ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

THỊ GIÁC MÁY TÍNH

Exercise 02
CNN

Giảng viên hướng dẫn Thầy Võ Hoài Việt

Thầy Nguyễn Trọng Việt

Thầy Phạm Minh Hoàng

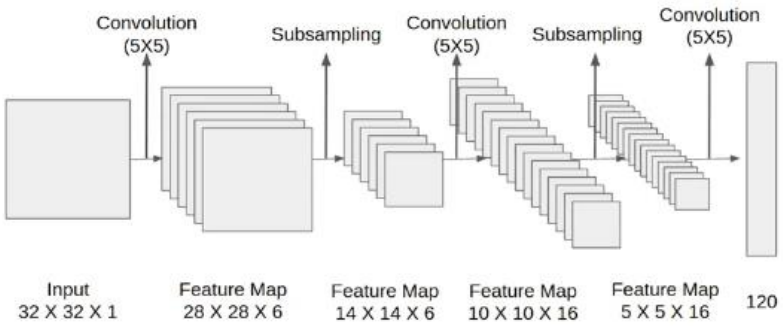
Sinh viên thực hiện Võ Nguyễn Hoàng Kim

21127090

Phụ lục

I. Mạng CNN Lenet-5

- Lenet-5 là mô hình mạng CNN (convolution neuron network) có 5 lớp với các tham số có thể học được.
- Mô hình gồm 3 lớp convolution, theo sau mỗi lớp convolution sẽ là các lớp average pooling.
- Sau các lớp convolution và average pooling sẽ đến hai lớp fully connected.
- Cuối cùng, bộ phân loại Softmax được sử dụng để phân loại hình ảnh thành lớp tương ứng.
- Dưới đây là hình ảnh trực quan về mô hình mạng CNN Lenet-5



II. Thực nghiệm

1. Xây dựng mã nguồn

- Chuẩn bị dữ liệu
  - Tập dữ liệu MNIST và MNIST Fashion được tải từ thư viện keras, được chia ngẫu nhiên thành các tập train và test, sau đó được thay đổi kích thước cũng như chuẩn hóa, biến đổi (tiền xử lý) để phù hợp với mô hình mạng.
- Xây dựng mô hình mạng với kích thước đầu vào là $(28, 28, 1)$
  - Xây dựng lớp Convolution đầu tiên và thiết lập các thông số quan trọng như số lượng filter là 6, kích thước kernel là 5×5, hàm kích hoạt là ***tanh***, kích thước đầu vào như trên, đồng thời thêm padding (với các giá trị bằng với các giá trị ở các cạnh) để đảm bảo kích thước của các lớp convolution.
  - Thêm lớp Average pooling với kích thước là 2×2 và bước di chuyển là 2 cho cả hai chiều
  - Tiếp tục xây dựng 1 lớp Convolution và Average pooling tương tự như trên.
  - Đến đây, thực hiện làm phẳng đầu ra để có thể kết nối nó với các lớp fully connected (chuyển đổi nó về mảng 1 chiều).
  - Thêm vào 2 lớp fully connected (lần lượt với số lượng nơ-ron là 120 và 84) với hàm kích hoạt được sử dụng là ***tanh***.
  - Thêm lớp cuối cùng là lớp đầu ra có 10 nơ-ron với hàm kích hoạt là ***softmax*.**
  - Mô hình sau khi xây dựng sẽ có kiến trúc như sau:

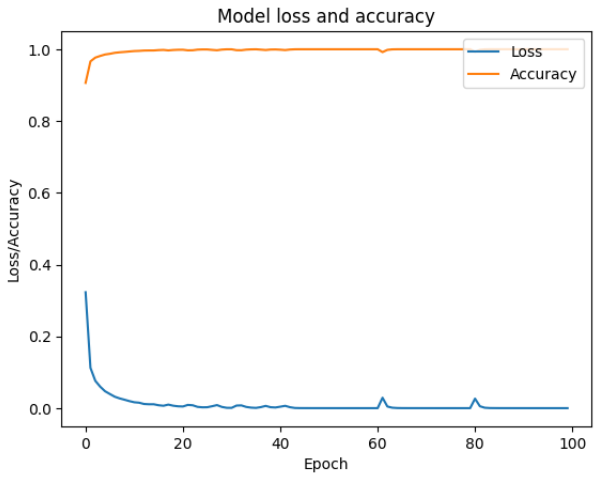
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 6)         156

 average_pooling2d (Average  (None, 14, 14, 6)         0
 Pooling2D)

 conv2d_1 (Conv2D)           (None, 10, 10, 16)        2416

 average_pooling2d_1 (Avera  (None, 5, 5, 16)          0
 gePooling2D)

 flatten (Flatten)           (None, 400)               0

 dense (Dense)               (None, 120)               48120

 dense_1 (Dense)             (None, 84)                10164

 dense_2 (Dense)             (None, 10)                850

=================================================================
Total params: 61706 (241.04 KB)
Trainable params: 61706 (241.04 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

- Huấn luyện mô hình:
    - Cài đặt các thông số cần thiết cho việc huấn luyện: kích thước batch là 128, số lượng epochs là 100
- Dự đoán dựa trên tập kiểm tra (test)

2. Yêu cầu 1

a. MNIST

- Sau khi huấn luyện mô hình với tập dữ liệu MNIST, ta thu được độ lỗi và độ chính xác mà mô hình có được như sau:



- Độ chính xác:
    - Đối với tập dữ liệu MNIST, mô hình CNN Lenet-5 gần như cho độ chính xác tuyệt đối dù là trên tập thử nghiệm hay huấn luyện
- Độ mất mát:
    - Dù ban đầu độ mất mát khá cao, song có thể thấy mô hình nhanh chóng cải thiện trong quá trình học tập, minh chứng là độ lỗi được giảm mạnh sau những epochs đầu tiên.

- Tuy nhiên, có thể thấy, sử dụng số lượng epochs là 100 đối với tập dữ liệu này gây lãng tốn chi phí cũng như tài nguyên do mô hình đã nhanh chóng hội tụ chỉ sau khoảng 20 epochs. Do đó, ta có thể dừng việc huấn luyện tại đó.
- Thực hiện dự đoán với mô hình trên, ta thu được kết quả khá cao (như hình dưới), điều này minh chứng cho việc mô hình đã học tốt từ tập dữ liệu

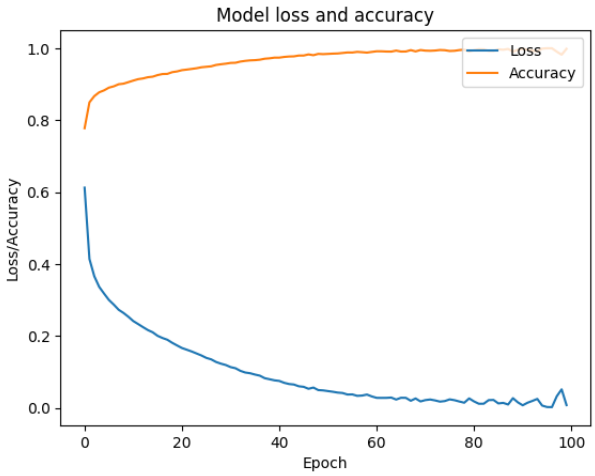
```
313/313 [==============================] - 2s 5ms/step - loss: 0.0668 - accuracy: 0.9883
Test Loss: 0.06676889210939407
Test accuracy: 0.9883000254631042
```

- Sau khi thực hiện huấn luyện trên tập dữ liệu này, ta lưu trọng số của nó để phục vụ cho yêu cầu 2 với tên là **"*MNIST_weights.h5*"**

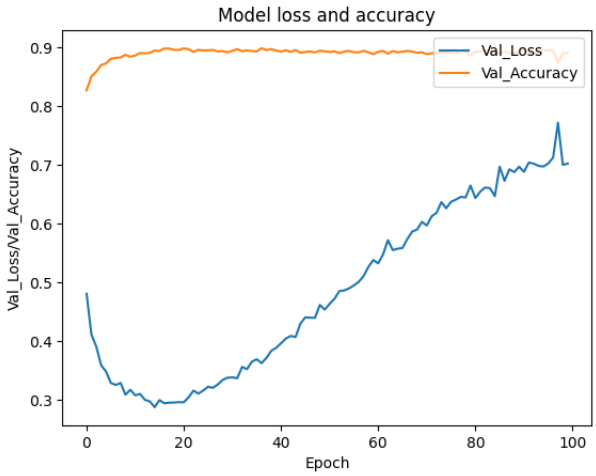
b. MNIST Fashion

- Sau khi huấn luyện mô hình với tập dữ liệu MNIST Fashion, ta thu được độ lỗi và độ chính xác mà mô hình có được như sau (được biểu diễn theo tập dữ liệu huấn luyện (training) và tập dữ liệu kiểm thử (Validation)):

Tập dữ liệu



Tập thử nghiệm



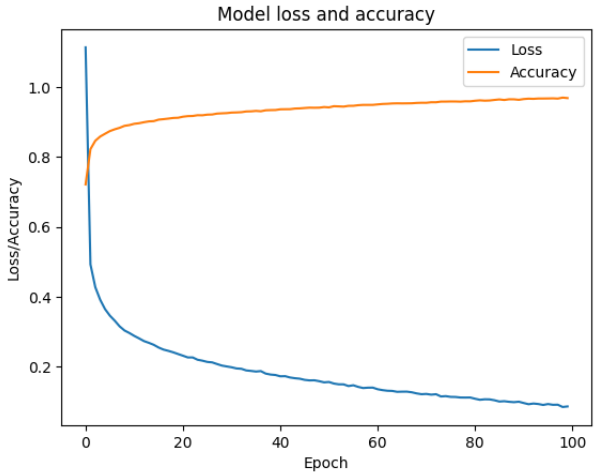
- Độ chính xác:
  - Đối với tập dữ liệu huấn luyện, phần trăm về độ chính xác được tăng khá cao, bắt đầu với 0.77 nhưng khi kết thúc lại đạt đến gần như tuyệt đối 0.99.
  - Tuy nhiên, đối với tập dữ liệu kiểm tra, phần trăm về độ chính xác dù có tăng nhưng không ổn định (dù độ chênh lệch không đáng kể). Kết quả được ghi nhận đối với epoch đầu là 0.82 và kết thúc là 0.89
- Độ mất mát:
  - Đối với tập dữ liệu huấn luyện, độ mất mát được cải thiện đáng kể, giảm dần qua từng epoch và mô hình gần như hội tụ (với độ lỗi cuối cùng khoảng 0.0076)
  - Đối với tập dữ liệu kiểm tra, ban đầu độ lỗi được cải thiện (khoảng 20 epochs đầu), tuy nhiên lại tăng cao đối với các epochs sau, điều này khiến ta cần lưu ý về mô hình (mô hình có khả năng bị overfitting).
- Kết quả dự đoán trên tập test cũng chỉ dừng ở độ chính xác là 0.891 và khoảng mất mát là 0.7

```
313/313 [==============================] - 2s 7ms/step - loss: 0.7022 - accuracy: 0.8912
Test Loss: 0.7022039890289307
Test accuracy: 0.8912000060081482
```

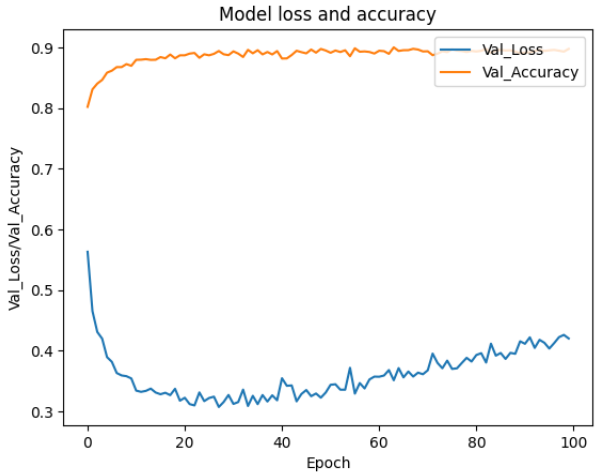
3. Yêu cầu 2

- Sau khi huấn luyện mô hình với tập dữ liệu MNIST Fashion kèm với trọng số ***MNIST_weights.h5*** từ việc huấn luyện tập dữ liệu MNIST trên, ta thu được độ lỗi và độ chính xác mà mô hình có được như sau (được biểu diễn theo tập dữ liệu huấn luyện (Training) và tập dữ liệu kiểm thử (Validation)):

Tập dữ liệu



Tập thử nghiệm



- Độ chính xác
  - Đối với tập dữ liệu huấn luyện, độ chính xác có xu hướng tăng lên đều và ổn định, bắt đầu với 0.72 và đạt đến gần như tuyệt đối, 0.96 ở epochs cuối cùng.
  - Đối với tập dữ liệu thử nghiệm, độ chính xác dù có tăng mạng ở khoảng 10 epochs đầu, tuy nhiên lại có sự dao động (tăng rồi giảm) ở những epochs còn lại, và kết thúc với độ chính xác cuối cùng là 0.89.
- Độ mất mát
  - Đối với tập dữ liệu, mô hình có dấu hiệu của sự hội tụ khi độ mất mát giảm mạnh ở khoảng 20 epochs đầu và dần dần tiến về 0 đối với các epochs sau.
  - Còn với tập dữ liệu thử nghiệm, dù có sự giảm mạnh ở khoảng 10 epochs đầu, nhưng sau đó lại có sự biến động ở những epochs sau (có dấu hiệu tăng lại). Điều này cho thấy mô hình có khả năng không thể học được nữa.
- Kết quả dự đoán trên tập test cũng chỉ dừng ở mức 0.897 đối với độ chính xác và khoảng mất mát là 0.4

```
313/313 [==============================] - 2s 7ms/step - loss: 0.4202 - accuracy: 0.8974
Test Loss: 0.42024779319763184
Test accuracy: 0.8974000215530396
```

- So với kết quả đạt được từ mô hình huấn luyện không có trọng số, dường như kết quả của chúng không có sự chênh lệch quá nhiều (so với độ chính xác trên tập dữ liệu và thử nghiệm). Tuy nhiên, đối với mô hình có sử dụng trong số, độ lỗi có sự cải thiện tốt hơn
- Từ kết quả thực nghiệm, có thể thấy việc thêm trọng số ở đây không giúp cho mô hình tốt hơn về độ chính xác nhưng lại giúp cho mô hình cải thiện được độ lỗi đáng kể.