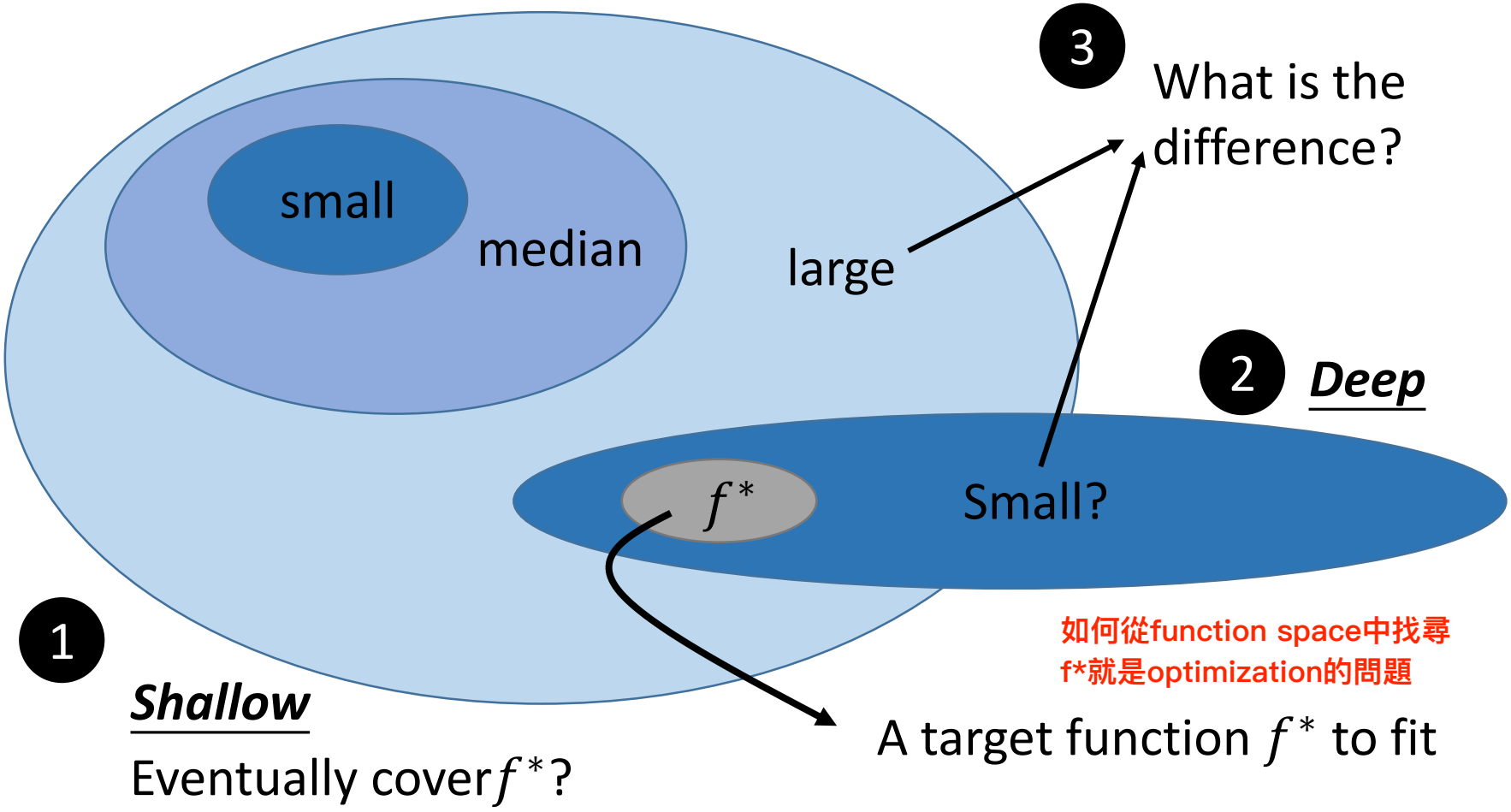
Optimization

李宏毅

Hung-yi Lee

Last time …

Optimization: Is it possible to find f^* in the function space.



Optimization

Optimization ≠ Learning

optimization: fit training data (minimize loss function)
learning: can apply on testing data without overfitting

Network: $f_\theta(x)$

Training data:

$(x^1, \hat{y}^1)$

$(x^2, \hat{y}^2)$

$\vdots$

$(x^R, \hat{y}^R)$

$$L(\theta) = \sum_{r=1}^{R} l(f_\theta(x^r) - \hat{y}^r)$$

$$\theta^* = arg \min_\theta L(\theta)$$

In Deep Learning, $L(\theta)$ is not convex.

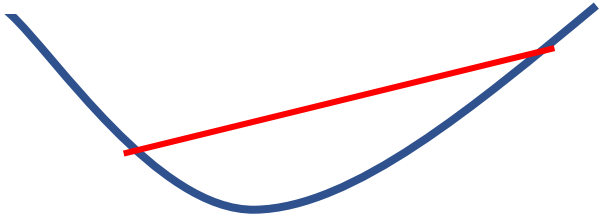
Non-convex optimization is NP-hard.

Why can we solve the problem by gradient descent?

loss function通常都不是凸函數，因此找到不一定是global min

Loss of Deep Learning is not convex

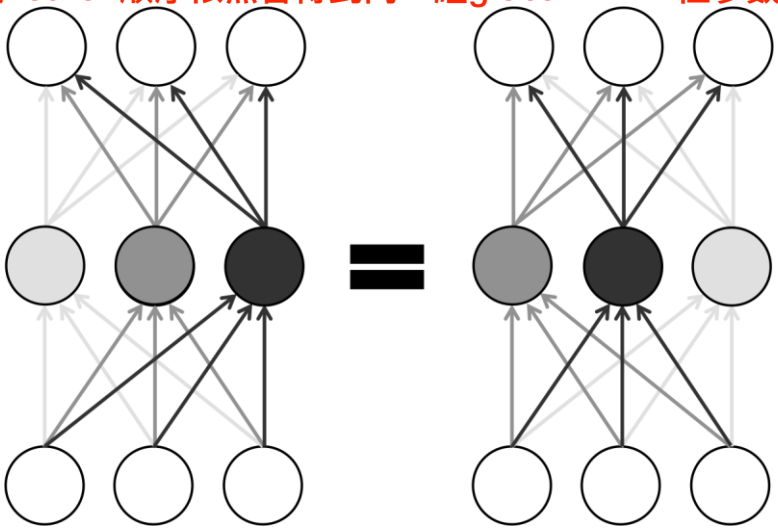
convex定義：任意兩點連線，所有直都大於function



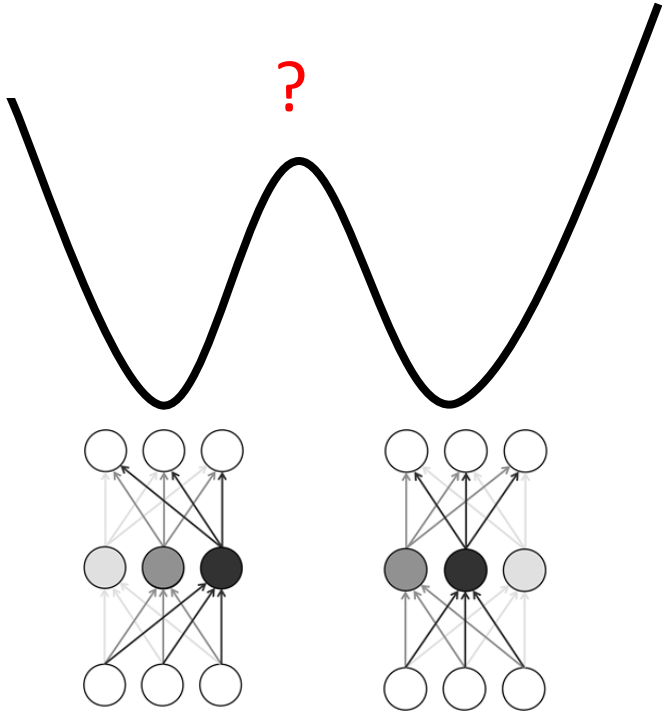
There are at least exponentially many global minima for a neural net.

Permutating the neurons in one layer does not change the loss.

即使調換neuron順序依然會得到同一組global min，但參數是不同的

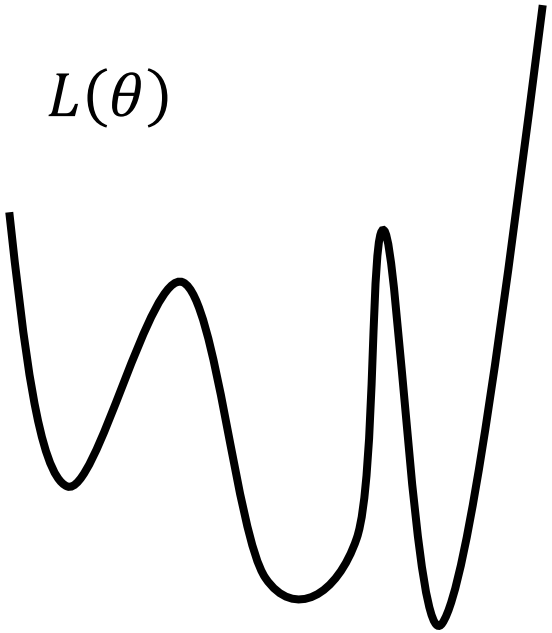


假設找到一組global min的參數

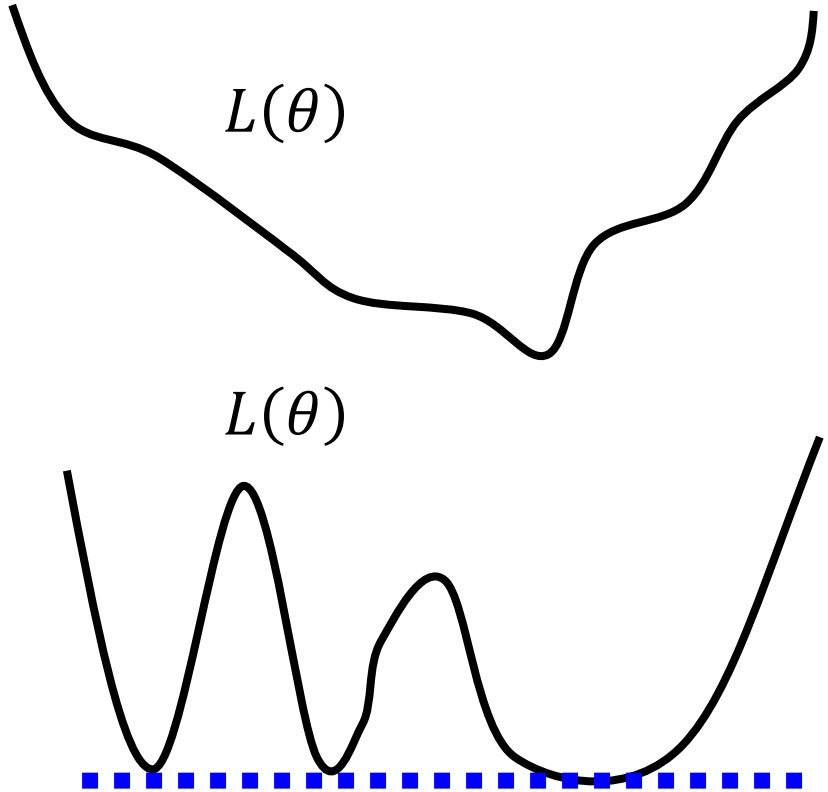


Non-convex ≠ Difficult

有沒有可能deep learning所定義出來的non convex function是比較簡單的



Not guarantee to find optimal solution by gradient descent



但是由於上一頁投影片，我們知道deep learning應該會有很多的global min，因此雖然參數不同但我只要找到其中一個即可

Outline

- Review: Hessian
- Deep Linear Model
- Deep Non-linear Model
- Conjecture about Deep Learning
- Empirical Observation about Error Surface

Hessian Matrix: When Gradient is Zero

Some examples in this part are from:
https://www.math.upenn.edu/~kazdan/312F12/Notes/max-min-notesJan09/max-min.pdf

http://www.deeplearningbook.org/contents/optimization.html

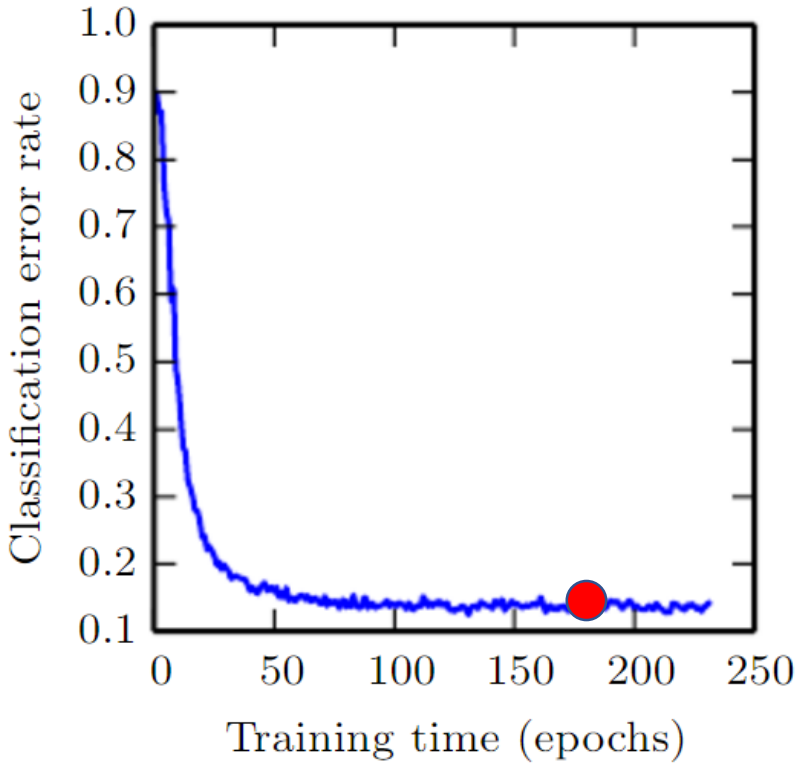
Training stops ….

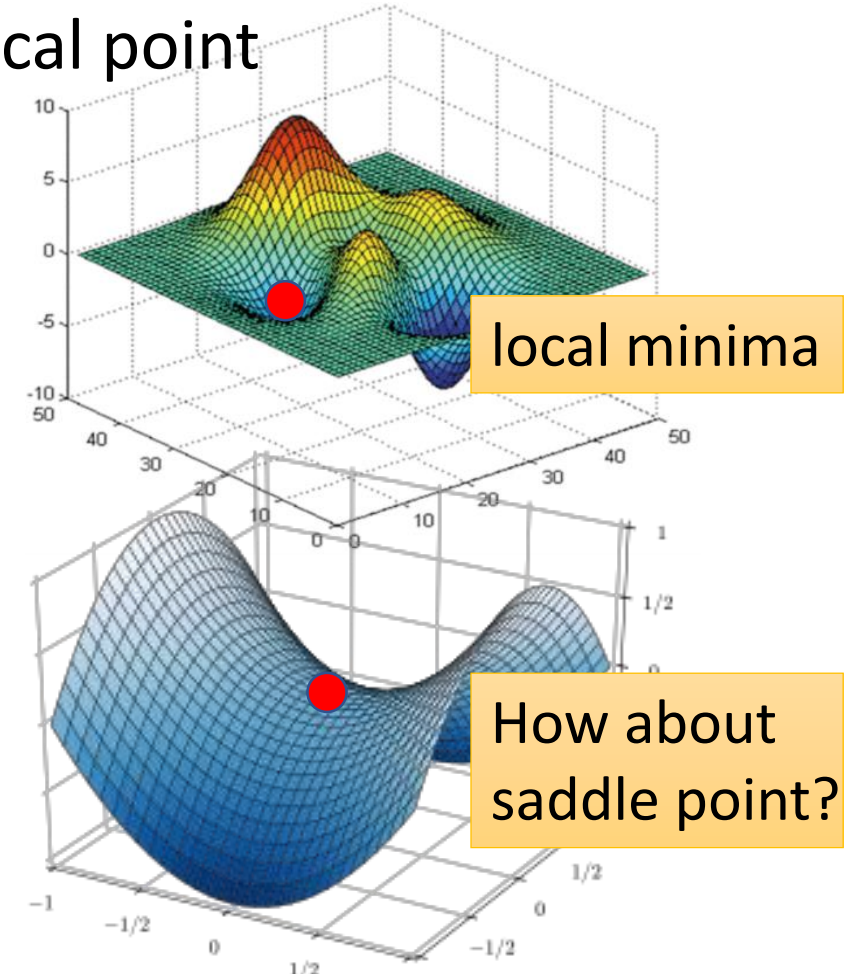
critical point: gradient is zero

training卡住有可能是因為到了global/local min，也可能走到saddle point

為了辨別是哪一個，可以使用hessian matrix

- People believe training stuck because the parameters are near a critical point





local minima

How about saddle point?

When Gradient is Zero

任何算式皆滿足

$$f(\theta) = f(\theta^0) + (\theta - \theta^0)^T g + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0) + \cdots$$

Gradient g is a ***vector***

$$g_i = \frac{\partial f(\theta^0)}{\partial \theta_i}$$

gradient $\nabla f(\theta^0)$

Hessian H is a ***matrix***

nxn dim

$$H_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} f(\theta^0)$$

symmetric

$$= \frac{\partial^2}{\partial \theta_j \partial \theta_i} f(\theta^0) = H_{ji}$$

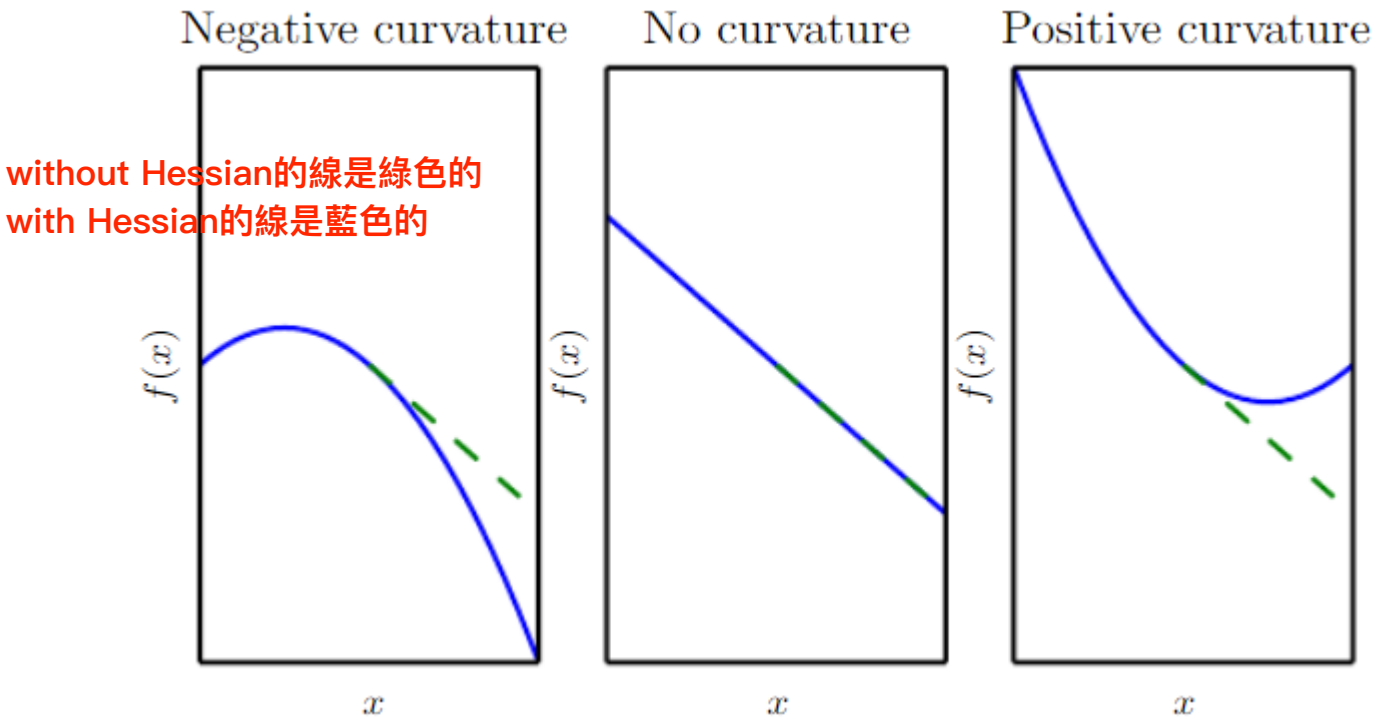
誰先微都一樣，因此是對稱矩陣

Hessian

Source of image:
http://www.deeplearningbook.org/contents/numerical.html

$$f(\theta) = f(\theta^0) + (\theta - \theta^0)^T g + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0) + \cdots$$

H determines the curvature

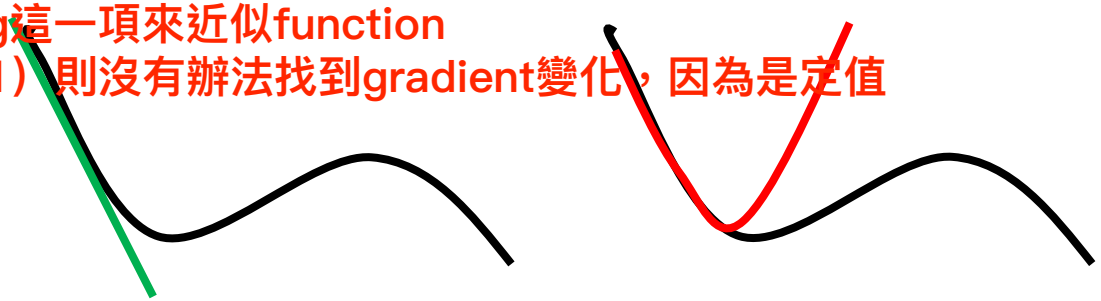


黑色的線是f(θ)，綠色的線是不考慮Hessian項，紅色是考慮Hessian項

gradient descent就是只利用g這一項來近似function

但如果只考慮前兩項（不包含H），則沒有辦法找到gradient變化，因為是定值

Hessian



$$f(\theta) = f(\theta^0) + (\theta - \theta^0)^T g + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0) + \cdots$$

快速照出min的點

Newton's method ➡ Find the space such that $\nabla f(\theta) = 0$

newton's method希望能一步就找到critical point

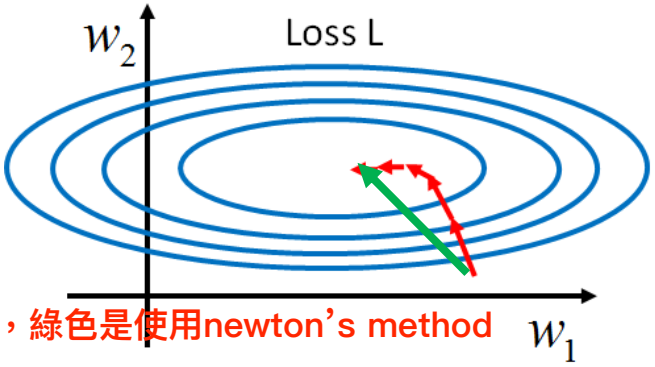
$$\nabla f(\theta) \approx \underbrace{\nabla[(\theta - \theta^0)^T g]}_{= g} + \underbrace{\nabla\left[\frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0)\right]}_{H(\theta - \theta^0)}$$

$$\frac{\partial[(\theta - \theta^0)^T g]}{\partial \theta_i} = g_i$$

拿θi對每一微做偏微分

$$\frac{\partial\left[\frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0)\right]}{\partial \theta_i}$$

Hessian



紅色是只考慮gradient的結果，綠色是使用newton's method

$$f(\theta) = f(\theta^0) + (\theta - \theta^0)^T g + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0) + \cdots$$

Newton's method

$$\nabla f(\theta) \approx \underbrace{\nabla[(\theta - \theta^0)^T g]}_{= g} + \underbrace{\nabla\left[\frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0)\right]}_{H(\theta - \theta^0)}$$

vector 未知

$$\nabla f(\theta) \approx g + H(\theta - \theta^0) = 0$$

matrix 已知

$$H(\theta - \theta^0) = -g$$

$$\theta - \theta^0 = -H^{-1} g$$

$$\theta = \theta^0 - \boxed{H^{-1}} g \quad \text{v.s.} \quad \theta = \theta^0 - \eta g$$

H的inverse可以改變方向跟大小，比learning rate厲害

假設Hessian invertible，但是並不是所有hessian都可以用這招

Change the direction, determine step size

newton's method不好用，主因有2個：第一個是deep learning的參數很多（幾百萬個），要算幾百萬x幾百萬的inverse matrix是很耗費計算的，地兒個是H想找的是gradient為0的地方，有可能帶你到maximum或是saddle point

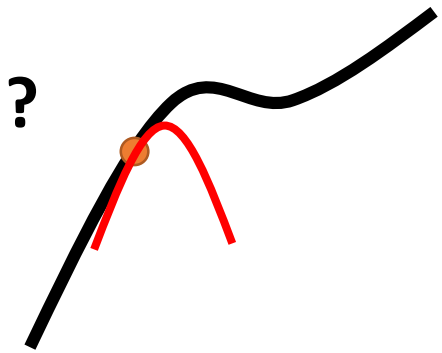
Hessian

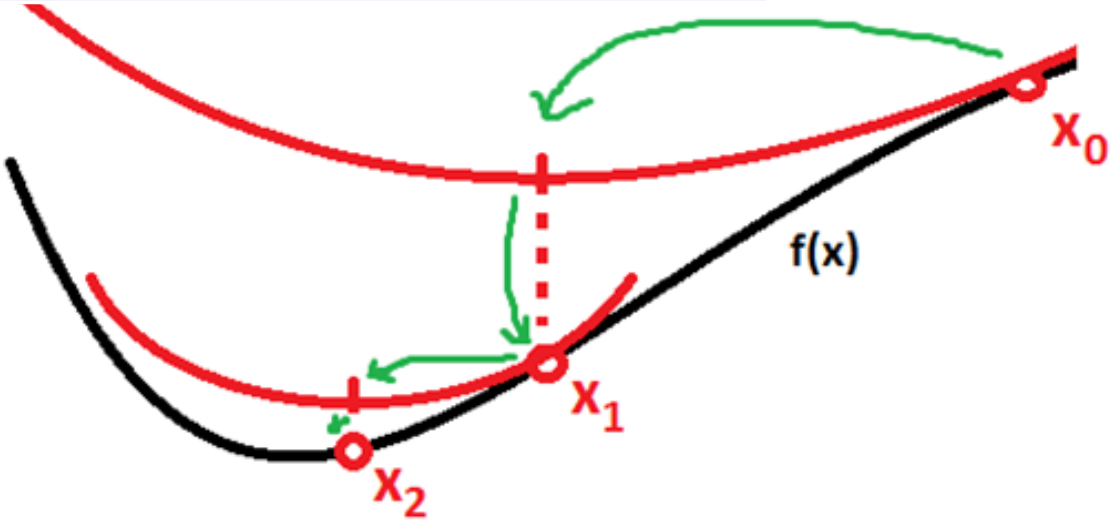
Source of image: https://math.stackexchange.com/questions/609680/newtons-method-intuition

$$f(\theta) = f(\theta^0) + (\theta - \theta^0)^T g + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0) + \cdots$$

Newton's method Not suitable for Deep Learning

Consider that $\theta = x$





If $f(x)$ is a quadratic function, obtain critical point in one step.

What is the problem?

Source of image:
http://www.offconvex.org/2016/03/22/saddlepoints/

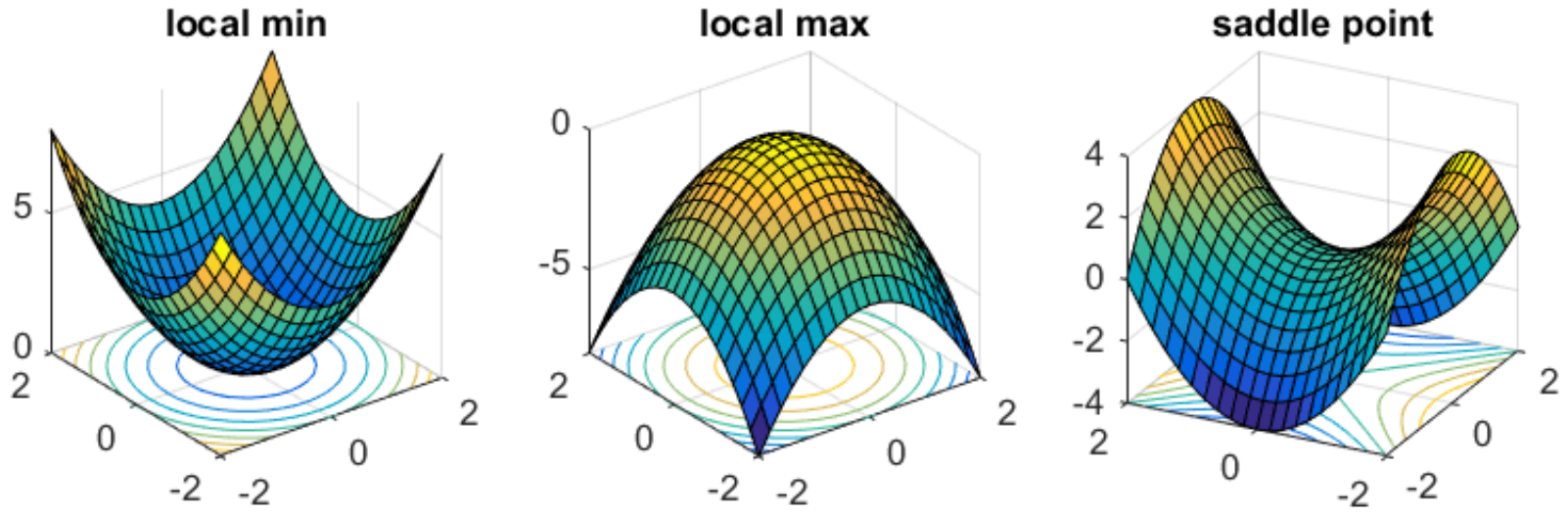
Hessian

$$f(\theta) = f(\theta^0) + (\theta - \theta^0)^T g + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0) + \cdots$$

At critical point ($g = 0$)

H tells us the properties of critical points.

究竟是min/max/saddle point?



http://speech.ee.ntu.edu.tw/~tlkagk/courses/LA_2016/Lecture/eigen.pdf

Review: Linear Algebra

- If $Av = \lambda v$ (v is a vector, λ is a scalar)
  - v is an eigenvector of A excluding zero vector
  - λ is an eigenvalue of A that corresponds to v

A must be square

Eigen value

$$\begin{bmatrix} 5 & 2 & 1 \\ -2 & 1 & -1 \\ 2 & 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ 4 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Eigen vector

Review: Positive/Negative Definite

- An nxn matrix A is symmetric.
- For every non-zero vector x ($x \neq 0$)

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

positive definite: $x^T Ax > 0$ ⟺ All eigen values are positive.

positive semi-definite: $x^T Ax \geq 0$ ⟺ All eigen values are non-negative.

negative definite: $x^T Ax < 0$ ⟺ All eigen values are negative.

negative semi-definite: $x^T Ax \leq 0$ ⟺ All eigen values are non-positive.

Hessian

At critical point:

$$f(\theta) \approx f(\theta^0) + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0)$$

$x^T Hx$

H is positive definite ➡ $x^T Hx > 0$

All eigen values are positive.

$x^T Hx \geq 0$?

當H是半正定/半負定矩陣，則無法確定，取決於之後的項是正還是負

➡ Around θ^0: $f(\theta) > f(\theta^0)$ ➡ Local minima

H is negative definite ➡ $x^T Hx < 0$

$x^T Hx \leq 0$?

All eigen values are negative

➡ Around θ^0: $f(\theta) < f(\theta^0)$ ➡ Local maxima

Sometimes $x^T Hx > 0$, sometimes $x^T Hx < 0$

$f(\theta^0)$ ➡ Saddle point

Hessian

At critical point:

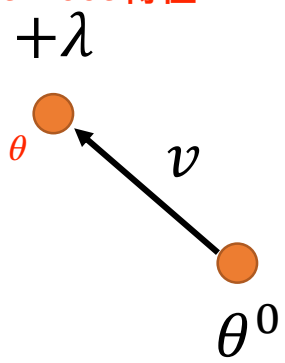
$$f(\theta) \approx f(\theta^0) + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0)$$

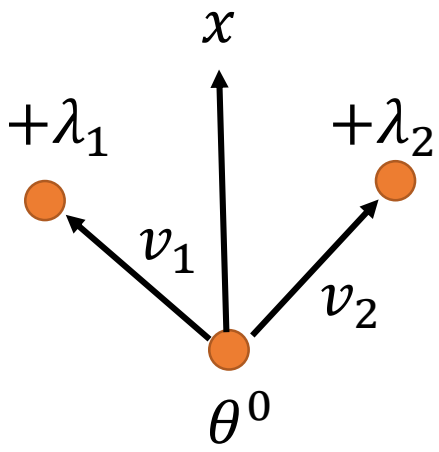
v is an eigen vector ➡ $v^T H v = v^T(\lambda v) = \lambda \|v\|^2$

Unit vector

$= \lambda$

根據某一個eigen vector的方向移動，則function會增加1/2*laambda得值





$x = a_1 v_1 + a_2 v_2$

假設v1,v2皆為eigen vector

$x^T H x$

$= (a_1 v_1 + a_2 v_2)^T H (a_1 v_1 + a_2 v_2)$

v_1 and v_2 are orthogonal

$= (a_1)^2 \lambda_1 + (a_x)^2 \lambda_2$?

Because H is an nxn symmetric matrix,

H can have eigen vectors $\{v_1, v_2, \ldots, v_n\}$ form a orthonormal basis.

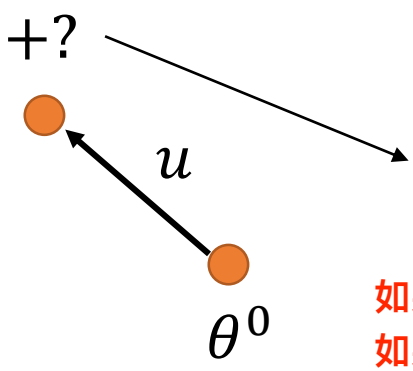
Hessian

At critical point:

$$f(\theta) \approx f(\theta^0) + \frac{1}{2}(\theta - \theta^0)^T H(\theta - \theta^0)$$

v is an eigen vector ➡ $v^T H v = v^T(\lambda v) = \lambda \|v\|^2$

Unit vector $= \lambda$



$$u = a_1 v_1 + a_2 v_2 + \cdots + a_n v_n$$

$$(a_1)^2 \lambda_1 + (a_2)^2 \lambda_2 + \cdots + (a_n)^2 \lambda_n$$

如果是正定矩陣，則往任何方向走function value一定會增加，因此是local min
如果是負定矩陣，則往任何方向走function value一定會減少，因此是local max
其餘則是saddle point

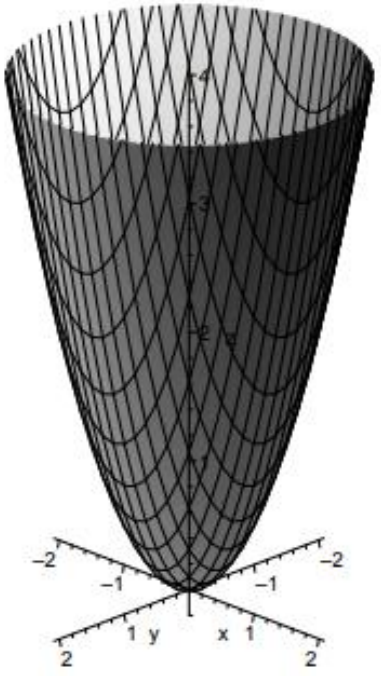
Because H is an nxn symmetric matrix,

H can have eigen vectors $\{v_1, v_2, \ldots, v_n\}$ form a orthonormal basis.

can span R^n

Examples

$$f(x,y) = x^2 + 3y^2$$



$$\frac{\partial f(x,y)}{\partial x} = 2x \qquad \frac{\partial f(x,y)}{\partial y} = 6y$$

$$x = 0, y = 0$$

走到critical point後，算hessian判斷min/max/saddle

$$\frac{\partial^2}{\partial x \partial x} f(x,y) = 2 \qquad \frac{\partial^2}{\partial x \partial y} f(x,y) = 0$$

$$\frac{\partial^2}{\partial y \partial x} f(x,y) = 0 \qquad \frac{\partial^2}{\partial y \partial y} f(x,y) = 6$$

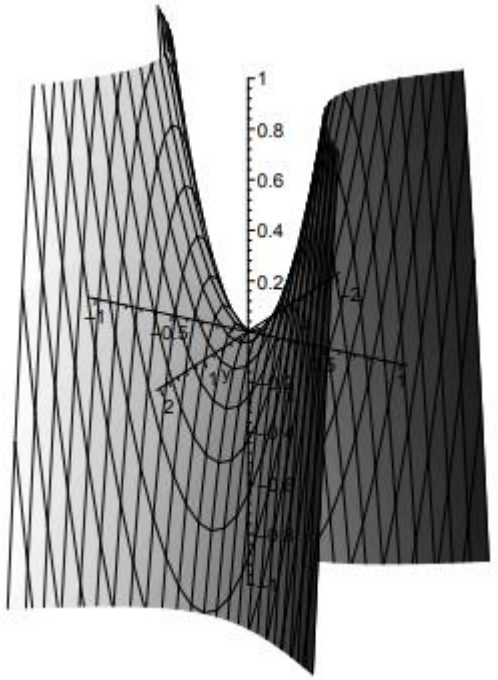
$$H = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}$$

Positive-definite

Local minima

Examples

$$f(x,y) = -x^2 + 3y^2$$



$$\frac{\partial f(x,y)}{\partial x} = -2x \quad \frac{\partial f(x,y)}{\partial y} = 6y$$

$$x = 0, y = 0$$

$$\frac{\partial^2}{\partial x \partial x} f(x,y) = -2 \qquad \frac{\partial^2}{\partial x \partial y} f(x,y) = 0$$

$$\frac{\partial^2}{\partial y \partial x} f(x,y) = 0 \qquad \frac{\partial^2}{\partial y \partial y} f(x,y) = 6$$

$$H = \begin{bmatrix} -2 & 0 \\ 0 & 6 \end{bmatrix}$$

Saddle

Degenerate

至少有一個eigen value是0

- Degenerate Hessian has at least one zero eigen value

$$f(x,y) = x^2 + y^4$$

$$\frac{\partial f(x,y)}{\partial x} = 2x \qquad \frac{\partial f(x,y)}{\partial y} = 4y^3 \qquad x = y = 0$$

$$\frac{\partial^2}{\partial x \partial x} f(x,y) = 2 \qquad \frac{\partial^2}{\partial x \partial y} f(x,y) = 0 \qquad H = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\frac{\partial^2}{\partial y \partial x} f(x,y) = 0 \qquad \frac{\partial^2}{\partial y \partial y} f(x,y) = 12y^2$$

Degenerate

- Degenerate Hessian has at least one zero eigen value

$$f(x,y) = x^2 + y^4 \qquad g(x,y) = x^2 - y^4$$

$x = y = 0$ $\qquad$ $x = y = 0$ saddle point

$$g = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad g = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

Hessian影響是0，因此無法判斷，因為要考慮後面向的正負才能知道function的值是增加還是減少

No Difference

Degenerate

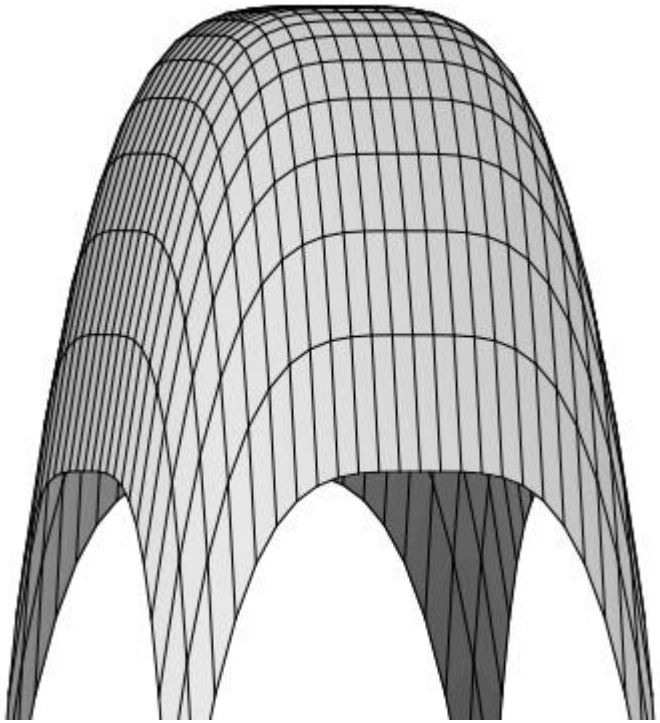
$$h(x,y) = 0$$

$$x = y = 0$$

$$g = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$f(x,y) = -x^4 - y^4$$



$$\frac{\partial f(x,y)}{\partial x} = -4x^3 \quad \frac{\partial f(x,y)}{\partial y} = -4y^3$$

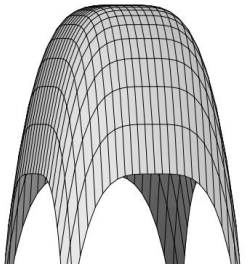
$$\frac{\partial^2}{\partial x \partial x} f(x,y) = -12x^2 \quad \frac{\partial^2}{\partial x \partial y} f(x,y) = 0$$

$$\frac{\partial^2}{\partial y \partial x} f(x,y) = 0 \quad \frac{\partial^2}{\partial y \partial y} f(x,y) = -12y^2$$

$$H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

http://homepages.math.uic.edu/~julius/monkeysaddle.html

c.f.



Monkey Saddle

$$\frac{\partial f(x,y)}{\partial x} = 3x^2 - 3y^2$$

$$\frac{\partial f(x,y)}{\partial y} = -6xy$$

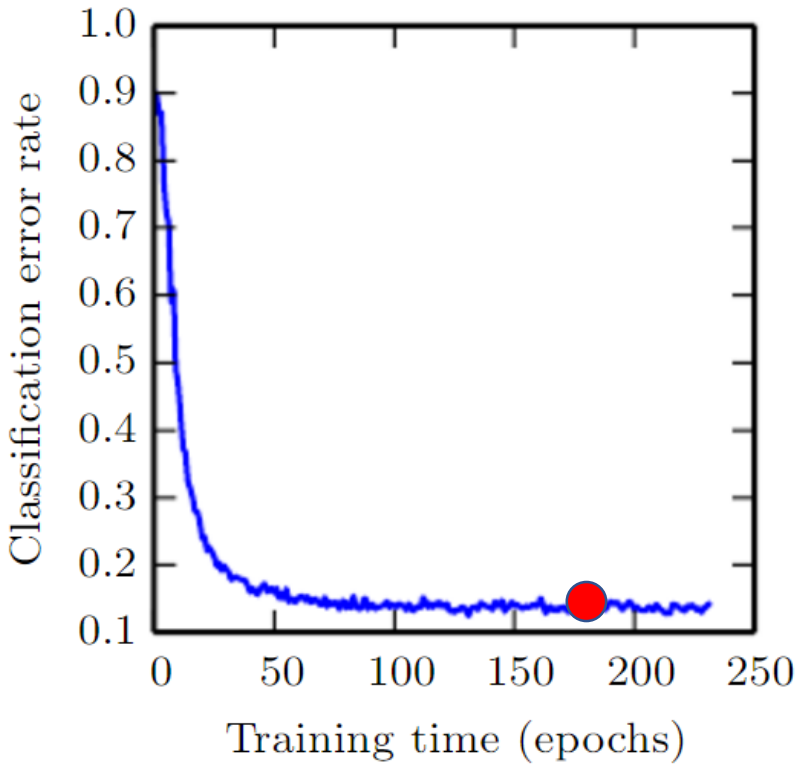
$$\frac{\partial^2}{\partial x \partial x} f(x,y) = 6x \qquad \frac{\partial^2}{\partial x \partial y} f(x,y) = -6y$$

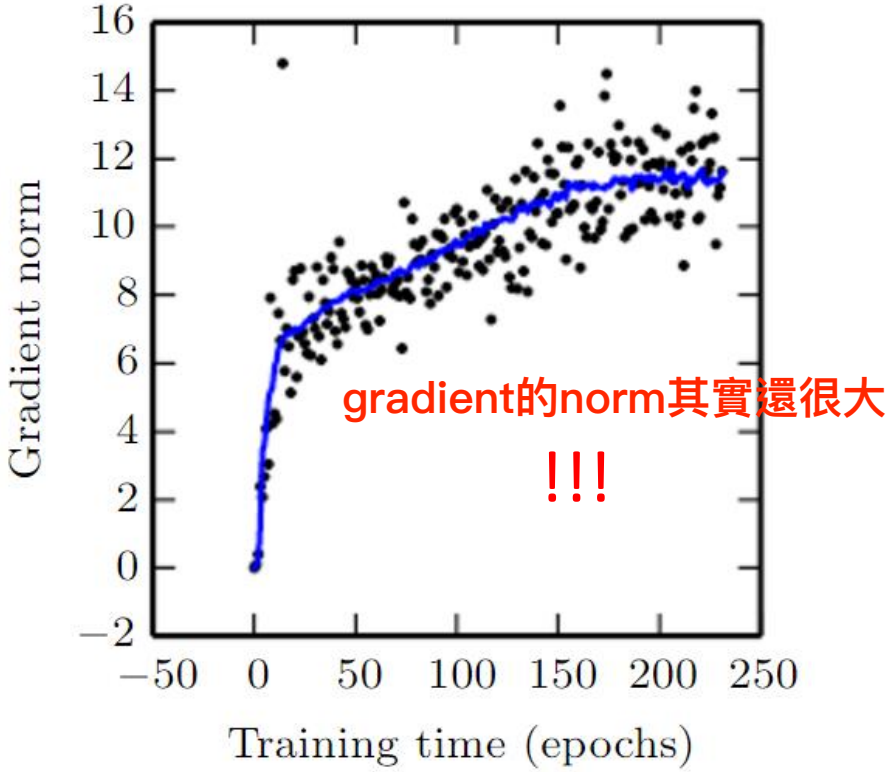
$$\frac{\partial^2}{\partial y \partial x} f(x,y) = -6y \qquad \frac{\partial^2}{\partial y \partial y} f(x,y) = -6x$$

http://www.deeplearningbook.org/contents/optimization.html

Training stuck ≠ Zero Gradient

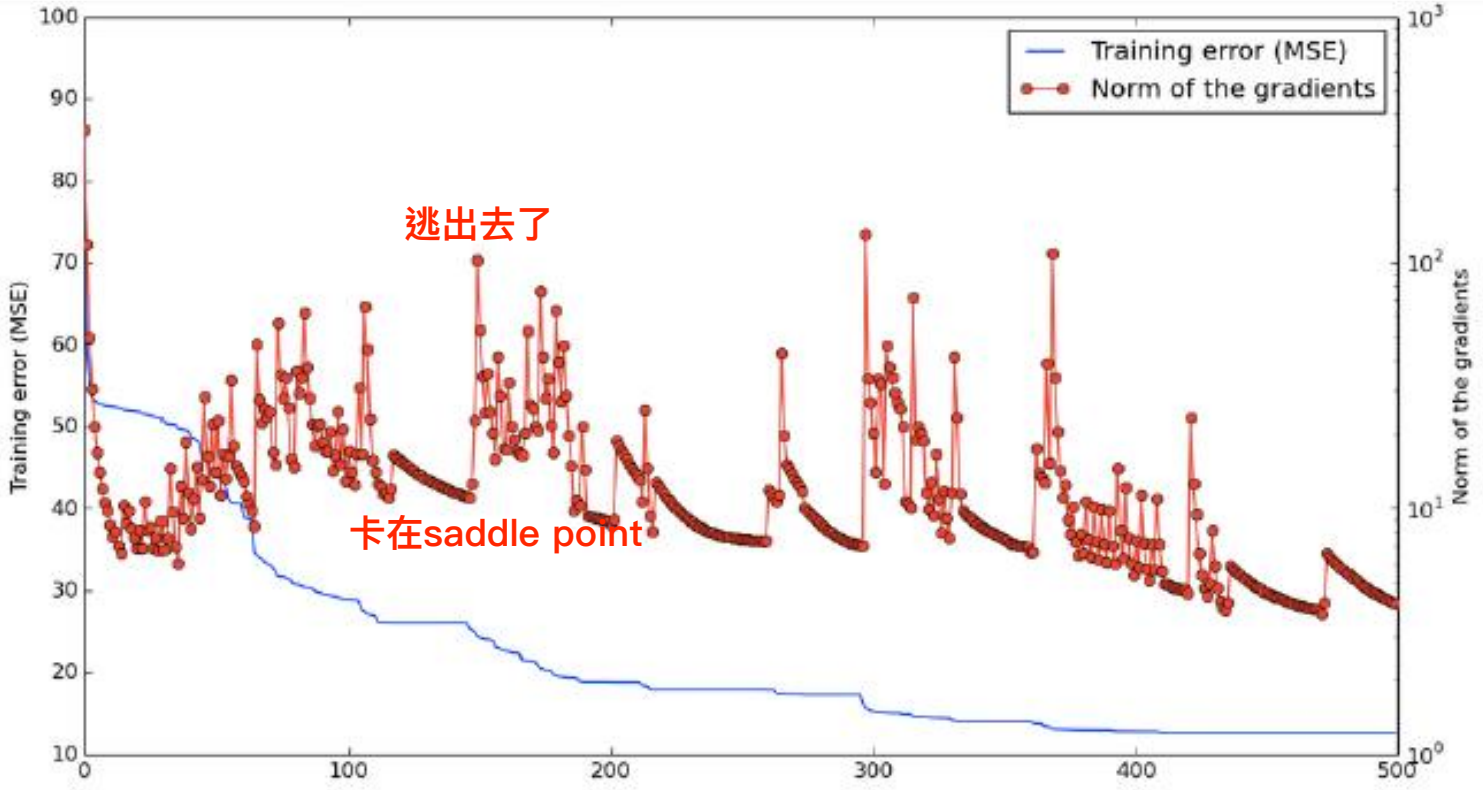
- People believe training stuck because the parameters are around a critical point





Goodfellow：Deeplearning教科書提到，training卡住不見得是因為zero gradient

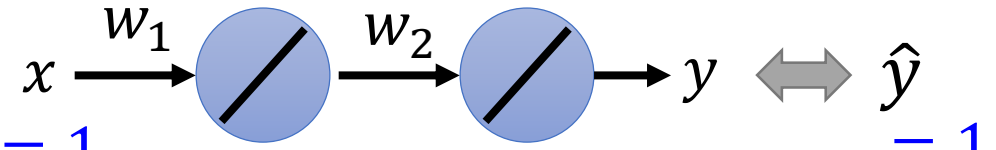
Training stuck ≠ Zero Gradient



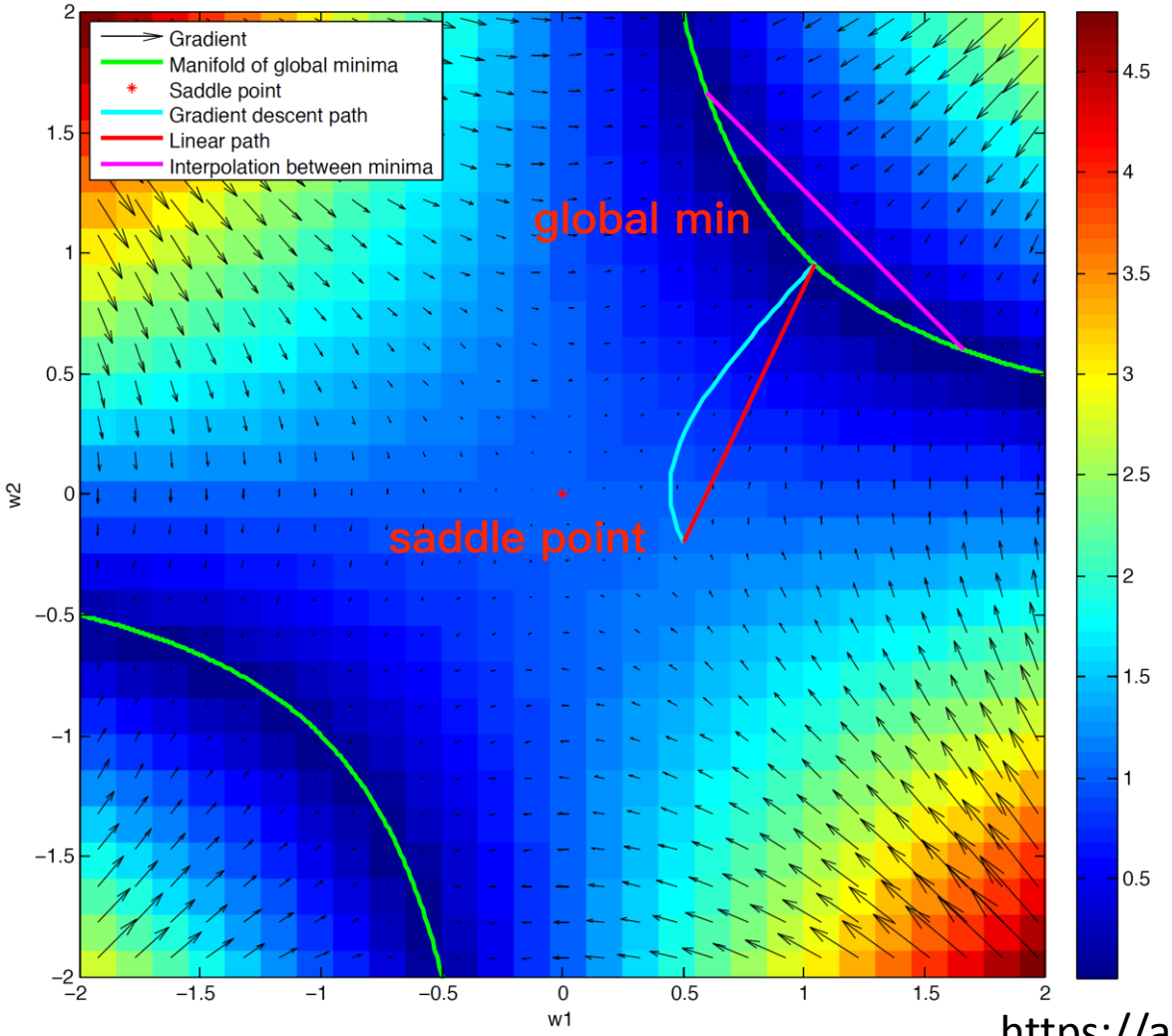
Approach a saddle point, and then escape

http://videolectures.net/deeplearning2015_bengio_theoretical_motivations/

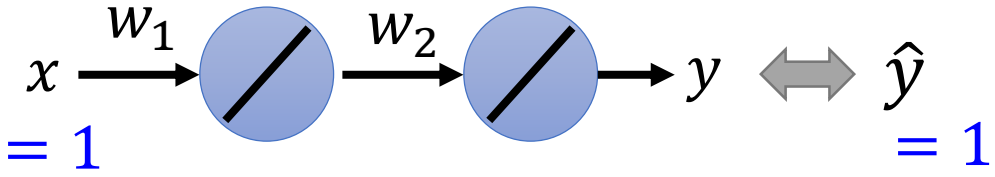
Deep Linear Network

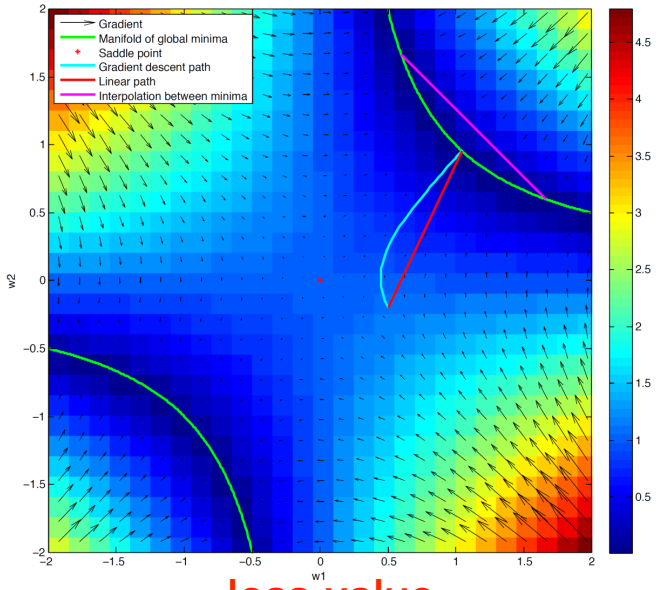


loss value



https://arxiv.org/abs/1412.6544





loss value

$$L = (\hat{y} - w_1 w_2 x)^2 = (1 - w_1 w_2)^2$$

$$\frac{\partial L}{\partial w_1} = 2(1 - w_1 w_2)(-w_2)$$

$$\frac{\partial L}{\partial w_2} = 2(1 - w_1 w_2)(-w_1)$$

The probability of stuck as saddle point is almost zero.

Easy to escape

如果要逃出saddle point很簡單，只要計算一下hessian就可以知道往哪裡走會使loss下降

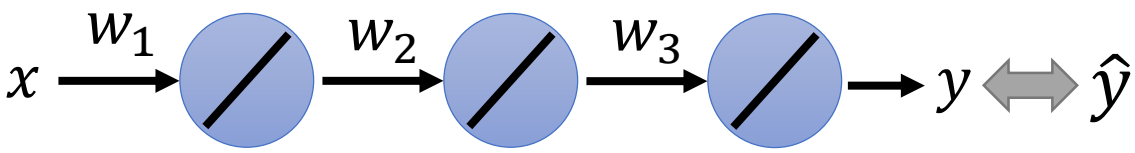
$$\frac{\partial^2 L}{\partial {w_1}^2} = 2(-w_2)(-w_2) \qquad \frac{\partial^2 L}{\partial w_1 \partial w_2} = -2 + 4 w_1 w_2$$

$$\frac{\partial^2 L}{\partial w_2 \partial w_1} = -2 + 4 w_1 w_2 \qquad \frac{\partial^2 L}{\partial {w_2}^2} = 2(-w_1)(-w_1)$$

Hessian Matrix

2-hidden layers

$$L = (1 - w_1 w_2 w_3)^2$$



$$\frac{\partial L}{\partial w_1} = 2(1 - w_1 w_2 w_3)(-w_2 w_3)$$

$$\frac{\partial L}{\partial w_2} = 2(1 - w_1 w_2 w_3)(-w_1 w_3)$$

$$\frac{\partial L}{\partial w_3} = 2(1 - w_1 w_2 w_3)(-w_1 w_2)$$

$$\frac{\partial^2 L}{\partial {w_1}^2} = 2(-w_2 w_3)^2$$

$$\frac{\partial^2 L}{\partial w_2 \partial w_1} = -2w_3 + 4w_1 w_2 (w_3)^2$$

All minima are global, some critical points are “bad”.

critical point

$w_1 w_2 w_3 = 1$ global minima

critical point

$w_1 = w_2 = w_3 = 0$

error surface很平坦，完全逃不出

saddle point

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

So flat

critical point

$w_1 = w_2 = 0, w_3 = k$

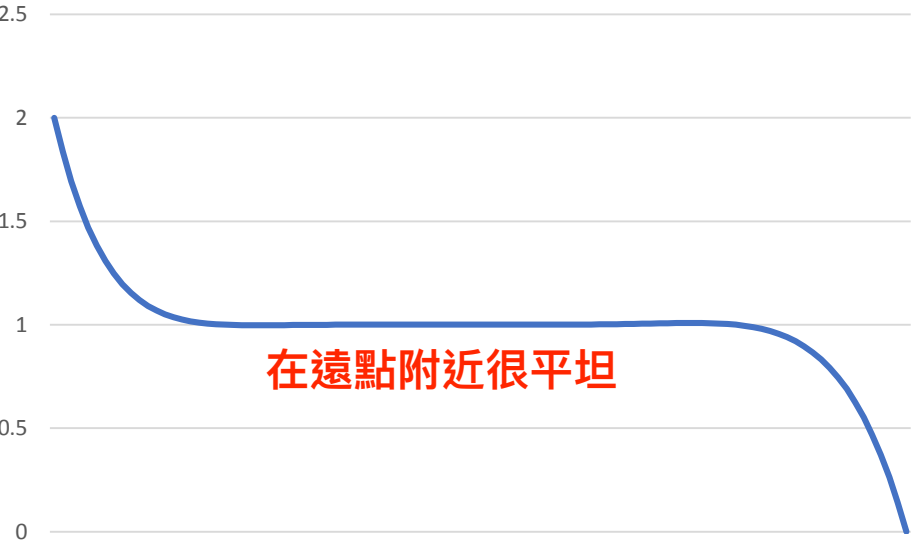
eigenvalue有正有負，因此可以算hessian得知往哪個方向走可以下降loss，便可以逃出

$$H = \begin{bmatrix} 0 & -2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Saddle point

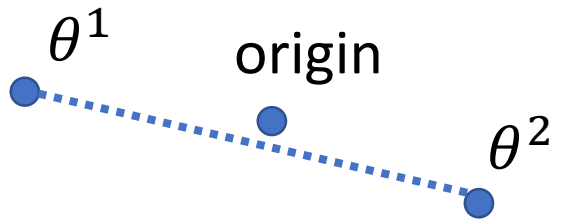
10 hidden layers

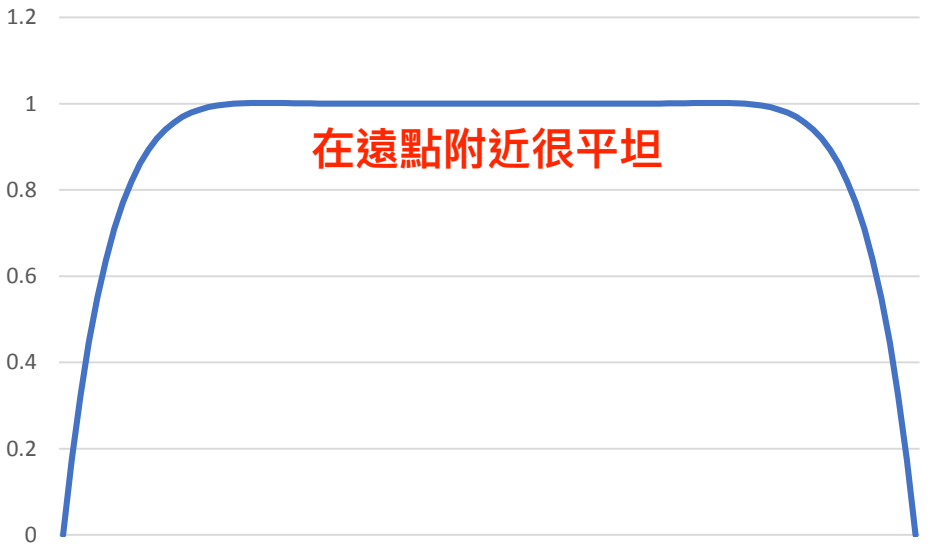
11個參數space上面取正負兩個點，origin就是在中間的點



在遠點附近很平坦

取任意兩個正負值





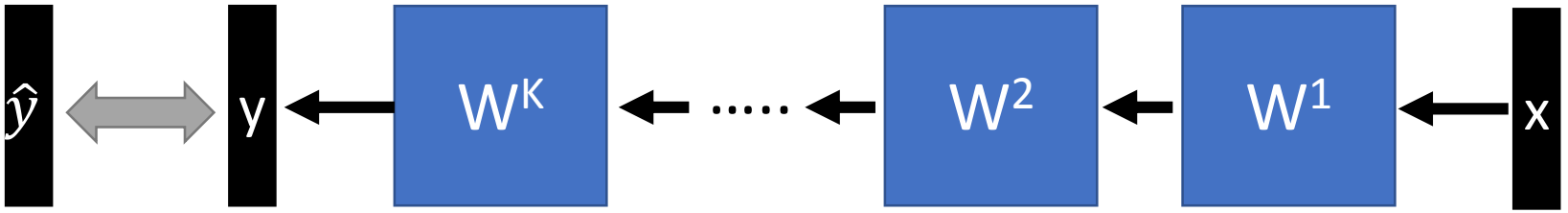
在遠點附近很平坦

取任意兩個正負值

10-hidden layers

Demo

Deep Linear Network



$$y = W^K W^{K-1} \cdots W^2 W^1 x \qquad L = \sum_{n=1}^{N} (x^n - \hat{y}^n)^2$$

寬鬆條件： Hidden layer size ≥ Input dim, output dim

只要滿足一些寬鬆的條件，就可以證明找到的local minimun一定是global minimum

More than two hidden layers can produce saddle point without negative eigenvalues.

當hidden layer大於兩層，則容易產生不好的saddle point (所有eigen value = 0)

Reference

證明linear network 沒有local minimum

- Kenji Kawaguchi, Deep Learning without Poor Local Minima, NIPS, 2016
- Haihao Lu, Kenji Kawaguchi, Depth Creates No Bad Local Minima, arXiv, 2017
- Thomas Laurent, James von Brecht, Deep linear neural networks with arbitrary loss: All local minima are global, arXiv, 2017
- Maher Nouiehed, Meisam Razaviyayn, Learning Deep Models: Critical Points and Local Openness, arXiv, 2018

證明linear的network沒有local minimum，ICLR reject

Non-linear Deep Network

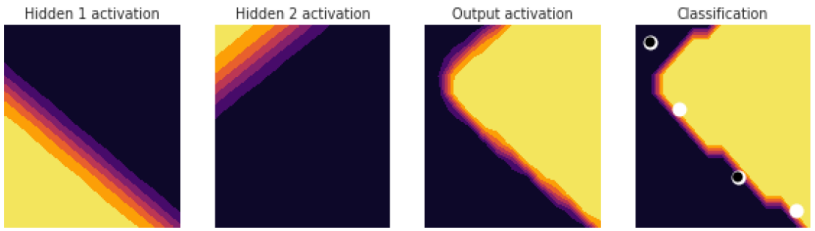
Dose it have local minima?

證明事情不存在很難，證明事情存在相對容易

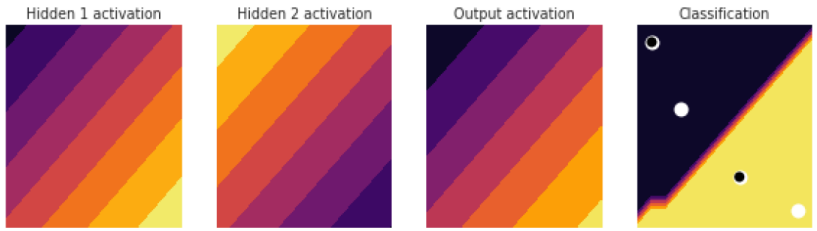
Even Simple Task can be Difficult

neuron數h in one hidden layer

h		XOR ReLU	XOR Sigmoid	Jellyfish ReLU	Jellyfish Sigmoid		XOR ReLU	XOR Sigmoid	Jellyfish ReLU	Jellyfish Sigmoid
2	Adam	28%	79%	7%	0%	GD	23%	90%	16%	62%
3	Adam	52%	98%	34%	0%	GD	47%	100%	33%	100%
4	Adam	68%	100%	50%	2%	GD	70%	100%	66%	100%
5	Adam	81%	100%	51%	27%	GD	80%	100%	68%	100%
6	Adam	91%	100%	61%	17%	GD	89%	100%	69%	100%
7	Adam	97%	100%	69%	58%	GD	89%	100%	86%	100%

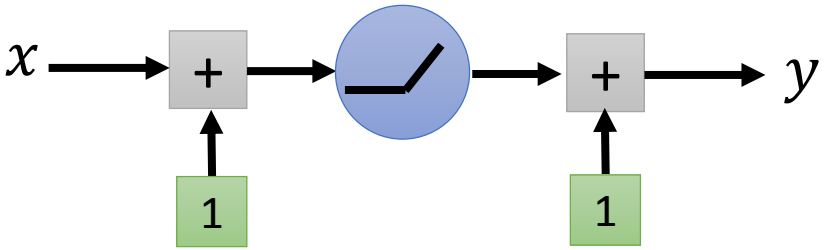


(a) Optimally converged net for Jellyfish.

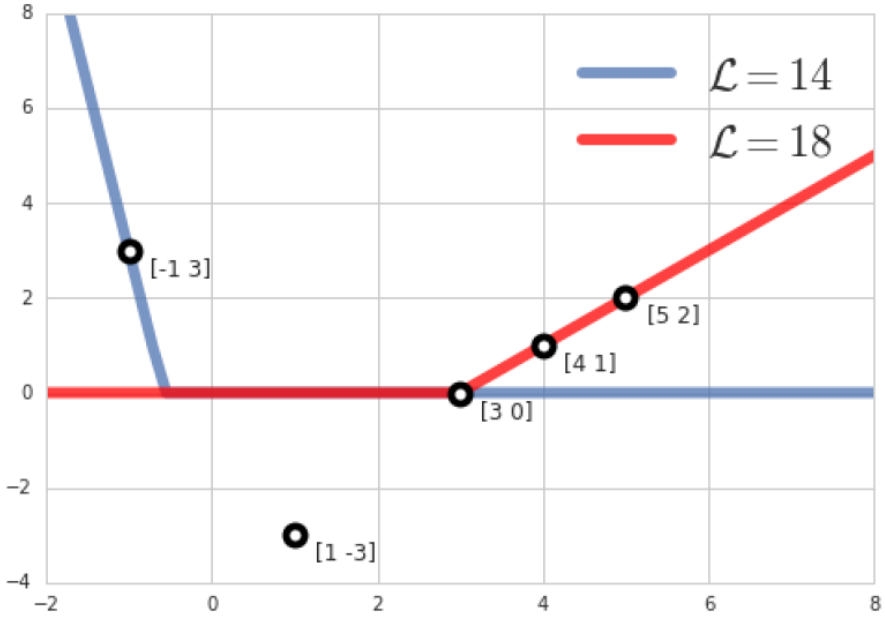


(b) Stuck net for Jellyfish.

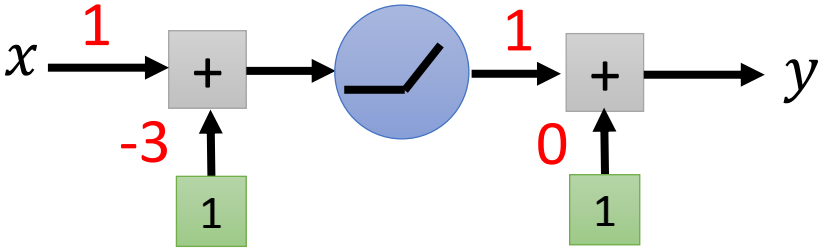
ReLU has local



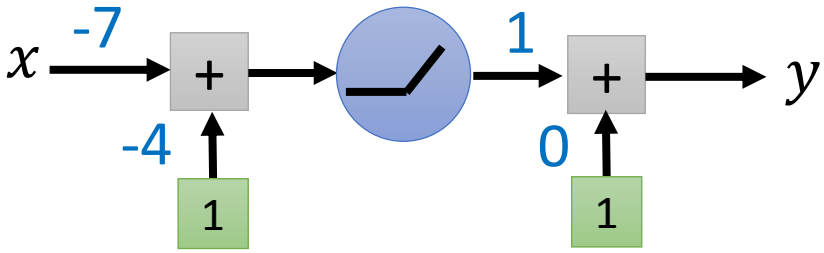
local minimum，因為對其參數調整一點點delta都會使loss變大



(-1,3) (1,-3) (3,0) (4,1) (5,2)



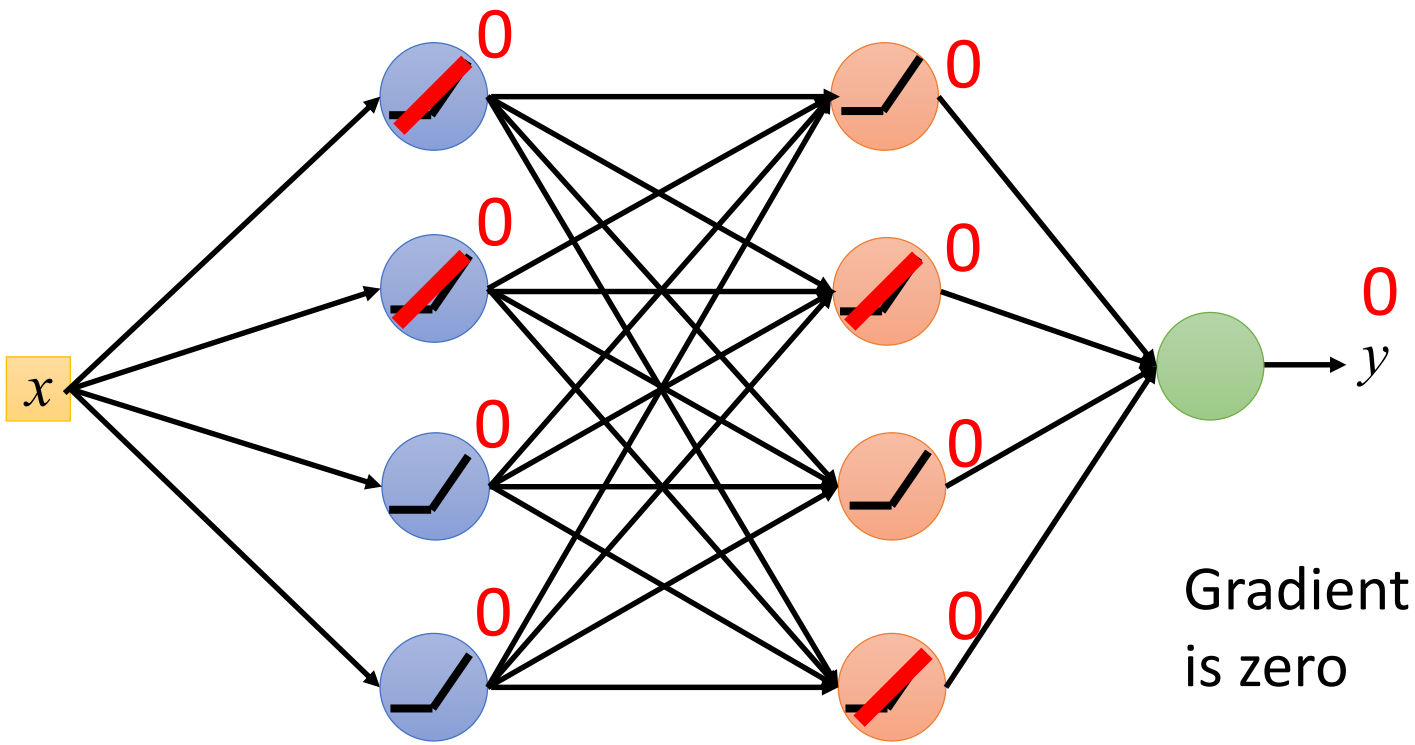
但隨便找一組又可以找到更低的loss value



This relu network has local minima.

“Blind Spot” of ReLU

假設每個relu都走到0 output，則y=0，gradient也為0，因此為critical point



It is pretty easy to make this happens ……

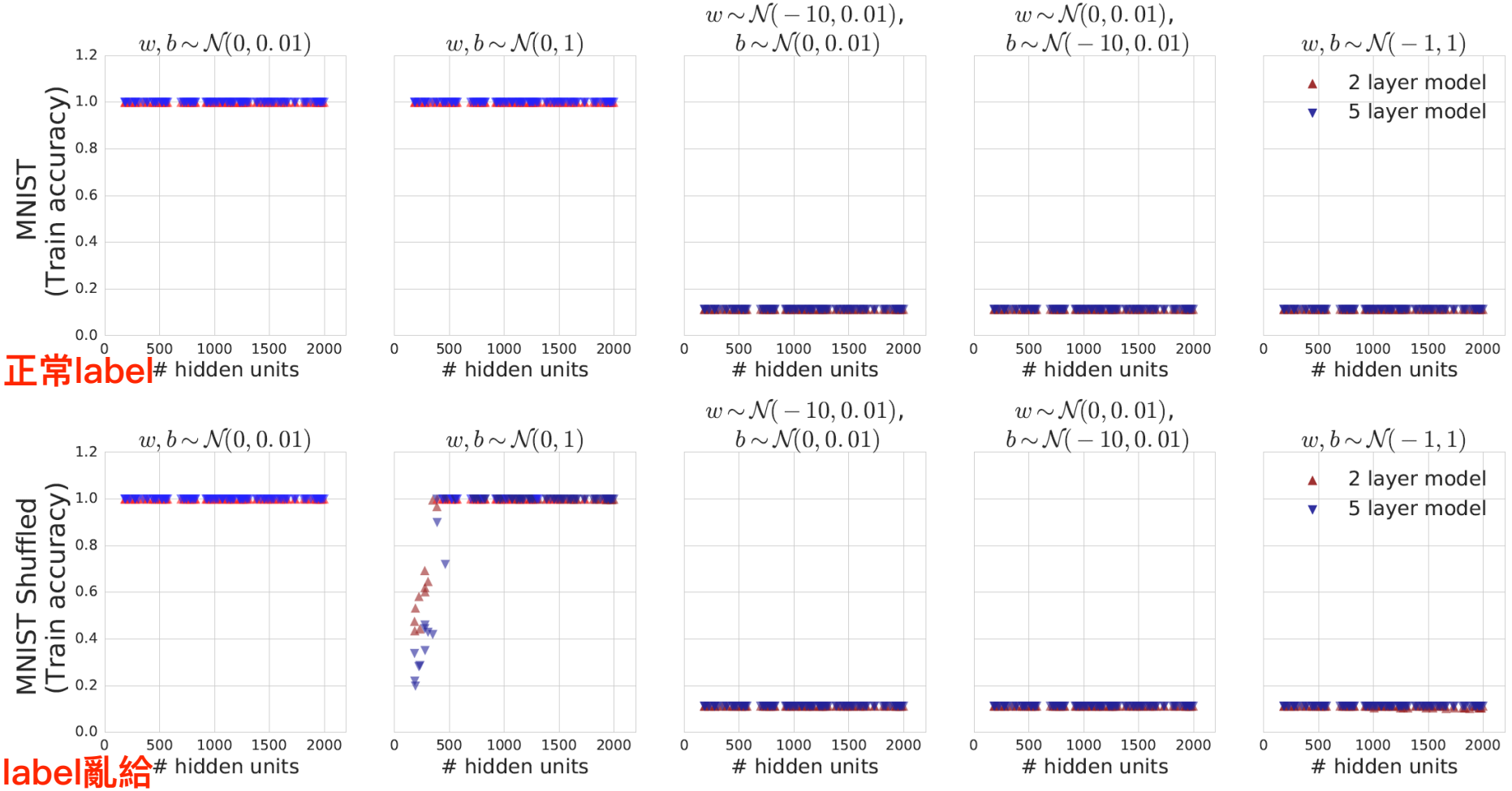
“Blind Spot” of ReLU

每個點表示一個network

Consider your initialization

在初始化的時候故意給怪怪的值都會導致train不起來

- MNIST, Adam, 1M updates



假設training data是由generator產生

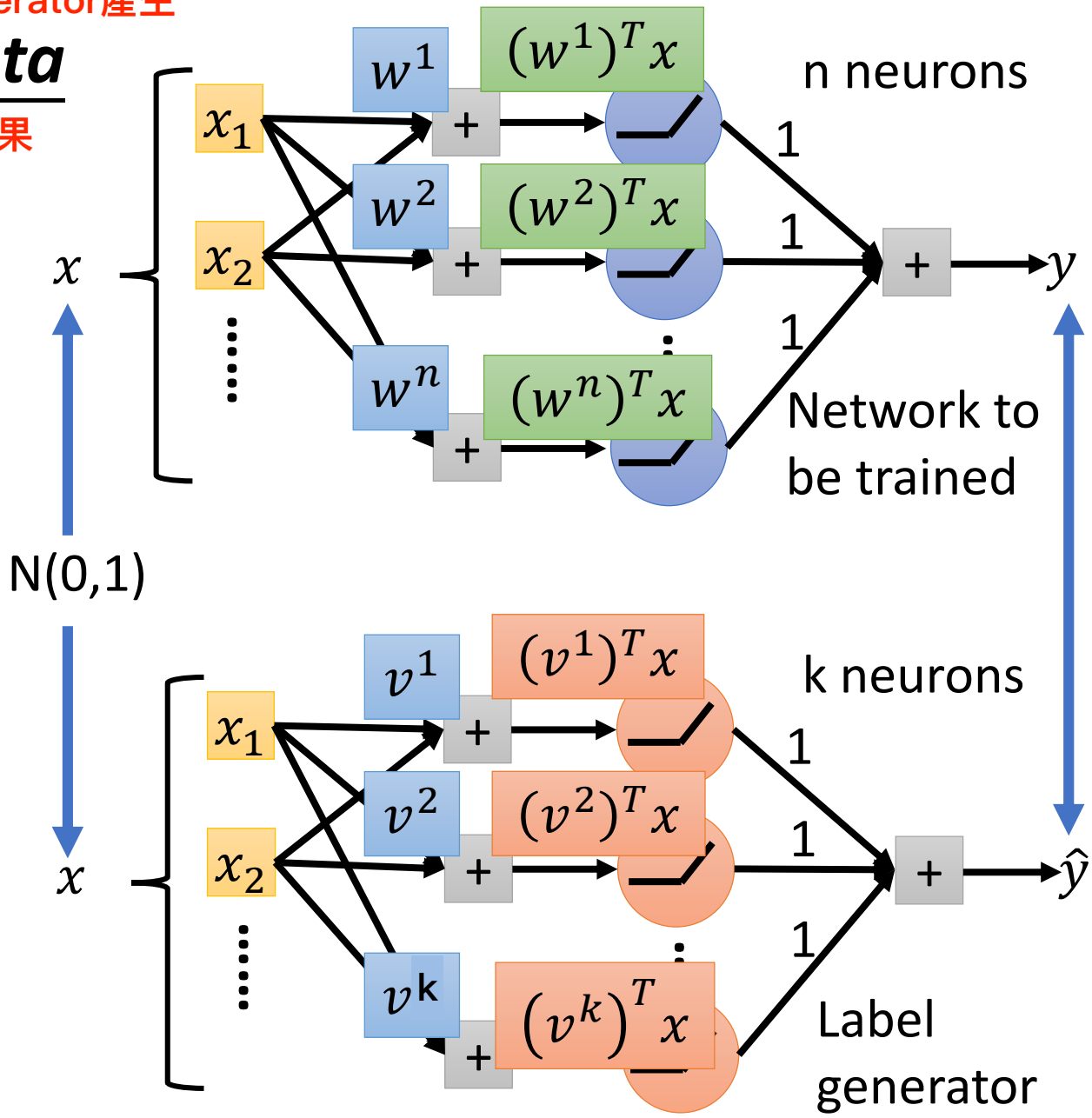
Considering Data

但是n=k以及n>k得到的結果是很不一樣的

If $n \geq k$ and $w^i = v^i$

We obtain global minima

The number of k and n matters



Considering Data

Table 1: Spurious local minima found for $n = k$

k	n	% of runs converging to local minima	Average minimal eigenvalue	Average objective value
6	6	0.3%	0.0047	0.025
7	7	5.5%	0.014	0.023
8	8	12.6%	0.021	0.021
9	9	21.8%	0.027	0.02
10	10	34.6%	0.03	0.022
11	11	45.5%	0.034	0.022
12	12	58.5%	0.035	0.021
13	13	73%	0.037	0.022
14	14	73.6%	0.038	0.023
15	15	80.3%	0.038	0.024
16	16	85.1%	0.038	0.027
17	17	89.7%	0.039	0.027
18	18	90%	0.039	0.029
19	19	93.4%	0.038	0.031
20	20	94%	0.038	0.033

很大的機率停在
local minimum

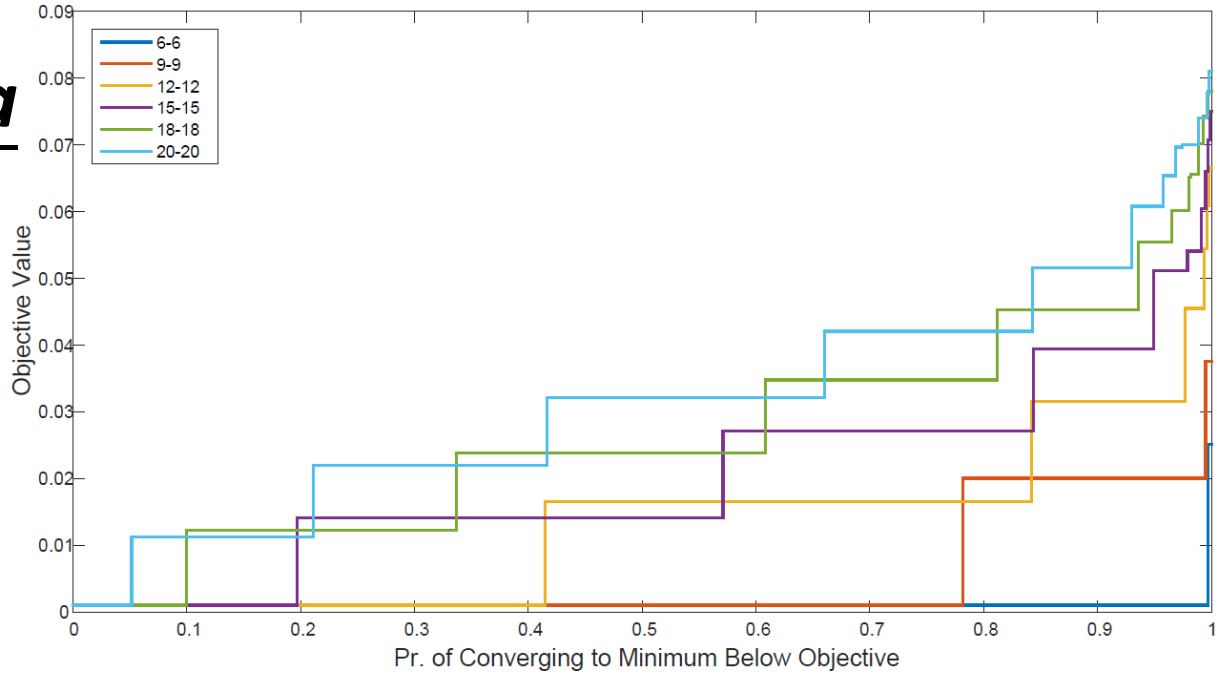
Table 2: Spurious local minima found for $n \neq k$

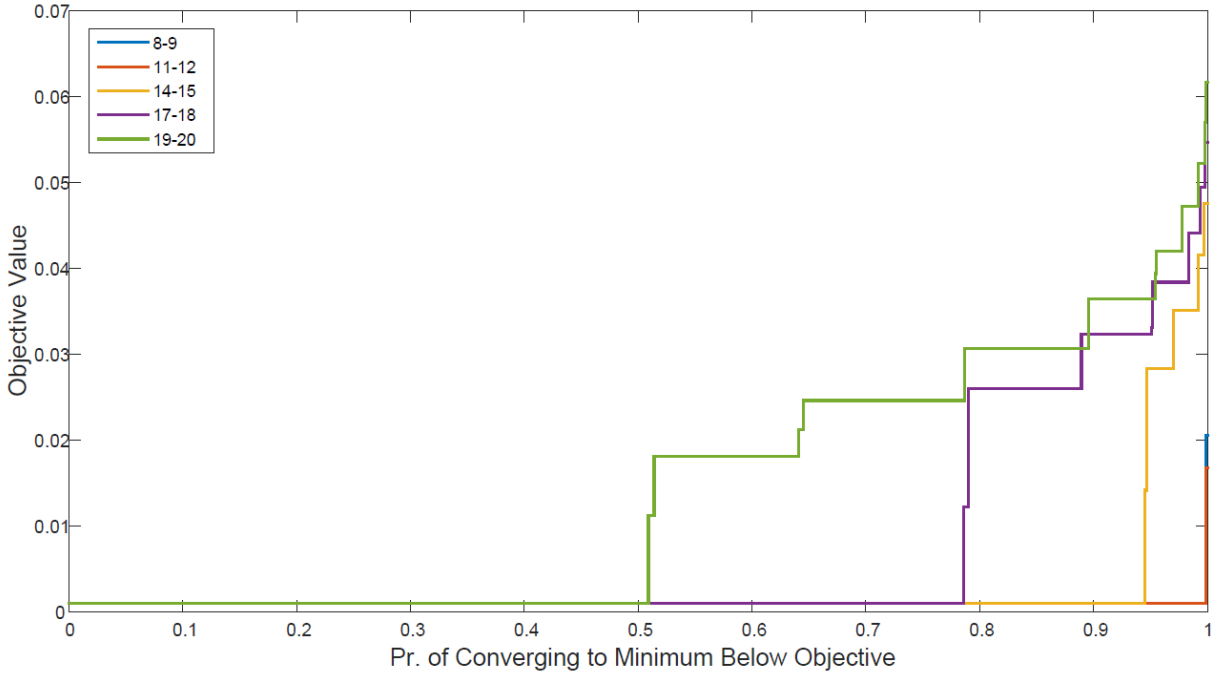
k	n	% of runs converging to local minima	Average minimal eigenvalue	Average objective value
8	9	0.1%	0.0059	0.021
10	11	0.1%	0.0057	0.018
11	12	0.1%	0.0056	0.017
12	13	0.3%	0.0054	0.016
13	14	1.5%	0.0015	0.038
14	15	5.5%	0.002	0.033
15	16	10.1%	0.004	0.032
16	17	18%	0.0055	0.031
17	18	20.9%	0.007	0.031
18	19	36.9%	0.0064	0.028
19	20	49.1%	0.0077	0.027

No local for $n \geq k + 2$

當n>k則可以大幅降低找到local minumin機率
越大越容易找到global minimum

Considering Data





Reference

- Grzegorz Swirszcz, Wojciech Marian Czarnecki, Razvan Pascanu, “Local minima in training of neural networks”, arXiv, 2016
- Itay Safran, Ohad Shamir, “Spurious Local Minima are Common in Two-Layer ReLU Neural Networks”, arXiv, 2017
- Yi Zhou, Yingbin Liang, “Critical Points of Neural Networks: Analytical Forms and Landscape Properties”, arXiv, 2017
- Shai Shalev-Shwartz, Ohad Shamir, Shaked Shammah, “Failures of Gradient-Based Deep Learning”, arXiv, 2017

無法證明說找不到local minimum，應該是在某種假設下會找到local minimum，那我們就反過來做就可以找到global了

The theory should looks like …

Under some conditions (initialization, data, ……),

We can find global optimal.

Conjecture about Deep Learning

Y. Lecun在2007說通常local min很接近global min

Almost all local minimum have very similar loss to the global optimum, and hence finding a local minimum is good enough.

Analyzing Hessian

- When we meet a critical point, it can be saddle point or local minima.
- Analyzing H

If the network has N parameters

eigen vector	v_1	v_2	v_3	……	v_N
eigen value	λ_1	λ_2	λ_3	……	λ_N

假設

We assume λ has 1/2 (?) to be positive, 1/2 (?) to be negative.

Analyzing Hessian

- If N=1: v_1 λ_1

  1/2 local minima, 1/2 local maxima,
  Saddle point is almost impossible

- If N=2: v_1 v_2 λ_1 λ_2

  + + - -
  1/4 local minima, 1/4 local maxima,
  1/2 Saddle points
  + -, - +

- If N=10:

  1/1024 local minima, 1/1024 local maxima,
  Almost every critical point is saddle point

When a network is very large,

It is almost impossible to meet a local minima.

Saddle point is what you need to worry about.

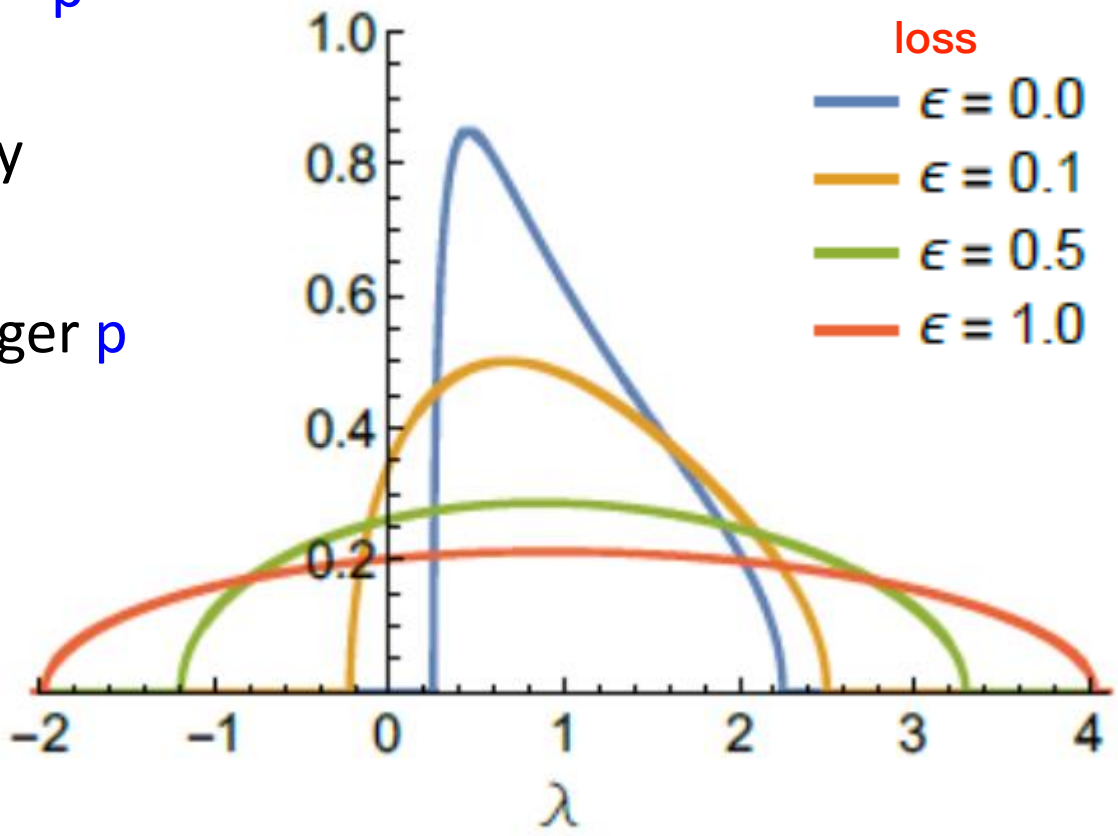
Error v.s. Eigenvalues

如果loss越來越小，則eigen value會逐漸往政的方向篇

We assume λ has ~~1/2 (?)~~ p to be negative.

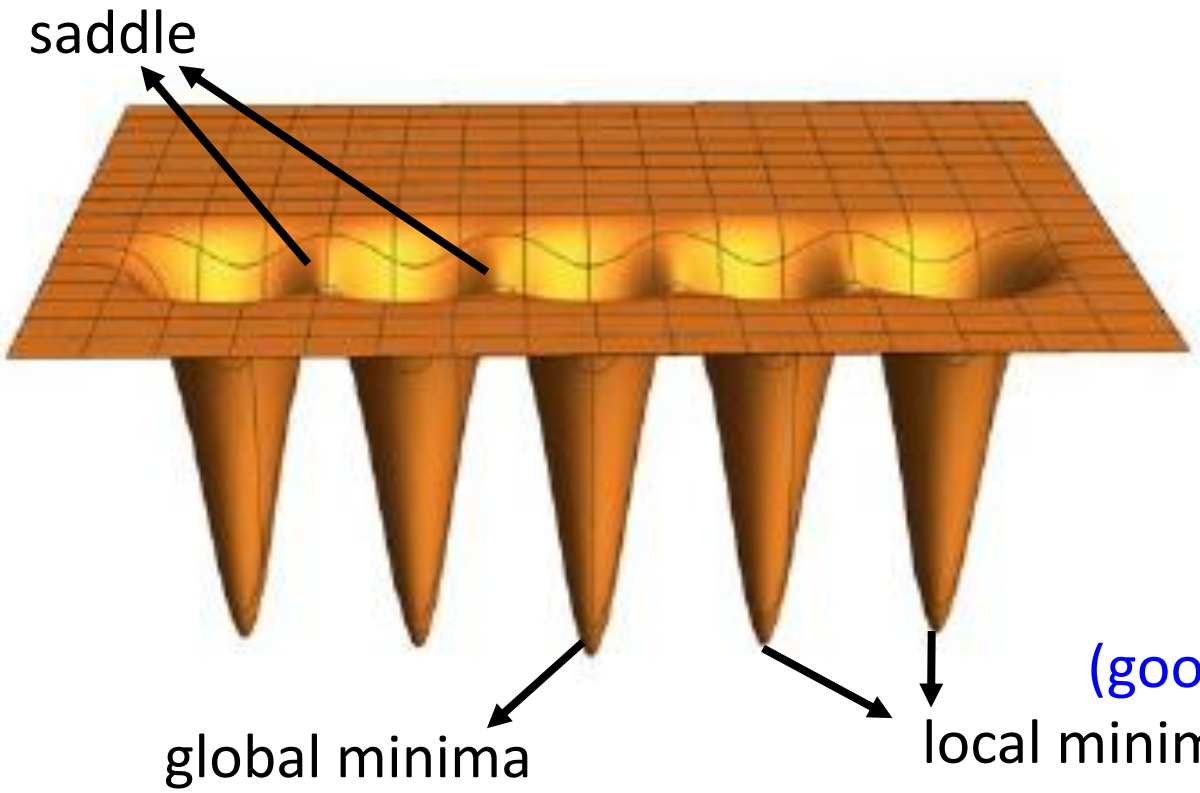
p is a probability related to error

Larger error, larger p



Source of image:
http://proceedings.mlr.press/v70/pennington17a/pennington17a.pdf

Guess about Error Surface

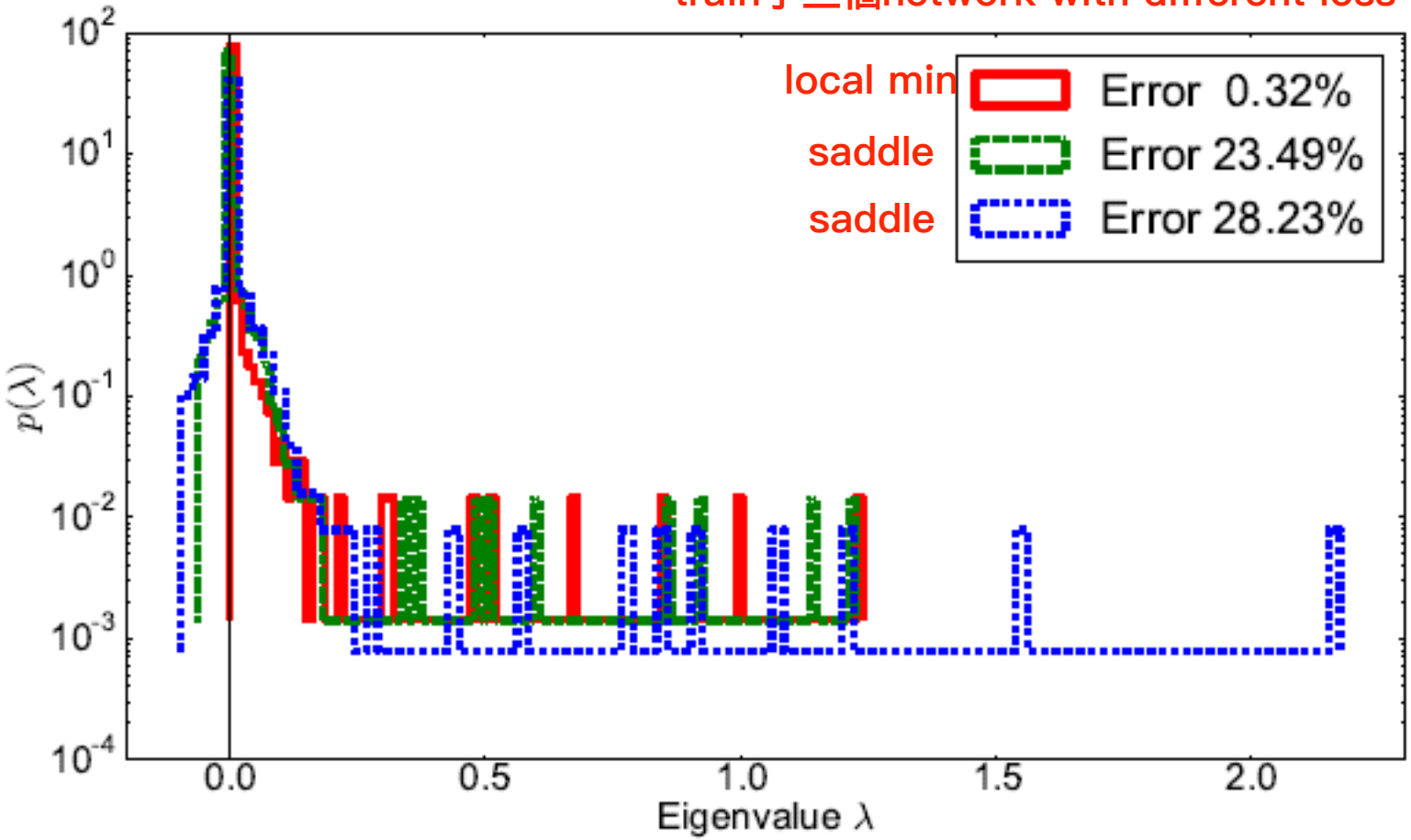


https://stats385.github.io/assets/lectures/Understanding_and_improving_deep_learing_with_random_matrix_theory.pdf

Training Error v.s. Eigenvalues

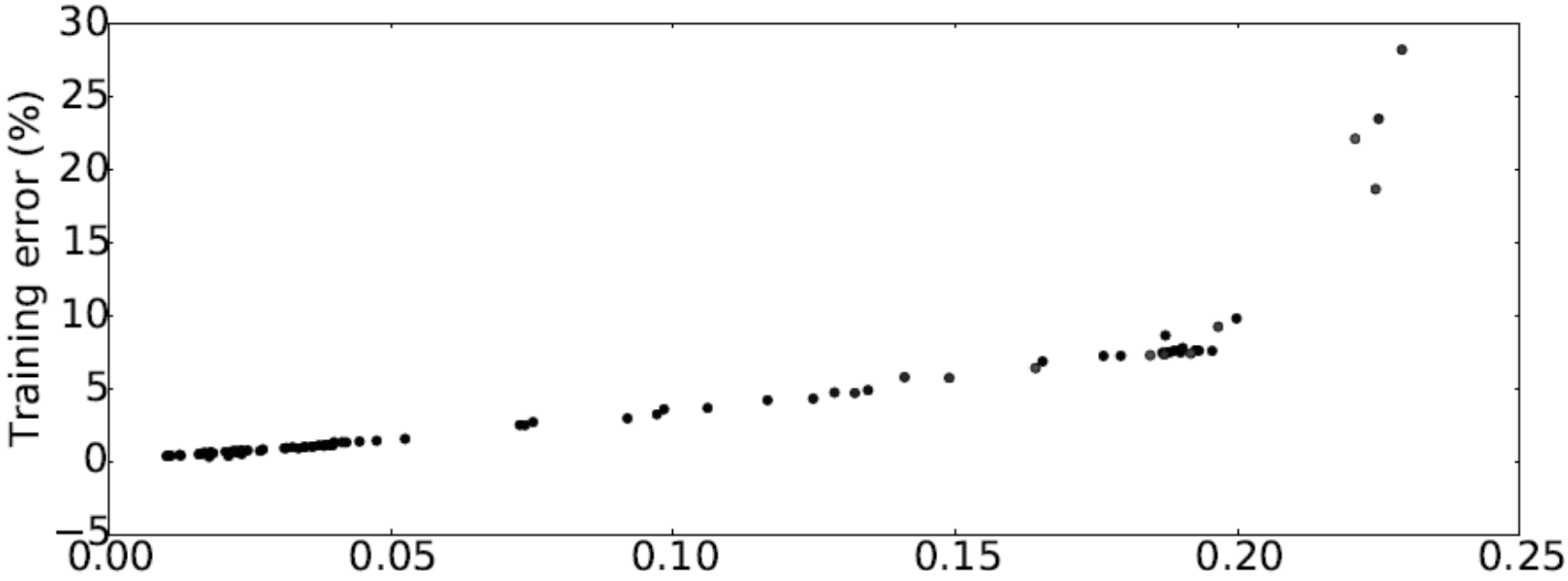
各找了critical point解他的hessian eigen value

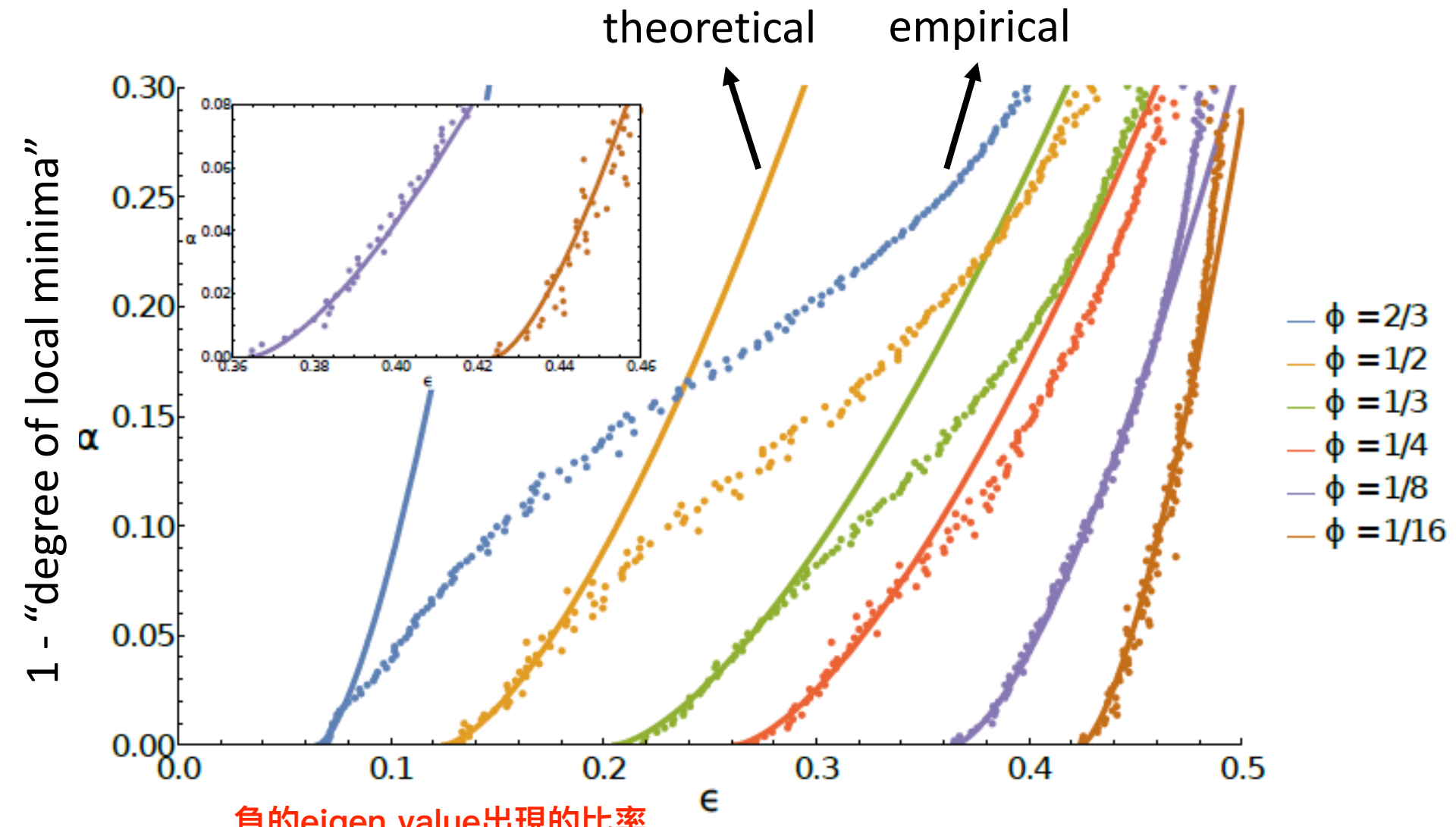
train了三個network with different loss



Training Error v.s. Eigenvalues

Portion of positive eigenvalues → "Degree of Local Minima"





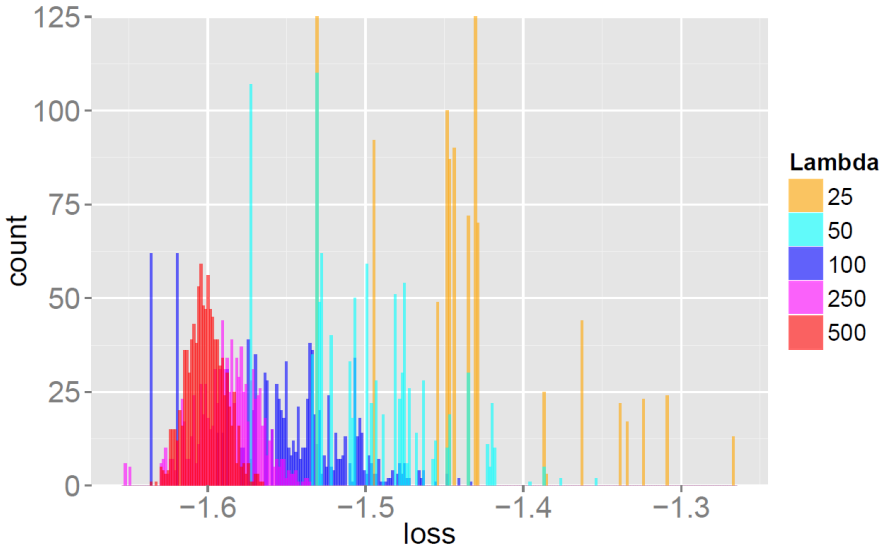
負的eigen value出現的比率

1 - “degree of local minima”
(portion of negative eigen values)

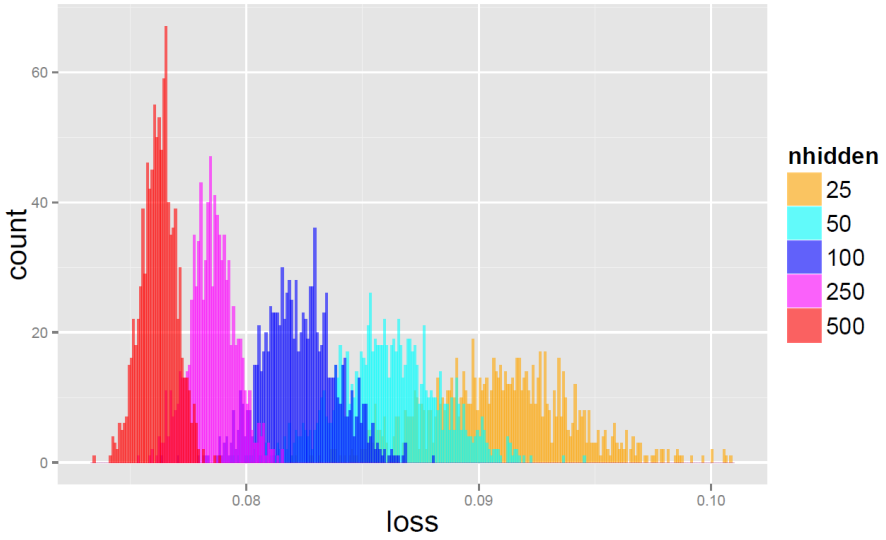
$$\alpha \propto \left(\frac{\varepsilon}{c} - 1\right)^{3/2}$$

Spin Glass v.s. Deep Learning

- Deep learning is the same as spin glass model with ***7 assumptions***.



spin glass model



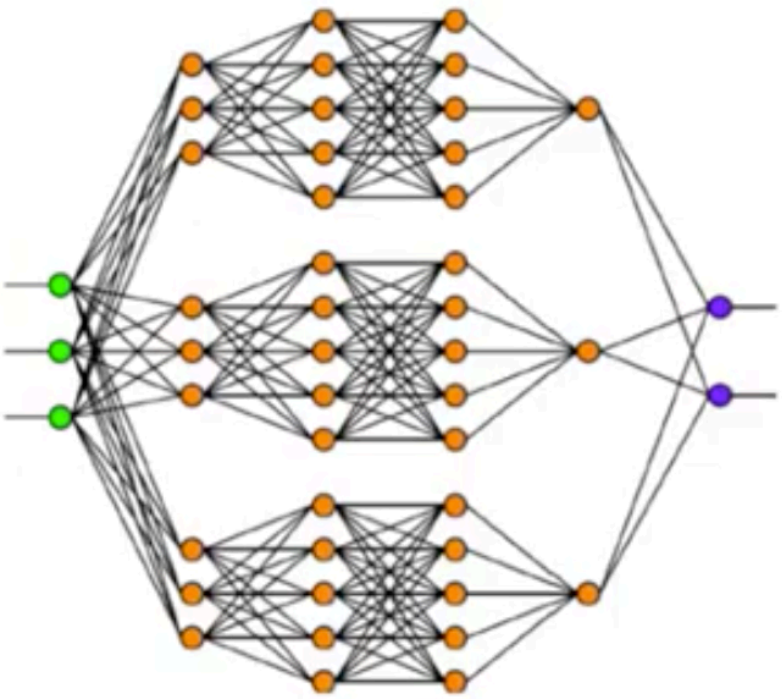
network

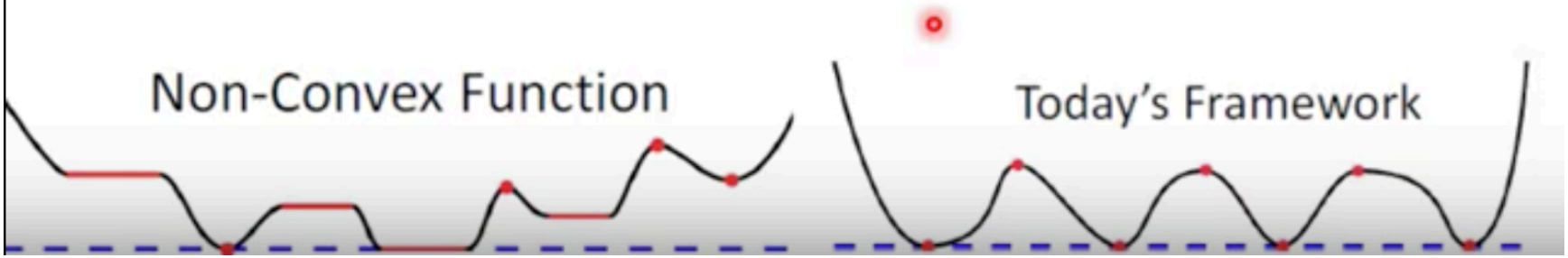
Benjamin D. Haeffele, Rene Vidal, ” Global Optimality in Neural Network Training”, CVPR, 2017

More Theory

- If the size of network is **large enough**, we can find global optimal by gradient descent
  - Independent to initialization

無視initial





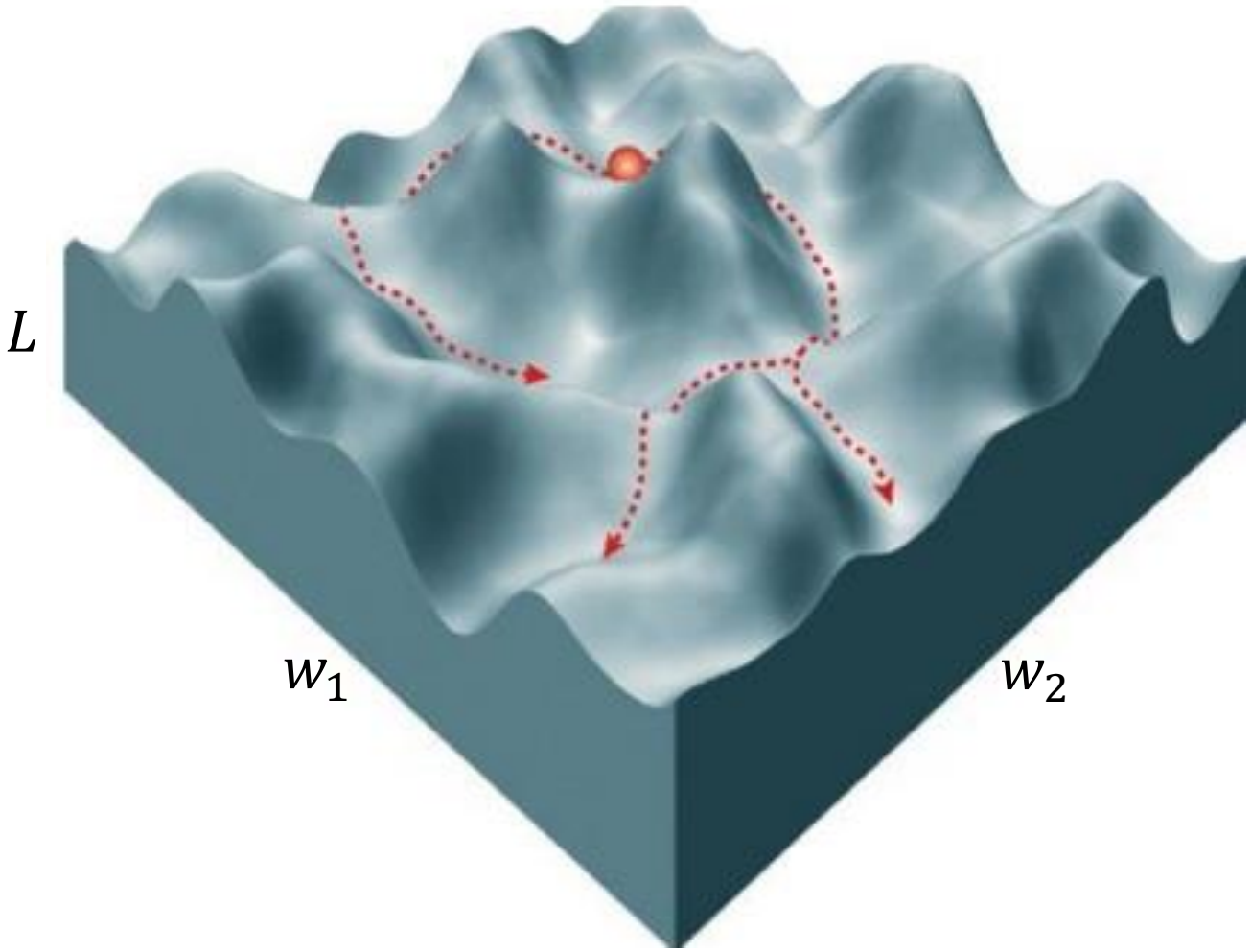
Reference

hessian

- Razvan Pascanu, Yann N. Dauphin, Surya Ganguli, Yoshua Bengio, On the saddle point problem for non-convex optimization, arXiv, 2014
- Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, Yoshua Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization”, NIPS, 2014
- Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, Yann LeCun, “The Loss Surfaces of Multilayer Networks”, PMLR, 2015
- Jeffrey Pennington, Yasaman Bahri, “Geometry of Neural Network Loss Surfaces via Random Matrix Theory”, PMLR, 2017 證明了local min接近global min
- Benjamin D. Haeffele, Rene Vidal, ” Global Optimality in Neural Network Training”, CVPR, 2017

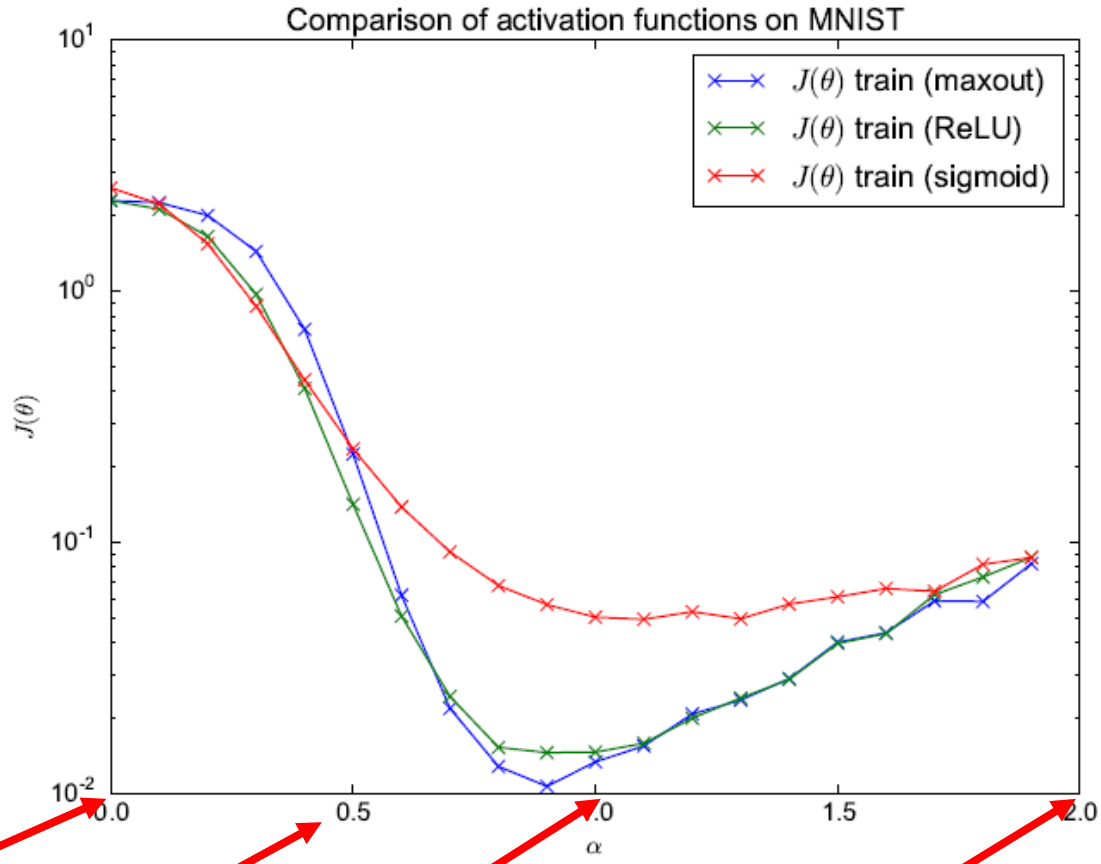
What does the Error Surface look like?

Error Surface



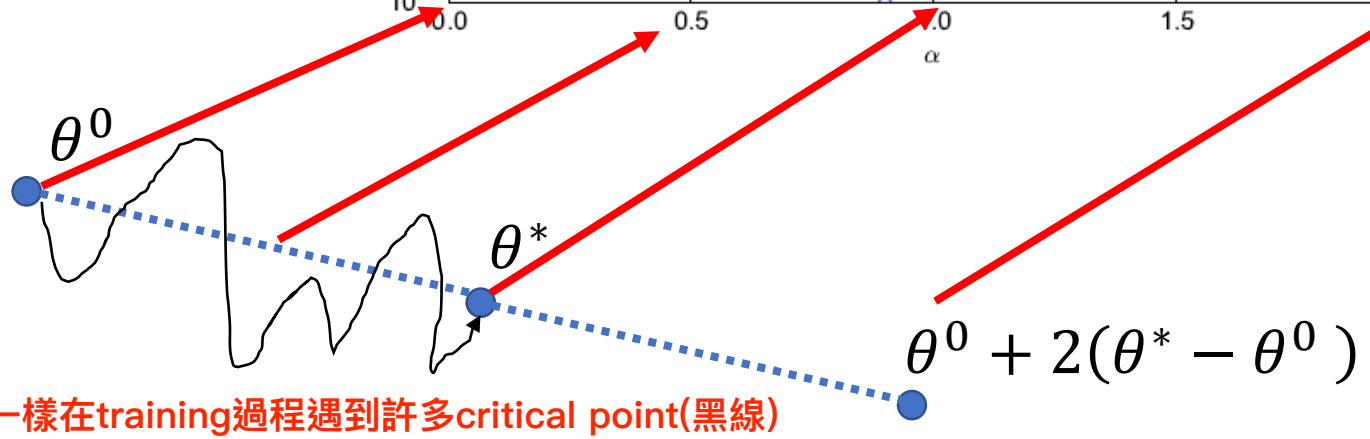
Profile

Visualize the loss value through the specific direction of parameters



local minima is rare?

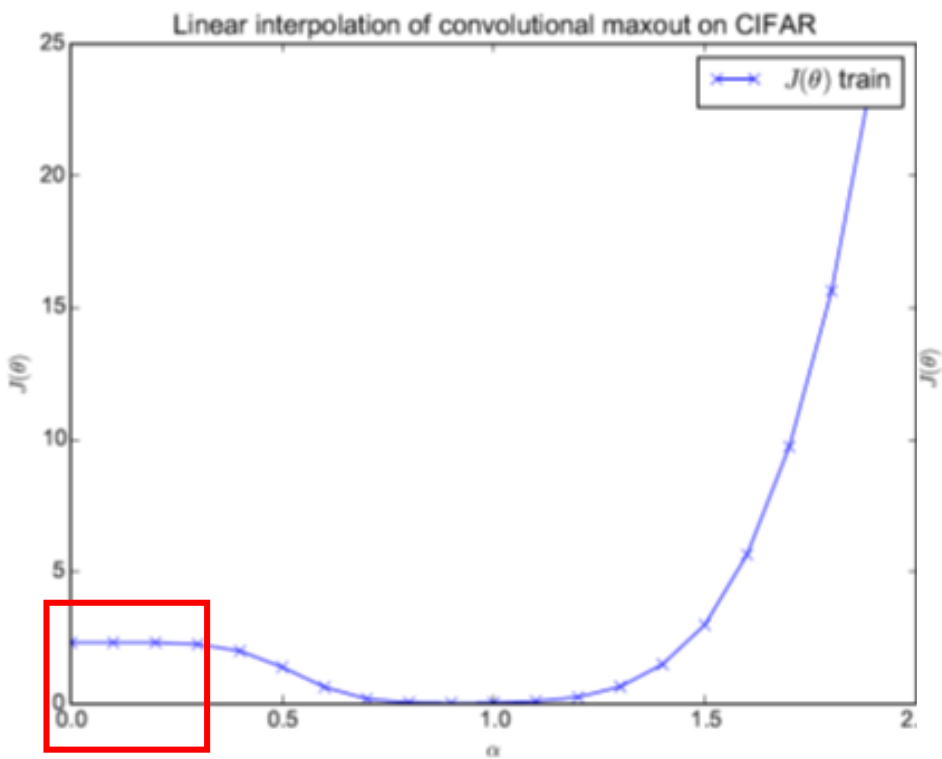
θ: 高維空間中的點，代表參數

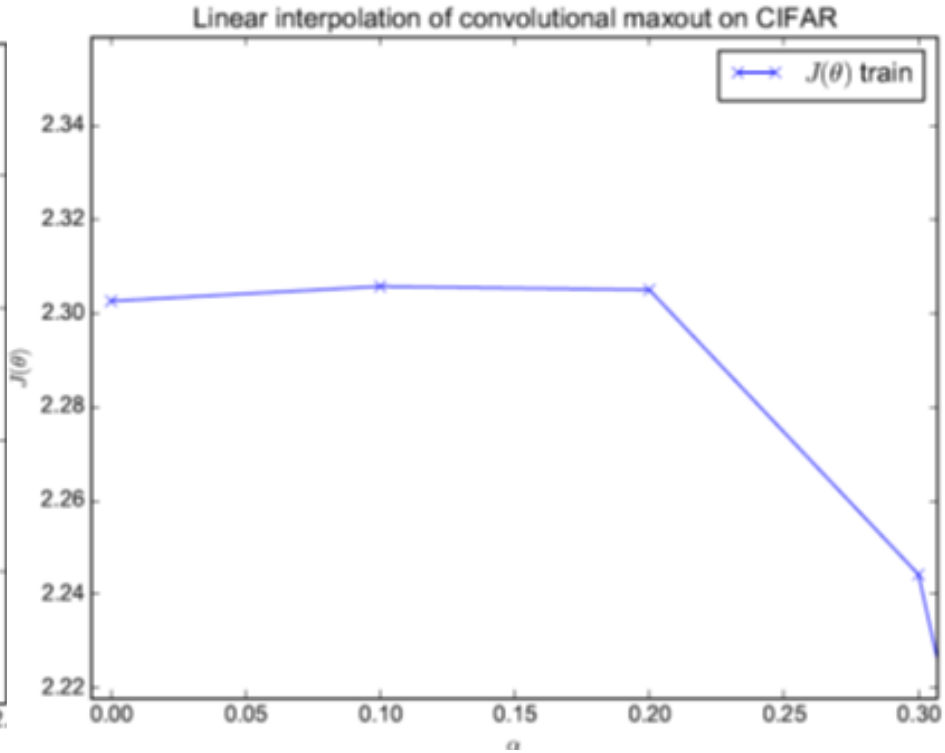


並非像想像的一樣在training過程遇到許多critical point(黑線)
反而是平坦的

Profile

要確認是critical point要先檢查gradient=0
然後利用hessian檢查是否為local min/ max

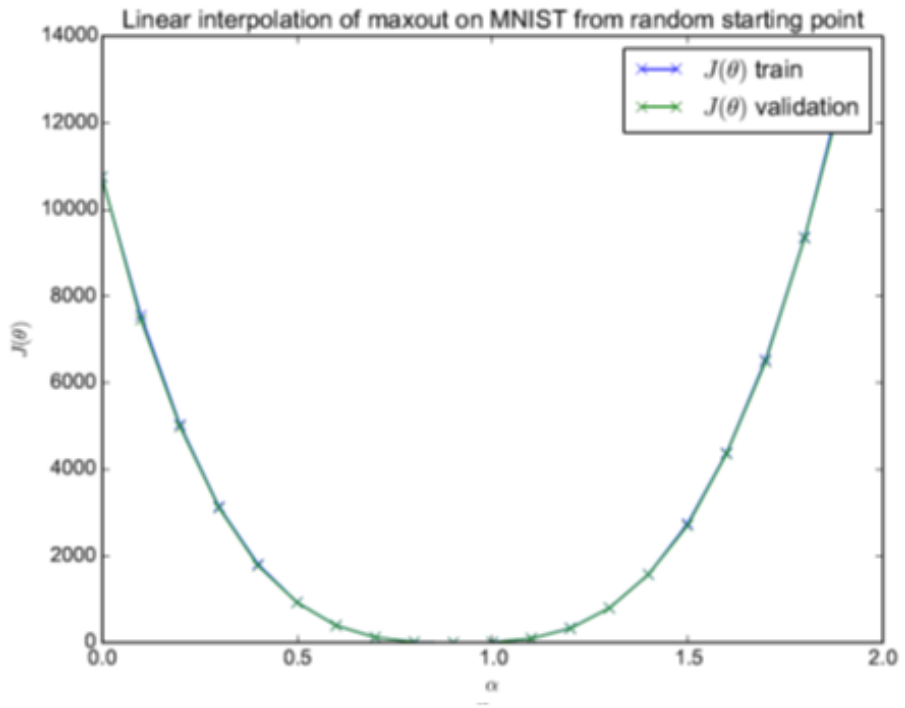




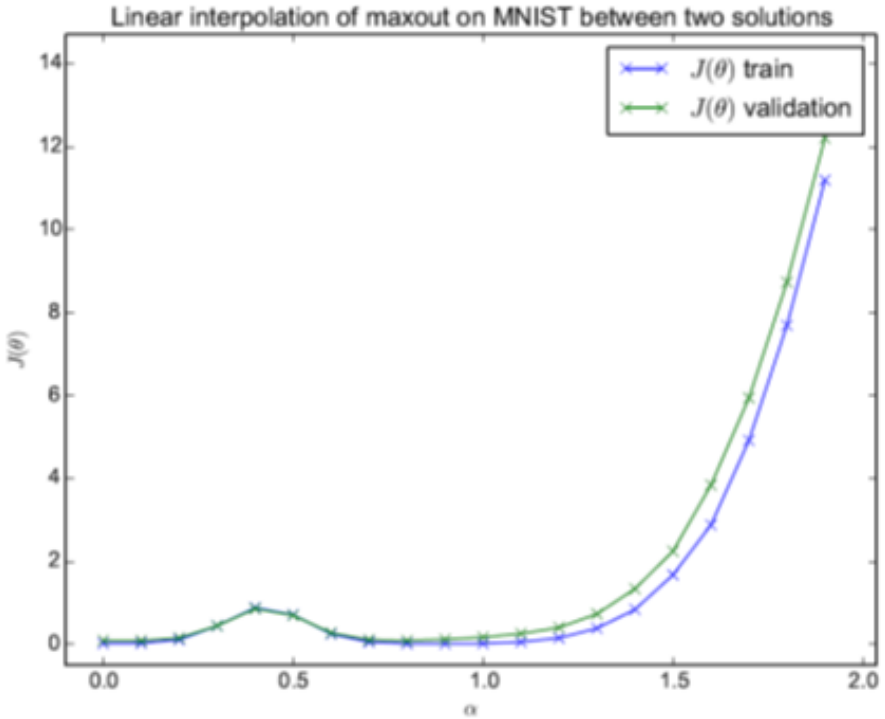
algorithm solution

Profile

從不兩個不同的initialization train可以得到兩種solution



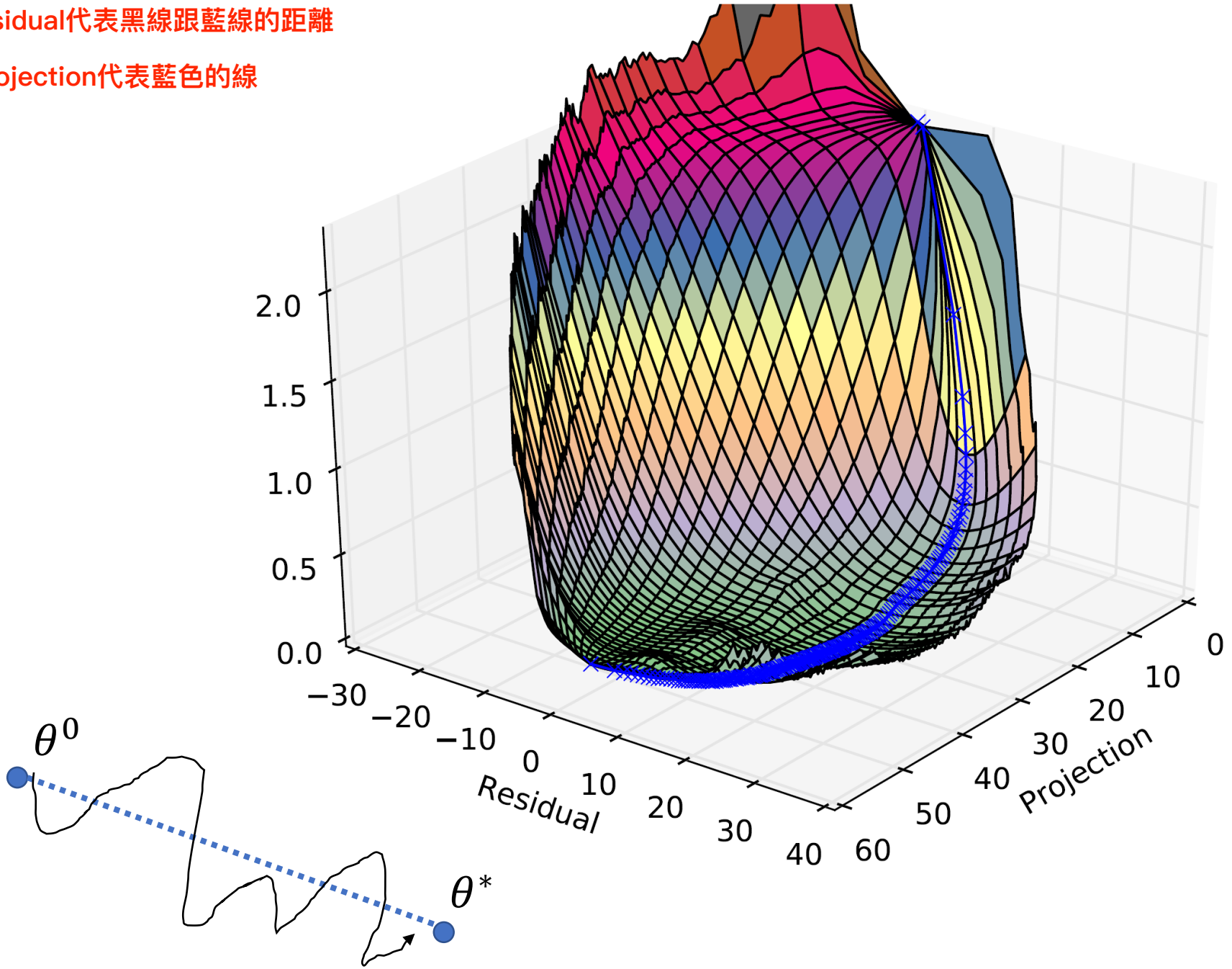
two random starting points



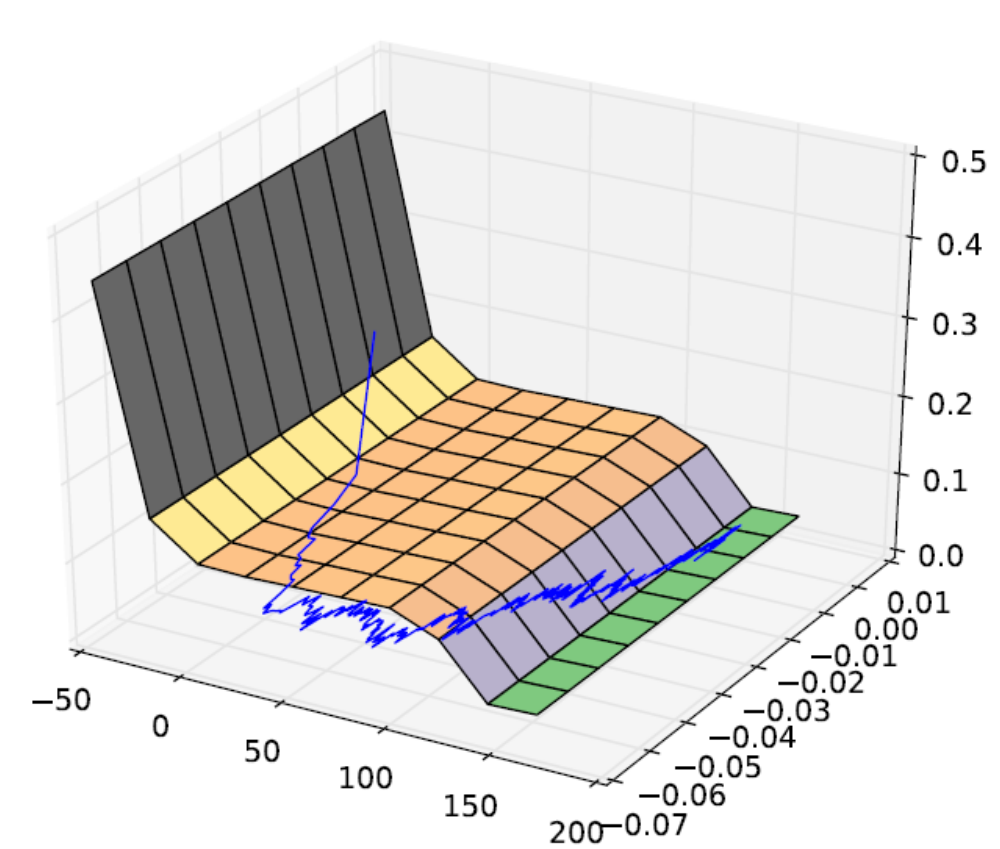
two “solutions”

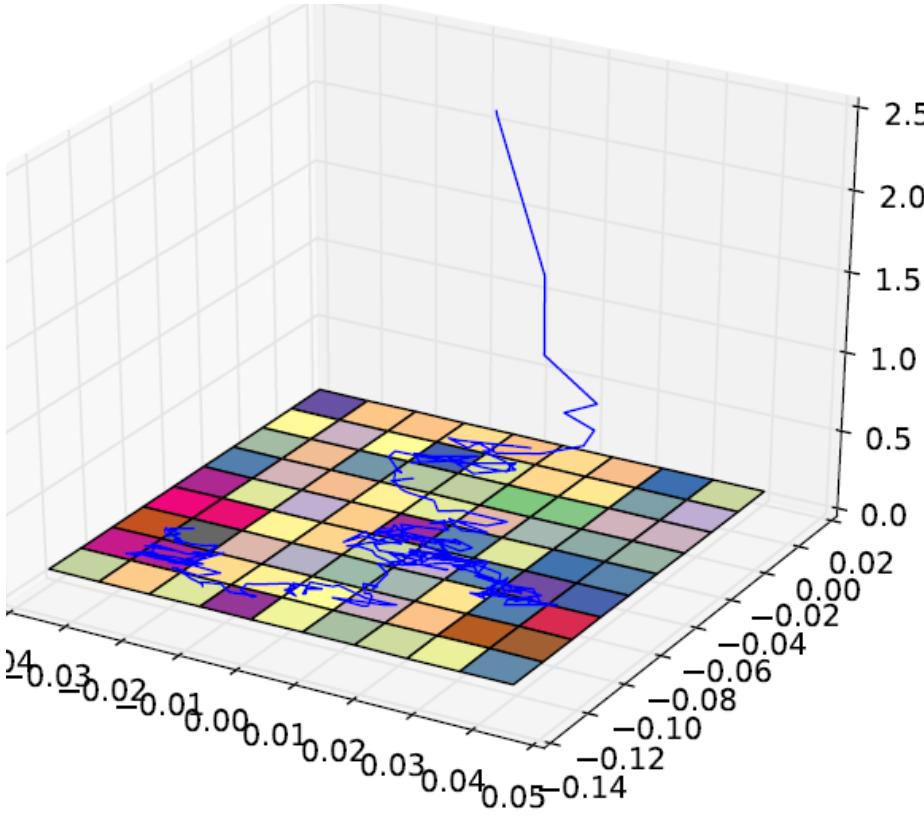
residual代表黑線跟藍線的距離

projection代表藍色的線

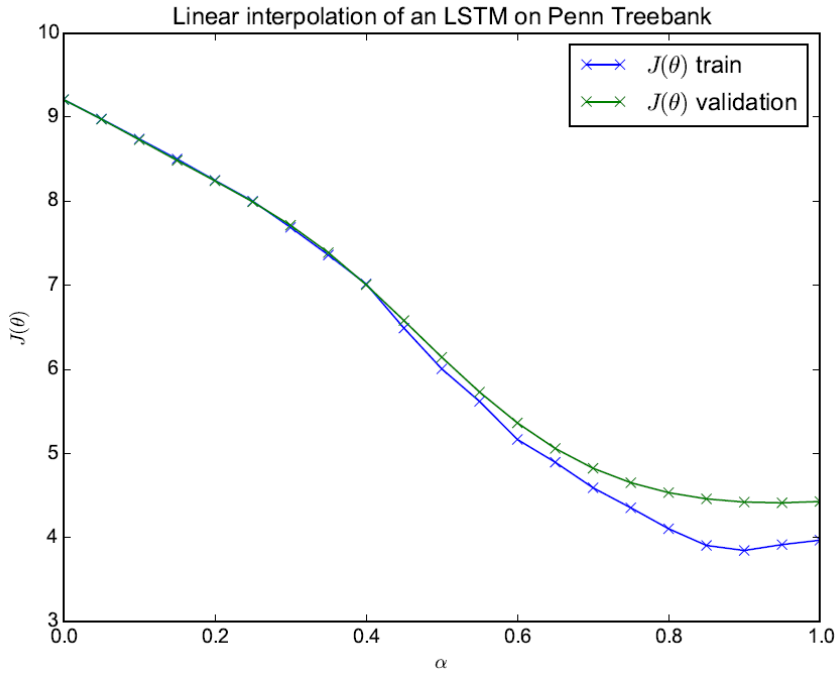


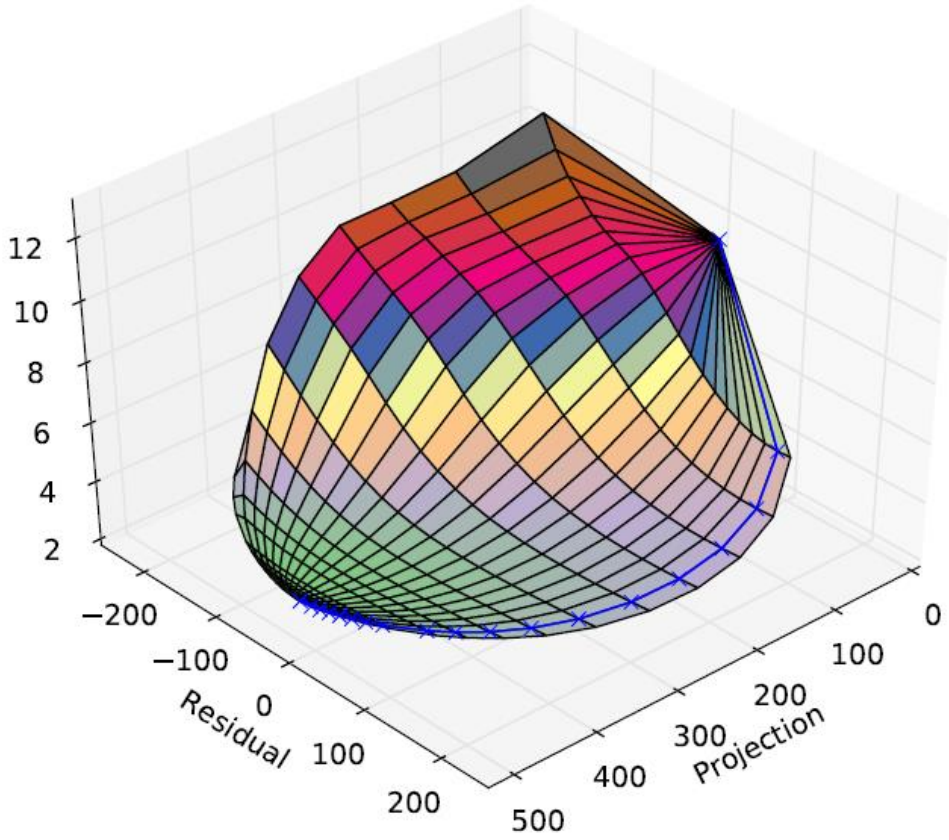
固定其他參數，只抓random兩個參數
調整並做visualization





Profile - LSTM





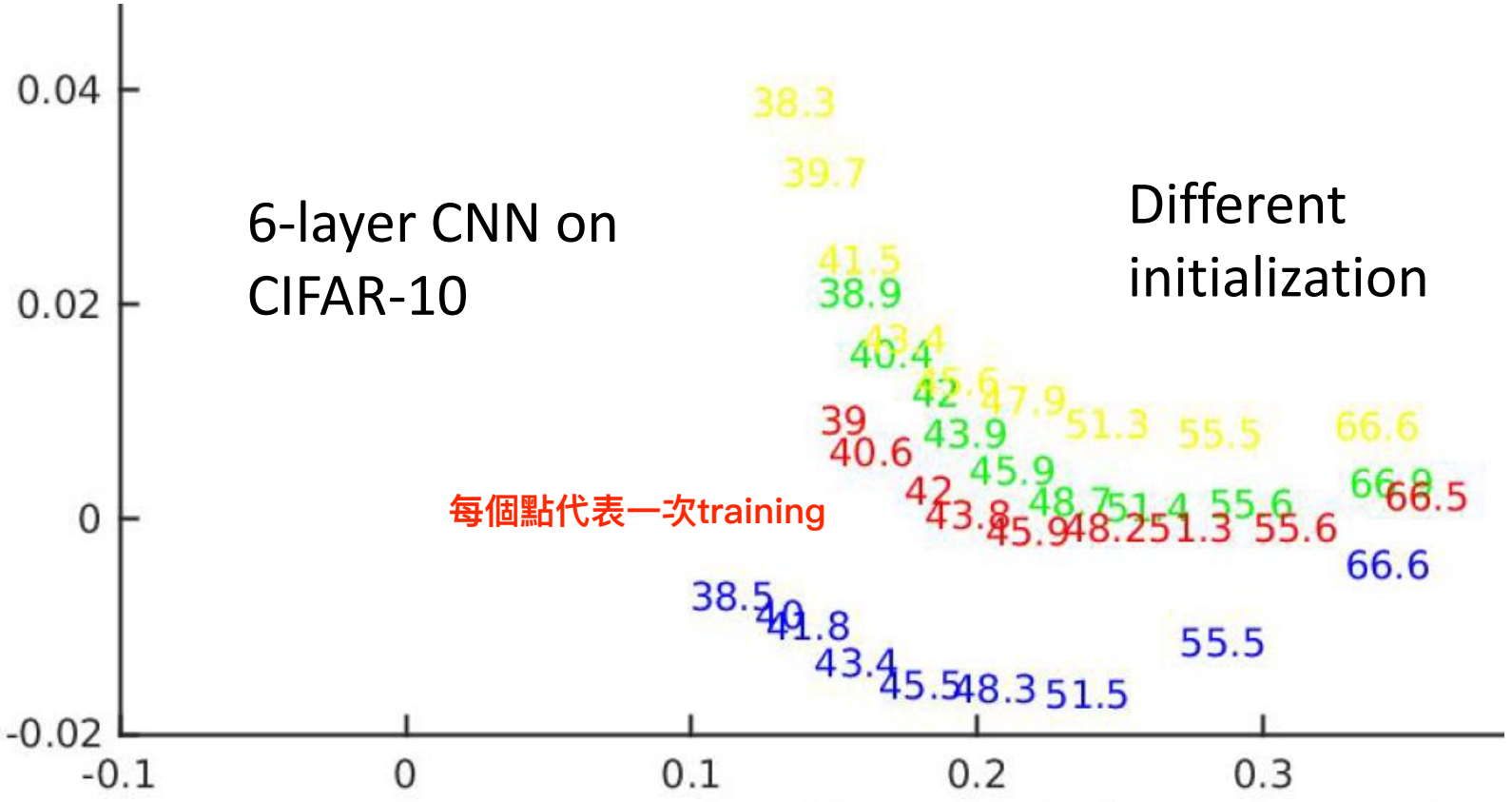
QianliLiao,TomasoPoggio,"TheoryII:LandscapeoftheEmpiricalRiskin Deep Learning", arXiv 2017

在不同的initialization Train Network，並且記錄下所有參數，並且將它做PCA降維

同一個model with different initial 結果差不多，但不同model之間差很多

Training Processing

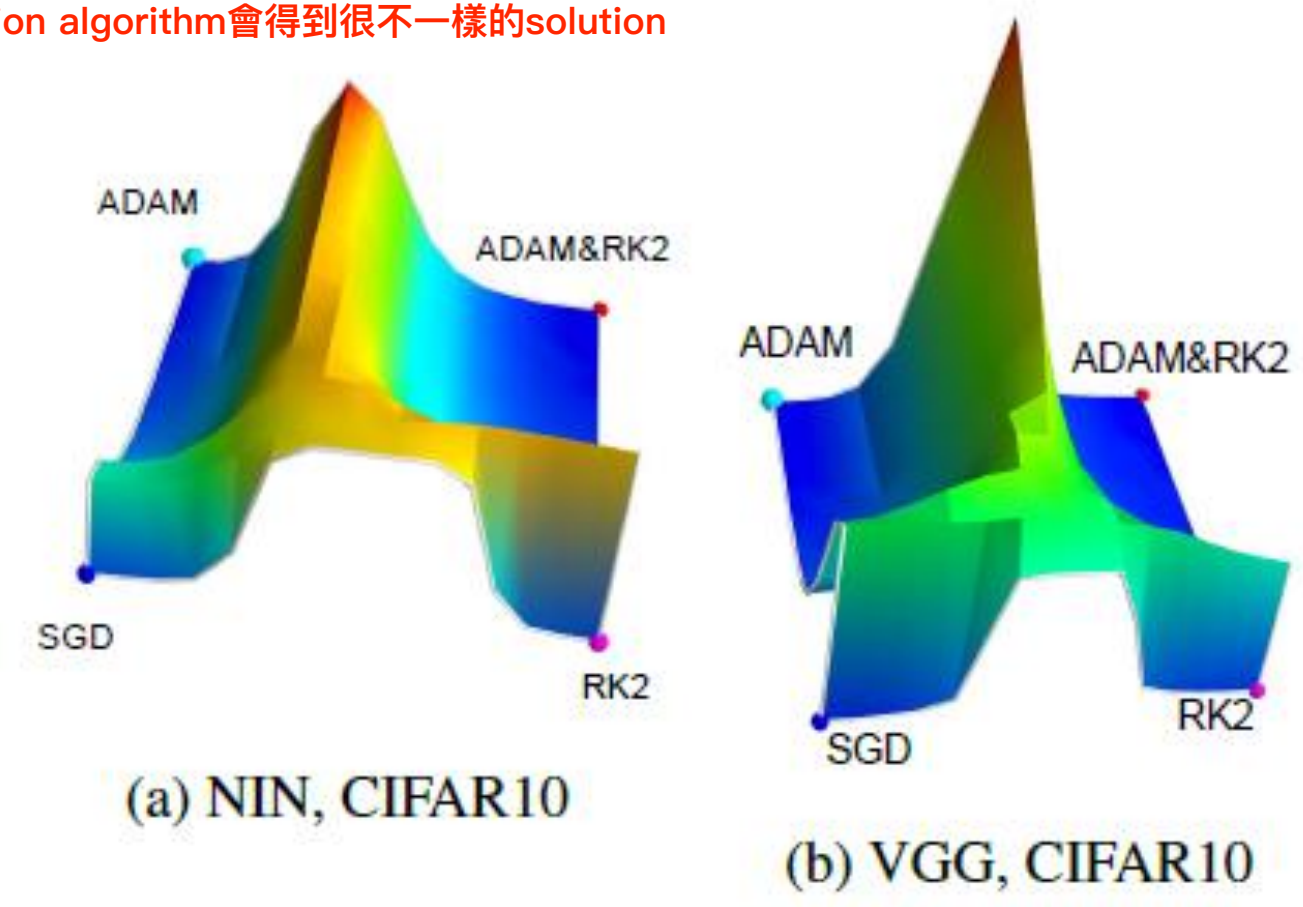
Different initialization / different strategies usually lead to similar loss (there are some exceptions).

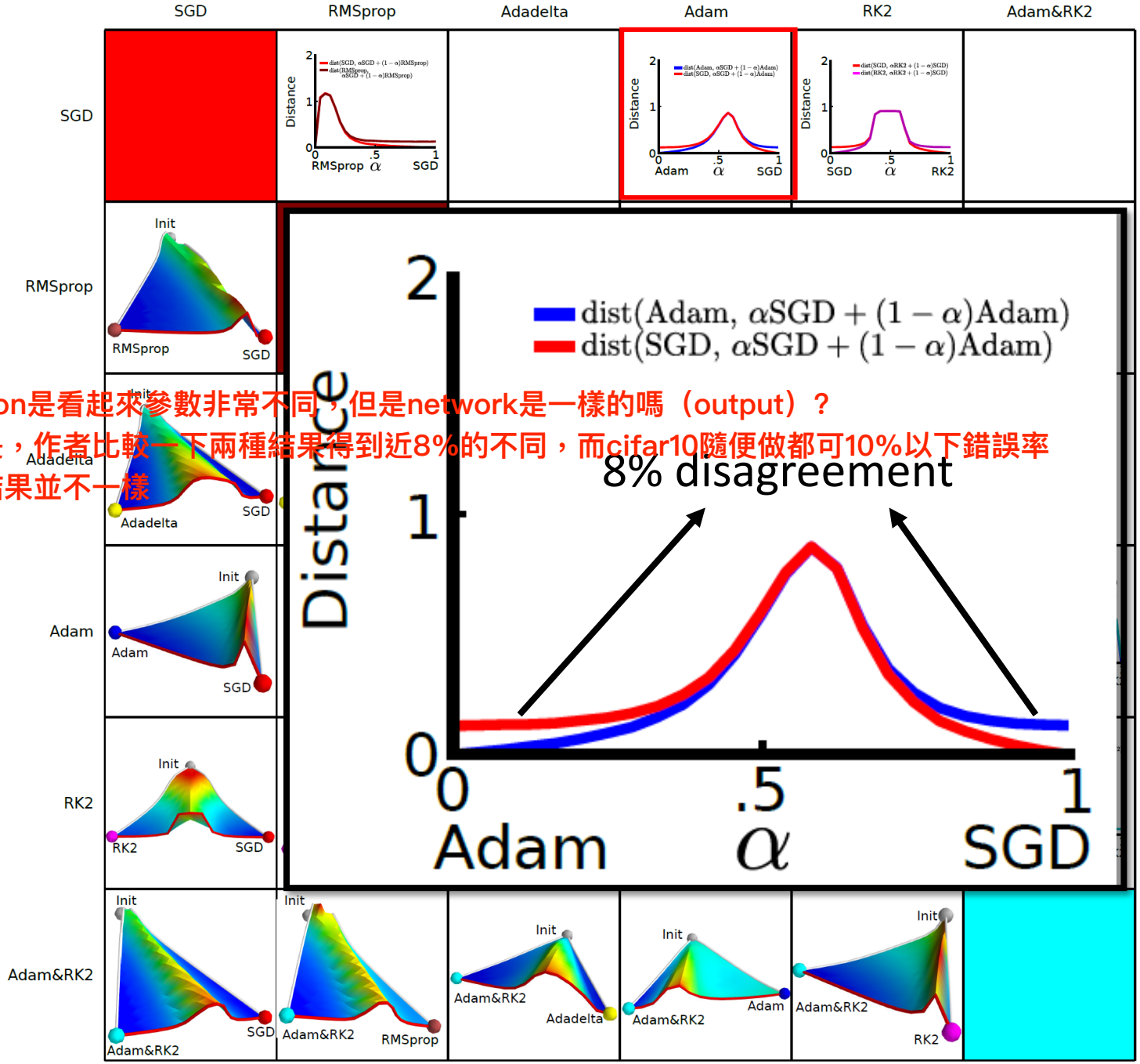


Training Processing

- Different strategies (the same initialization)

四種不同的optimization algorithm會得到很不一樣的solution





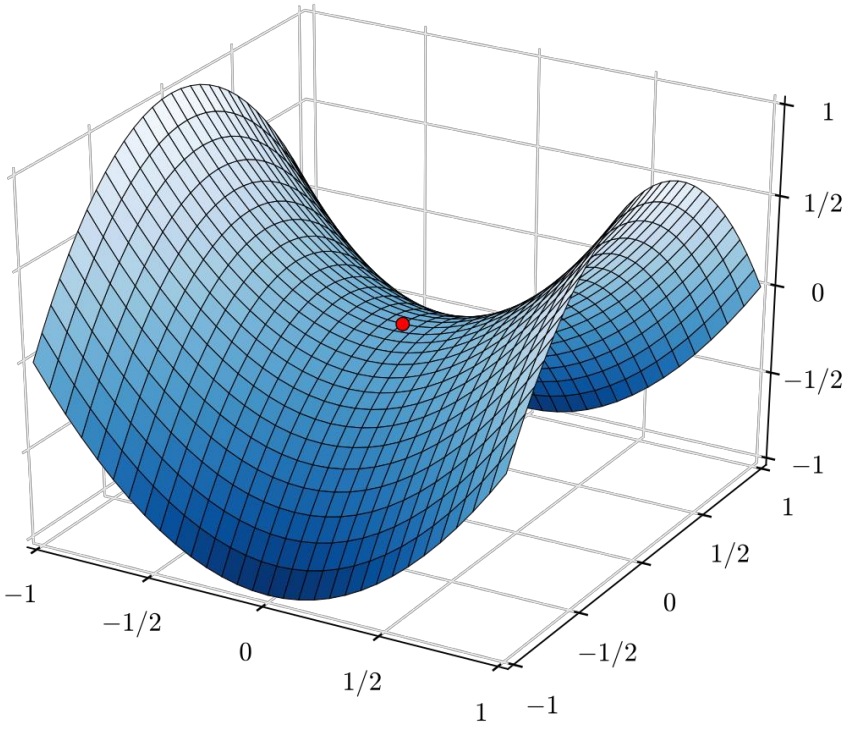
solution是看起來參數非常不同，但是network是一樣的嗎（output）？
並不是，作者比較一下兩種結果得到近8%的不同，而cifar10隨便做都可10%以下錯誤率
因此結果並不一樣

(c) VGG, CIFAR10

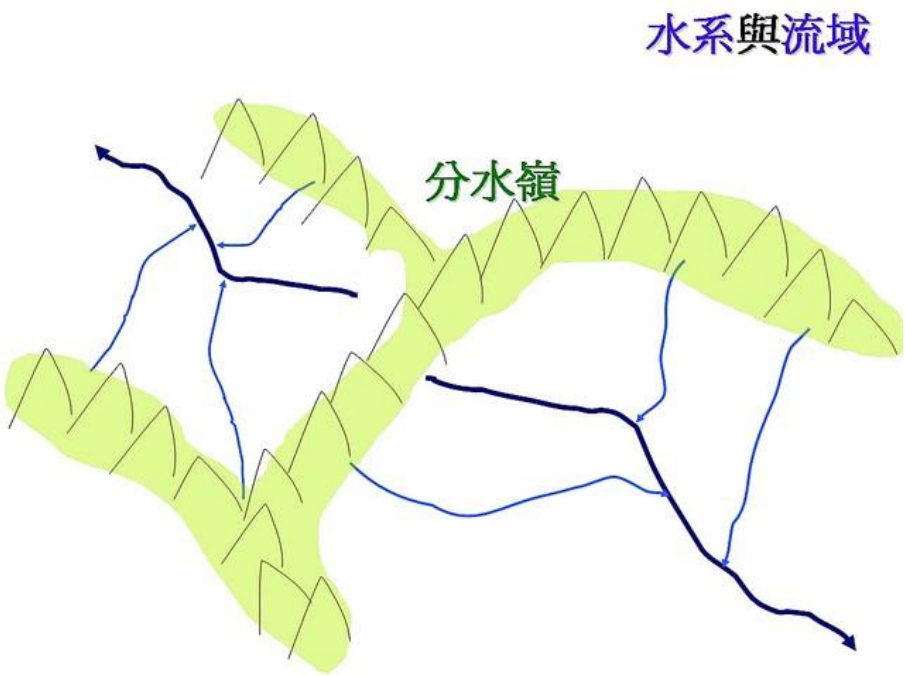
http://mypaper.pchome.com.tw/ccschoolgeo/post/1311484084

Training Processing

initialize不同的時候就已經會影響到不同的結果



何時分道揚鑣？



Different training strategies

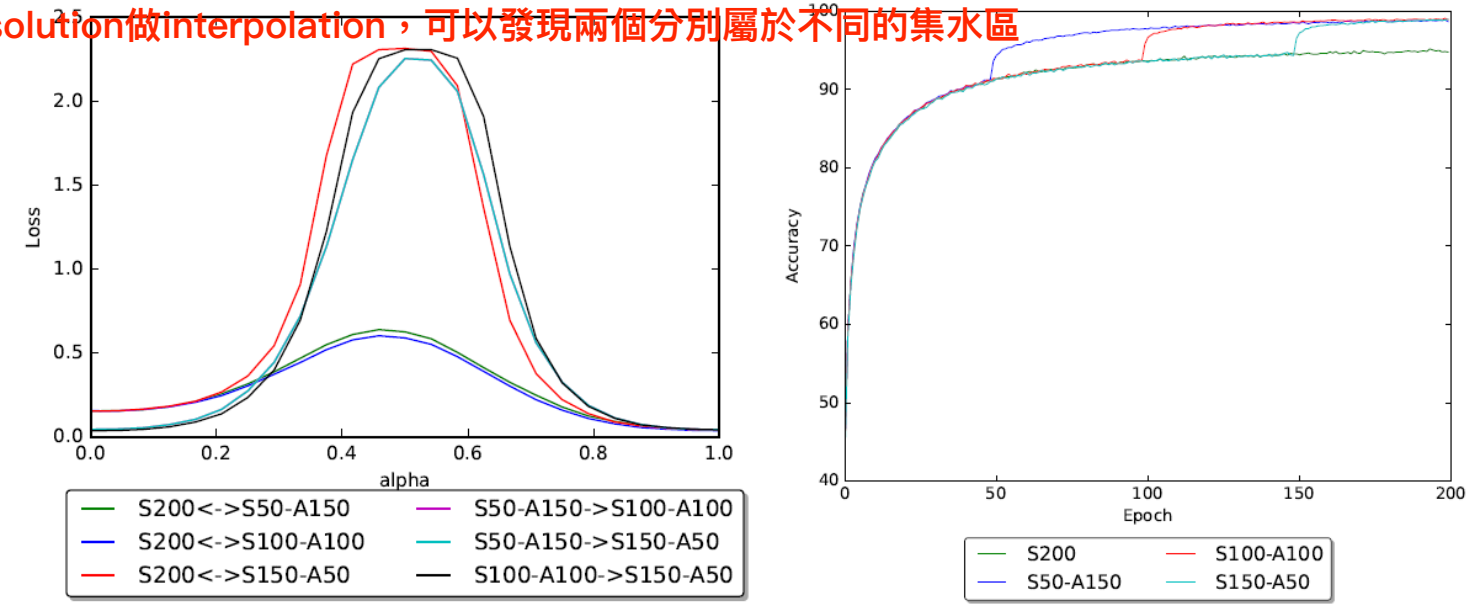
Different basins

Training Processing

- Training strategies make difference at all stages of training

不管在什麼時候轉換成adam，最後得到的結果都差不多（流域都相同）

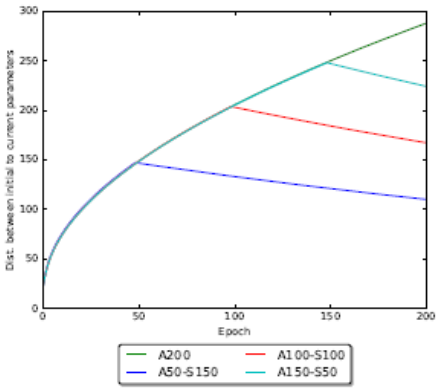
兩個solution做interpolation，可以發現兩個分別屬於不同的集水區



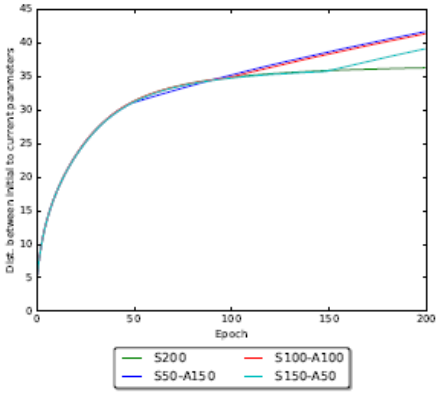
不同的algorithm最後走到的區域是非常不同的

(b) NIN: Switching from SGD (S, $\eta = .1$) to Adam (A, $\eta = .0001$).

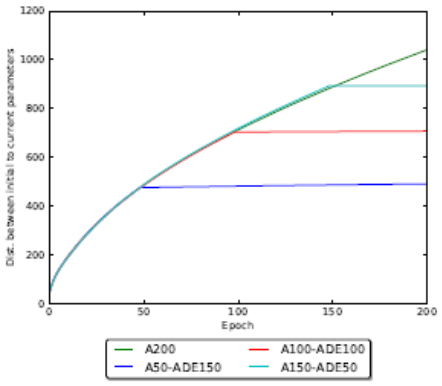
不同的optimization algorithm有各自不同的特性



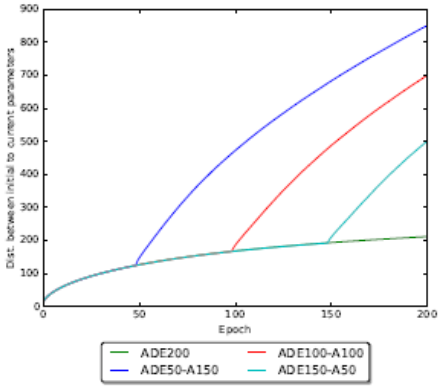
(a)



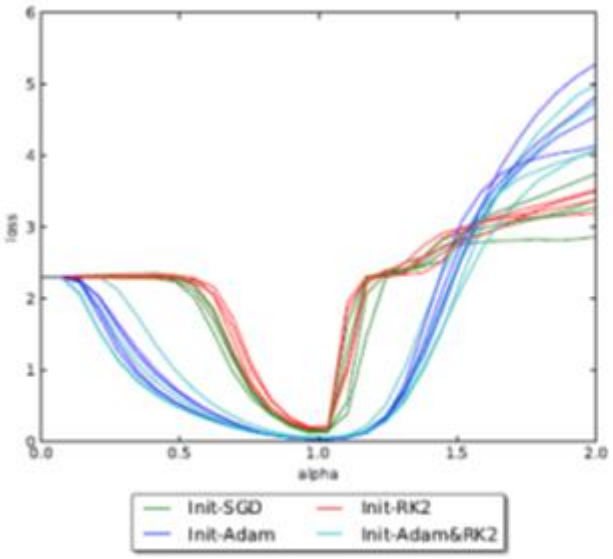
(b)



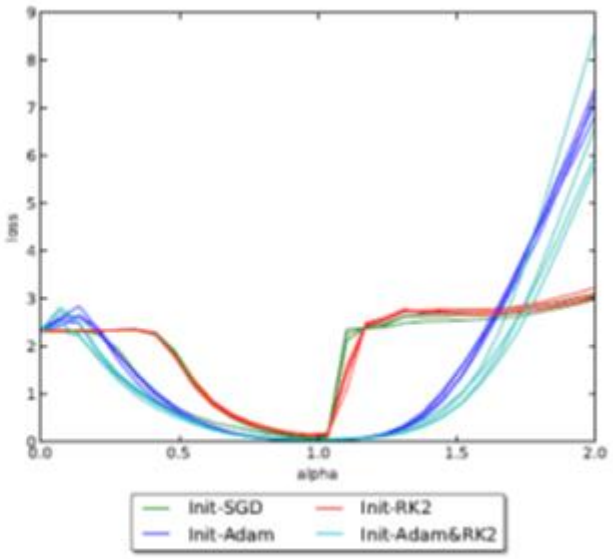
(c)



(d)



(a) CIFAR10, NIN



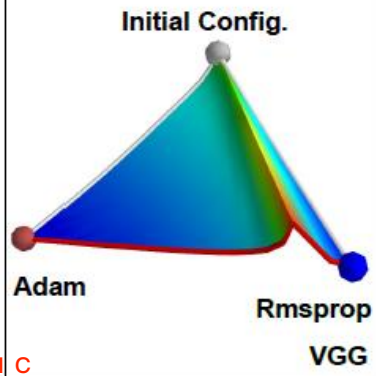
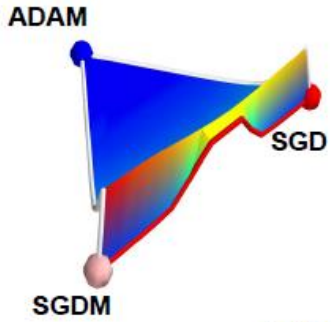
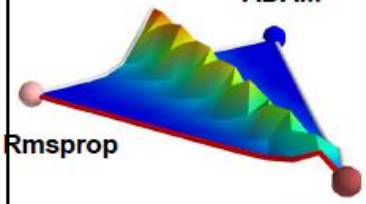
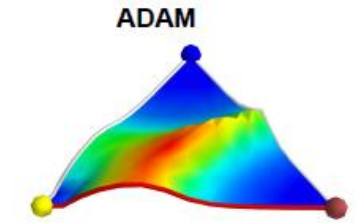
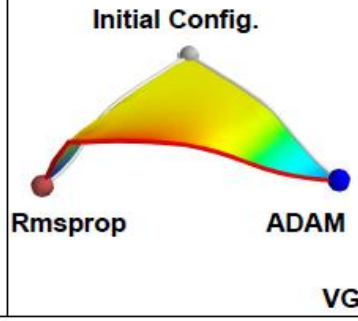
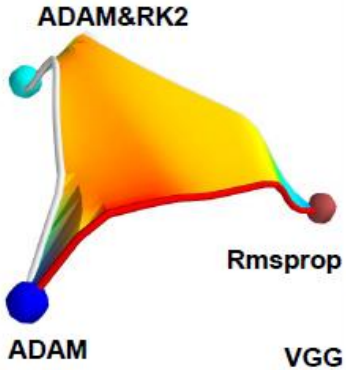
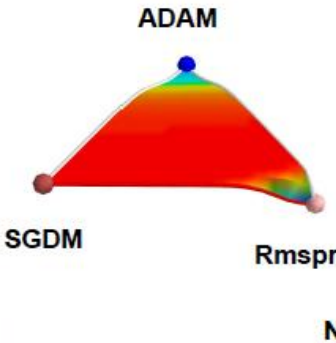
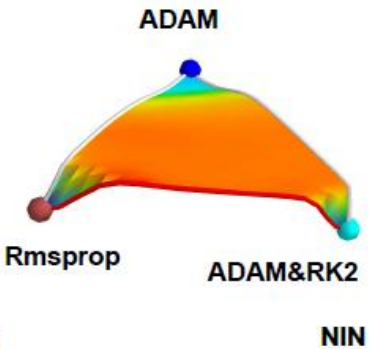
(b) CIFAR10, VGG

每個顏色代表一種algorithm

Larger basin for Adam

Batch Normalization

不同solution間有一個很尖銳的分水嶺

With Batch Normalization				
Without Batch Normalization				

不同的solution間有一個很平坦的高原, which may cause sucking in the saddle point

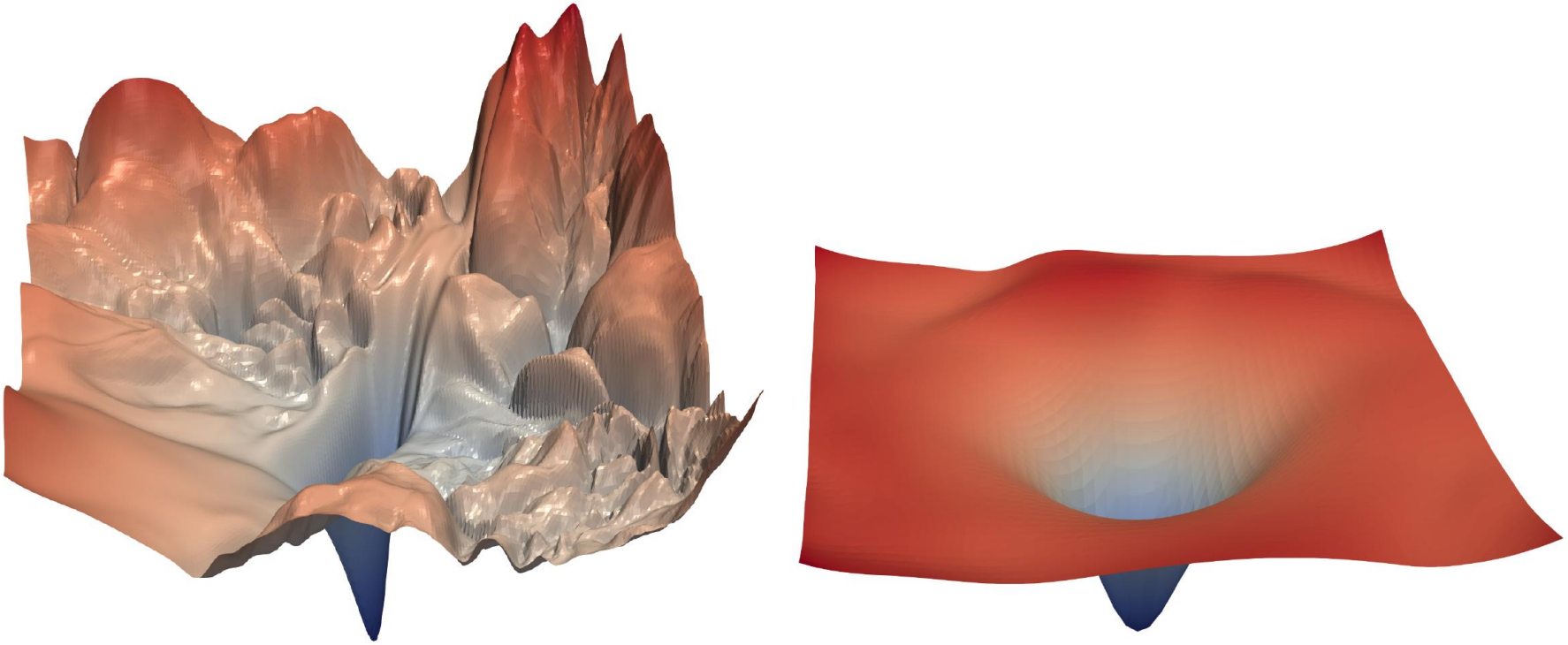
Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein, "Visualizing the Loss Landscape of Neural Nets", arXiv 2017

Skip Connection

ICLR 2018 reject

選兩個參數去visualize他的變化情況

network 要很深才能觀察到這個現象



(a) without skip connections

50 layers

(b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The vertical axis is logarithmic to show dynamic range. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Reference

- Ian J. Goodfellow, Oriol Vinyals, Andrew M. Saxe, “Qualitatively characterizing neural network optimization problems”, ICLR 2015
- Daniel Jiwoong Im, Michael Tao, Kristin Branson, “An Empirical Analysis of Deep Network Loss Surfaces”, arXiv 2016
- Qianli Liao, Tomaso Poggio, “Theory II: Landscape of the Empirical Risk in Deep Learning”, arXiv 2017
- Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein, “Visualizing the Loss Landscape of Neural Nets”, arXiv 2017

Concluding Remarks

Concluding Remarks

- Deep linear network is not convex, but all the local minima are global minima.
  - There are saddle points which are hard to escape
- Deep network has local minima.
  - We need more theory in the future
- Conjecture: 推測
  - When training a larger network, it is rare to meet local minima.
  - All local minima are almost as good as global
- We can try to understand the error surface by visualization.
  - The error surface is not as complexed as imagined.