

何時給作業: 12/10, 2018 何時交作業: 01/4, 2019

遲交不收, 抄襲 0 分。

- (a) Given $f(i \Delta t) = 0.1, i=1, \dots, 4$, and $g(i \Delta t) = 0.1, i=1, \dots, 4, \Delta t = 0.1$ second, find 1D convolutional result of f and g .
Sequence of samples of $f = \{0.1, 0.1, 0.1, 0.1\}$ $\Delta t=0.1$ seconds.
Sequence of samples of $g = \{0.1, 0.1, 0.1, 0.1\}$ $\Delta t=0.1$ seconds.
(Use toolbox and Self-programming)
- (b) Given $f(i \Delta x, j \Delta y) = 0.1, i=1, \dots, 4, j=1, \dots, 4$, and $g(i \Delta x, j \Delta y) = 0.1, i=1, \dots, 4, j=1, \dots, 4, \Delta x=0.1, \Delta y=0.1$ find 2D convolutional result of f and g .
(Use toolbox and Self-programming)
- (c) 利用 Matlab 的 Convolutional neural network (CNN) 的 toolbox 做實驗。

CNN 有兩個主要的部分, 第一部分為 convolution layers 做 feature extraction, 第二部分為 classification layers, 即為多層的 perceptrons (MLP)。由於有很多層需要學習, 所以稱為 deep learning。

訓練影像是 MNIST Image Dataset (提供了 0~9 手寫數字的灰階影像集), 使用 Matlab 的 Neural Network Toolbox 提供的 Convolutional neural network 的功能, 去建立並訓練 CNN classification model:

<https://www.mathworks.com/help/nnet/examples/create-simple-deep-learning-network-for-classification.html>

- (1) 選取 1000 個 examples 做 training, 1000 個做 testing。
- (2) 全部的 60000 個 examples 做 training, 10000 個做 testing。

先找 Y. LeCun 的 papers, 及網頁上 Stanford 大學的研究成果。
參考文獻要寫出來。

*報告要用中文寫, 不要抄英文的, 會的, 才寫, 重點步驟要寫得詳細!!!

***可各個問題的 program 分開做。

Outputs:

- (1) Flowchart of programming.
- (2) Plot the average error vs. iteration, CPU time in learning.
- (3) 輸出各層抽取的特徵圖。
- (4) 輸出 150 個測試的圖形, 放成 15x10 的排列, 每一個圖形的下面要標示 desired class, 最後辨認的 class, 對或錯。
- (5) 辨認結果的 10x10 的 confusion matrix, i.e., table of correction and error rate.

要交的東西:

- (1) 在指定日的上課前, 交紙本報告 (包含敘述如何做, flowchart, 結果, 討論, 參考文獻, 及 Matlab programs).

(2) 將要交紙本報告的 doc file 及分開的 MATLAB program file 建成一個 directory (資料夾)，壓縮後，上傳到 e3 system.

Directory name 的名稱: Proj#3_姓名_PR_2018_Fall。

(做一個 presentation。)