**Báo cáo giữa kỳ Machine Learning**

# **Introduction – Giới thiệu**

* 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 lâp 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.

# **Reason – Lý do**

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 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 củng cố kiến thức và tự tìm hiểu thêm về mảng kiến thức mới này.

# **3. Meaning – Ý Nghĩa**

* Nắm được khái quát mối quan hệ giữa kiến thức rời rạt 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.
* 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 kiên thức về ngon ngữ lập trình Python và tìm hiểu các thư viện
* Prastice được những kiến thức tự tìm hiểu và là cơ sở để tìm hiểu sâu về lĩnh vực artificial intelligence, data analyst

# **Goal – Kết quả**

 Xây dựng được chương tình nhận diện được chữ số viết tay đơn giản.

# **Objectives – Các mục tiêu cần đạt được**

## 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 đã kiến thức tìm hiểu ở tìm ra phương pháp, quy trình hoạt động và luông xử lý của chương trình

* Neutrak 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.

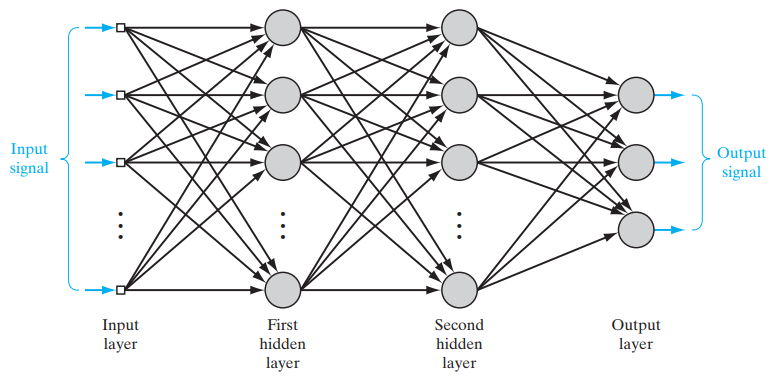
# **Theory – Cơ sở lý thuyết**

6.1. Neutral Network

Dựa trên mô hình liên kết giữa các tế bào thần kinh trong não bộ, mô hình neural network là tập hợp các đơn vị kết nối gọi là neuron, các neuron liên kết truyền tín hiệu cho nhau.

Neutral Network được phát triển dựa trên nền tảng Perceptron Learning Algorithm (PLA) hay còn gọi là Perceptron.

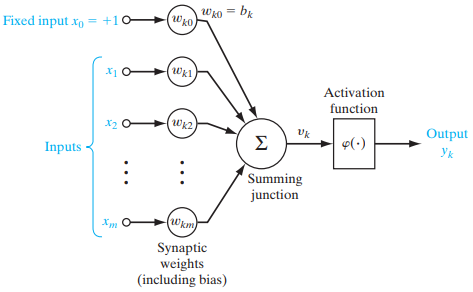
Neural network có cấu trúc Multi-layer Perceptrons. Perceptron có thể hiểu tương đương single-layer neural network.



Hình: Neural network structure

Neuron là một đơn vị xử lý thông tin cơ bản cho hoạt động của neural network

Một neuron trong bất kỳ layer nào trong network đều được kết nối với tất cả các neuron (node) trong layer trước đó – *fully connected*.



Hình : Mô hình non-linear của một neuron, có nhãn k

Phương trình viết lại từ mô hình:

và

Với

* : input
* : trọng số tương ứng của neuron k
* : bias
* : activation function
* : output

6.1.1. Activation Function

Mục đích: giới hạn phạm vi biên độ cho phép của tín hiệu đầu ra tới một số giá trị hữu hạn.

Một số hàm activation thường dùng : Sigmoid, Rectified linear unit (ReLU), Softmax

6.1.1.1. Sigmoid function

Sigmoid function là một logistic function.

Phạm vi (0, 1). Do đó, nó đặc biệt được sử dụng cho các mô hình mà phải dự đoán xác suất. Vì vậy xác suất của bất cứ điều gì chỉ tồn tại giữa phạm vi 0 và 1.

Áp dụng tốt nhất có bài toán classification với chỉ hai class.



Hình: Đồ thị của Sigmoid Function

6.1.1.2. Rectified linear unit (ReLU)

ReLU là activation function được sử dụng nhiều nhất trên thế giới hiện nay, nó được sử dụng trong Convolutional Neural Networks (CNN) hay Deep Learning.

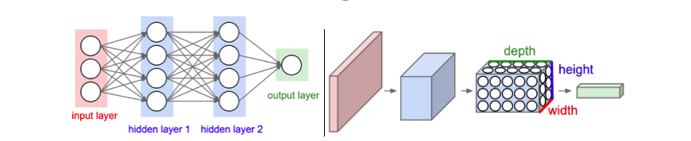
Phạm vi:

Với bất kỳ input có giá trị âm đưa vào cho hàm ReLU sẽ biến giá trị thành 0, do đó ảnh hưởng đến biểu đồ kết quả khi không ánh xạ các giá trị âm một cách thích hợp.

6.1.1.3. Softmax Function

6.2 Convolutional Neutral Network

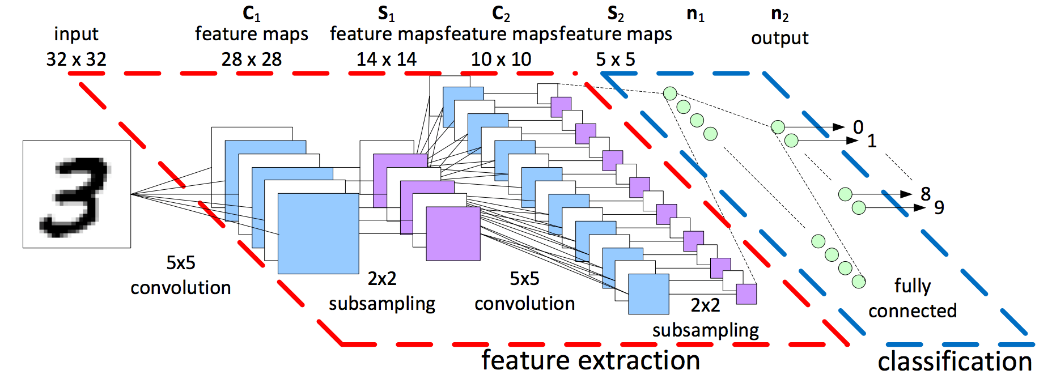
Convolutional Neural Network –CNN là một dạng Artificial Neural Network, giống như Neural Network truyền thống nhận input, biến đổi input qua hidden layer nhưng khác nhau ở chỗ các layer được tổ chức theo không gian 3 chiều và những neurons trong 1 layer không liên kết đến tất cả các neuron ở layer sau



Normal NN vs CNN. — Source: <http://cs231n.github.io/convolutional-networks/>

CNNs có 2 thành phần chính

2 ảnh đều bổ sung điểm thiếu cho nhau nên để 2 luôn ko

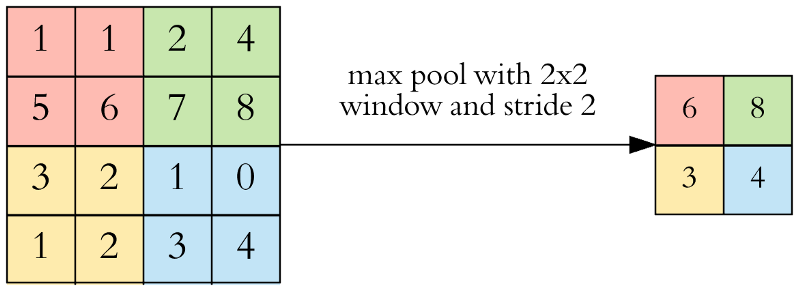




-Hidden layer/feature extraction gồm convolutional layers, pooling layers

* + Convolutional layer :
* Tập các feature map, mỗi feature map là 1 bản scan thu nhỏ đặc tính cụ thể của dữ liệu input
* Scan thì dựa vào convolution filter/kernel, là ma trận sẽ quét qua ma trận dữ liệu đầu vào, từ trái qua phải, trên xuống dưới.
* Kết quả đưa qua activation funciton (sigmoid, relu,…), kết quả sẽ là một con số cụ thể
* Tập hợp các con số này lại là 1 ma trận nữa, chính là **feture map**



* + Pooling : giảm số hyperparameter cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting, thường gặp nhất là max pooling, lấy giá trị lớn nhất trong một pooling window
  + 

- Classification : fully connected layers

Layer này cũng chính là 1 fully connected ANN (the traditional multi-layer perceptron neural network )

Nguồn

<https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>

<https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM>

https://en.wikipedia.org/wiki/Convolutional\_neural\_network

# 6.5 Cost Function

Là độ lệch sai nếu C(w,b)≈0, precisely thì khi đó y(x) gần như sấp 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)≈0. Ngược lại, khi C(w,b) càng lớn, điều đó có nghĩa rằng y(x) có trá 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 neutral network có thể tìm ra được weights và bises nó nhỏ nhất với hàm bình phương C(w,b).

# **7.UML**

## 7.1 Block Diagram

## 7.2  Flow chart