Mạng neuron
và ứng dụng trong xử lý tín hiệu

Giảng viên

Trần Thị Thanh Hải

International Research Institute MICA

Multimedia, Information, Communication & Applications

UMI 2954

Hanoi University of Science and Technology

1 Dai Co Viet - Hanoi - Vietnam

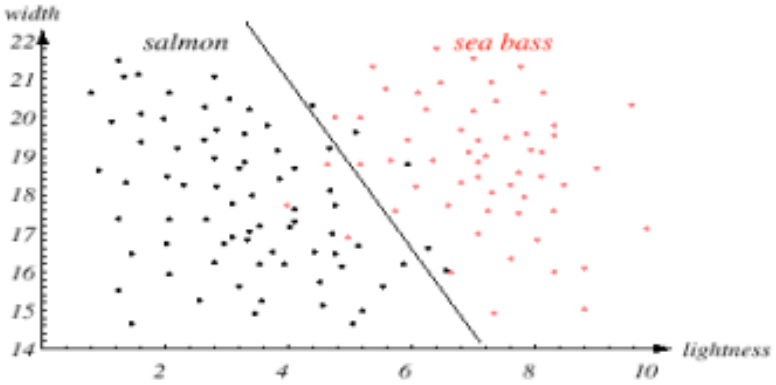
Bài 7:

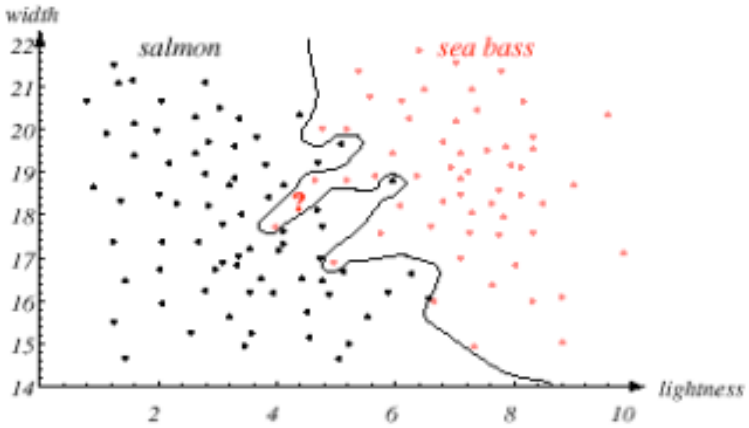
Đánh giá một hệ thống phân lớp

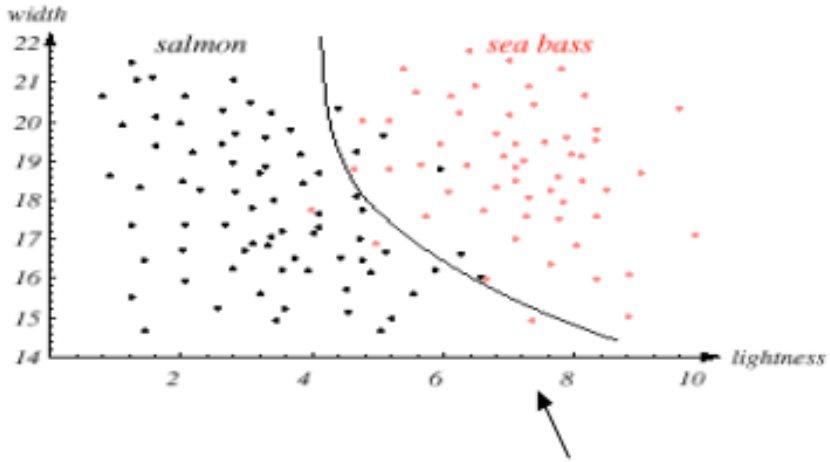
Overfitting and underfitting

Giới thiệu

Mô hình nào tốt nhất ?







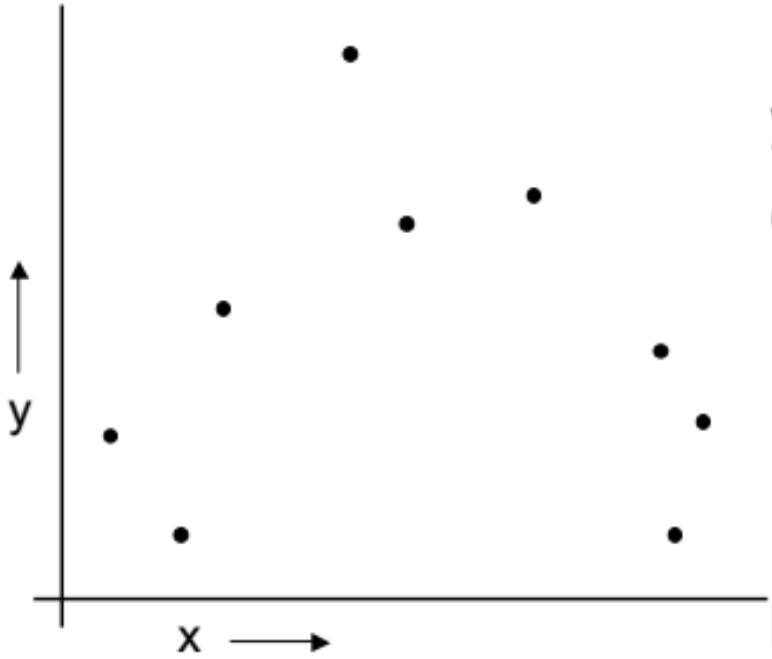
People like this one. Why?

Giới thiệu

Mô hình nào tốt nhất ?

- Hiệu năng của mô hình trên tập huấn luyện là tốt
- Tuy nhiên hiệu năng của mô hình trên dữ liệu mới là vô cùng quan trọng

Một số ví dụ

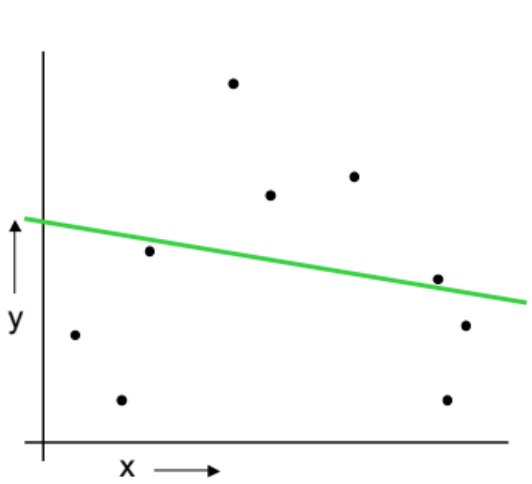


y = f(x) + noise

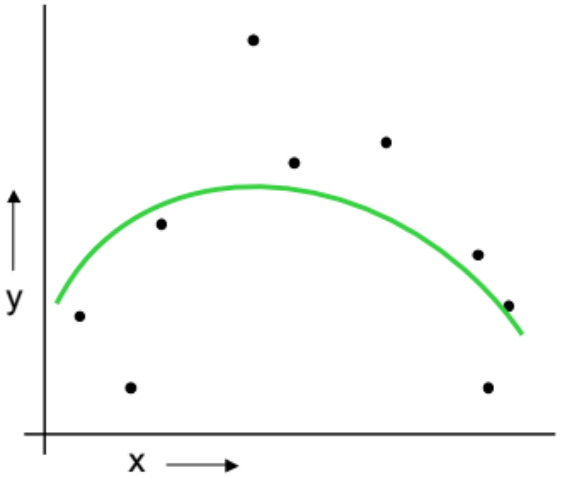
Can we learn f from this data?

Let's consider three methods…

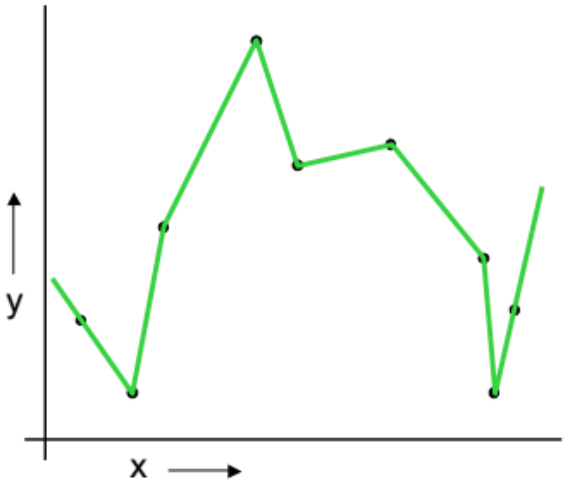
Mô hình nào tốt nhất ?



$y = w_0 + w_1 \cdot x$



$y = w_0 + w_1 \cdot x + w_2 \cdot x^2$



y = complicated

Mục tiêu

- Khi xây dựng một mạng neuron để thực hiện bài toán phân lớp, cần đánh giá xem mô hình huấn luyện được có tốt hay không ?
- Hiệu năng của mô hình thường được đánh giá trên tập dữ liệu kiểm thử (test data)

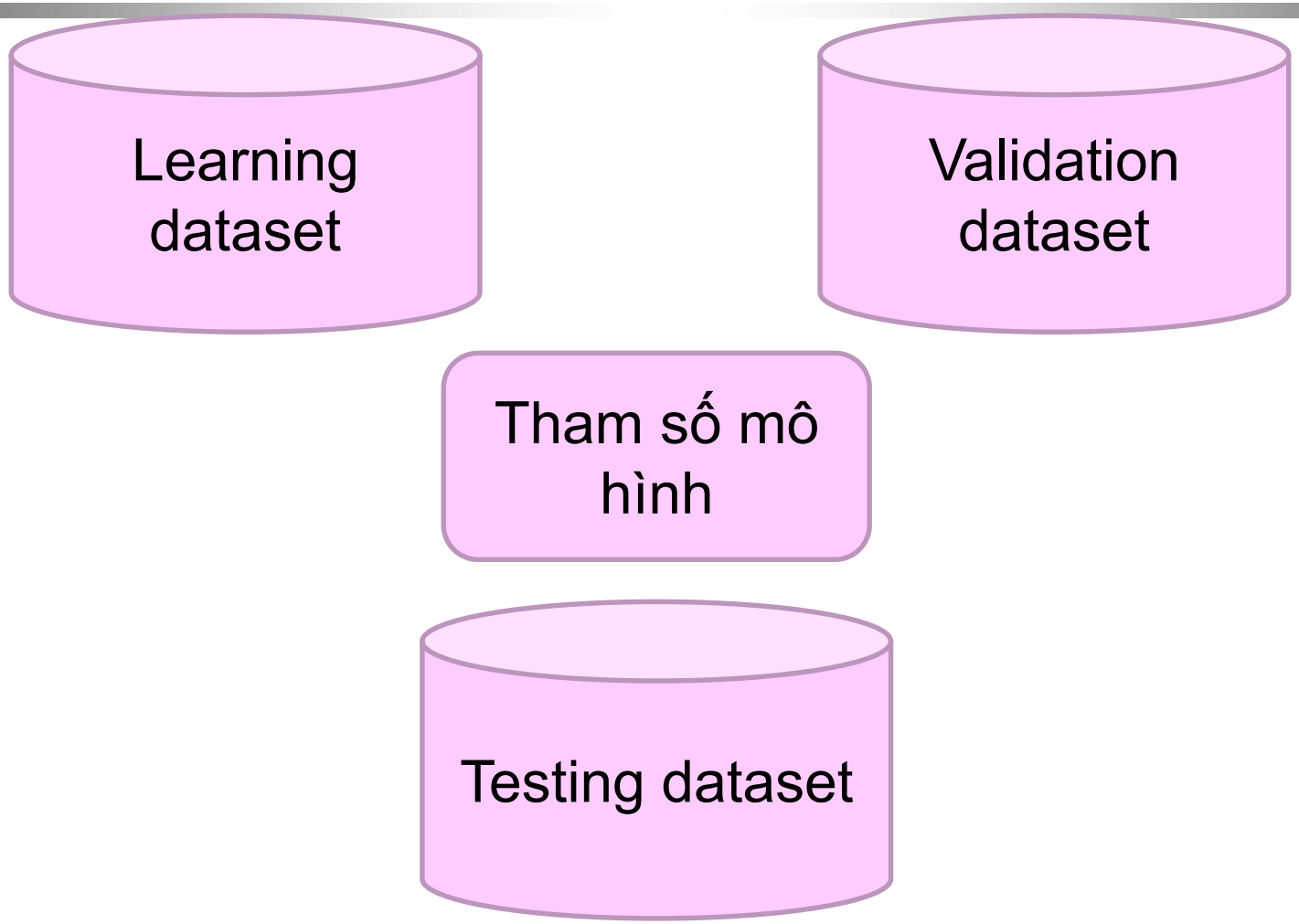
Giới thiệu chung

- **[Haykin94] đã chứng minh:** với một tập dữ liệu bất kỳ, với một sai số bất kỳ, hoàn toàn có thể xây dựng một mạng neuron đạt được sai số này.
- Tuy vậy trong những trường hợp này, mạng neuron có cấu trúc đồ sộ và chưa chắc đã cho đáp ứng tốt với một đầu vào mới: nghĩa là mạng có khả năng tổng quát hóa thấp
- Câu hỏi đặt ra: làm thế nào để xây dựng mạng đáp ứng được các yêu cầu đã biết và có xác suất thành công cao với đầu vào mới.

Giải pháp

- **Sử dụng hai tập dữ liệu**
  - Một tập sử dụng để xây dựng (học) các tham số của mô hình mạng (learning dataset)
  - Một tập sử dụng để kiểm tra các tham số (testing dataset)
  - Lưu ý: tập dữ liệu kiểm tra không chứa các mẫu trong tập học
- **Hiện nay, một số phương pháp sử dụng tập dữ liệu kiểm chứng (validation test)**

Dữ liệu huấn luyện và thử nghiệm



"F:\Mang Neuron - 2017\refs\Training_Testing_Validation_Dataset\mod_10_eval_train_test_xval.ppt"

Cấu trúc mạng

- Ánh xạ phi tuyến phải bao trùm tốt nhất các khu vực của không gian đầu vào để biết được cần phải ánh xạ từng khu vực như thế nào
- Như vậy đòi hỏi cấu trúc mạng đủ lớn, bậc phi tuyến của mạng phải tương đương với bậc phi tuyến của ánh xạ cần tìm
- Tuy nhiên trên thực tế ta không dự báo trước được bậc phi tuyến của ánh xạ cần tìm
- Nếu thiết kế mạng quá phức tạp hoặc quá đơn giản đều có thể không đạt hiệu năng mong muốn

Cấu trúc mạng

- Khả năng tổng quát hóa của mạng
- Hiện tượng học quá khớp (overfitting) và học quá ít (underfitting)

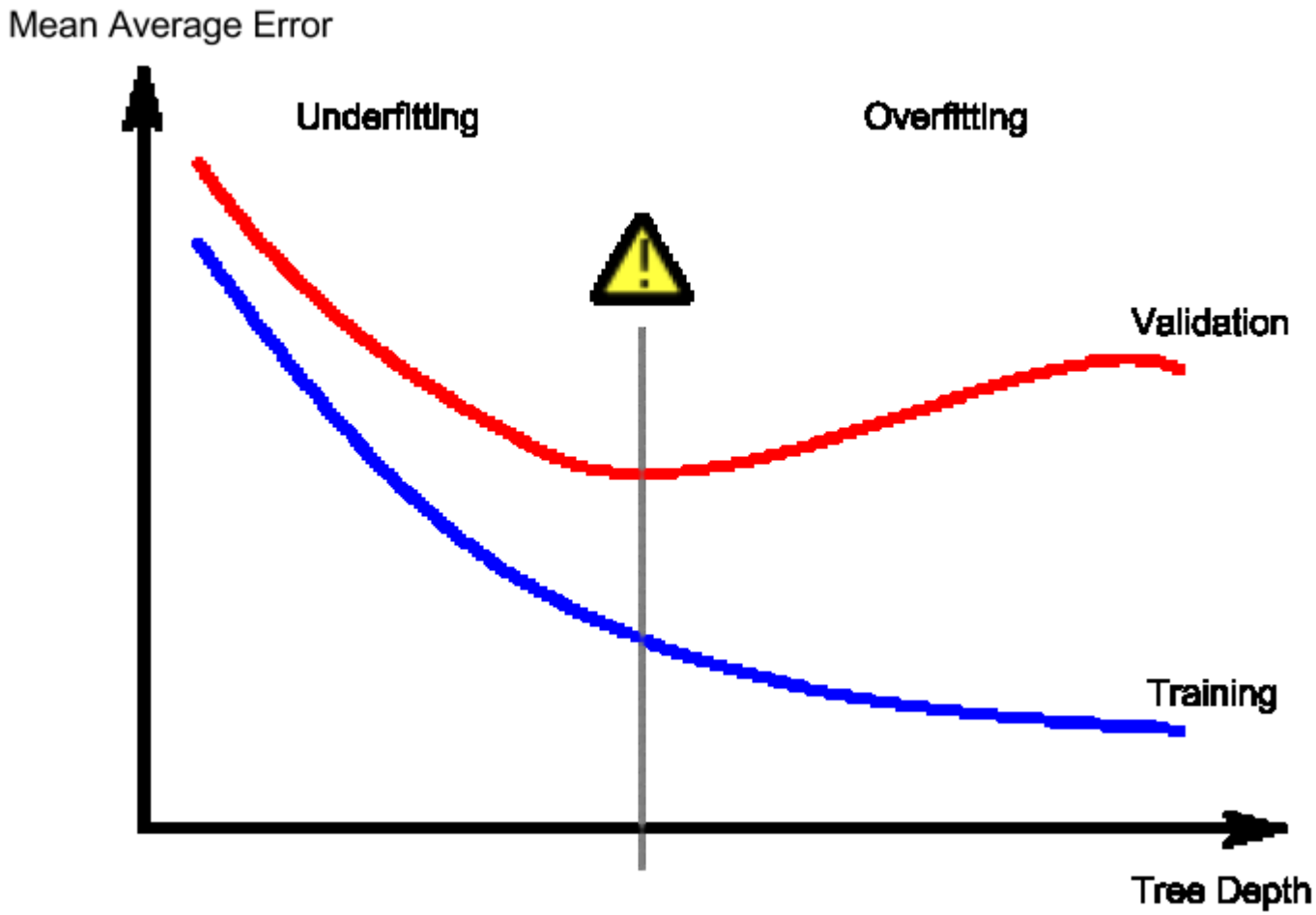
Overfitting

- Một mạng quá phức tạp có thể học rất chính xác các mẫu đã có đến mức không thể giải quyết được các mẫu mới (hiện tượng tương tự như học tủ, học lệch) sẽ dẫn tới trường hợp học quá khớp
- Học quá khớp nguy hiểm vì nó có thể đưa ra đáp ứng quá lệch so với giá trị mong muốn
  - Khi dữ liệu ít, các thông số của mạng lớn. Học quá fit dẫn đến tính khái quát hóa không tốt và sẽ gây lỗi khi thử nghiệm với dữ liệu mới
- Làm thế nào để tránh overfitting ?

Overfitting

- **Sử dụng nhiều mẫu**
- **Nếu số lượng mẫu**
  - Trong trường hợp có nhiễu: lớn hơn 30 lần số lượng tham số
  - Trong trường hợp không nhiễu: lớn hơn khoảng 5 lần
- **Tuy nhiên nếu như ta đã có sẵn trước một bộ số liệu mẫu thì ta không thể giảm số lượng tham số cần điều chỉnh đi được**

Under fitting



Source: https://www.kaggle.com/dansbecker/underfitting-overfitting-and-model-optimization

Các yếu tố ảnh hưởng đến độ phức tạp của mạng

- **Yếu tố ảnh hưởng đến độ phi tuyến của mạng là số lớp ẩn**
- **Số neuron ẩn và lớp ẩn phụ thuộc vào nhiều yếu tố**
  - Số đầu vào và đầu ra của mạng
  - Số cặp mẫu trong tập số liệu
  - Lượng nhiễu trong dữ liệu đầu vào
  - Độ phức tạp của ánh xạ cần tìm
  - Cấu trúc của mạng
  - Dạng hàm truyền đạt
  - Thuật toán học
  - Các quá trình hỗ trợ

Đề xuất lựa chọn neuron ẩn

- [Blum92]: Số neuron ẩn giới hạn trong khoảng đầu vào và số đầu ra của mạng
- [Swingler96]: Mạng neuron có một lớp ẩn thì không sử dụng quá 2N neuron ẩn. Trong đó N là số đầu vào của mạng
- [Boger97]: Số neuron lớp ẩn sẽ tương đương với số thành phần chính PCA được sử dụng để tái tạo từ 70 đến 90% mức độ biến thiên của dữ liệu.

Một số cách để tránh overfitting

- **Regularization**: kỹ thuật giảm thiểu phức tạp của mô hình
- **Early stopping (Dừng sớm)**: dừng thuật toán trước khi giá trị mà mất mát quá nhỏ.
  - Sử dụng một tập train và một tập validation
  - Sau N bước lặp, đều tính train error và validation error
  - Thuật toán dừng khi validation error có chiều hướng tăng lên. Khi đó ta dừng thuật toán và quay lại mô hình với điểm validation error nhỏ

Một số cách để tránh overfitting

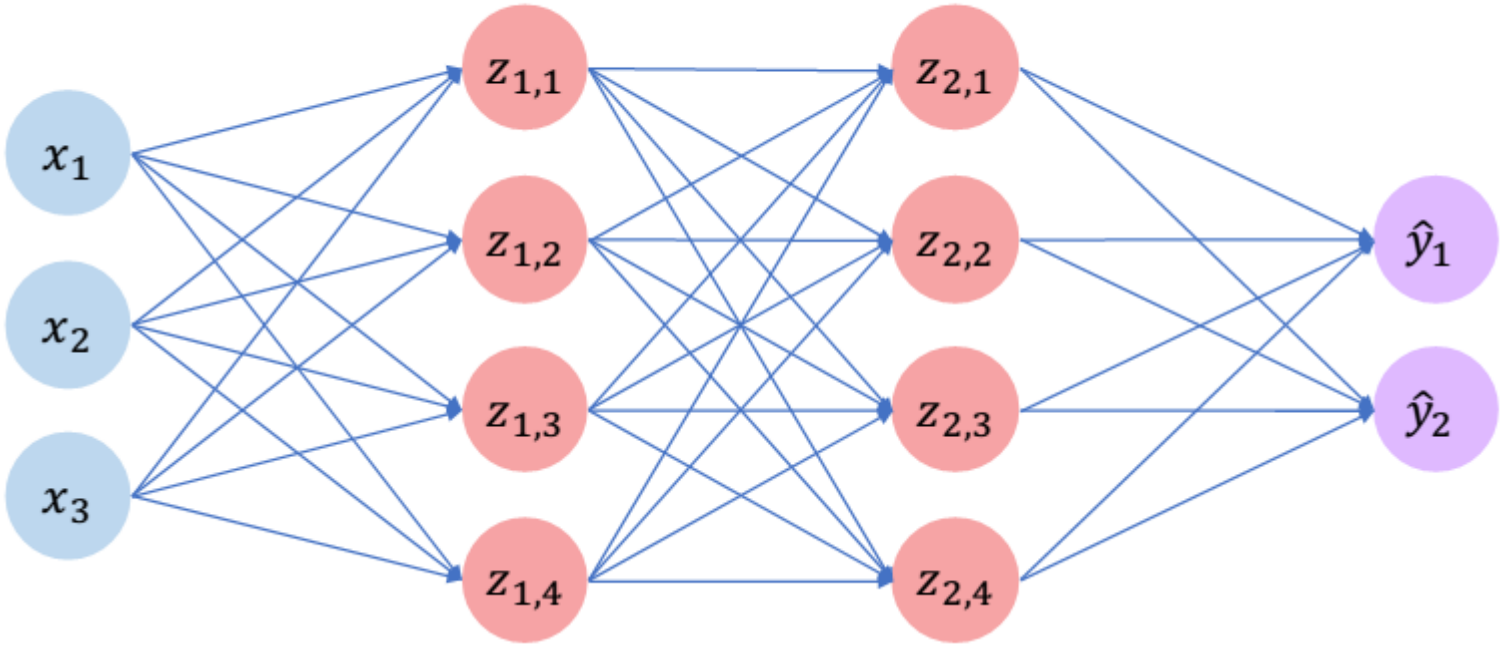
- **Regularization**: kỹ thuật giảm thiểu phức tạp của mô hình
- **Thêm số hạng vào hàm mất mát (regularized loss function)**

$$J_{\text{reg}}(\theta) = J(\theta) + \lambda R(\theta)$$

  - Số hạng R(θ) càng lớn càng thể hiện mô hình phức tạp và ngược lại
  - λ: một số dương để cân bằng cho hai đại lượng J và R. Thông thường λ được chọn là nhỏ để không làm sai khác giữa nghiệm của J và J_{reg}

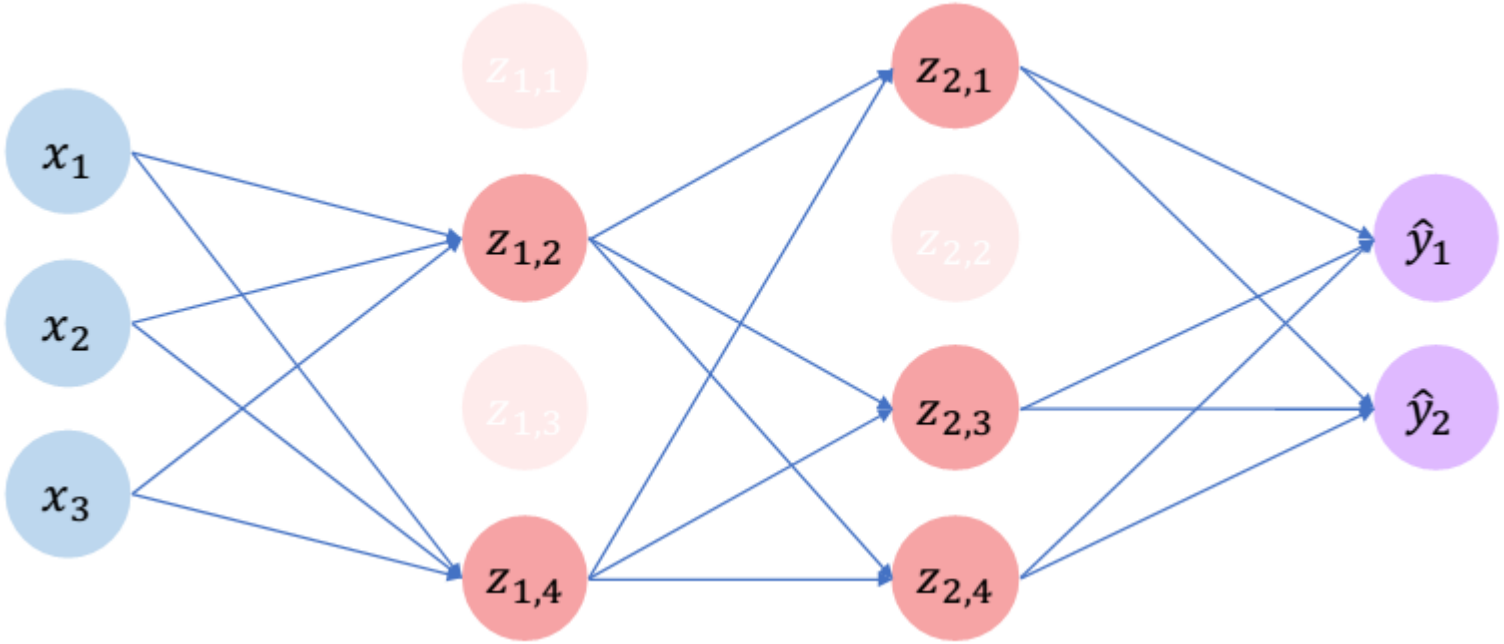
Một số cách để tránh overfitting

- **Regularization**: kỹ thuật giảm thiểu phức tạp của mô hình
- **Drop out:** trong quá trình huấn luyện mạng, desactivate (set to 0) ngẫu nhiên một số nút của mạng



Một số cách để tránh overfitting

- **Regularization**: kỹ thuật giảm thiểu phức tạp của mô hình
- **Drop out:** trong quá trình huấn luyện mạng, desactivate (set to 0) ngẫu nhiên một số nút của mang



Một số cách để tránh overfitting

- **Regularization**: kỹ thuật giảm thiểu phức tạp của mô hình
- **Drop out:** trong quá trình huấn luyện mạng, desactivate (set to 0) ngẫu nhiên một số nút của mạng
  - Thông thường: có thể dropout đến 50% trong các lớp
  - Hướng mạng không dựa trên duy nhất một số nút nào đó

Tài liệu tham khảo

- Giáo trình Mạng neuron – PGS.TSKH Trần Hoài Linh
- Một số tài liệu khác (link trong slide tương ứng)
- Cross-validation for detecting and preventing overfitting
  - Andrew W. Moore
  - CMU
- https://machinelearningcoban.com/2017/03/04/overfitting/#-regularization