Week 1

một vài định nghĩa và note

high bias (underfiting): kiểu linear chỉ cover đc một phần của data

high variance (overfit): kiểu quá uốn lượn khiến train thì ngon predict thì dở

hình ảnh cụ thể hơn từ coursera

Diagram, scatter chart

Description automatically generated

hình ảnh trung của việc huấn luyện một model và thực thi chúng

A picture containing text, person

Description automatically generated

hình trên là cơ bản quá trình lập của việc train model và cải thiện cho tới khi hoàn tất

nếu bị rơi vào high bias (train + predict tệ)

thì nên tăng kích thước mạng

hoặc tăng thời gian train

nói chung là kiểu phức tạp hóa model lên

nếu bị rơi vào high variance (overfit)

thì nên tăng số lượng tập train

ngoài ra còn regulazation

giải thích tạm thời regularziation

thêm biến này vào có thể giảm overfit vì

cảm giác như là chấp nhận sai số trong giới hạn cho phép

việc này có thể loại đc điểm ngoại lai

tăng tính tổng quát

nhưng nếu biến này quá lớn (lớn hơn hàm loss)

khi đó model trở nên vô dụng

toàn bộ trọng số là 0 do muốn hàm min

dẫn tới underfit

một ví dụ về ảnh hưởng của lamda tới kết quả thực nghiệm, lamda nhỏ và lamda quá lớn

Chart, scatter chart

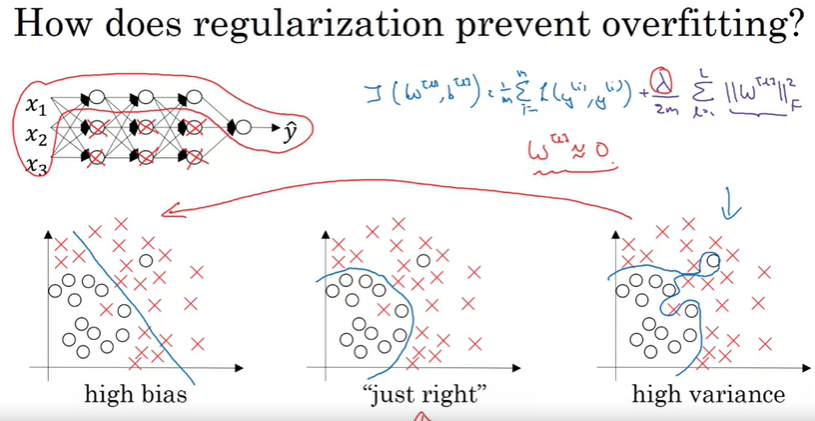
Description automatically generated

hình ảnh dưới thể hiện việc để lambda quá lớn có thể dẫn tới việc

toàn bộ trọng số bị mất đi giá trị, việc này dẫn tới có thể trở thành high bias

mạng sẽ trở về trạng thái đơn giản (linear) (underfit)

hay chính là việc quá “phò”, “dễ dãi” cho phép sai số lớn hơn cần thiết



giải thích việc có thể trở thành linear như sau

do lamda lớn nên w sẽ nhỏ

dẫn tới việc các activation node sẽ có giá trị nhỏ

các node có giá trị nhỏ nên cho vào hàm activate

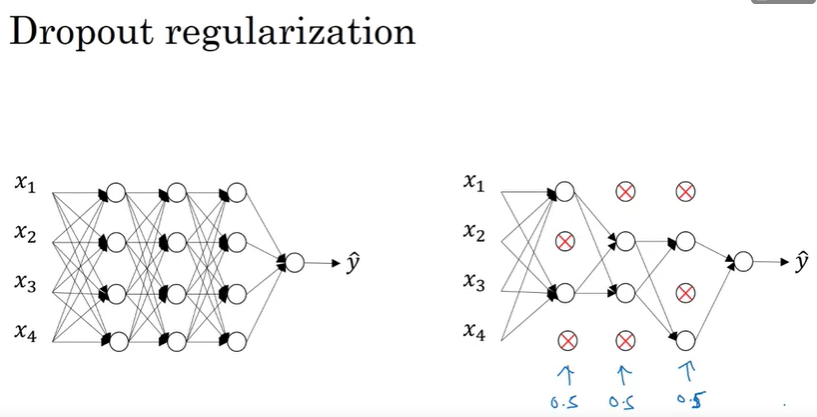
sẽ nằm ở khoảng giá trị gần 0 mà đoạn này gần như linear

dẫn tới toàn mạng sẽ trở thành một linear

Chart, line chart

Description automatically generated

kĩ thuật dropout regulaziation



ý tưởng với một phần tử train, sẽ random loại một phần của một lớp

việc này sẽ làm giảm overfit vì mạng đang trở nên đơn giản (train một mạng con nhỏ hơn)

dẫn dần sẽ tìm đc các nút cần thiết và loại đi các nút không cần 😁

ý tưởng dropout cũng đi kèm vs một vài khó khăn, chú ý

do việc mất “bi” ngẫu nhiên nên activation node cũng sẽ có những kết quả không ổn định

việc không ổn định này có thể ảnh hưởng tới grandient decent

một cách nhỏ là “inverted dropout” bằng cách như sau

VD: giả sử tỉ lệ mất là 0.5 thì toàn bộ giá trị của nút còn lại đều đc chia cho 0.5

việc này sẽ giảm thiểu đc một phần sự mất mát của các nút đó

hình ảnh giải thích (dùng ma trận random {0,1} nhân vs các nút sẽ làm “mất” (về 0) các nút dropout

Text, letter

Description automatically generated

một vài methods regularziation (tránh overfit)

VD: nhận diện con mèo

muốn thêm data những tốn công

flip ảnh con mèo, xoay ảnh con mèo, nghiên ngả

A picture containing timeline

Description automatically generated

early stopping

thay vì chỉ quan tâm tới hàm loss của tập train giờ ta quan tâm tới cả tập dev (test)

ngay thì thấy có dấu hiện overfit (tập train thì giảm, mà tập dev (test) lại tăng) thì dừng

Chart, line chart

Description automatically generated

downside của early stopping lại chính là việc quan tâm tới 2 yếu tố cùng một lúc

điều này nếu “may mắn” thì sẽ đạt đc mục tiêu

nhưng cũng không có gì chắc chắn là nó sẽ đạt đc mục tiêu

do đang có 2 mục tiêu là giảm chi phí + check overfit

dẫn tới việc chưa đạt tới mức “hoàn hảo”

chưa kể việc triển khai còn khoai hơn các method đơn giản và OK khác

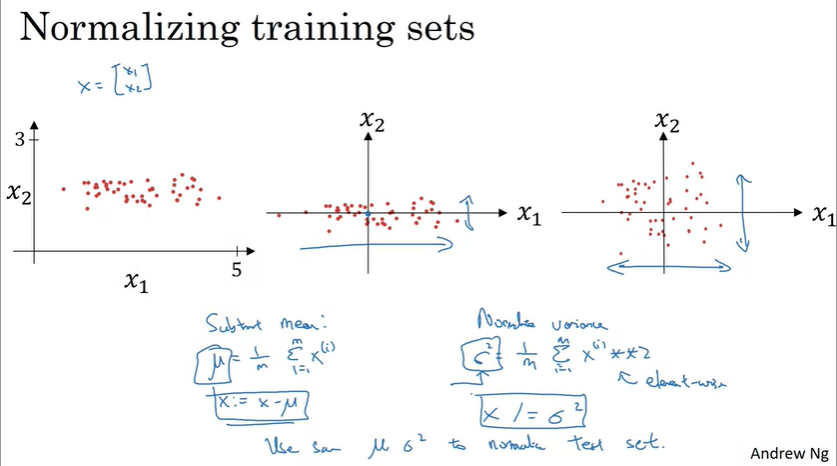
thay vào đó ta chỉ nên follow một method

VD: việc dùng L2 vào trong giảm chi phí chính ra lại nhất quán và dễ thực thi

nhưng cũng có nhược điểm là phải thử rất nhiều lamda :V

nói cung là tính nhất quán trong code =)))

normalize inputs



phát đầu tiên co các điểm về quỹ đạo chuẩn bằng cách tính trung bình và sau đó dịch toàn bộ giá trị về điểm đó

(subtract mean)

phát thứ 2 làm căng kích thước sao cho range của các chiều same same nhau (VD: quy đồng hết về 1)

(normalize the variance)

việc normalize input khiến

dễ nhìn và so sánh các thuộc tính vs nhau trên biểu đồ

gradient decent có thể hoạt động chính xác hơn

Diagram

Description automatically generated

Vanishing / Exploding gradients

xảy ra do độ sâu của mạng quá lớn và đa số trọng số trong ma trận Weight

nhỏ hơn 1, VD: 0.5, 0.2 sẽ khiến giá trị siêu nhỏ (vs máy tính là về 0)

lớn hon 1, VD: 1.5, 2,... sẽ khiến giá trị siêu to (máy tính thì tràn số)

chỉ cần độ sâu khoảng > 100 là điều này sẽ sảy ra

do vậy khi init trọng số sẽ cần cẩn trọng hơn

VD: một mạng mà nhiều feature (hoặc giữa 2 layer)

thì mình sẽ init trọng số nhỏ đi để không bị toang

giá trị Z = W\*X

Text, letter

Description automatically generated

việc này có thể check bằng tay việc gradient decent của mình đang hoạt động đúng (giảm) hay không

cái này sẽ dùng lúc gradient checking, gradient checking chỉ dùng để debug không dùng trong lúc train

gradient không dùng vs dropout vì bị mất nút ngẫu nhiên

Programming Exercise

What is L2-regularization actually doing?

* L2-regularization relies on the assumption that a model with small weights is simpler than a model with large weights. Thus, by penalizing the square values of the weights in the cost function you drive all the weights to smaller values. It becomes too costly for the cost to have large weights! This leads to a smoother model in which the output changes more slowly as the input changes.

What you should remember about dropout:

* Dropout is a regularization technique.
* You only use dropout during training. Don't use dropout (randomly eliminate nodes) during test time.
* Apply dropout both during forward and backward propagation.
* During training time, divide each dropout layer by keep\_prob to keep the same expected value for the activations. For example, if keep\_prob is 0.5, then we will on average shut down half the nodes, so the output will be scaled by 0.5 since only the remaining half are contributing to the solution. Dividing by 0.5 is equivalent to multiplying by 2. Hence, the output now has the same expected value. You can check that this works even when keep\_prob is other values than 0.5.