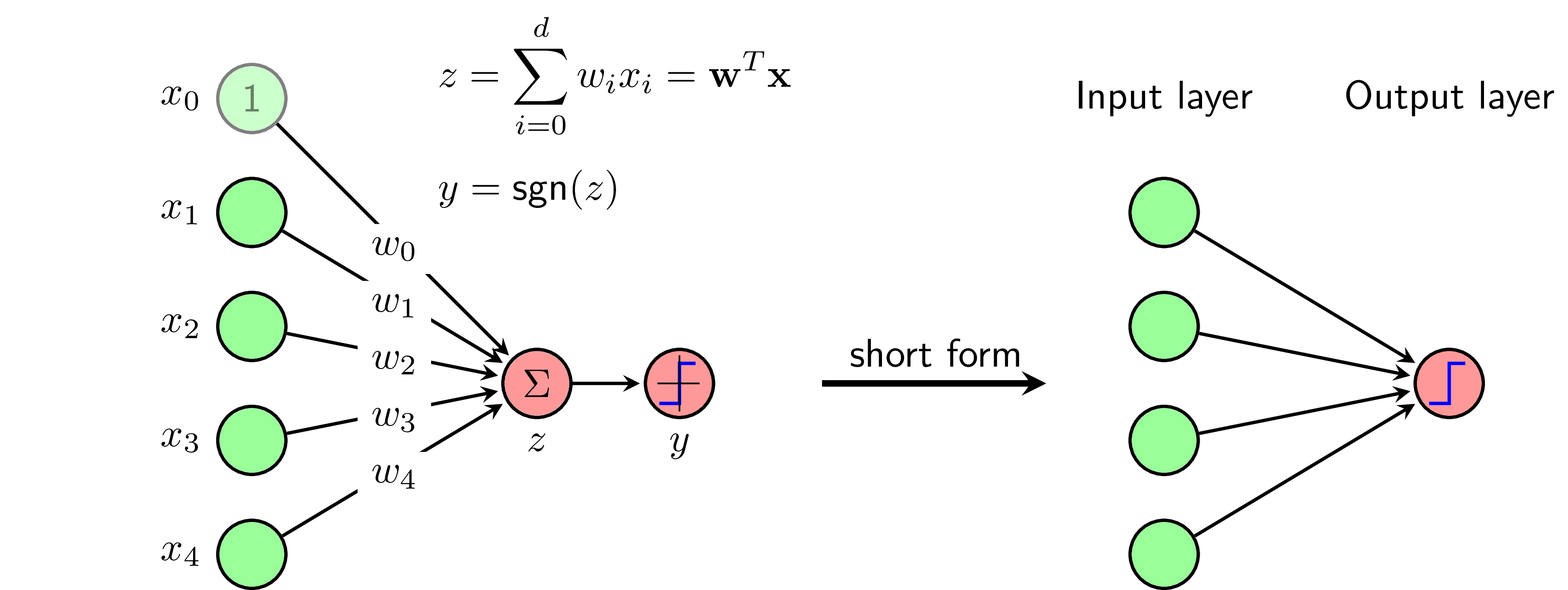


1. **Perceptron Algorithms (PLA)**

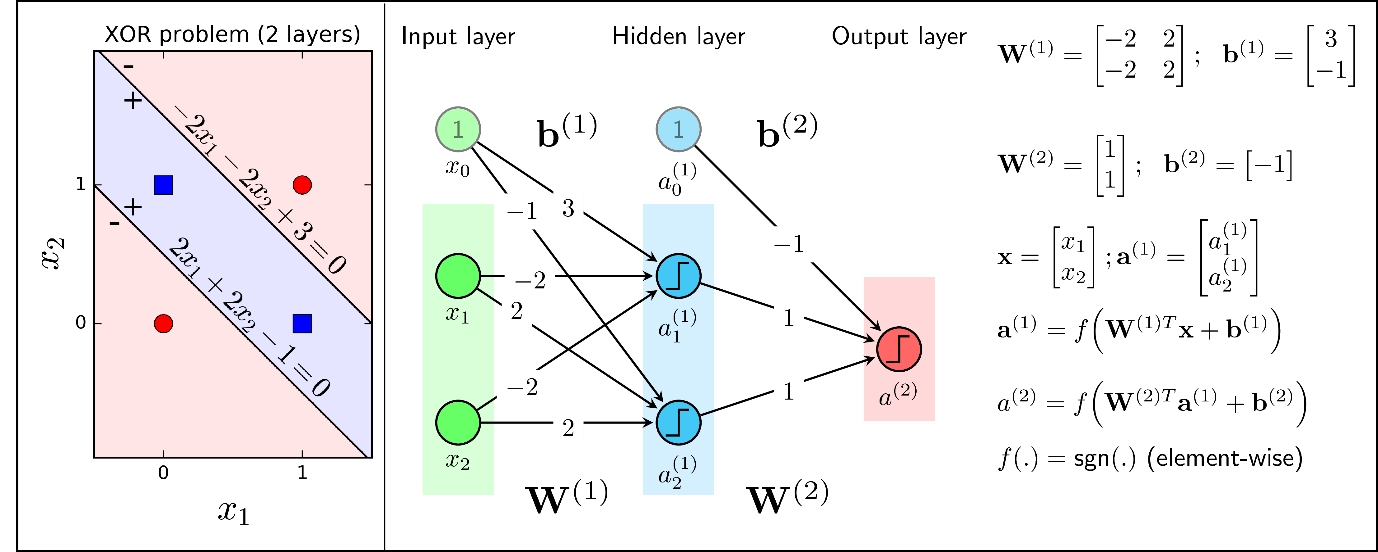
* Đây là 1 thuật toán binara regession nhưng có thể nậng cấp thành **multi-perceptron** để gải quyết vần đề nhiều class và đều có thể biểu diễn dạng **Neural Network**. Có thể giải quyết được cả 2 đề xuất của bài toán **.**

***. PA*** *không thích hợp để giải bài toán, do thuật toán đòi hỏi dữ liệu phải linearly separable, đòi hỏi độ chênh lệch số lượng giữa 2 class kô quá lớn, boundary là linear.*

**

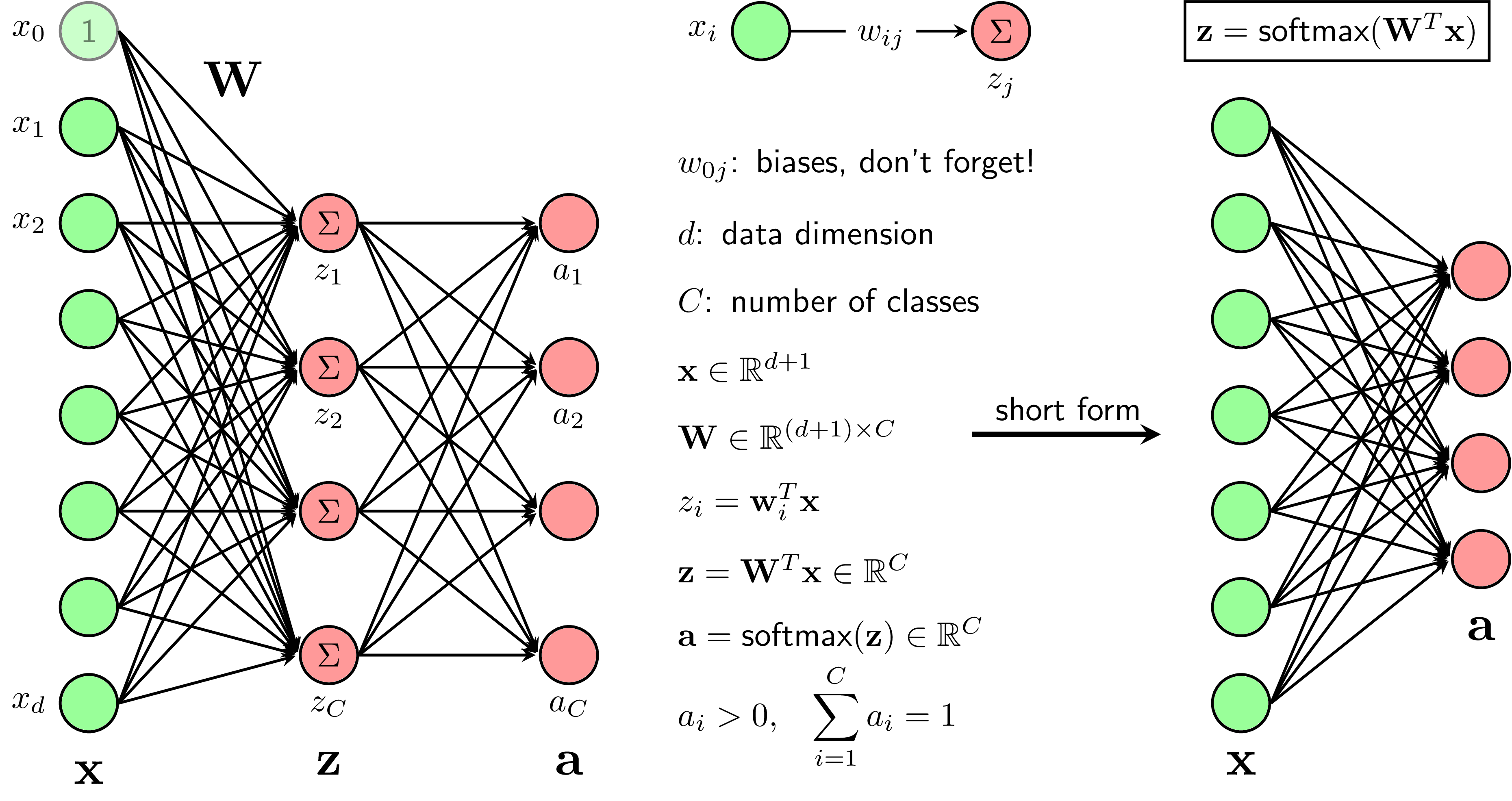
**

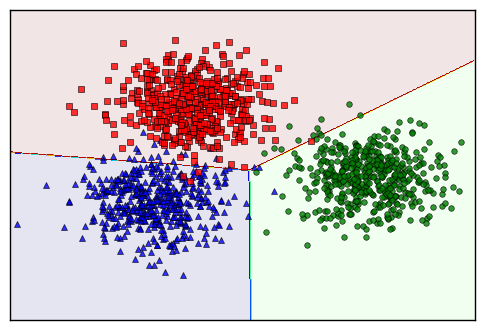
***. MLP*** *ngược lại* ***có thể áp dụng*** *cho bài toán được. Do thuật toán không giới hạn dữ liệu thuộc linear hay không, boundary phụ thuộc vào dữ liệu train.*

******

1. **Softmax Regresstion**

* Thuật toán có thể giải quyết bài toán nhiều class, tốc độ train, học nhanh, thương được áp dụng ở layer cuối của mô hình DLN (layer output) do tính tiện dụng + áp dụng xác suất nên độ tin cậy.
* Logictis Regresstion là 1 trường hợp đặc biệt của Softmax.

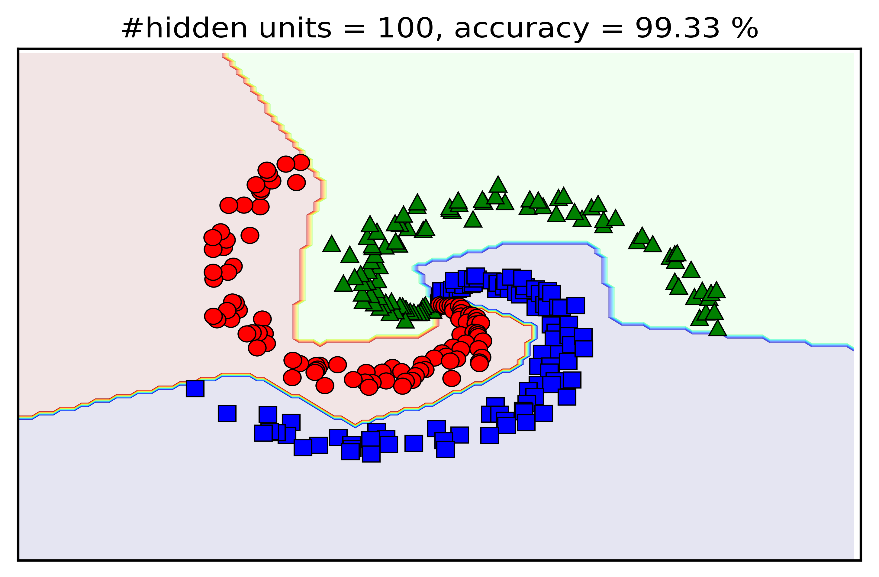
****

****

* Có thể dùng đẻ giải bài toán.

1. **Back-propagation**

* Thuật toán này có thể coi như là bản nâng cấp của MLP nó cải thiện tốc đọc train , học , tối ưu độ chính xác, và tính ứng dụng rông hơn.



* Có thể áp dụng để giải bài toán.

**Ta thấy có 3 thuật toán có thể áp dụng để giải bài toán ( có thể còn nhiều thuật toán ML khác có thể nhưng chỉ xét ở những thuật toán thông dụng nhất)**

**Overfitting**

Overfitting là một hiện tượng không mong muốn thường gặp, người xây dựng mô hình Machine Learning cần nắm được các kỹ thuật để tránh hiện tượng này.

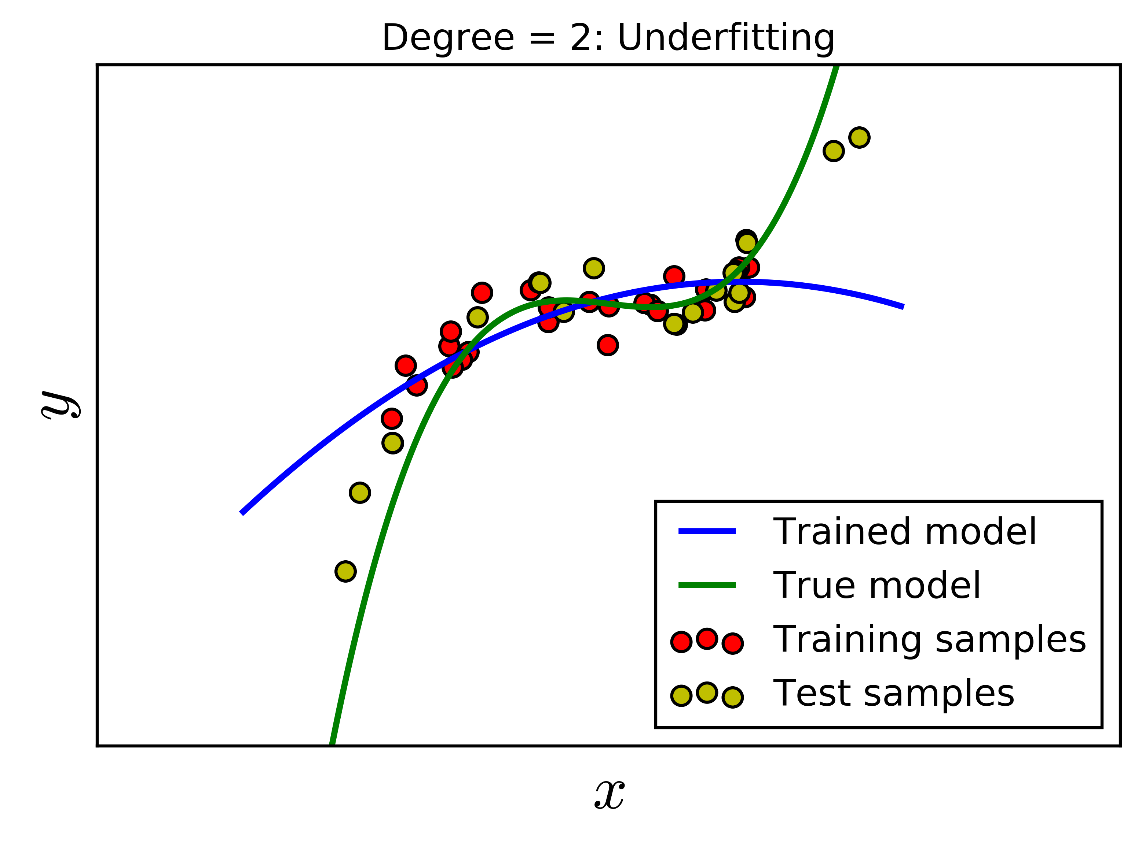
Nếu một mô hình quá fit với dữ liệu thì nó sẽ gây phản tác dụng! Hiện tượng quá fit này trong Machine Learning được gọi là overfitting.

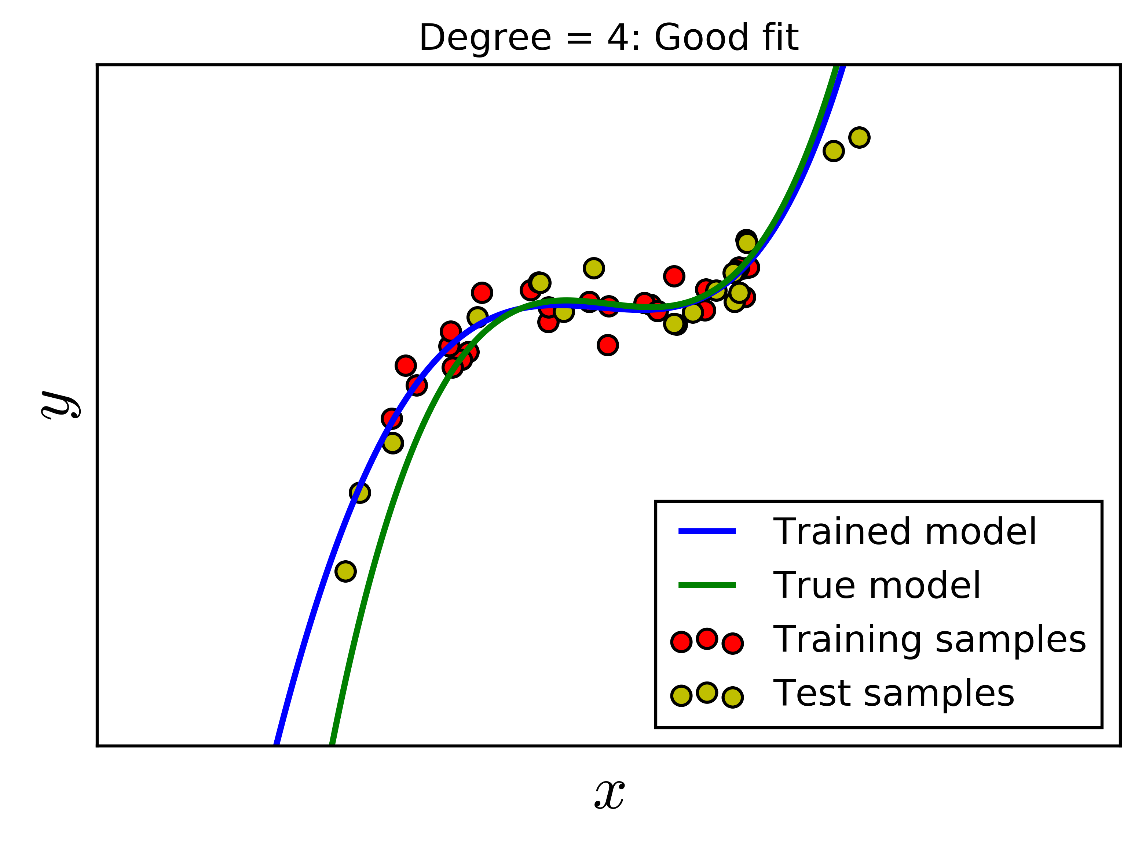
Vi dụ với bài toán:

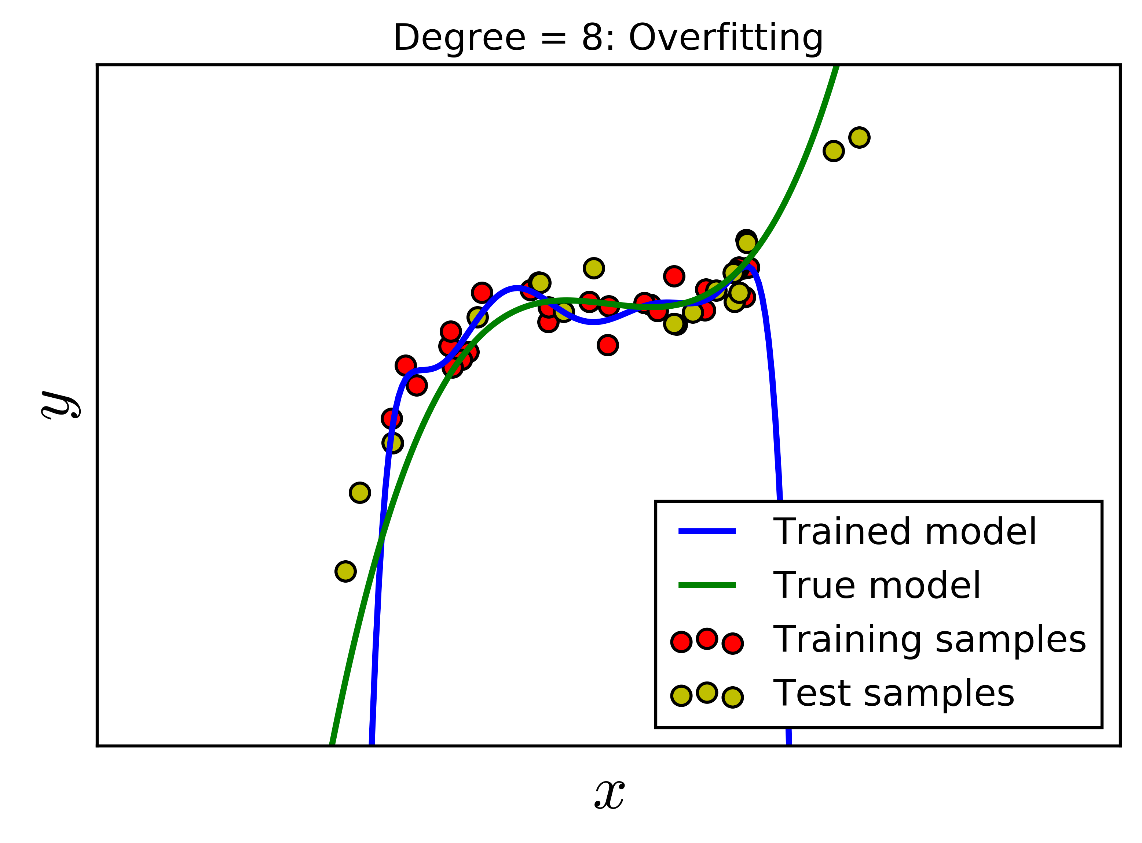
*Có 50 điểm dữ liệu được tạo bằng một đa thức bậc ba cộng thêm nhiễu. Tập dữ liệu này được chia làm hai, 30 điểm dữ liệu màu đỏ cho training data, 20 điểm dữ liệu màu vàng cho test data. Đồ thị của đa thức bậc ba này được cho bởi đường màu xanh lục. Bài toán của chúng ta là giả sử ta không biết mô hình ban đầu mà chỉ biết các điểm dữ liệu, hãy tìm một mô hình “tốt” để mô tả dữ liệu đã cho.*

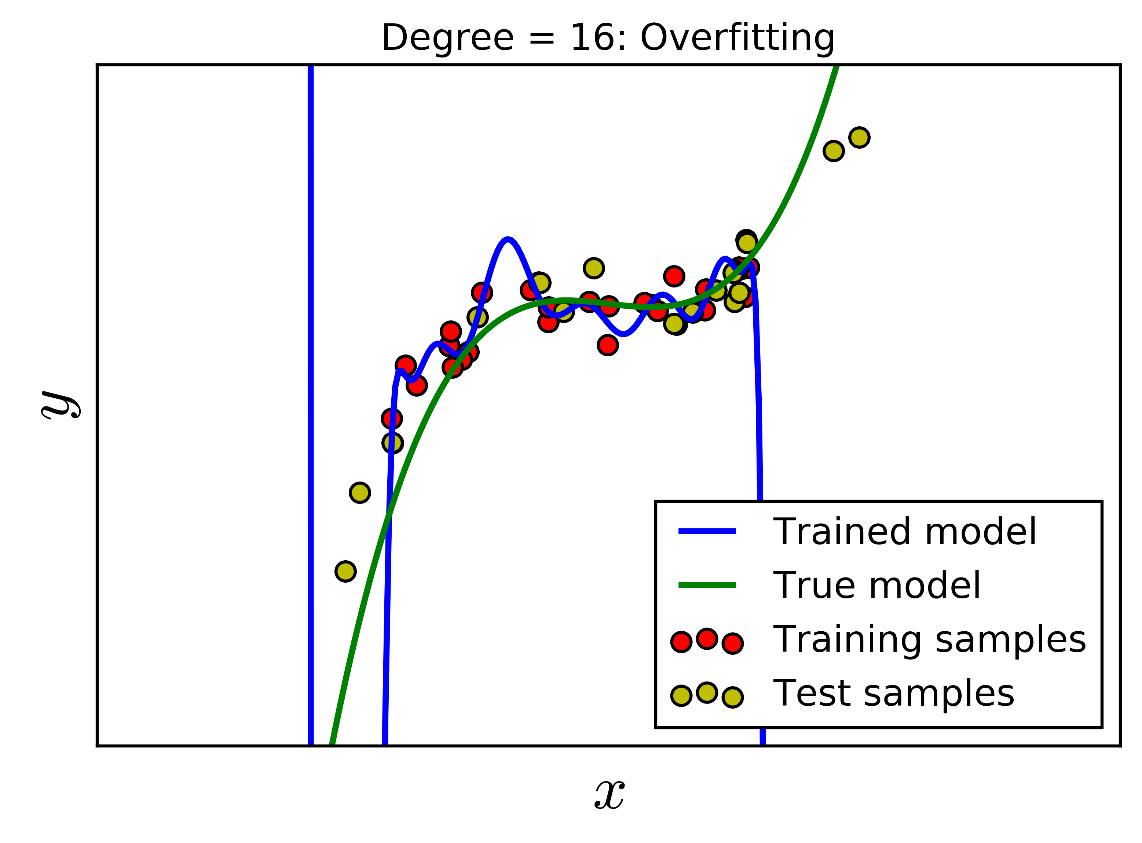
x=[1,x,x^2,x^3,…,x^d]^T có d bật. Một đa thức bậc không vượt quá 29 có thể fit được hoàn toàn với 30 điểm trong training data

Ta xét d = 2, 4, 6, 8, 16 được









**train error**

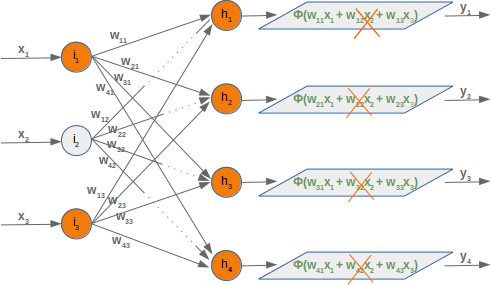
**test error**

Một mô hình được coi là tốt (fit) nếu cả train error và test error đều thấp.

Nếu train error thấp nhưng test error cao, ta nói mô hình bị overfitting. Nếu train error cao và test error cao, ta nói mô hình bị underfitting.

**Dropout**

* Một cách giải quyết vấn đề overfitting (gần đây)
* Dropout là cách thức mà chúng ta giả định một phần các unit bị ẩn đi trong quá trình training, qua đó làm giảm tích hòa trộn (hay nói cách khác là 1 hidden unit không thể dựa vào 1 unit khác để sửa lỗi lầm của nó, dễ cho chúng ta thấy các hidden unit không đáng tin cậy).
* Tại mỗi step trong quá trình training, khi thực hiện Forward Propagation (Lan truyền xuôi) đến layer sử dụng Drop-Out, thay vì tính toán tất cả unit có trên layer, tại mỗi unit ta “gieo xúc xắc” xem unit đó có được tính hay không dựa trên xác suất p



* Cách thức hoạt động của dropout là để đạt được kết qủa trung bình của việc train nhiều mạng con trong network (bằng việc giả định ẩn đi % unit) thay vì chỉ lấy kết quả dựa trên việc train 1 mạng mẹ.