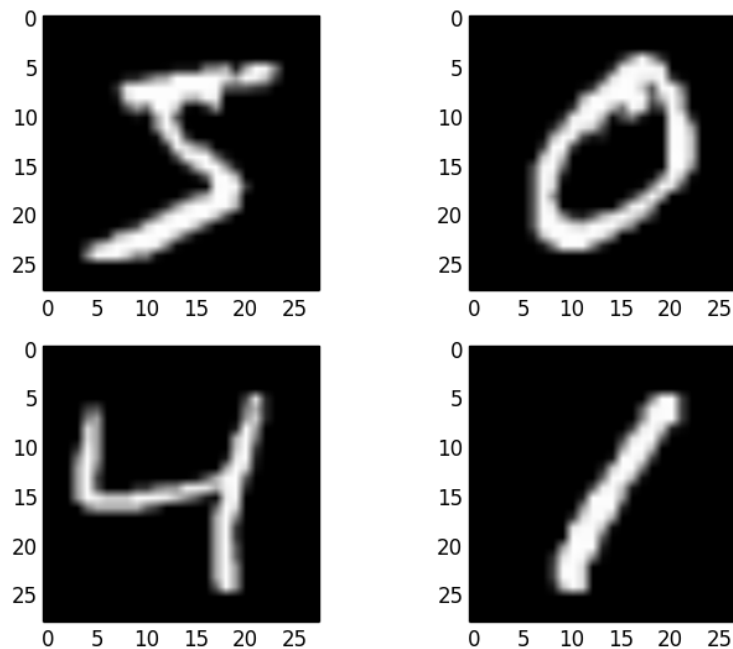


# Ứng dụng Active Learning trong bài toán nhận diện ảnh chữ số MNIST



Sinh viên: Đồng Xuân Toàn

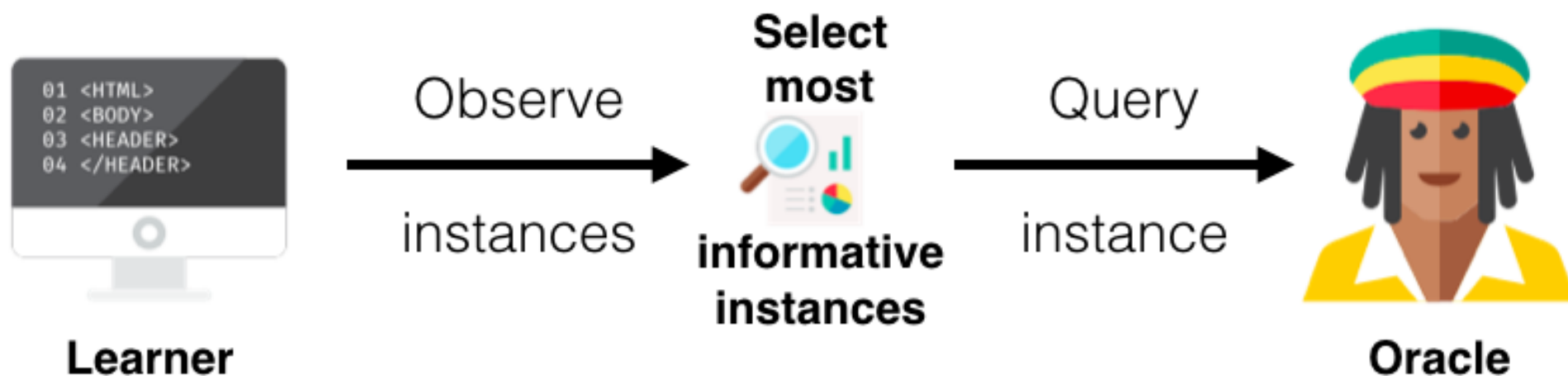


# Outline

- 1. Kịch bản Active Learning
- 2. Mã giả
- 3. Kiến trúc mô hình
- 4. Activation Function: Softmax
- 5. Loss function
- 6. Optimizer : Gradient Descent
- 7. Early Stop
- 8. Sample selection function
- 9. Tổng hợp
- 10. Kết quả

# 1. Kịch bản Active Learning

- Mô hình sử dụng kịch bản Pool-based sampling. Khi đó, mẫu được chọn từ các nhóm chưa được dán nhãn (sao cho các mẫu được chọn chứa nhiều thông tin nhất) với mục đích dán nhãn chúng.



## 2. Mã giả

- 1. Chia data thành một nhóm và test set.
- 2. Chọn k mẫu từ nhóm để khởi tạo training set, phần còn lại làm validation set.
- 3. Normalize tất cả dataset.
- 4. Train model với train set.
- 5. Dùng train model dự đoán với validation set, lấy xác suất với mỗi mẫu.
- 6. Dùng train model dự đoán với test set, đo hiệu suất.
- 7. Từ validation set, Chọn k mẫu thông tin nhất dựa trên xác suất (vd: các mẫu mà model dự đoán không chắc chắn)
- 8. Chuyển k mẫu từ validation set xuống train set và yêu cầu nhãn
- 9. Đảo ngược normalize với toàn bộ dataset
- 10. Kiểm tra điều kiện dừng, nếu ko thỏa mãn chuyển xuống bước 3.

Điều kiện dừng có thể là số vòng lặp đạt đến max\_iteration cho trước, hoặc chênh lệch độ chính xác trên test set hoặc validation set quá nhỏ.

### 3. Kiến trúc mô hình

Ảnh đầu vào ( $28 \times 28$ ) được Flatten về kích thước (784,). Các nhãn ban đầu (từ 0  $\rightarrow$  9) được chuyển về dạng one hot vector (vector xác suất để mẫu thuộc về từng lớp). Sau đó đưa vào mô hình mạng

Input(784)  $\rightarrow$  Fully Connected (10)  $\rightarrow$  Activation Function: Softmax (10).

Từ đó, với input ban đầu, mô hình cho đầu ra là vector độ dài 10 tương ứng với xác suất mà mô hình dự đoán mẫu thuộc về từng lớp.

## 4. Activation Function: Softmax

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Hệ thống sử dụng Softmax Function, giúp từ 10 giá trị đầu ra của lớp Fully Connected, chuyển thành 10 giá trị xác suất (tổng bằng 1) đồng biến với các giá trị đầu vào của hàm Softmax.

# 5. Loss function

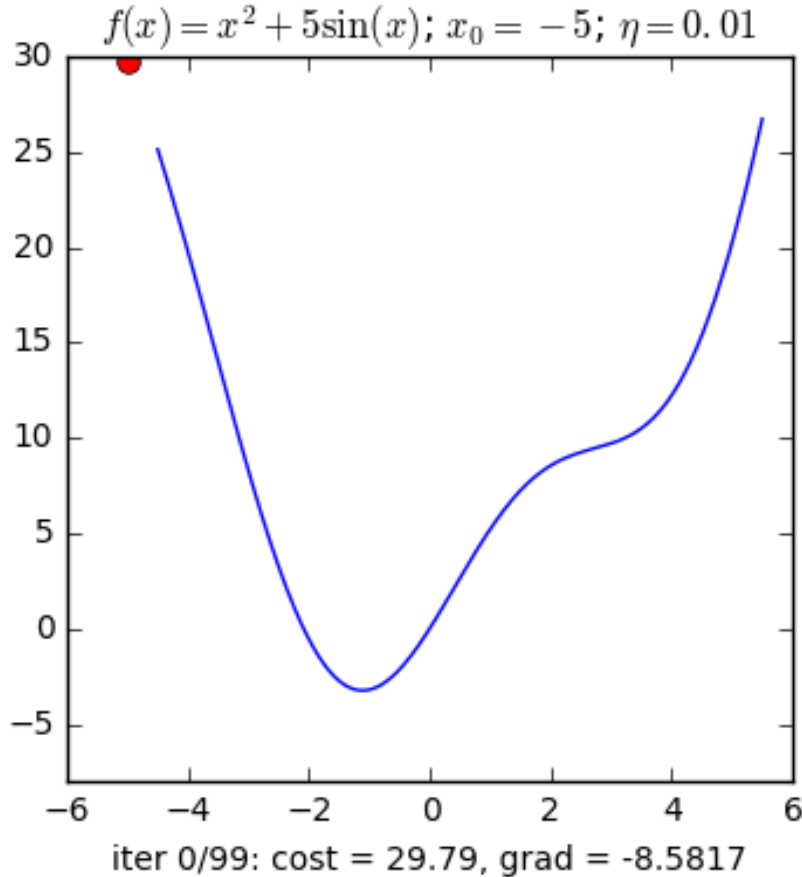
Cross entropy là một khái niệm khá quen thuộc trong lý thuyết thông tin, dùng để so sánh 2 phân phối.

$$\text{Cross Entropy} = - \overset{\textcircled{3}}{\frac{1}{N}} \overset{\textcircled{2}}{\sum_j} \overset{\textcircled{1}}{y_j} * \log(\hat{y}_j)$$

- ① true label \* log(predicted)
- ② sum over all sequences in each batch
- ③ divide by the number of samples

Trên keras, ta sử dụng hàm `keras.losses.categorical_crossentropy`

# 6. Optimizer : Gradient Descent



Nhận thấy, với đồ thị hàm số trên, dù bắt đầu ở điểm nào, chỉ cần đi ngược chiều đạo hàm, ta đều có thể tiến về điểm cực trị của hàm số (cực tiểu).

Từ đó, ta có công thức tối ưu:

$$\theta = \theta - \eta \nabla_{\theta} J(\theta)$$

Stochastic Gradient Descent (SGD) là một biến thể của Gradient Descent, thích ứng với các bài toán dữ liệu lớn (Deep Learning).

Trong phần code, ta sử dụng SGD với learning rate= $10^{-4}$  và momentum=0.9



## 7. Early Stop

Do kịch bản Active Learning yêu cầu mô hình cần train nhiều lần để thích ứng với dữ liệu thay đổi, để giảm thời gian training, với mỗi lần train, ta chỉ train tối đa 30 epochs với điều kiện dừng sớm:

```
es = EarlyStopping(monitor='val_loss',  
mode='min', verbose=1)
```

Model sẽ dừng việc training ngay khi val\_loss giảm. Nhờ đó tiết kiệm thời gian training, tránh overfit.

## 8. Sample selection function

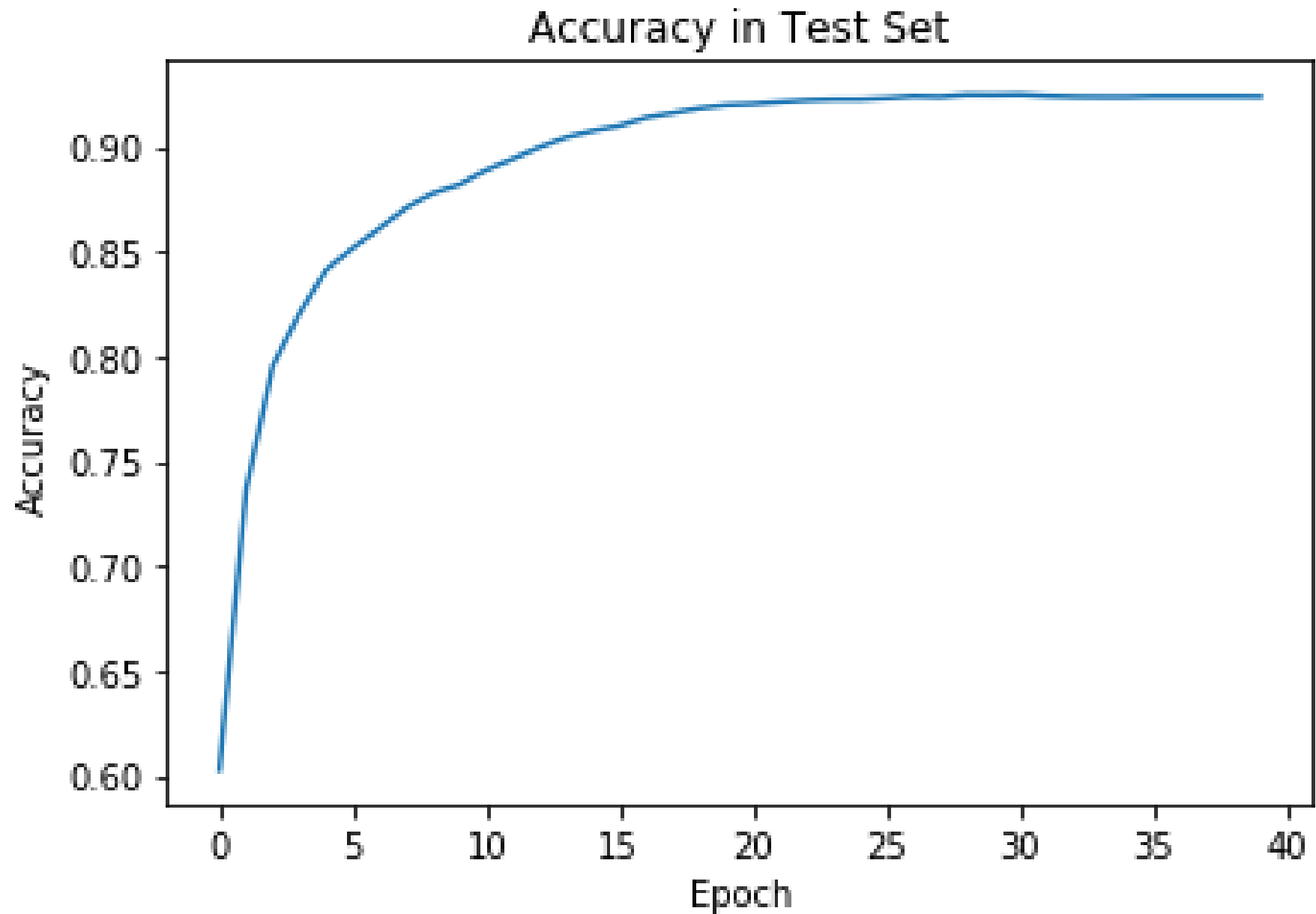
Hàm lựa chọn mẫu giúp lựa chọn những mẫu chứa nhiều thông tin hữu ích nhất, có ý nghĩa quyết định tới hiệu suất thuật toán. 2 hàm lựa chọn mẫu thường dùng

- Entropy selection: Lựa chọn mẫu có entropy dựa trên phân phối model dự đoán cao nhất. Entropy càng cao tức phân phối càng gần phân phối đều, tức model dự đoán không chắc chắn.
- Margin Selection: Chọn k mẫu sao độ chênh lệch xác suất giữa 2 lớp có xác suất cao nhất là nhỏ nhất. Cao khi rất chắc chắn, thấp khi các lớp tương tự nhau.
  - Hàm chọn mẫu giúp chọn ra các đối tượng mà model đưa ra dự đoán không chắc chắn, bỏ qua các đối tượng mà model đã dự đoán tốt.

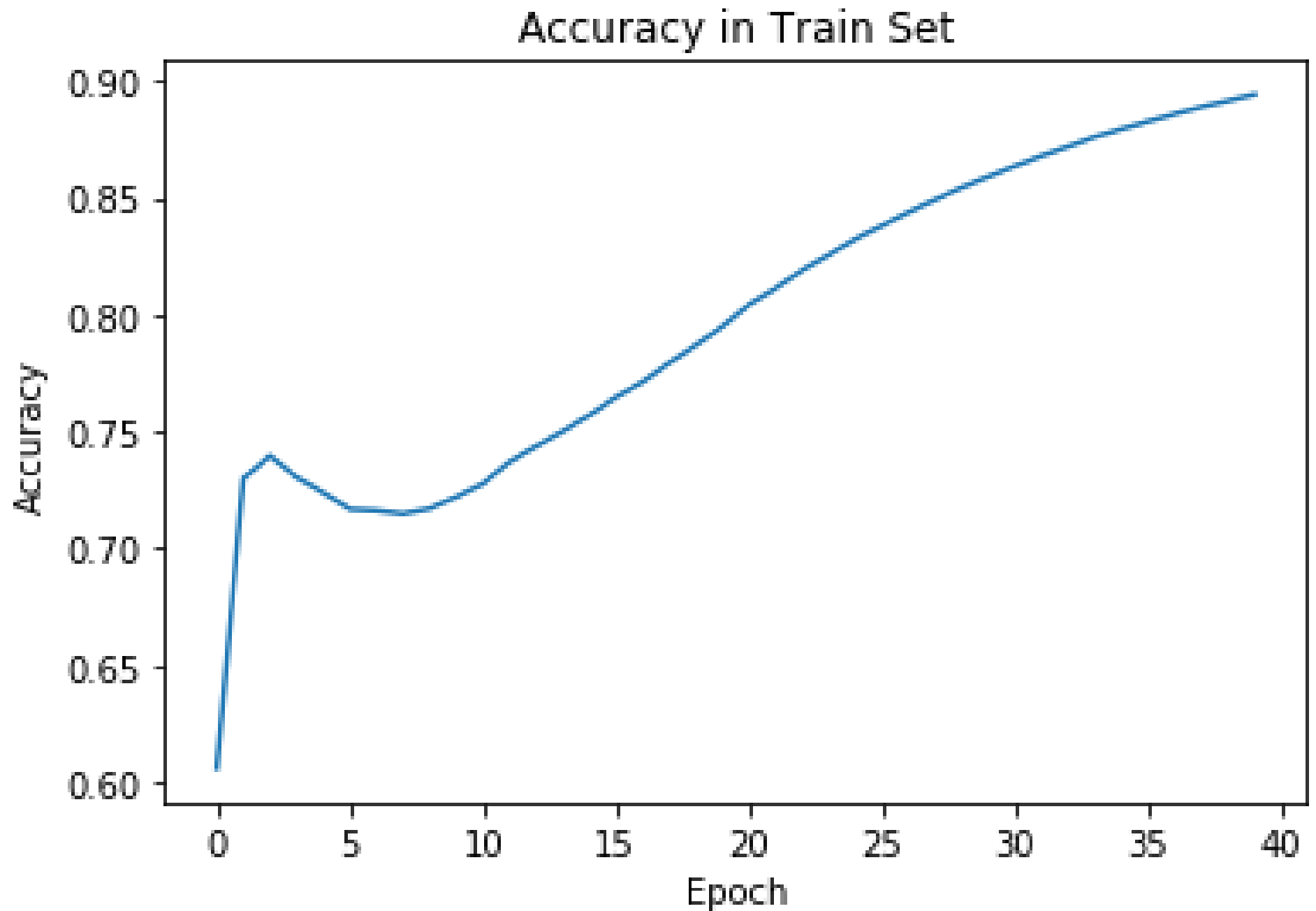
# 9. Tổng hợp

Bắt đầu từ mô hình mạng nơ ron với 1 lớp FullyConnected và activation function Softmax, loss function categorical\_crossentropy, sử dụng optimizer Stochastic Gradient Descent (SGD), mỗi lần training ta train tối đa 30 epoch, với batchsize=32. Trước tiên, ta chia bộ dữ liệu thành tập train\_full (60000 ảnh), tập test (10000 ảnh). Tách tập train\_full thành tập train (10000 ảnh) và tập validation (50000 ảnh còn lại) để khởi tạo. Sau đó, theo kịch bản Active Learning, mỗi lần ta chọn ra 10000 mẫu chứa nhiều thông tin nhất theo hàm Entropy\_Selection từ tập validation để chuyển xuống tập train và tiến hành training. Trải qua 40 vòng lặp, ta được kết quả độ chính xác trên tập test đạt 92,4% chỉ với bộ train\_set 40000 mẫu. Khá ấn tượng!

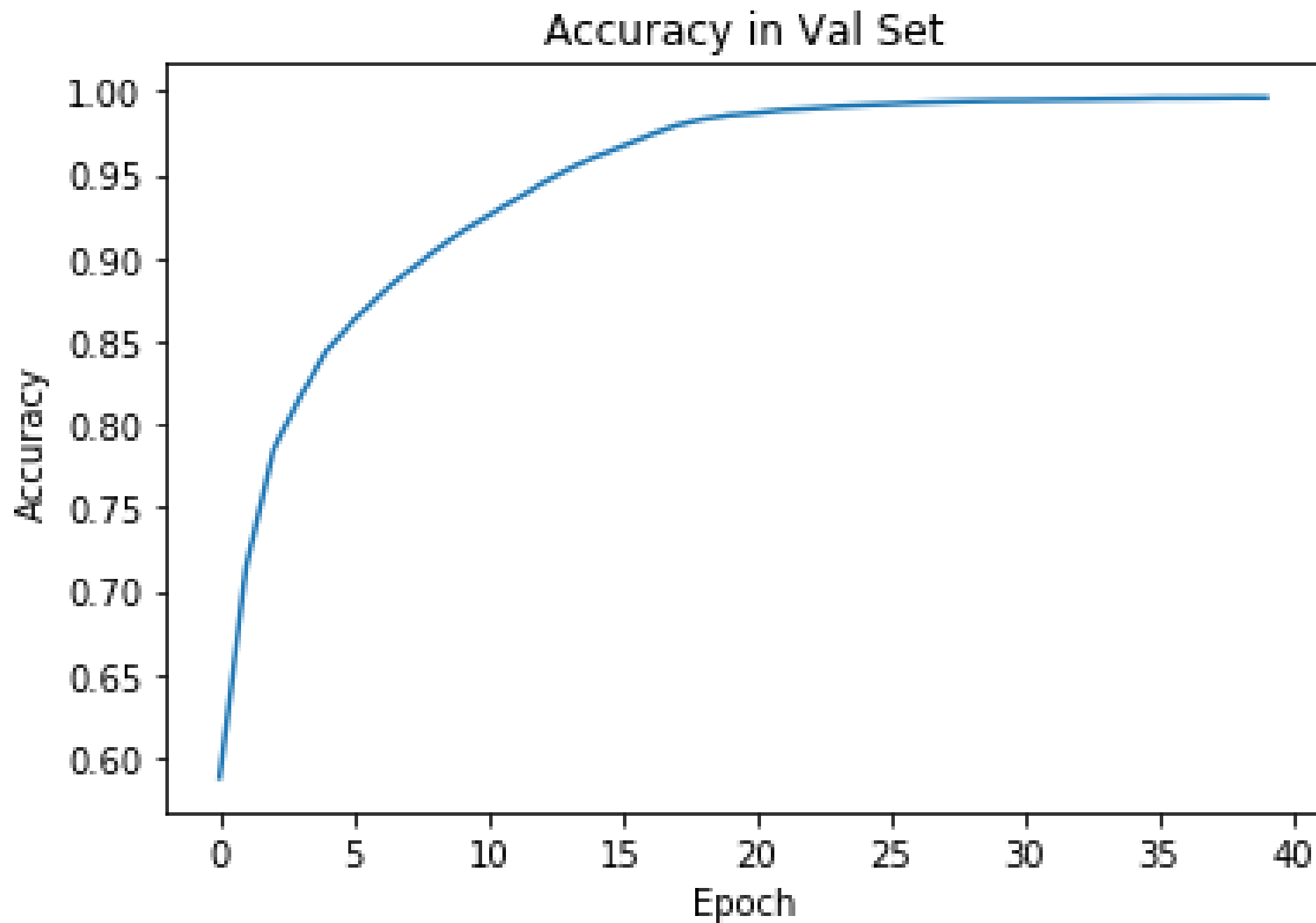
## 10. Kết quả



## 10. Kết quả



## 10. Kết quả



## 10. Kết quả

