

Link github: [https://github.com/rotsutc/DEEP530517\\_Deep\\_Learning.git](https://github.com/rotsutc/DEEP530517_Deep_Learning.git)

## 1. Xây dựng một kiến trúc Deep Learning

Sử dụng CNN (mạng nơ-ron tích chập) sử dụng thư viện TensorFlow để giải quyết bài toán phân loại hình ảnh.

Kiến trúc mạng CNN là một mạng sequential (tuần tự), được xây dựng từ 4 khối tích chập (Conv2D) + hợp nhất (MaxPooling2D) và kết thúc bằng các lớp Fully-Connected (Dense) để phân loại.

| Lớp        | Loại lớp     | Tham số quan trọng       | Số nơ-ron/đầu ra   | Hàm kích hoạt |
|------------|--------------|--------------------------|--------------------|---------------|
| Đầu vào    | Input        | (32, 32, 3)              | 32x32x3            | -             |
| Lớp 1      | Conv2D       | 32 bộ lọc, kernel (3,3)  | 32x32x32           | relu          |
|            | MaxPooling2D | (2,2)                    | 16x16x32           | -             |
| Lớp 2      | Conv2D       | 64 bộ lọc, kernel (3,3)  | 16x16x64           | relu          |
|            | MaxPooling2D | (2,2)                    | 8x8x64             | -             |
| Lớp 3      | Conv2D       | 128 bộ lọc, kernel (3,3) | 8x8x128            | relu          |
|            | MaxPooling2D | (2,2)                    | 4x4x128            | -             |
| Lớp 4      | Conv2D       | 128 bộ lọc, kernel (3,3) | 4x4x128            | relu          |
|            | Flatten      |                          | 2048 (4x4x128)     | -             |
| Lớp ẩn     | Dense        |                          | 256 nơ-ron         | relu          |
| Lớp đầu ra | Dense        |                          | num_classes nơ-ron | softmax       |

Thuật toán tối ưu hóa (Optimizer): Adam

Hàm mất mát: categorical\_crossentropy

Chỉ số đánh giá: accuracy

Epochs = 10 (*Do thời gian chạy trên máy cá nhân lâu nên em thiết lập epochs nhỏ*)

Batch size = 128

## 2. Dữ liệu CIFAR100

Sử dụng bộ dữ liệu CIFAR-100, một bộ dữ liệu phổ biến trong thị trường học máy, chứa 100 lớp khác nhau, mỗi lớp có 600 hình ảnh (500 ảnh huấn luyện và 100 ảnh kiểm tra).

Dữ liệu sau khi tải về cần được chia thành hai phần:

- 80% dùng làm dữ liệu huấn luyện để mô hình học.

- 20% dùng làm dữ liệu kiểm tra để đánh giá hiệu suất của mô hình sau khi huấn luyện.

### 3. Kết quả

Các chỉ số Accuracy, Precision, Recall và F1-score đều thấp (khoảng 39% - 41%), cho thấy mô hình cần được cải thiện mạnh mẽ trong quá trình huấn luyện.

| STT | Đại lượng        | Nhận xét  |
|-----|------------------|---|
| 1   | Accuracy=0.3977  | Độ chính xác này rất thấp. Trong 100 lần dự đoán, mô hình chỉ đưa ra dự đoán đúng cho lớp có xác suất cao nhất chưa đến 40 lần. Đây là dấu hiệu rõ ràng cho thấy mô hình chưa học đủ đặc trưng hoặc quá đơn giản so với độ phức tạp của bài toán 100 lớp (CIFAR-100).   |
| 2   | Precision=0.4175 | Precision cao hơn một chút so với Accuracy và Recall. Điều này ngụ ý rằng, khi mô hình đưa ra một dự đoán tích cực (rằng một mẫu thuộc về một lớp cụ thể), thì dự đoán đó có độ tin cậy hơi cao hơn so với tỷ lệ phát hiện tổng thể. Tuy nhiên, mức 41% vẫn là thấp, nghĩa là phần lớn các dự đoán của mô hình là không đáng tin cậy. |
| 3   | Recall=0.3977    | Recall gần như tương đương với Accuracy. Chỉ số này cho thấy mô hình chỉ phát hiện được khoảng 40% các trường hợp tích cực thực tế (các nhãn đúng) trong tập dữ liệu.   |
| 4   | F1-score=0.3954  | F1-score là trung bình điều hòa của Precision và Recall. Vì cả hai chỉ số này đều thấp và gần bằng nhau, F1-score cũng ở mức thấp (dưới 40%). Chỉ số này củng cố kết luận rằng mô hình có hiệu suất tổng thể kém trong việc cân bằng giữa độ tin cậy và khả năng phát hiện.   |
| 5   | F1@Top-5=0.6929  | Đây là chỉ số đáng chú ý nhất. Sự chênh lệch lớn giữa F1-score (~39.5%) và F1@Top-5 (~69.3%) chỉ ra rằng:   |

|  |  |   |
|--|--|---|
|  |  | <ul style="list-style-type: none"><li>• Mặc dù mô hình gặp khó khăn trong việc xác định chính xác lớp mục tiêu là dự đoán số 1 (Top-1), nhưng nhận đúng thường xuyên nằm trong 5 dự đoán hàng đầu.</li><li>• Điều này ngụ ý rằng mô hình có khả năng thu hẹp phạm vi tìm kiếm một cách tương đối tốt, tức là nó có thể phân biệt được các siêu lớp (superclasses) hoặc các lớp có đặc trưng gần nhau, nhưng lại không đủ "tinh tế" để chọn ra dự đoán chính xác nhất (Top-1).</li></ul> |
|--|--|---|

#### 4. Đề xuất hướng cải thiện

Để đạt được mục tiêu độ chính xác cao hơn 90% như đề bài yêu cầu, vì thời gian huấn luyện quá lâu trên máy cá nhân nên em đề xuất các hướng cải thiện như sau:

- Tăng độ phức tạp của mô hình: Thêm các lớp, tăng số lượng bộ lọc, hoặc chuyển sang các kiến trúc mạnh mẽ hơn (như ResNet hoặc DenseNet).
- Kỹ thuật Regularization: Thêm các kỹ thuật như Dropout (đặc biệt trong lớp Dense) để ngăn chặn hiện tượng quá khớp (overfitting) do kiến trúc phức tạp hơn.
- Tăng Epochs và tối ưu hóa: Huấn luyện mô hình trong nhiều epochs hơn và thử nghiệm với các tham số tối ưu hóa như thay đổi tốc độ học - Learning Rate Schedule).
- Data Augmentation nâng cao: Sử dụng các kỹ thuật tăng cường dữ liệu mạnh mẽ hơn như Cutout, CutMix, hoặc Mixup để làm phong phú dữ liệu huấn luyện và cải thiện khả năng tổng quát hóa (generalization).