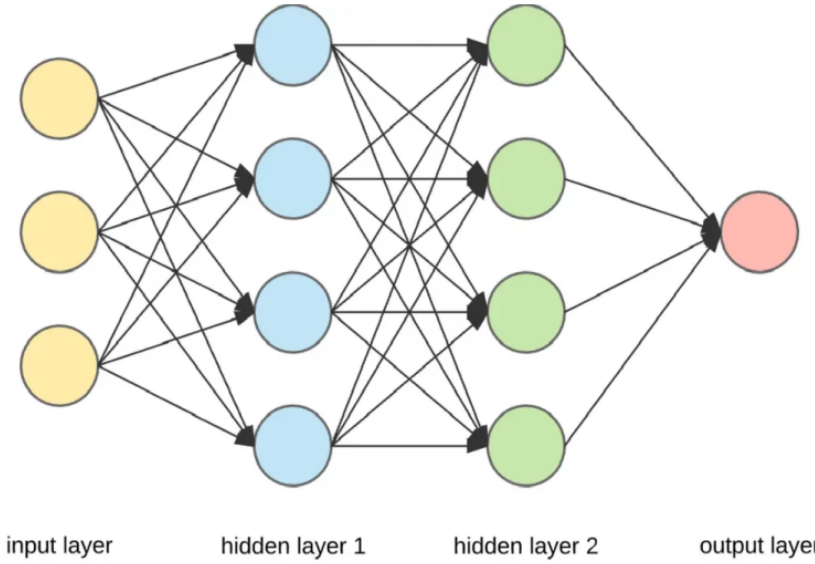
DS201 - BÀI THỰC HÀNH 1
XÂY DỰNG MẠNG NEURON ĐƠN GIẢN

Ngày 10 tháng 10 năm 2025

Mục lục

1. Ôn lại khái niệm cơ bản



Hình 1: Minh họa cấu trúc của một mạng neural cơ bản

1.1 Cấu trúc cơ bản

Layer đầu tiên là *input layer*, các layer ở giữa được gọi là *hidden layer*, layer cuối cùng được gọi là *output layer*. Các hình tròn được gọi là node.
Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không các hidden layer. Tổng số layer trong mô hình được quy ước là số layer – 1 (Không tính input layer).
Ví dụ như ở hình trên có 1 input layer, 2 hidden layer và 1 output layer. Số lượng layer của mô hình là 3 layer.
Mỗi node trong *hidden layer* và *output layer*:

- Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng.
- Mỗi node có 1 hệ số bias b riêng.

1.2 Hàm kích hoạt (Activation Function)

- ReLU: f(x)=max(0,x) thường dùng cho hidden layer.
- Softmax: Biến đầu ra thành xác suất cho các lớp, dùng cho output layer đa lớp.
- Sigmoid: thường dùng cho bài toán nhị phân.

1.3 Quy trình huấn luyện mạng neuron

- Forward pass (Lan truyền tiến): Tính toán đầu ra từ đầu vào qua các lớp bằng cách nhân ma trận trọng số và áp dụng hàm kích hoạt.
- Loss function: Đo độ sai lệch giữa dự đoán và thực tế (ví dụ: cross-entropy).
- Backward pass (Lan truyền ngược): Tính toán gradient để cập nhật trọng số, sử dụng thuật toán lan truyền ngược (backpropagation).
- Optimizer: Thuật toán tối ưu hóa như SGD, Adam để cập nhật trọng số.

2. Dataset MNIST

2.1 Giới thiệu về MNIST

Trong bài thực hành này, chúng ta sẽ sử dụng bộ dữ liệu MNIST. Đây là bộ dữ liệu sử dụng cho bài toán nhận dạng chữ số viết tay. MNIST gồm 70,000 ảnh số viết tay (0-9), mỗi ảnh kích thước 28x28 pixels, chia thành tập train (60,000 ảnh) và test (10,000 ảnh). Ảnh đen trắng, mỗi pixel có giá trị từ 0 đến 255.

Hình 2: Minh họa bộ dữ liệu MNIST

2.2 Khám phá dữ liệu

Tải dữ liệu:

1. Download trực tiếp: http://yann.lecun.com/exdb/mnist/
2. Load qua thư viện Keras cung cấp (bao gồm tập train và tập test), sử dụng câu lệnh sau:

```
from keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Chuẩn bị dữ liệu:

- Sử dụng thư viện Matplotlib để trực quan hóa ảnh train[0] và ảnh test[0]

```
import matplotlib.pyplot as plt

plt.imshow(X_train[0], cmap='gray')
plt.title(f'Label: {y_train[0]}')
plt.show()
```

- Xem kích thước hai tập train, test (sử dụng thuộc tính shape)
- Đếm số lượng ảnh mỗi lớp trong tập train/test để kiểm tra phân bố dữ liệu
- Sử dụng thư viện Matplotlib để vẽ các biểu đồ sau:
  - Biểu đồ cột thể hiện sự phân bố các nhãn có trong tập train
  - Biểu đồ tròn thể hiện tỉ lệ các nhãn có trong tập train

3. Bài tập thực hành

3.1 Bài 1: Xây dựng mô hình 1-layer MLP

- Sử dụng hàm Softmax làm activation function cho output layer
- Huấn luyện mô hình với SGD optimizer
- Đánh giá mô hình trên các độ đo: accuracy, precision, recall, F1-macro
- Đánh giá kết quả với từng chữ số (0-9)

3.2 Bài 2: Xây dựng mô hình 3-layer MLP

- Hai layer đầu dùng ReLU, layer cuối dùng Softmax
- Huấn luyện mô hình với SGD optimizer
- Đánh giá mô hình trên các độ đo: accuracy, precision, recall, F1-macro
- Đánh giá kết quả với từng chữ số (0-9)