

NEURAL NETWORK USING FASHION MNIST DATASET

Nguyễn Việt Đức, Nguyễn Kim Toàn

ABSTRACT

Xây dựng mạng nơ-ron truyền thống để phân loại các lớp trên bộ dữ liệu Fashion Mnist.

Index Terms— Neural network from scratch, Fashion mnist, Artificial Neural Network, ANN,...

1. INTRODUCTION

Mạng thần kinh nhân tạo được lấy cảm hứng từ mạng neural sinh học. Bản thân một nơ-ron đơn lẻ tương đối vô dụng, nhưng khi kết hợp với hàng trăm hoặc hàng nghìn (hoặc nhiều hơn) các nơ-ron khác, sự liên kết tạo ra các mối quan hệ và kết quả có thể vượt trội hơn bất kỳ phương pháp học máy nào khác.

Sử dụng Mạng nơ-ron truyền thống này trên bộ dữ liệu Fashion mnist có thể nhận biết được hình ảnh đầu vào là áo, quần, đầm, áo choàng,... dựa trên việc huấn luyện một tập các hình ảnh đầu vào.

Bài toán này có tính tổng quát hơn việc bắt đầu xây dựng mô hình trên bộ chữ số viết tay Mnist.

Mô hình gồm 2 lớp ẩn sử dụng hàm kích hoạt ReLU và lớp đầu ra sử dụng hàm kích hoạt softmax. Bên cạnh đó sử dụng Cross-entropy loss, Adam optimizer, và Categorical accuracy.

2. METHOD

Tiền xử lý dữ liệu: đầu tiên, quét thư mục huấn luyện, lặp qua các thư mục và các hình ảnh bên trong, sau đó thêm vào danh sách có tên là X cùng với các nhãn tương ứng vào một biến danh sách y để tạo thành các mẫu. Chúng ta cần thực hiện thao tác này trên cả dữ liệu thử nghiệm và huấn luyện.

Chia tỷ lệ dữ liệu: mạng nơ-ron có xu hướng hoạt động tốt nhất với dữ liệu trong phạm vi từ 0 đến 1 hoặc -1 đến 1. Ở đây, dữ liệu hình ảnh nằm trong phạm vi 0 đến 255 vì vậy bằng cách lấy mỗi giá trị pixel, trừ một nửa giá trị tối đa của tất cả các giá trị pixel (tức là $255/2 = 127,5$), sau đó chia cho nửa này để tạo ra phạm vi giới hạn bởi -1 và 1.

Làm phẳng dữ liệu: chúng ta cần lấy những hình ảnh 28x28 này và làm phẳng chúng, có nghĩa là lấy từng hàng của một mảng hình ảnh và nối nó vào hàng đầu tiên của mảng đó. Cuối cùng thu được 60.000 mẫu gồm 784 tính năng. Sau đó xáo trộn đồng thời dữ liệu cả trên tập dữ liệu mẫu và nhãn tương ứng để tránh các lớp lệch khi huấn luyện.

Kiến trúc mạng sử dụng: 2 lớp ẩn, mỗi lớp 64 nơ-ron sử dụng hàm kích hoạt ReLU. Lý do sử dụng hàm kích hoạt ReLU: đơn giản, mạnh mẽ và được sử dụng rộng rãi với tốc độ tính toán nhanh, hiệu quả. Lớp đầu ra có 10 nơ-ron sử dụng hàm kích hoạt softmax.

Bước 1: Khi đã có input x, đặt hàm kích hoạt a cho lớp đầu vào:

$$z = wx + b$$

$$a = \sigma(z)$$

Kết quả này sẽ làm đầu vào cho lớp tiếp theo và tương tự.

Bước 2: Với mỗi L(layer đang xét)

$$z^L = w^L a^{L-1} + b^L$$

$$a^L = \sigma(z^L)$$

Bước 3: Tính vector lỗi

$$\delta^L = \nabla_a C \odot \sigma'(z^L)$$

trong đó: $\nabla_a C = a^L - y$ biểu thị độ thay đổi của hàm chi phí C với đầu ra a.

Bước 4: Lan truyền ngược: ở mỗi lớp, tính toán

$$\delta^l = (w^{l+1})^T \delta^{l+1} \odot \sigma'(z^l)$$

Đây là lỗi tổng quát cho bất kỳ lớp nào. Điều này cho phép chúng ta điều chỉnh trọng số và độ lệch để giúp giảm thiểu hàm chi phí. Sử dụng thuật toán tối ưu Adam để tự điều chỉnh tốc độ học bắt đầu với các bước lớn hơn, sau đó nhỏ hơn khi nhận thấy độ dốc gần bằng không. Đây được gọi là Adaptive gradient descent.

Để huấn luyện một mô hình, chúng ta điều chỉnh trọng số và độ lệch để cải thiện độ chính xác và độ tin cậy của mô hình. Để làm điều này, chúng ta tính toán mô hình có bao nhiêu lỗi. Hàm lỗi, còn được gọi là hàm chi phí, là thuật toán xác định độ sai của một mô hình.

Một trong những hàm mất mát được sử dụng phổ biến là hàm lỗi bình phương. Với bài toán này, chúng ta đang phân loại vì vậy chúng ta nên sử dụng một hàm lỗi khác. Hàm lỗi được sử dụng phổ biến nhất với lớp đầu ra sử dụng hàm softmax là Categorical cross-entropy. Minh họa bằng giả mã:

```
softmax output = [0.7, 0.1, 0.2]
target output = [1, 0, 0]
loss = -(log(softmax output[i])*target output[i])
Ngắn gọn hơn: loss = -log(softmax output[i])
```

Độ chính xác (accuracy) là một thang đo thường được dùng trong thực tế. Chúng ta sẽ sử dụng các giá trị argmax từ các đầu ra softmax và sau đó so sánh các giá trị này với các target. Minh hoạ bằng giả mã:

```
accuracy = mean(predictions==class targets)
```

3. EXPERIMENTAL RESULTS AND ANALYSIS

Để kiểm tra tính hiệu quả của mạng, sử dụng tập dữ liệu Fashion MNIST là tập hợp của 60.000 mẫu đào tạo-6.000 mẫu cho mỗi phân loại. và 10.000 mẫu thử nghiệm với hình ảnh 28x28 của 10 mặt hàng quần áo khác nhau như giày, ủng, áo sơ mi, túi xách, v.v Tập dữ liệu gốc bao gồm các tệp nhị phân chứa dữ liệu hình ảnh ở một định dạng cụ thể. Những hình ảnh này cũng được nhóm theo nhãn và được tách thành các nhóm huấn luyện và thử nghiệm. Các mẫu là hình ảnh của các mặt hàng quần áo, và nhãn là các phân loại. Dưới đây là các nhãn số và mô tả tương ứng của chúng:

| LABEL | DESCRIPTION |
|-------|---------------------------------|
| 0 | T-shirt/top (áo phông/áo sơ mi) |
| 1 | Trouser (Quần) |
| 2 | Pullover (áo chui đầu) |
| 3 | Dress (Đầm) |
| 4 | Coat (áo choàng) |
| 5 | Sandal (dép) |
| 6 | Shirt (áo sơ mi) |
| 7 | Sneaker (giày) |
| 8 | Bag (túi xách) |
| 9 | Ankle boot (giày cao cổ) |

Bảng 1: Mô tả bộ dữ liệu

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| T-shirt/Top | 0.84 | 0.82 | 0.83 | 1000 |
| Trouser | 0.98 | 0.97 | 0.97 | 1000 |
| Pullover | 0.78 | 0.80 | 0.79 | 1000 |
| Dress | 0.86 | 0.89 | 0.88 | 1000 |
| Coat | 0.79 | 0.79 | 0.79 | 1000 |
| Sandal | 0.92 | 0.96 | 0.94 | 1000 |
| Shirt | 0.69 | 0.67 | 0.68 | 1000 |
| Sneaker | 0.95 | 0.90 | 0.92 | 1000 |
| Bag | 0.96 | 0.97 | 0.96 | 1000 |
| Ankle Boot | 0.94 | 0.95 | 0.95 | 1000 |
| accuracy | | | 0.87 | 10000 |
| macro avg | 0.87 | 0.87 | 0.87 | 10000 |
| weighted avg | 0.87 | 0.87 | 0.87 | 10000 |

Bảng 2: Kết quả đánh giá mô hình trên bộ dữ liệu thử nghiệm

Mô hình đạt được độ chính xác khá tốt là 87% tức là dự đoán đúng 87% trên tổng số 10.000 mẫu trong tập thử nghiệm, một con số khá ấn tượng. Việc tăng số nơ-ron trên mỗi lớp ẩn từ 64 lên 128, tăng số lượng epoch từ 5 lên 10 làm mô hình trở nên lớn hơn cũng làm tăng độ chính xác của mô hình.

Việc tăng số lớp ẩn từ 2 lớp lên 3 lớp không làm thay đổi đáng kể độ chính xác của mô hình.

4. CONCLUSION

Ngày nay với sự phát triển của công nghệ, việc xây dựng mạng nơ-ron nhân tạo không còn là một ý tưởng quá xa vời, đây là cách xây dựng mô hình mạng nơ-ron truyền thống từ đầu nên không thể tránh khỏi việc những dòng mã rất dài và có phần khó hiểu.

Tóm lại, bài báo cáo này chỉ tóm lược cách một mạng neural truyền thống hoạt động và không đề cập đến những khía cạnh như chính quy hóa, dropout, những hàm kích hoạt ít phổ biến hơn, cũng như một số thuật toán tối ưu như SGD, RMSProp,...

5. REFERENCES

[1] Harrison Kinsley Daniel Kukieta. Neural Network from scratch in Python[Online]. Available from: https://www.youtube.com/watch?v=Wo5dMEP_BbI&list=PLQVvva0QuDcjD5BAw2DxE6OF2tius3V3

[2] Fashion Mnist dataset [Online]. Available from: <https://www.kaggle.com/zalando-research/fashionmnist>

[3] Jose Portilla. Python for Data Science and Machine Learning Bootcamp [Online]. Available from: <https://www.udemy.com/course/python-for-data-science-and-m>