Day 14

深度學習與電腦視覺

學習馬拉松

cupoy 陪跑專家：楊哲寧

CUPOY

深度學習理論與實作

CNN原理：BatchNormalization

重要知識點

CUPOY

- 理解 Batch Normalization 原理。
- Batch Normalization 用來解決什麼問題。

BN (Batch Normalization)

CUPOY

Batch Normalization 是 2015 年 Google 研究員在論文《Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift》一文中提出的，並同時將 BN 應用於 Inception-v2 的框架中。

BN解決的問題

CUPOY

- **神經網路深度**：近期的 CNN 結構有越疊越深的趨勢，然而更深的網路往往伴隨著難以收斂的問題，透過 BN 歸一化輸入資料，使其分佈更穩定，進而加速收斂。

- **梯度消失**：過去常用 Sigmoid 作為激勵函數，然而 Sigmoid 容易造成梯度消失，主要是由於 Sigmoid 的導數最大值為 0.25，使用 BN 再使用 Sigmoid 函數能有效降低梯度消失的可能性。

- **正則化**：常見降低 Overfitting 的方式包含 Dropout 層與 L1、L2 正則，而BN也能達到一定的正則化效果。

BN算法

首先計算輸入 Batch 的平均值與標準差

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$ 平均值 // mini-batch mean

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$$ 標準差 // mini-batch variance

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$ // normalize

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$$ // scale and shift

BN算法

CUPOY

透過平均值與標準差將輸入資料歸一化

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

https://arxiv.org/pdf/1502.03167.pdf

BN算法

透過學習 Gamma 與 Beta 做縮放與平移， Gamma 與 Beta 為 BN 層內唯二需要學習的參數。

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

資料分佈

CUPOY

一般來說我們都是以 **Mini Batch的方式訓練資料**，然而每一個 Batch 間的**資料分佈**可能不太相同，而輸入每一層神經元的資訊分布也都可能會改變，造成收斂上的困難。

資料分佈

CUPOY

透過 BN，將每一層輸入資料的分佈歸一化為**平均值為0**，**方差為1**，確保資料分佈的穩定性。

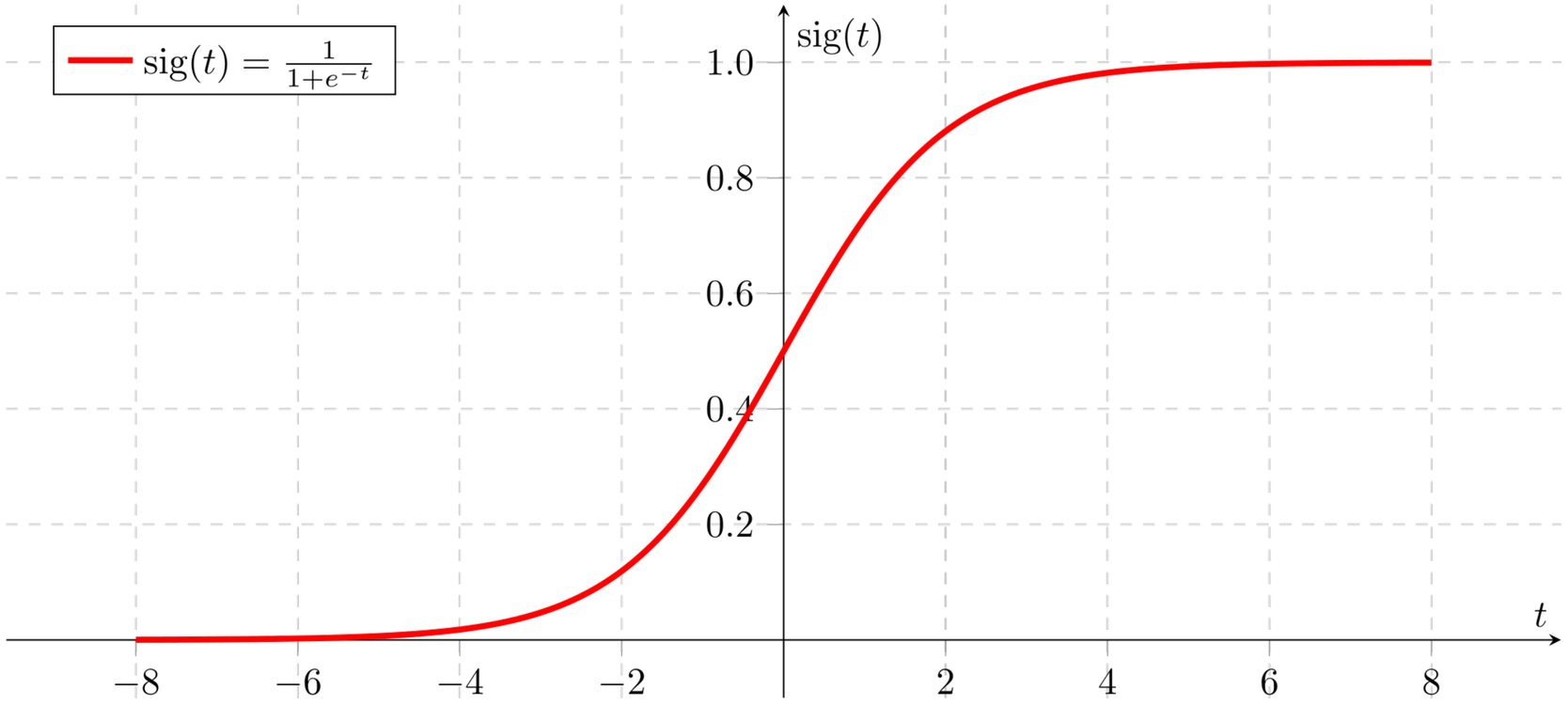
資料分佈

CUPOY

然而 Normalize 改變資料的分佈，可能會造成**上一層學到的特徵消失**，因此 BN 的最後一步透過學習 Beta、Gamma，去微調 Normalize 後資料的分佈 (參考資料)。

梯度消失

- Sigmoid會將數值較大與較小的值通通壓在一起，並且由於其導函數最大值為0.25，容易發生**梯度消失**的情形，透過BN，我們將資料分布歸一化，能有效降低梯度消失的可能性。



參考來源： Derivative of the Sigmoid function

推薦延伸閱讀

CUPOY

Competitions Datasets Notebooks Discussion Courses ...

239 views · 22d ago

Tutorial – Covariate Shift –Sberbank Housing EDA

Introduction

You may have heard from various people that data science competitions are a good way to learn data science, but they are not as useful in solving real world data science problems. Why do you think this is the case?

One of the differences lies in the quality of data that has been provided. In Data Science Competitions, the datasets are carefully curated. Usually, a single large dataset is split into train and test file. So, most of the times the train and test have been generated from the same distribution.

But this is not the case when dealing with real world problems, especially when the data has been collected over a long period of time. In such cases, there may be multiple variables / environment changes might have happened during that period. If proper care is not taken then, the training dataset cannot be used to predict anything about the test dataset in a usable manner.

In this kernel, we will see the different types of problems or Dataset Shift that we might encounter in the real world. Specifically, we will be talking in detail about one particular kind of shift in the Dataset (**Covariate shift**), the existing methods to deal with this kind of shift and an in depth demonstration of a particular method to correct this shift.

Table of Contents

Covariate Shift 解釋

連結

Competitions Datasets Notebooks Discussion Courses ...

A

6,240 views · 22d ago

Tutorial – Covariate Shift –Sberbank Housing EDA

Introduction

You may have heard from various people that data science competitions are a good way to learn data science, but they are not as useful in solving real world data science problems. Why do you think this is the case?

One of the differences lies in the quality of data that has been provided. In Data Science Competitions, the datasets are carefully curated. Usually, a single large dataset is split into train and test file. So, most of the times the train and test have been generated from the same distribution.

But this is not the case when dealing with real world problems, especially when the data has been collected over a long period of time. In such cases, there may be multiple variables / environment changes might have happened during that period. If proper care is not taken then, the training dataset cannot be used to predict anything about the test dataset in a usable manner.

In this kernel, we will see the different types of problems or Dataset Shift that we might encounter in the real world. Specifically, we will be talking in detail about one particular kind of shift in the Dataset (**Covariate shift**), the existing methods to deal with this kind of shift and an in depth demonstration of a particular method to correct this shift.

Table of Contents

- 1. What is Dataset Shift?
- 2. What causes Dataset Shift?
- 3. Types of Dataset Shift
- 4. Covariate Shift
- 5. Identification
- 6. Treatment
  - 6.1 Dropping of drifting features
  - 6.2 Importance Weight using Density Ratio Estimation

Batch Normalization 的用途

連結

Gradient Vanishing Problem — 以 ReLU / Maxout 取代 Sigmoid actvation function

tags: 李宏毅 Maching Learning

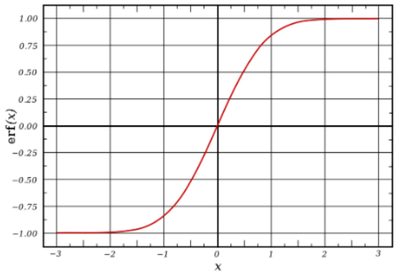
> - 本文內容參考自Hung-yi Lee , Machine Learning(2017) 課程內容 : Tips for Training DNN
> - 本文圖片部分來自於課程講義內容

梯度消失 Gradient Vanish

「類似」於 Sigmoid function 的激勵函數，普遍帶有梯度消失 (Gradient Vanish) 的隱憂，那究竟什麼是梯度消失?

$$Sigmoid\ function = \theta(s) = \frac{1}{1+e^{-s}}$$

此函數圖形為



(圖片取自： Wikipedia — Sigmoid function)

從圖上可知，其圖形切線斜率 (導數) 不會超過0.25，如此情況當我們在進行 Gradient Descent 的過程中，隨著迭代次數的增加，參數的更新會越來越緩慢 而整個 train 不起來。[1]

Sigmoid梯度消失原因與解決

連結

解題時間 Let’s Crack It

CUPOY

請跳出 PDF 至官網 Sample Code & 作業開始解題