## **1. Bài toán phân loại quần áo trên bộ dữ liệu Fashion MNIST**

### ***1.1. Mô tả dữ liệu***

Tập dữ liệu về quần áo và giày dép từ https://github.com/zalandoresearch/fashion-mnist, chứa khoảng 70,000 ảnh đen trắng phân thành 10 loại. Mỗi một ảnh là một loại quần áo hoặc giày dép với độ phân giải thấp (28 by 28 pixel), như hình minh hoạ bên dưới:



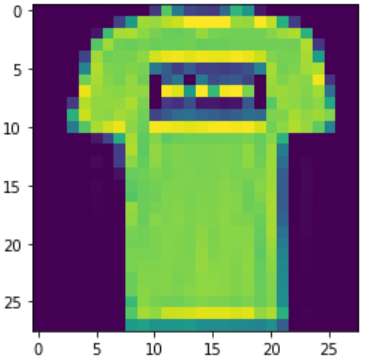
Bộ data này gồm 70,000 ảnh, trong đó training set có 60,000 ảnh, test set có 10,000 ảnh được phân vào 10 class như sau:

|  |  |
| --- | --- |
| **Label** | **Class** |
| 0 | T-shirt/top |
| 1 | Trouser |
| 2 | Pullover |
| 3 | Dress |
| 4 | Coat |
| 5 | Sandal |
| 6 | Shirt |
| 7 | Sneaker |
| 8 | Bag |
| 9 | Ankle boot |

### ***1.2. Mô tả bài toán học máy***

Bài toán phân lớp nhiều lớp dựa vào ảnh để phân loại quần áo, giày dép

Bộ dữ liệu có 60.000 ảnh trong tập huấn luyện, với mỗi ảnh được biểu diễn theo dạng 28x28 pixel:



Kỹ thuật sử dụng: Kỹ thuật học máy được áp dụng: Neural Network (C[onvolutional Neural Network](https://nttuan8.com/bai-6-convolutional-neural-network/))

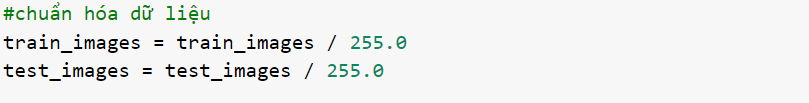
### ***1.3. Tiền xử lý dữ liệu***



Dữ liệu cần được tiền xử lý trước khi được dùng để huấn luyện mạng neuron. Phân tích ảnh đầu tiên trong tập dữ liệu, chúng ta sẽ thấy các pixel có giá trị từ 0 đến 255.



Tiền xử lý để mỗi một điểm ảnh có giá trị từ 0 đến 1. Để làm điều này, chúng ta chỉ cần lấy giá trị của pixel chia cho 255

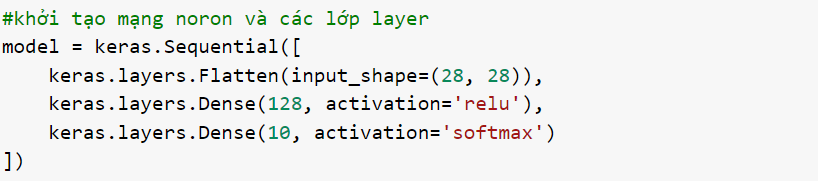


Để xây dựng mạng neuron, chúng tay cần cấu hình các layer của mô hình, và sau đó biên dịch mô hình.

*Thiết lập các layers*

Thành phần cơ bản của một mạng neuron là các *layer*. Các layer trích xuất các điểm đặc biệt từ dữ liệu mà chúng đón nhận. Khi thực hiện tốt, những điểm đặc biệt này mang nhiều ý nghĩa và phục vụ cho toán của chúng ta.

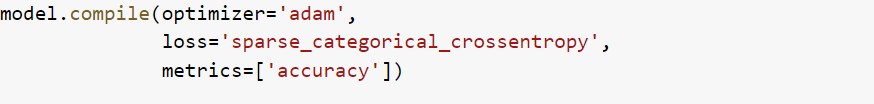
Đa số các mô hình deep learning đều chứa các layer đơn giản được xâu chuỗi lại với nhau. Đa số các layer, ví dụ tf.keras.layers.Dense, đều có các trọng số sẽ được học trong quá trình huấn luyện.



*Biên dịch mô hình*

Trước khi mô hình có thể được huấn luyện, chúng ta cần thêm vài chỉnh sửa. Các chỉnh sửa này được thêm vào trong bước *biên dịch* của mô hình:

* *Hàm thiệt hại* — dùng để đo lường mức độ chính xác của mô hình trong quá trình huấn luyện. Chúng ta cần giảm thiểu giá trị của hạm này "điều khiển" mô hình đi đúng hướng (thiệt hại càng ít, chính xác càng cao).
* *Trình tối ưu hoá* — Đây là cách mô hình được cập nhật dựa trên dữ liệu huấn luyện được cung cấp và hàm thiệt hại.
* *Số liệu* — dùng để theo dõi các bước huấn luyện và kiểm thử. Ví dụ sau dùng *accuracy*, tỉ lệ ảnh được phân loại chính xác.



*Huấn luyện mô hình*

Huấn luyện mô hình mạng neuron cần các bước sau:

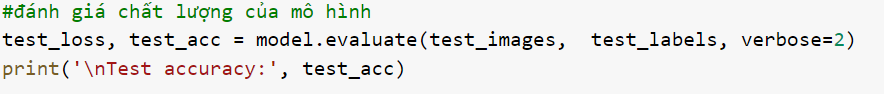
1. Cung cấp dữ liệu huấn luyện cho mô hình. Trong ví dụ này, dữ liệu huấn luyện năm trong 2 mảng train\_images và train\_labels
2. Mô hình sẽ học cách liên kết ảnh với nhãn.
3. Chúng ta sẽ yêu cầu mô hình đưa ra dự đoán từ dữ liệu của tập kiểm thử, trong ví dụ này là mảng test\_images, sau đó lấy kết quả dự đoán đối chiếu với nhãn trong mảng test\_labels.

Để bắt đầu huấn luyện, gọi hàm model.fit. Hàm này được đặt tên fit vì nó sẽ "fit" ("khớp") mô hình với dữ liệu huấn luyện:



*Đánh giá chất lượng mô hình*

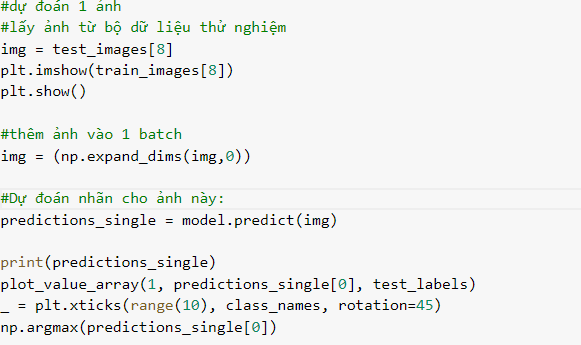
Đánh giá model với dữ liệu test set



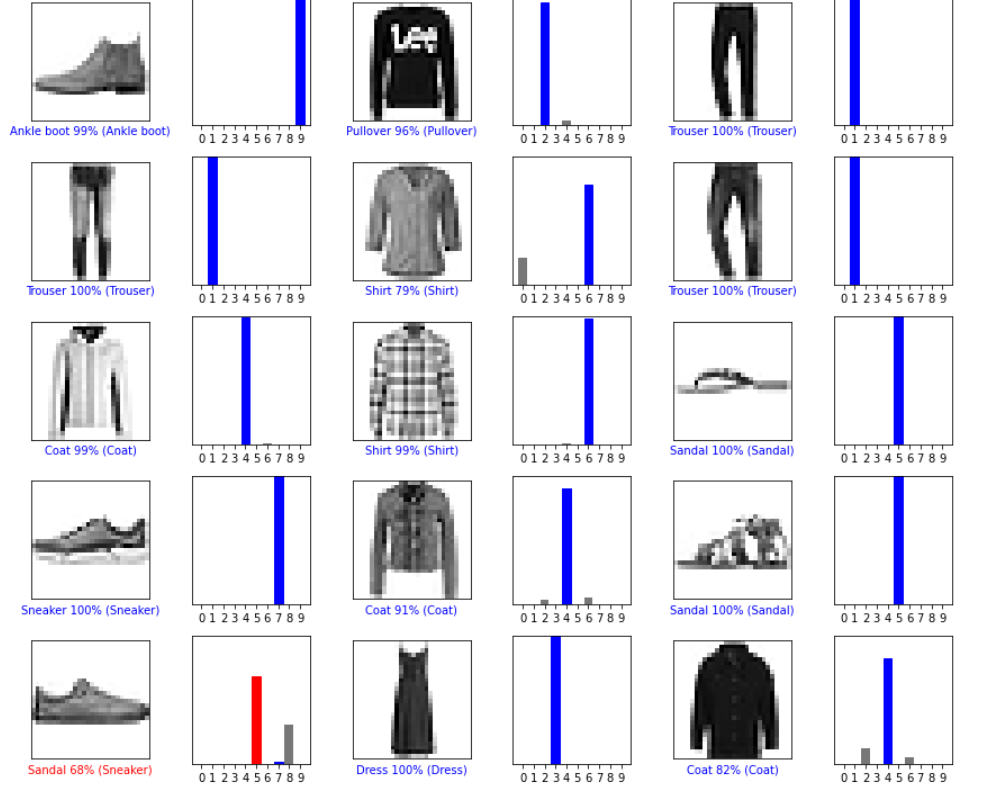
*Đưa ra dự đoán*

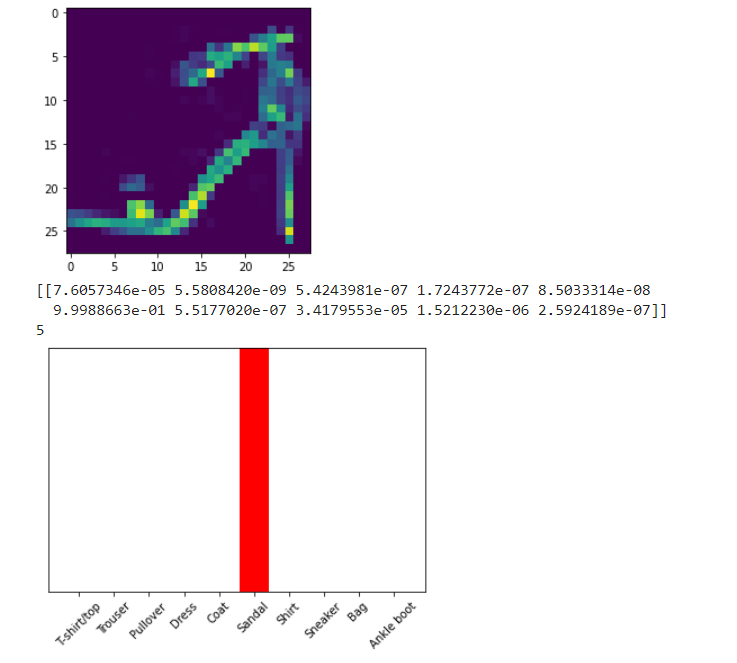


Dùng mô hình để đưa ra dự đoán về một ảnh duy nhất.



Kết quả chương trình sau khi chạy





### **1.4. Kết luận**

Chúng ta đã hoàn thành bài toán phân lớp từ load data, tạo model và tiến hành huấn luyện mô hình cho tập dữ liệu Fashion MNIST với độ chính xác là 88,4 % tương đối cao. Chính vì thế mà ngày nay việc áp dụng kỹ thuật này đang trở nên vô cùng phổ biến cũng như tầm quan trọng của nó

* Chuẩn đoán ảnh X-ray của bệnh nhân có bị ung thư hay không
* Phân loại, nhận diện được các chữ, số viết tay => tự động đọc được biển số xe, văn bản.
* Phân loại được các biển báo giao thông => hỗ trợ cho ô tô tự lái