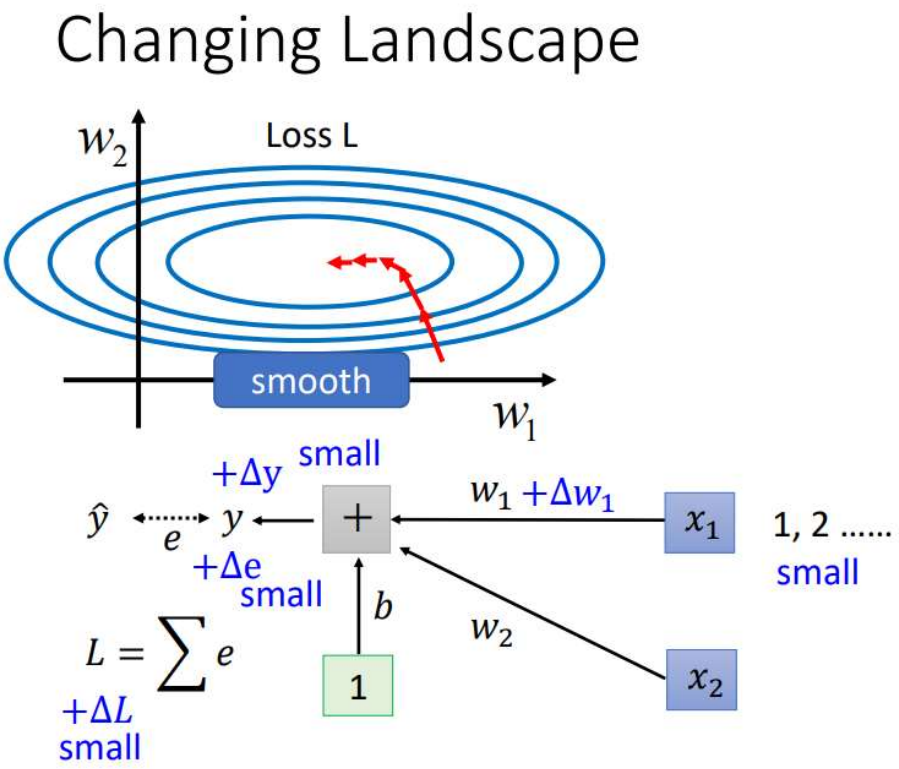
類神經網路訓練不起來怎麼辦 (五)： 批次標準化 (Batch Normalization) 簡介

Create at 2022/06/17

- 上課資源：
  1. 類神經網路訓練不起來怎麼辦 (五)： 批次標準化 (Batch Normalization) 簡介 (https://www.youtube.com/watch?v=BABPWOkSbLE)

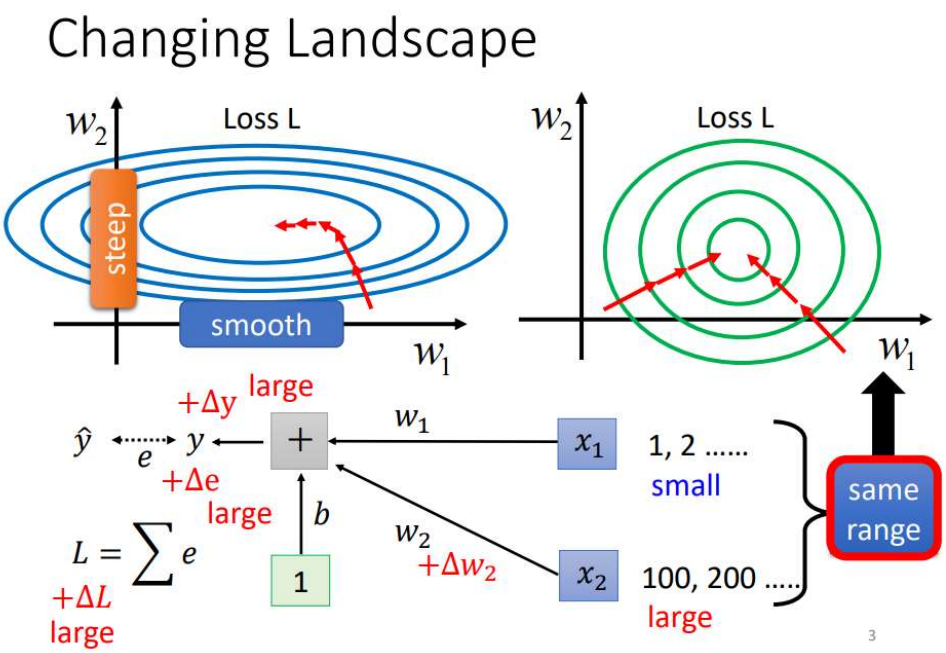
Quick Introduction of Batch Normalization



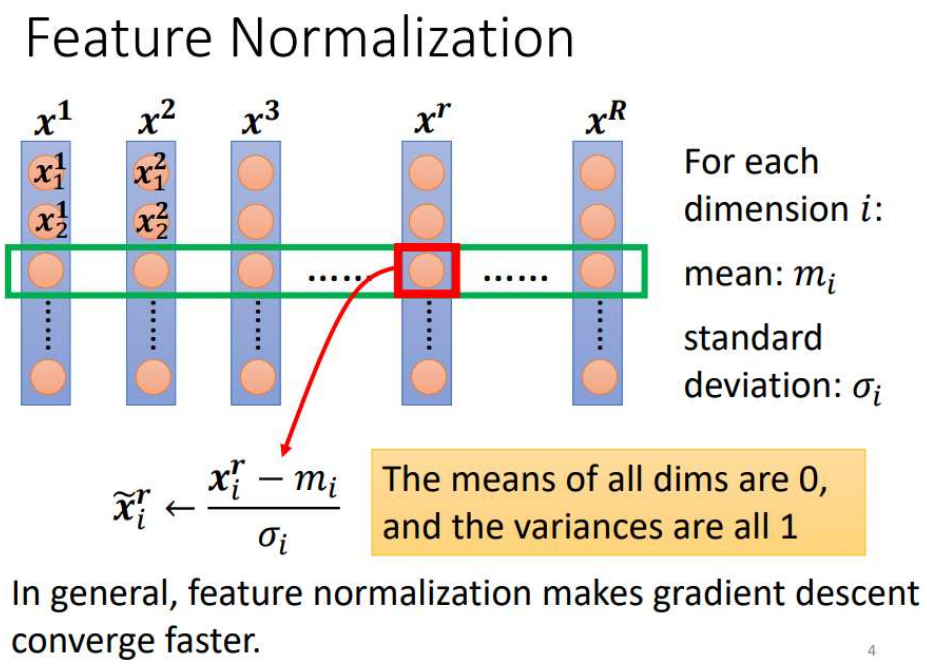
- error surface 崎嶇的時候比較難 train
  - 是否能直接把山剷平？
    - Batch Normalization 是其中一個把山剷平的想法
- 假設兩個參數，它們對 Loss 的斜率差別非常大
  - 在 W_1 方向上，斜率變化很小
  - 在 W_2 方向上，斜率變化比較大
- 如果是固定的 learning rate 很難得到好的結果
- 所以需要很 adaptive 的 learning rate
- 需要用 Adam 等等比較進階的 optimization 方法

- 換一個方向想
  - 直接把難做的 error surface 改掉，看能不能好做一點
- W_1、W_2 斜率差很多的狀況是怎麼來的？
  - 舉例：現在有一個非常簡單的 model
    - 輸入：x_1、x_2，分別對應的參數是 w_1、w_2
    - 是 linear model 沒有 activation function
    - $x_1w_1 + x_2w_2 + b = y$
    - 計算 y 跟 $\hat{y}$ 之間的差距，得到 e
    - 把所有 training data 的 e 加起來，就會得到 $Loss$

- 當 w_1 有小改變時，$w_1 + \Delta w_1$
  - 所以 y 也會改變，$y + \Delta y$
  - L 也會改變，$L + \Delta L$
- 什麼時候 w_1 的改變會對 L 的影響很小呢？
  - 當 x_1 input 很小的時候
- 假設 x_2 input 的值都很大
  - 當 w_2 有小變化的時候，y、e、L 的變化就會很大

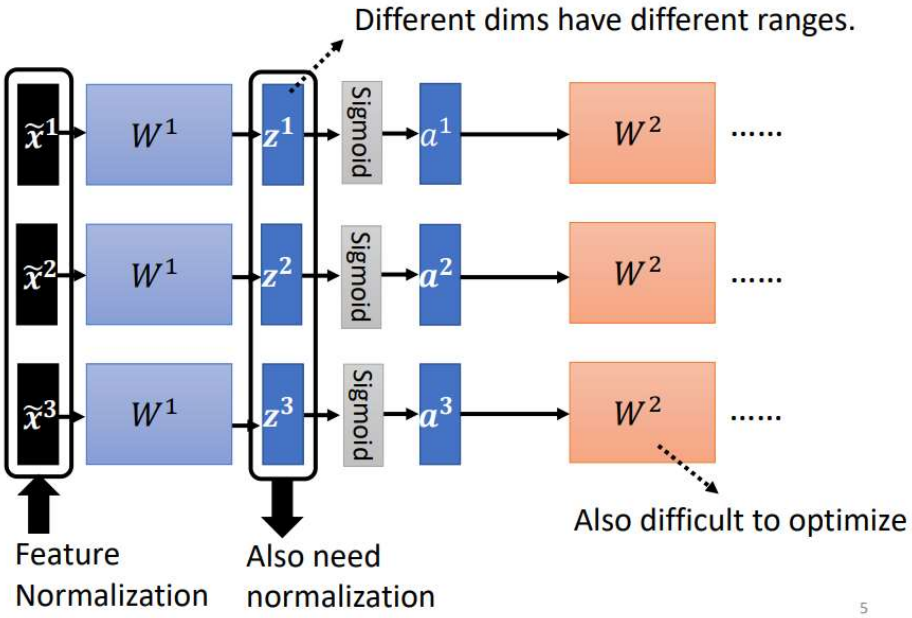


- 觀察到在 linear model 裡面
  - input 的 feature 每一個 dimension 值 scale 差距很大的時候，可能產生像這樣的 error surface，可能產生不同方向斜率非常不同的 error surface
- 解決辦法：
  - 給不同的 dimension 讓它有同樣的數值範圍
  - 可能就可以製造比較好的 error surface，讓 training 變得比較容易一些



- 如何讓不同的 dimension 有同樣的數值範圍
  - 方法統稱為 Feature Normalization
- 假設 x^1、x^2、x^3... x^R 是我們所有訓練資料的 feature vector
- 把不同 feature vector 的同一個 dimension 裡面的數值取出來
- 接著計算某一個 dimension 裡面的 mean
- 做完 normalize (standard) 之後，得到 $\tilde{x}$，dimension 上面的數值平均為 0，variance 是 1
- 把每一個 dimension 都做一樣的事情
- 所以所有 feature 不同 dimension 的數值都在 0 上下，可能就可以製造比較好的 error surface，可以在做 gradient descent 的時候 Loss 收斂更快一些

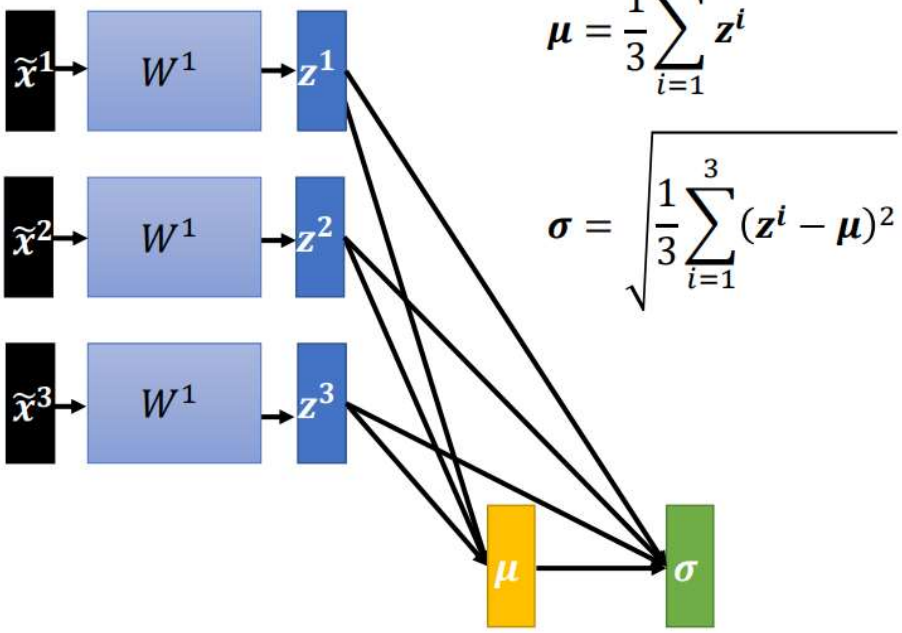
Considering Deep Learning



- $\tilde{x}$ 代表 normalize 的 feature 之後丟到 deep network 裡面去做接下來的計算
- $\tilde{x}^1$ 通過第一個 layer 得到 z^1，通過 activation function (Sigmoid / ReLU)，再得到 a^1，再通過下一層等等...
- 每一個 $\tilde{x}$ 都做一樣的事情

- 對 W^2 來說 a^1、a^3、z^1、z^3 也是另外一種 input，因為不同 dimension 有不同的 range
- z (a 也可以，activation function 前後做 normalization 差異不大，但比較推薦對 z 做) 需做 normalization，否則 W^2 一樣難 training

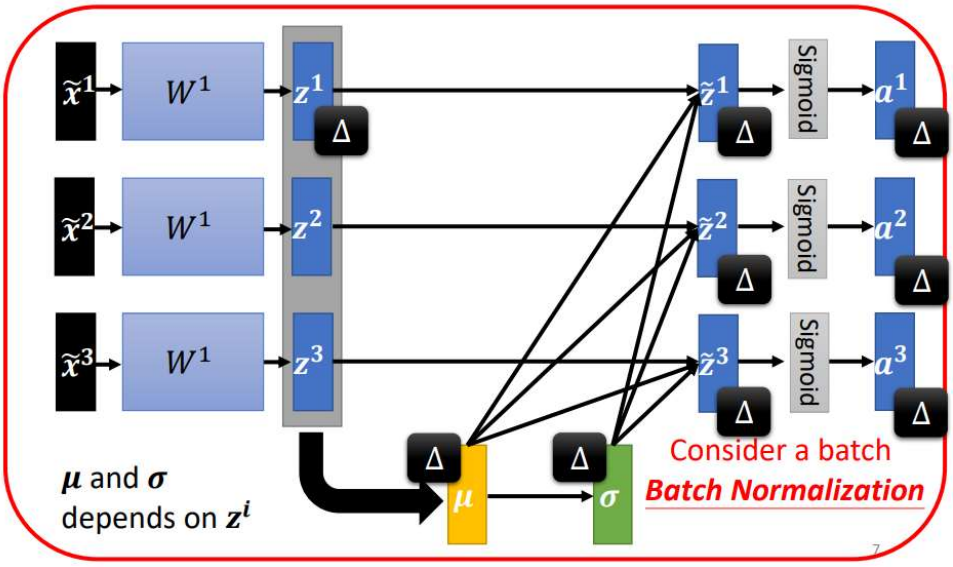
Considering Deep Learning



Considering Deep Learning

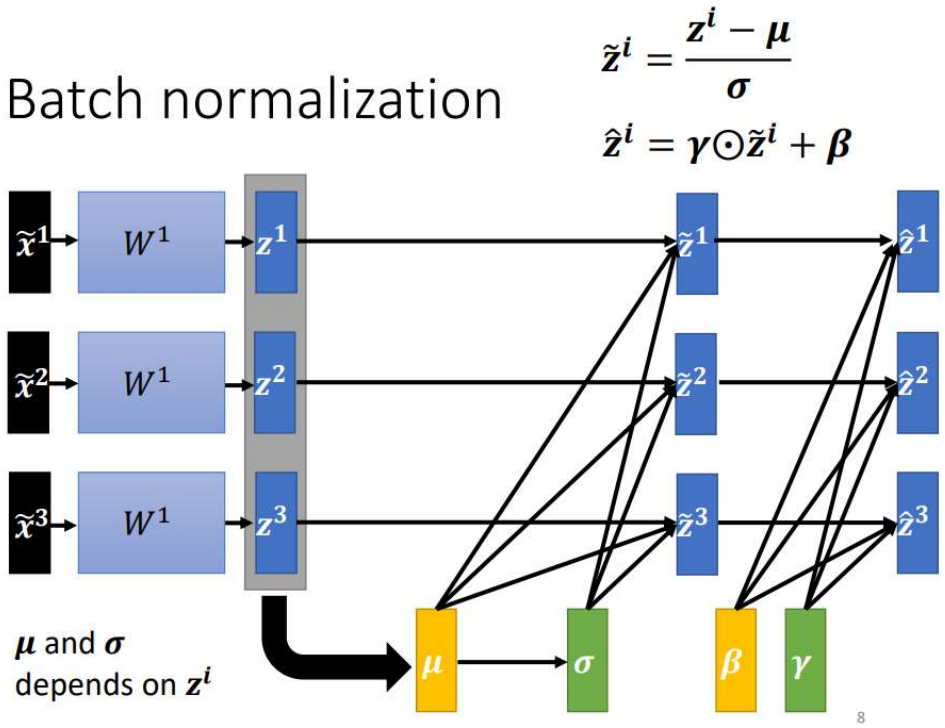
This is a large network!

$$\tilde{z}^i = \frac{z^i - \mu}{\sigma}$$



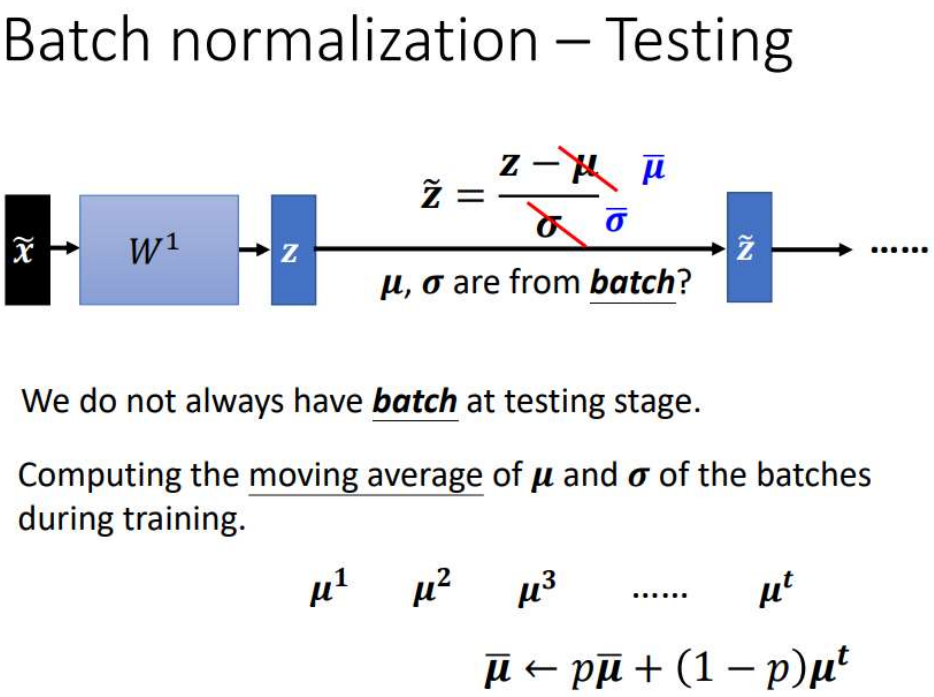
- 對 z 做 feature normalization
  - 每一個 dimension 算出 μ、σ
  - $\frac{z^1-\mu}{\sigma}=\tilde{z}^1$
  - $\frac{z^2-\mu}{\sigma}=\tilde{z}^2$
  - $\frac{z^3-\mu}{\sigma}=\tilde{z}^3$
  - $\tilde{z}^1$、$\tilde{z}^2$、$\tilde{z}^3$ 通過 activation function 得到其他 vector，再去通過其他 layer 等等
- 在沒有做 feature normalization 的時候
  - 當改變 z^1，只會改變 $\tilde{z}^1$、a^1
- 但現在改變 z^1，會改變 μ、σ
  - 改變 μ、σ，會跟著改變 $\tilde{z}^1$、a^1、$\tilde{z}^2$、a^2、$\tilde{z}^3$、a^3
- 當有做 feature normalization 的時候，要把整個 process 當作一個 large network
- 在實作的時候不會讓 network 考慮整個 training data 裡面的所有 example，只會考慮一個 batch 裡面的 example
  - 若 batch 設 64，network 就是把 64 筆 data 讀進去，算 64 筆 data 的 μ、σ
  - 對這 64 筆 data 做 normalization
  - 這招稱為 **Batch Normalization**
- 問題：
  - 要有一個夠大的 Batch，才算得出 μ、σ

Batch normalization

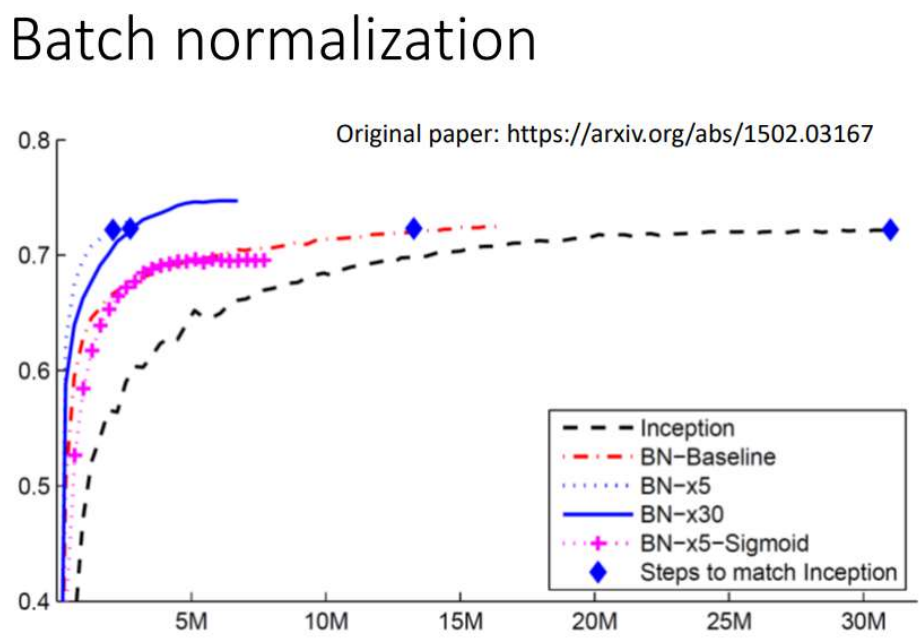


- 通常會把 $\tilde{z}^i$ 乘上一個向量 γ 加上向量 β 得到 $\hat{z}^i$
  - β、γ 想成是 network 的參數

Batch normalization - Testing



- 怎麼解決不會等每筆資料都讀進來之後，才開始做運算的問題 (怎麼先取得 μ、σ) ?
  - 在 training 的時候，每一個 batch 計算出來的 μ、σ，都會拿出來做 moving average，來更新 μ、σ
  - 在 testing 的時候，就不用算 batch 裡面的 μ、σ，直接拿 $\bar{\mu}$、$\bar{\sigma}$ 來取代這邊的 μ、σ



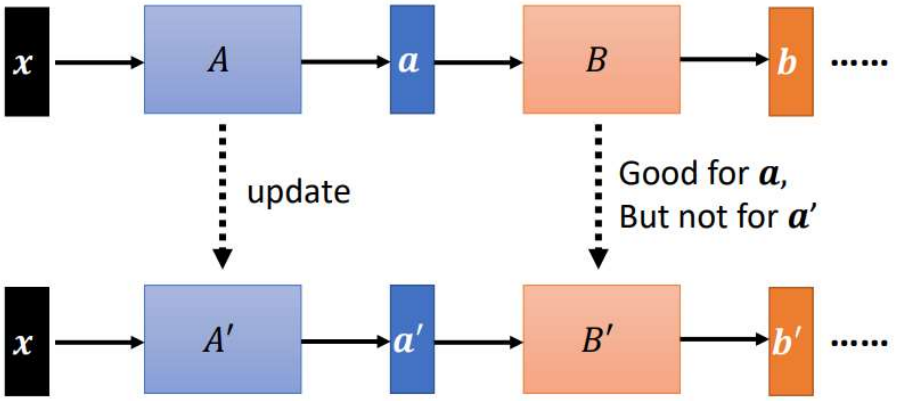
- 橫軸：訓練的過程
- 縱軸：validation set 上面的 accuracy

- 黑色虛線：沒有做 batch normalization 的結果，是用 inception network (CNN-based)
- 紅色虛線：用 batch normalization，訓練速度比黑色虛線快很多
- 粉紅線：sigmoid function + Batch normalization

Internal Covariate Shift?

How Does Batch Normalization Help Optimization?

https://arxiv.org/abs/1805.11604



Batch normalization make a and a' have similar statistics.

Experimental results do not support the above idea.

11

- Batch normalization 為什麼有幫助呢？
  - 可能會讓 a 與 a' 比較接近

Internal Covariate Shift?

How Does Batch Normalization Help Optimization?

https://arxiv.org/abs/1805.11604

Experimental results (and theoretically analysis) support batch normalization change the landscape of error surface.

and 12 of Appendix B.) This suggests that the positive impact of BatchNorm on training might be somewhat serendipitous. Therefore, it might be valuable to perform a principled exploration of the design space of normalization schemes as it can lead to better performance.

serendipitous（偶然的）

penicillin

- Batch normalization 為什麼有幫助呢？

- 其他 Normalization 的方法

tags: 2022 李宏毅_機器學習