

Báo cáo giữa kỳ Machine Learning

1. Introduction

- Handwritten Digit Recognition using Convolutional Neural Networks
- Limitation:
 - Đề tài: nhận diện chữ số từ 1 đến 10, sử dụng data set MNIST để training bộ dữ liệu.
 - Kiến thức: áp dụng những kiến thức cơ bản của Neural Networks, kiến thức của một số thuật toán, activation function, cost functions để phục vụ cho bài toán nhận dạng chữ số viết.
 - Sản phẩm: sản phẩm là chương trình được viết bằng ngôn ngữ lập trình Python và chạy trên trình soạn code Visual Studio 2017. Sử dụng bộ dữ liệu có sẵn MNIST để chạy thử nghiệm nhận diện chữ số.

2. Reason and Why do you choose this topic?

Việc lựa chọn đề tài dựa trên 2 yếu tố chính

2.1 Scope của đề tài :

Do thời gian chỉ giới hạn trong một học kỳ và phải tiếp thu nhiều kiến thức mới nên nhóm quyết định chọn một đề tài vừa sức, phù hợp với thời gian và độ phức tạp nhưng vẫn có thể áp dụng những kiến thức cơ bản đã học và tự tìm hiểu những kiến thức mới. Với đề tài Handwritten Digit Recognition không quá phức tạp về mặt nghiệp vụ nên nhóm sẽ có nhiều thời gian hơn để chú trọng vào lý thuyết và tìm hiểu kiến thức kỹ những kiến thức mới để áp dụng vào bài toán

2.2 Đề tài có thể áp dụng với Convolutional Neural Network

Convolutional Neural Network là kiến thức mà nhóm có thể áp dụng cho đề tài, đây cũng là nền tảng để tiếp cận với kiến thức sâu hơn về artificial intelligence như Deep Learning nên nhóm sẵn có củng cố kiến thức và tự tìm hiểu thêm về mảng kiến thức mới này.

3. Meaning ? When u finish this topic how does result you achieve?

- Hiểu được khái quát mối quan hệ giữa kiến thức rời rạc để hoàn thành được đề tài
- Hiểu được bản chất của các thuật toán: ưu điểm và nhược điểm, khi nào và áp dụng như thế nào, liên

- Rèn luyện khả năng đọc hiểu và áp dụng những kiến thức mới vào đề tài.
- Hiểu và áp dụng a simple Convolutional Neural Network, một vài activation function dùng cho bài toán,
- Có kiến thức về ngôn ngữ lập trình Python và tìm hiểu các thư viện
- Practice được những kiến thức tự tìm hiểu và là cơ sở để tìm hiểu sâu về lĩnh vực artificial intelligence, data analyst

4. Goal

- Xây dựng một chương trình nhận diện được chữ số viết tay, nguồn dữ liệu được cung cấp từ MNIST

5. Objectives

Bước 1: Tìm hiểu về bộ dữ liệu MNIST

Bước 2: Tìm hiểu về các phương pháp để biến dữ liệu thô thành dữ liệu cho quá trình training cụ thể là CNN model

Bước 3: Tìm hiểu về kiến thức để cho quá trình Training

- Neural Network
- Các kiến thức về Multiple-layer Perceptron
- Back-Propagation
- Activation Function: Sigmoid, Softmax, Rectified linear unit (ReLU)
- Cost function : Gradient Descent

Bước 4: Dựa vào những kiến thức tìm hiểu ở tìm ra phương pháp, quy trình hoạt động và luồng xử lý của chương trình

- Neural Network sẽ hoạt động như thế nào trong hệ thống?
- Dữ liệu đầu vào, lớp xử lý và dữ liệu đầu ra như thế nào?
- Cách áp dụng activation function nào cho phù hợp và áp dụng khi nào vào bài toán?
- Tìm kiếm weight và bias như thế nào với cost function.?

Bước 5: Tìm hiểu về quá trình testing phase.

Bước 6: Áp dụng lý thuyết vào chương trình nhận diện chữ viết số.

6. Theory

6.1 Dữ liệu training MNIST

MNIST là bộ dữ liệu là các con số viết tay từ 0 đến 9. Bộ dữ liệu này bao gồm 60.000 mẫu cho huấn luyện và 10.000 mẫu để kiểm thử. Các mẫu dữ liệu trong bộ MNIST đã được chuẩn hóa về kích thước: căn chỉnh chính giữa dữ liệu, và mỗi

mẫu sẽ là một ảnh có kích thước(shape) 28×28 pixel có nhãn là giá trị số của nó(0 đến 9).



Trong bài toán nhận diện chữ viết, MNIST sẽ được sử dụng cho bộ dữ liệu training và testing. Để đơn giản hóa thì mỗi mẫu dữ liệu(ảnh) sẽ được đưa về 1-D numpy array có 784 chiều($28 \times 28 = 784$).

6.2 CNN Model

6.3 Neural Network

6.4 Activation Function

6.4.1. Sigmoid function

Sigmoid neurons are similar to perceptrons, but modified so that small changes in their weights and bias cause only a small change in their output. That's the crucial fact which will allow a network of sigmoid neurons to learn.

Just like a perceptron, the sigmoid neuron has inputs, x_1, x_2, \dots . But instead of being just 0 or 1, these inputs can also take on any values *between* 0 and 1.

it's $\sigma(w \cdot x + b)$, where σ is called the *sigmoid function*

6.4.2. Softmax Function

6.4.3. Rectified Linear Unit Function (ReLU)

6.5 Cost Function

Là độ lệch sai nếu $C(w, b) \approx 0$, precisely thì khi đó $y(x)$ gần như sắp xỉ bằng với giá trị output. Do đó thuật training algorithm hoạt động tốt nếu thuật toán đó tìm được weights and biases để $C(w, b) \approx 0$. Ngược lại, khi $C(w, b)$ càng lớn,

điều đó có nghĩa rằng $y(x)$ có giá trị không sát với giá trị đúng thực sự và có độ sai số cao. Vì vậy cần làm tối thiểu chi phí mất mát thấp nhất có thể (quadratic cost), chúng ta có thể sử dụng để kiểm tra lại độ chính xác của phân loại.

Ta có thể tìm bộ weights và biases với chi phí mất mát càng nhỏ nhất có thể bằng cách sử dụng thuật toán *gradient descent*. Mục đích để training cho neural network có thể tìm ra được weights và biases nó nhỏ nhất với hàm bình phương $C(w,b)$.

7.UML

7.1 Block Diagram

7.2 Flow chart