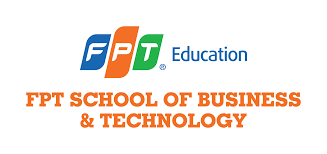
VIỆN QUẢN TRỊ & CÔNG NGHỆ FSB



ỨNG DỤNG MẠNG NEURAL NETWORK TRONG BÀI TOÁN PHÂN BIỆT CHỮ SỐ

Tác giả: Nguyễn Hoài Phương

Lớp: MSE09

# **1 – Giới thiệu**

Trong nghiên cứu này, tôi sẽ trình bày một phương pháp nhận dạng chữ số từ 0 đến 9 bằng phương pháp neural network. Ban đầu, một mạng neural cơ bản chỉ có 2 lớp và không áp dụng bất kỳ phương pháp optimizations nào sẽ được sử dụng để huấn luyện mô hình. Sau đó, dựa trên kết quả đạt được (training error và test error), các chiến thuật để nâng cao hiệu quả của mô hình sẽ được bàn bạc và ứng dụng. Cuối cùng, mô hình tốt nhất cho tập dữ liệu huấn luyện và kiểm thử sẽ được lựa chọn sau rất nhiều thử nghiệm được thực hiện trong nghiên cứu.

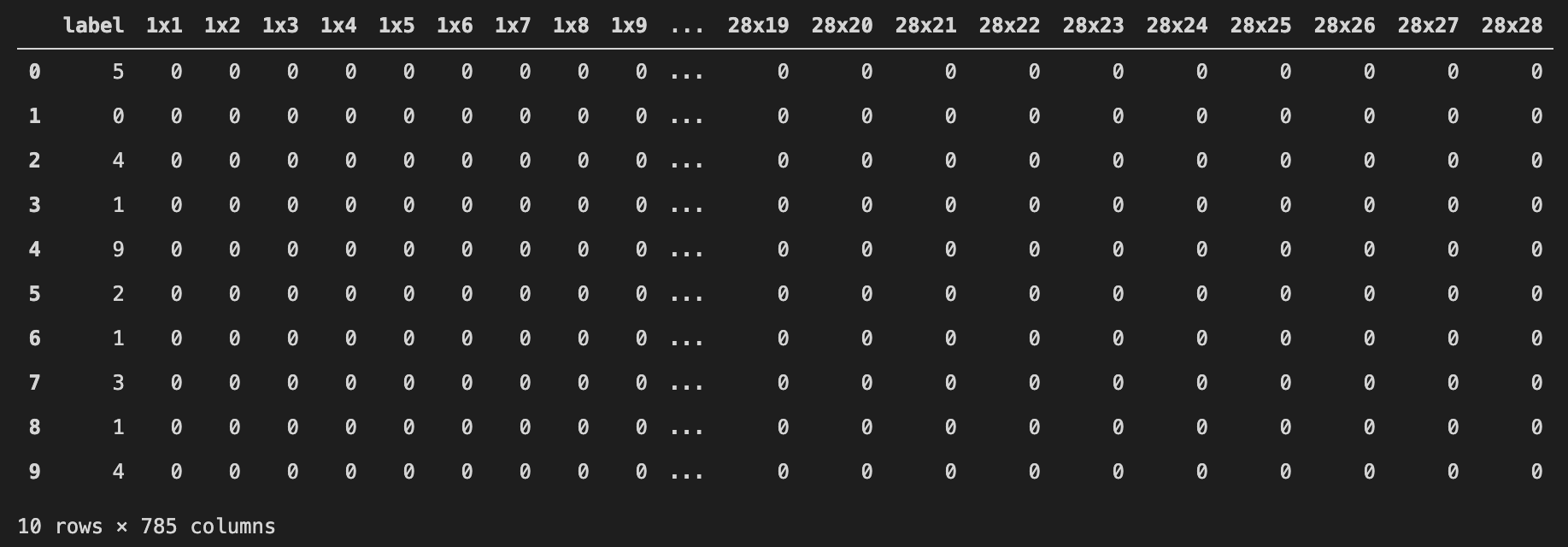
**2 – Tập dữ liệu**

Tập dữ liệu được sử dụng trong nghiên cứu này được download từ link https://www.kaggle.com/oddrationale/mnist-in-csv?select=mnist\_test.csv. Có 2 file csv được download là mnist\_train.csv và mnist\_test.csv tương ứng với 2 tập dữ liệu huấn luyện và kiểm thử. Tập hợp các ảnh 28x28 của các chữ số ở định dạng thang xám (gray) sau khi được flatten (chuyển đổi từ ma trận 28x28 sang vector hàng) thì lưu trữ trong 2 file excel trên. Mỗi dòng của file csv sẽ có 2 thông số chính:

- Label chính là chữ số của bức ảnh.

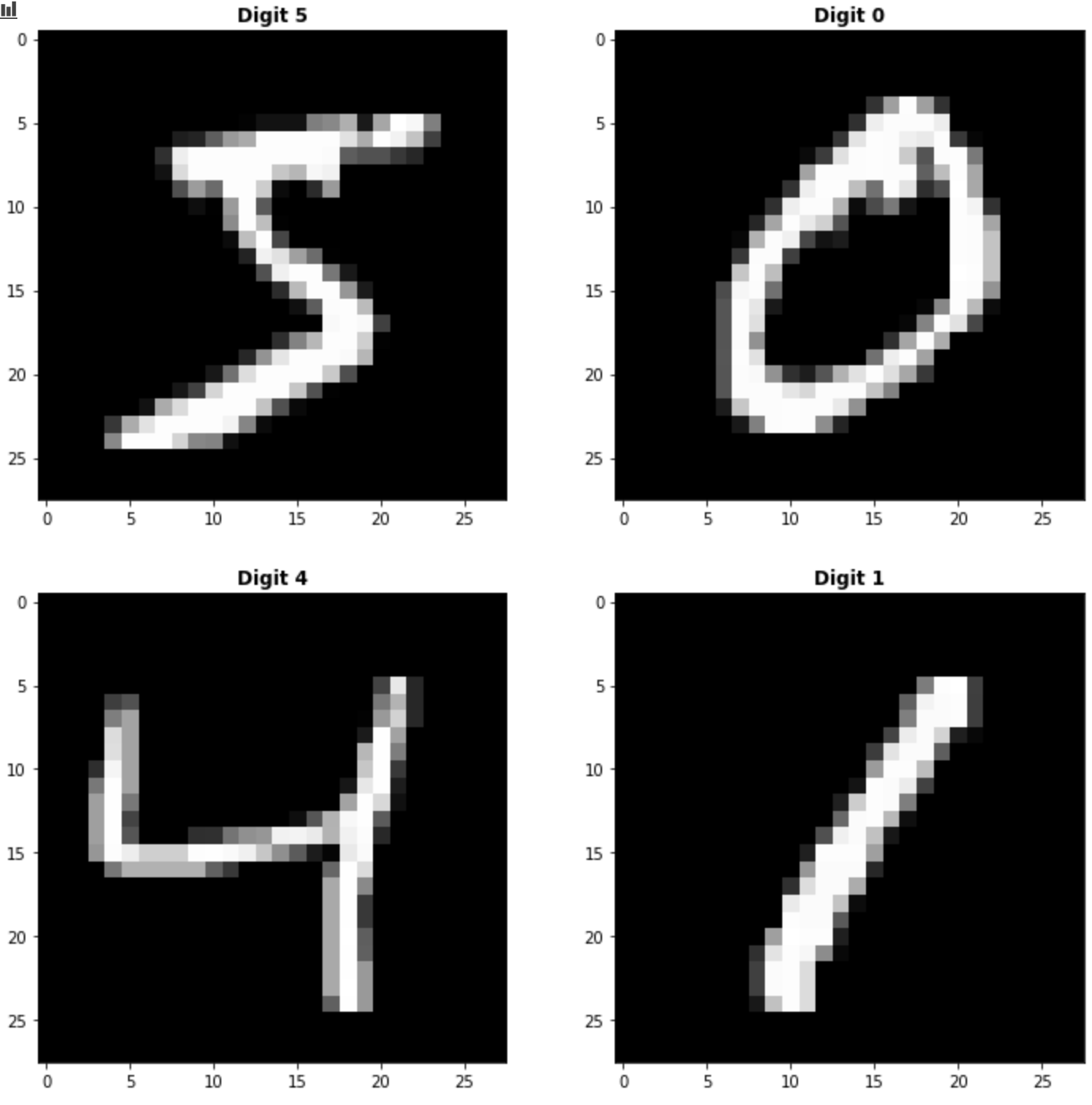
- Các cột 1x1, 1x2, ... 28x28 chính là giá trị pixel của bức ảnh

Có tổng cộng 60000 bức ảnh trong tập huấn luyện và 10000 bức ảnh trong tập kiểm thử. Với tập huấn luyện thì mỗi chữ số có khoảng 6000 bức ảnh, còn với tập kiểm thử thì mỗi chữ số có khoảng 1000 bức ảnh.

Hình 1: Data lưu trữ trong file csv.

|  |  |  |
| --- | --- | --- |
| **Chữ số** | **Training** | **Testing** |
| 0 | 5923 | 980 |
| 1 | 6742 | 1135 |
| 2 | 5958 | 1030 |
| 3 | 6131 | 1010 |
| 4 | 5842 | 982 |
| 5 | 5421 | 892 |
| 6 | 5918 | 958 |
| 7 | 6265 | 1028 |
| 8 | 5851 | 974 |
| 9 | 5949 | 1009 |
| **Tổng cộng** | **60000** | **10000** |

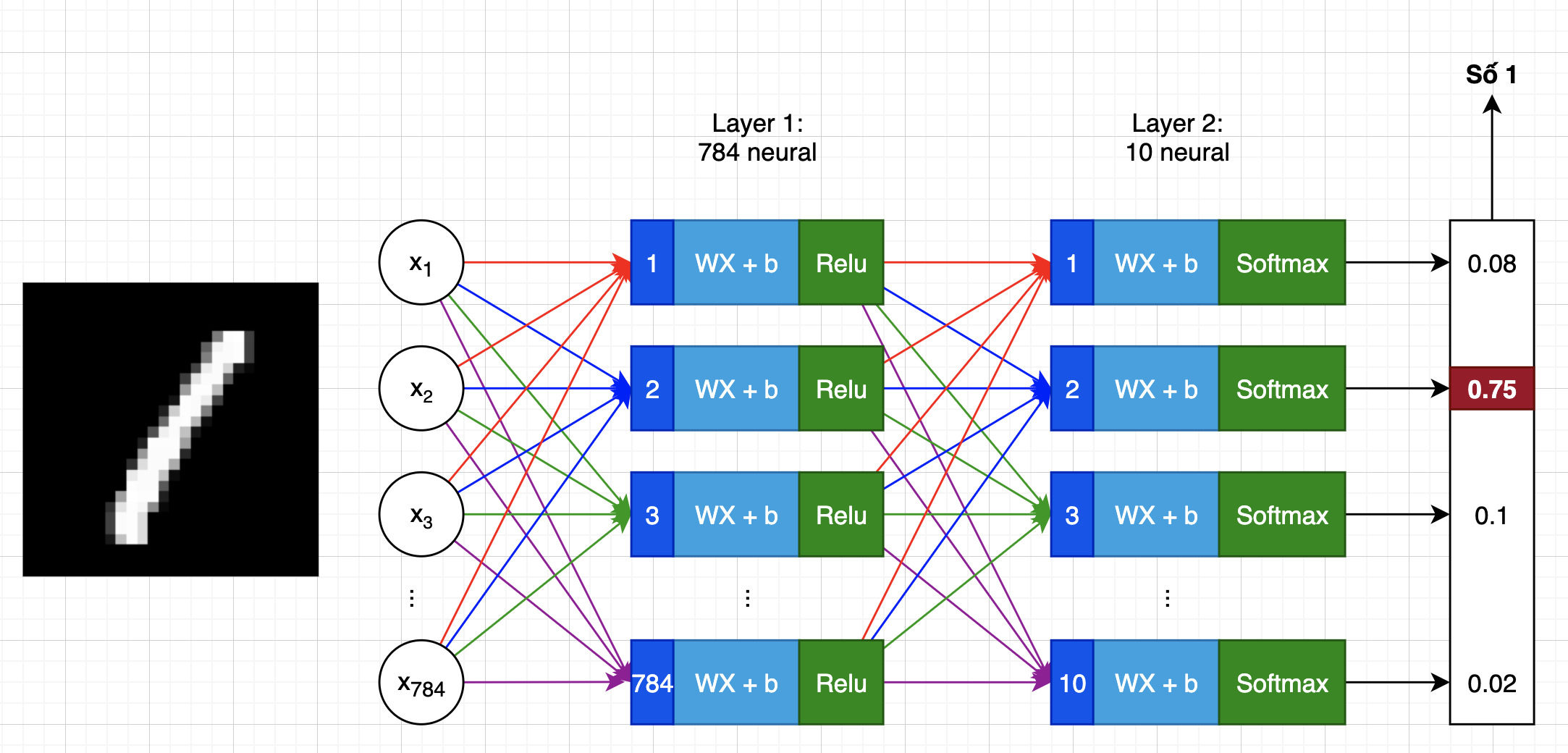
Bảng 1: tóm tắt tập dự liệu huấn luyện và kiểm thử

Hình 2: chữ số được tạo từ các giá trị pixel trong file csv.

# 3 – Mạng neural 2 lớp, không sử dụng optimization

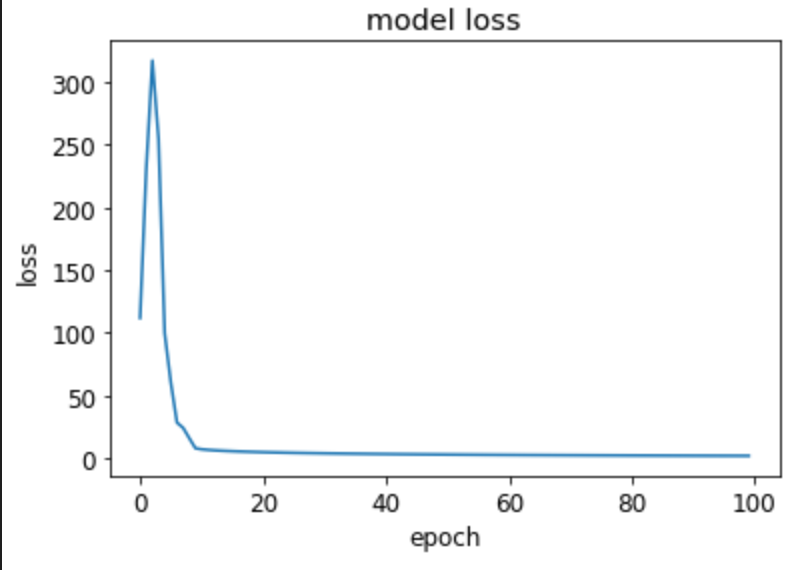
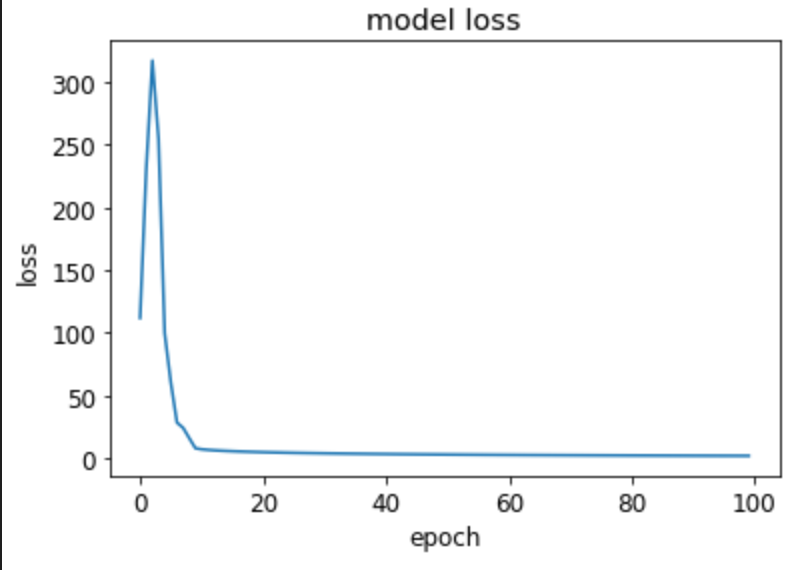
Đầu tiên, chúng ta sẽ thử giải bài toán nhận dạng chữ số bằng một mạng neural đơn giản :

* 2 layers: 1 hidden layer có 28x28 = 784 neural với activation là `Relu` và 1 output layer có 10 neural (tương ứng với 10 chữ số từ 0-9) với activation là `softmax`.
* Epochs = 100
* learning\_rate = 0.001
* batch\_size = 60000 (full batch của tập dữ liệu huấn luyện, không xài mini-batch)



Hình 3: mô hình mạng neural 2 layers

## 3.1 – Kết quả training



Hình 4: Loss và accuracy theo epoch

* Thời gian huấn luyện: 62(s)
* Loss tại epoch 100: 1.9657
* Training accuracy tại epoch 100: 91%
* Training error: 100 – 91 = 8%

## 3.2 – Kết quả training

* Accuracy : 89.91%
* Test error: 100 – 89.91 = 10.09%

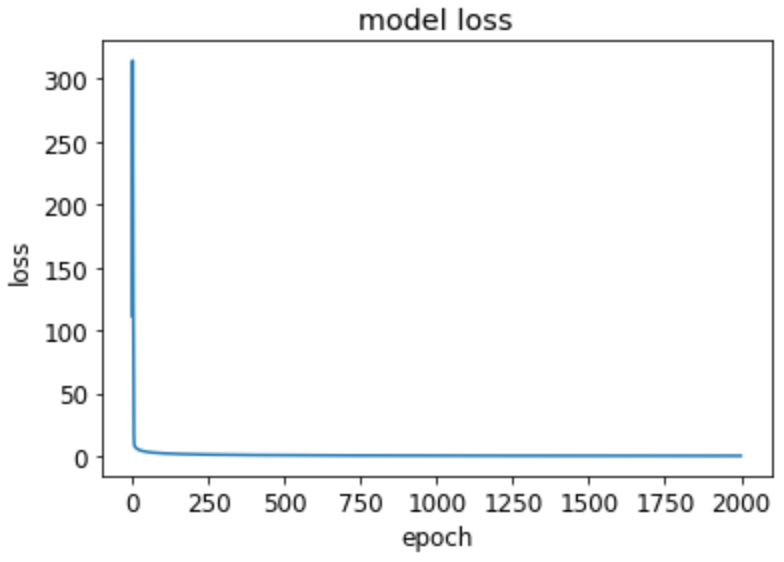
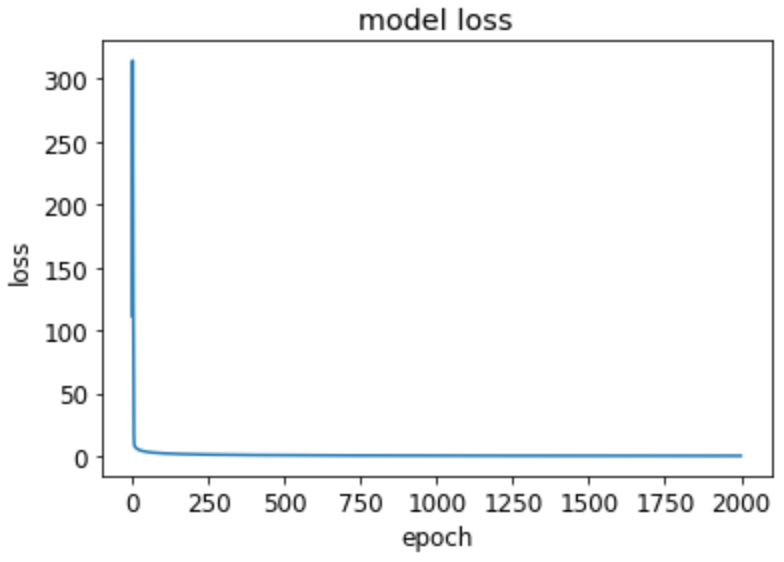
## 3.3 – Đánh giá mô hình:

Dựa vào training error và test error, chúng ta dễ dàng nhận ra mô hình đang bị bias. Hay nói 1 cách khác, mô hình chưa được huấn luyện đủ tốt. Như vậy, bước tiếp theo phải làm chính là tìm cách giảm training error của mô hình. Có 2 cách có thể thử nghiệm:

* Huấn luyện mô hình lâu hơn: tăng epochs từ 100 lên 2000
* Tăng só lượng layers của mô hình: tăng từ 2 layers lên 4 layers để mô hình có thể học “sâu” hơn.

# 4 – Huấn luyện mô hình lâu hơn: tăng epochs từ 100 lên 2000

## 4.1 - Kết quả training



Hình 5: Loss và accuracy theo epoch

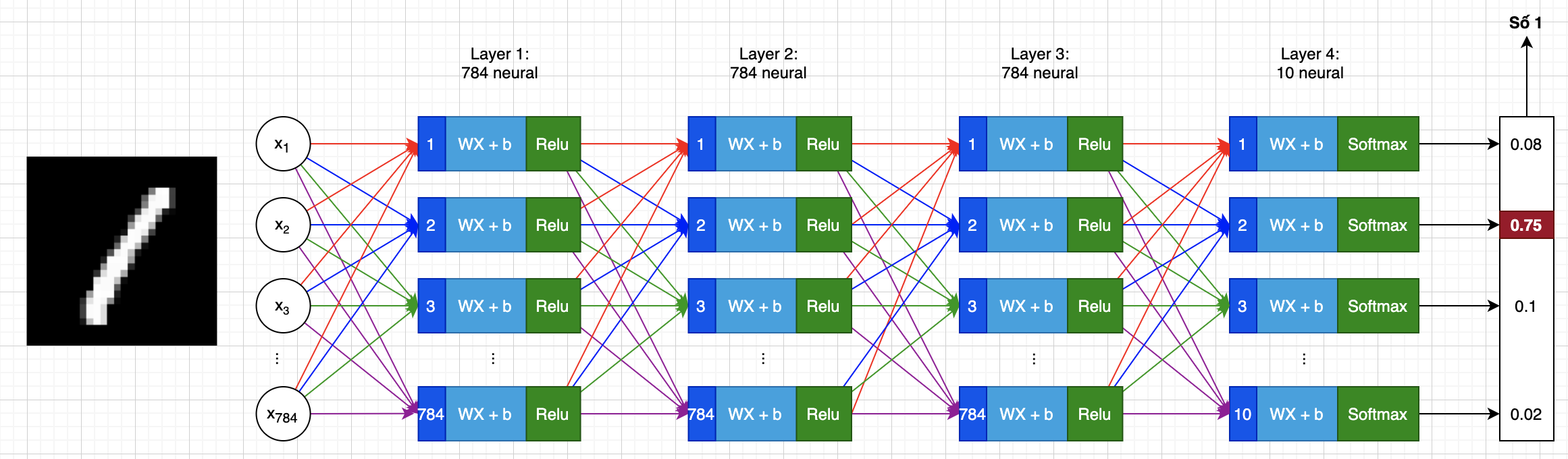
* Thời gian huấn luyện: 1747(s)
* Loss tại epoch 2000: 0.025
* Training accuracy tại epoch 2000: 100%
* Training error: 0%

## 4.2 - Kết quả test

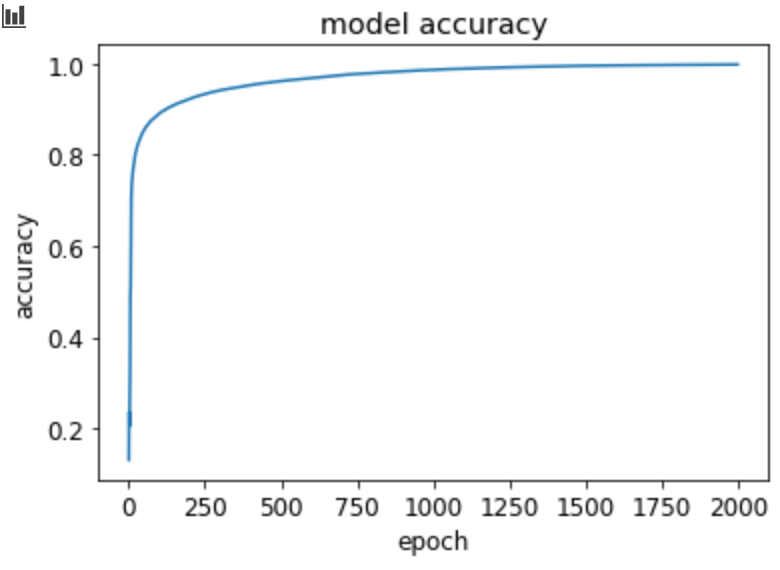
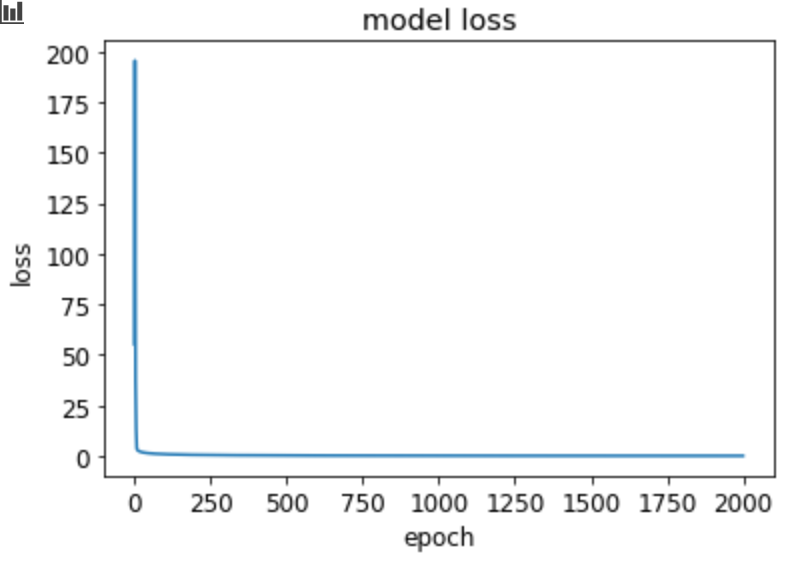
* Accuracy : 94.1%
* Test error: 100 – 94.1 = 5.9%

# 5 – Huấn luyện mô hình sâu hơn: sử dụng 4 layers

* 4 layers: 3 hidden layer có 28x28 = 784 neural với activation là `Relu` và 1 output layer có 10 neural (tương ứng với 10 chữ số từ 0-9) với activation là `softmax`.
* Epochs = 2000
* learning\_rate = 0.001
* batch\_size = 60000 (full batch của tập dữ liệu huấn luyện, không xài mini-batch)

Hình 6: mô hình mạng neural 4 layers

## 5.1 – Kết quả training



Hình 7: Loss và accuracy theo epoch

* Thời gian huấn luyện: 8308(s)
* Loss tại epoch 2000: 0.014
* Training accuracy tại epoch 2000: 99.85%
* Training error: 0.15%

## 5.2 – Kết quả test

* Accuracy : 93.25%
* Test error: 100 – 93.25 = 6.75%

## 5.3 – Đánh giá mô hình:

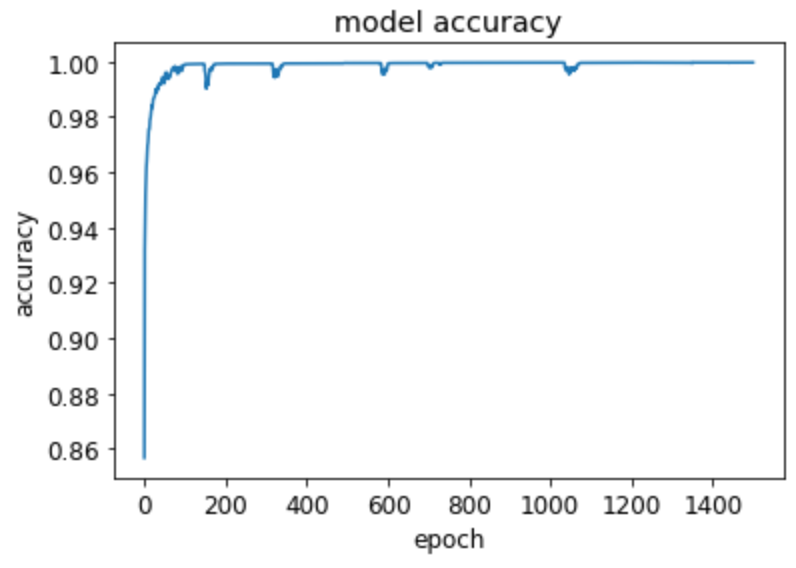
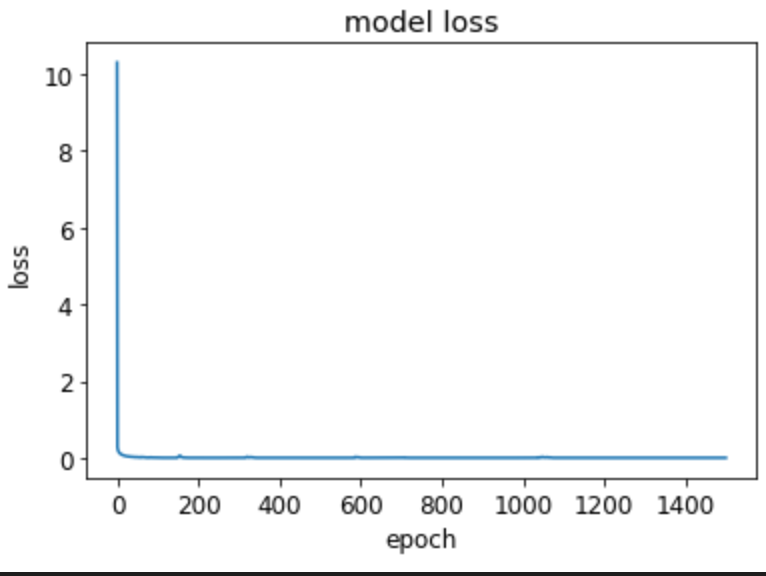
Mô hình mạng neural 4 layers này có performance không tốt bằng mô hình mạng neural 2 lớp khi được huấn luyện có cùng số vòng lặp như nhau (epochs = 2000). Thêm nữa, thời gian huấn luyện mô hình lâu hơn rất nhiều so với mô hình mạng neural 2 lớp. Như vậy, không phải lúc nào mạng càng sâu thì performance càng cao.

Mô hình mạng neural 2 lớp sẽ được sử dụng cho các bước optimizations tiếp theo.

# 6 – Optimize mô hình bằng phương pháp momentum và xài mini batch

* batch size : 128
* epochs : 1500
* learning rate : 0.001
* momentum : 0.9

## 6.1 – Kết quả training



Hình 8: Loss và accuracy theo epoch

* Thời gian huấn luyện: 3134(s)
* Loss tại epoch 1500: 0.009
* Training accuracy tại epoch 1500: 100%
* Training error: 0%

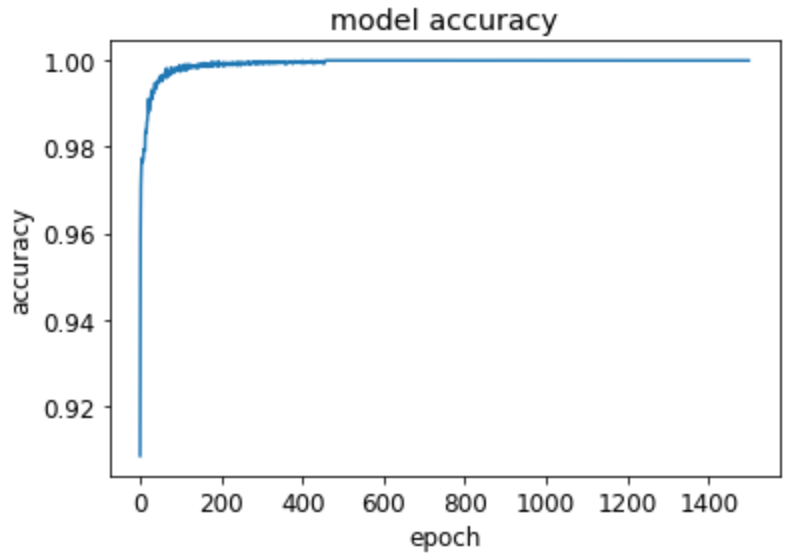
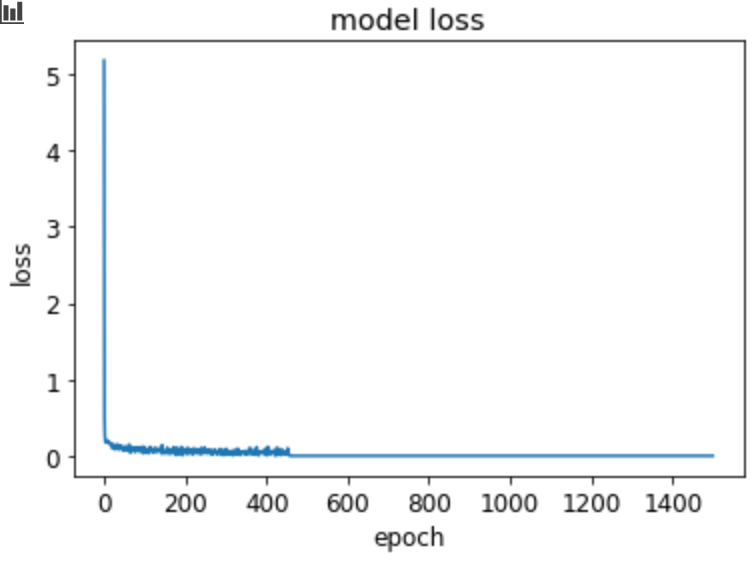
## 6.2 – Kết quả test

* Accuracy : 97.17%
* Test error: 2.83%

# 7 – Optimize mô hình bằng phương pháp Adam và xài mini batch

* batch size : 128
* epochs : 1500
* learning rate : 0.001

## 7.1 – Kết quả training



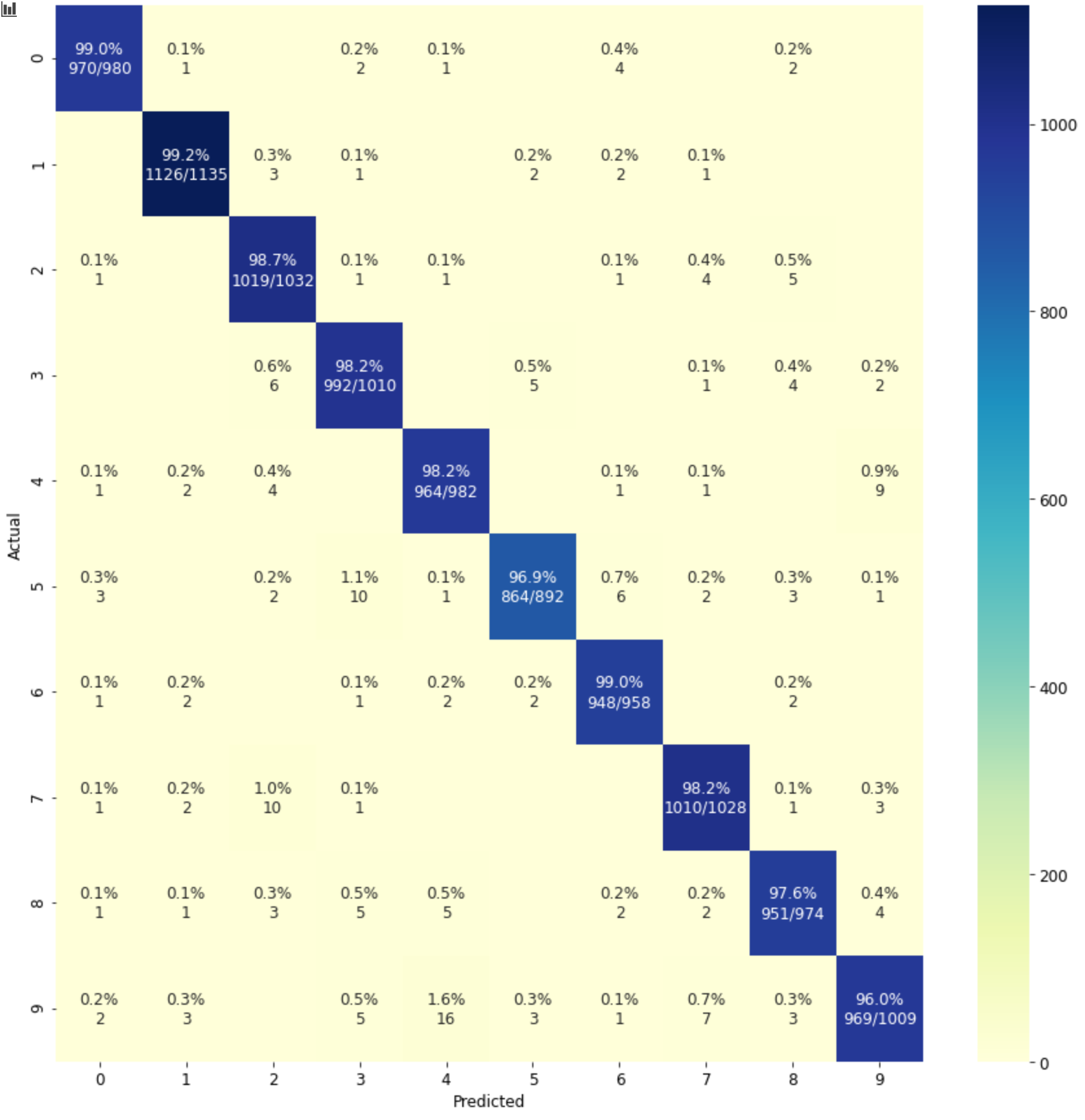
Hình 9: Loss và accuracy theo epoch

* Thời gian huấn luyện: 3083(s)
* Loss tại epoch 1500: 0.00018
* Training accuracy tại epoch 1500: 100%
* Training error: 0%

## 7.2 – Kết quả test

* Accuracy : 98.13%
* Test error: 1.87%

# 8 – Kiểm tra các metrics recall, precision, f1-score cho mô hình sử dụng Adam

Hình 10: confusion matrix và recall

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Digits | | | | | | | | | |
| Metrics | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Precision | 99% | 99% | 97.3% | 97.4% | 97.4% | 98.6% | 98.2% | 98.2% | 97.9% | 98.1% |
| Recall | 99% | 99.2% | 98.7% | 98.2% | 98.2% | 96.9% | 99% | 98.2% | 97.6% | 96% |
| f1-score | 99% | 99.1% | 98% | 97.8% | 97.8% | 97.7% | 98.6% | 98.2% | 97.8% | 97% |

Bảng 2: precision, recall và f1-score

# 9 – Kết luận

* Không phải lúc nào mạng neural càng sâu thì performance càng cao: mô hình mạng neural 2 lớp có performance tốt hơn mạng neural 4 lớp với tập dữ liệu sử dụng trong bài nghiên cứu này
* Optimization bằng phương pháp Adam cho kết quả hội tụ nhanh hơn phương pháp momentum.