Đồ Án Cuối Kỳ

ĐÁNH GIÁ TỰA SÁCH

BÁO CÁO ĐỒ ÁN CUỐI KỲ
KHOA HỌC DỮ LIỆU

Người thực hiện
Võ Phương Hòa - 1412192

Giáo viên phụ trách
Trần Trung Kiên

Mục lục

I. Giới thiệu đề tài

Đọc sách là sở thích, đồng thời cũng là nhu cầu cơ bản của con người trong cuộc sống thường nhật. Đáp ứng nhu cầu này, thị trường sách không ngừng cung cấp cho người đọc khối lượng sách khổng lồ cả về số lượng, nội dung, chủ đề, chất lượng… Sách càng nhiều dẫn đến những khó khăn khi người đọc muốn tìm kiếm một quyển sách ưng ý, xứng đáng để đọc. Bởi lẽ, có nhiều đầu sách không đáp ứng được kỳ vọng người đọc cả về chất lượng lẫn nội dung. Một câu hỏi được đặt ra là làm thế nào để biết trước một cuốn sách có đáng đọc hay không, nhằm tránh cho người đọc phải lãng phí thời gian lẫn tiền bạc.

Từ vấn đề nêu trên, chúng tôi quyết định thực hiện một bài toán về đánh giá tựa sách nhằm cung cấp cho người đọc một phương thức đánh giá tựa sách để lựa chọn đầu sách phù hợp.

Trước khi đi vào tìm hiểu và giải quyết bài toán, chúng tôi hi vọng các bạn có một cái nhìn tổng quan về học máy – cách thức mà chúng tôi hướng đến trong vấn đề giải quyết bài toán này.

II. Đôi nét về học máy

Học máy, có tài liệu gọi là Máy học, (tiếng Anh: *machine learning*) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác (spam) hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ.

Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán **NP-khó**, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được.

Học máy có hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt (*robot locomotion*).

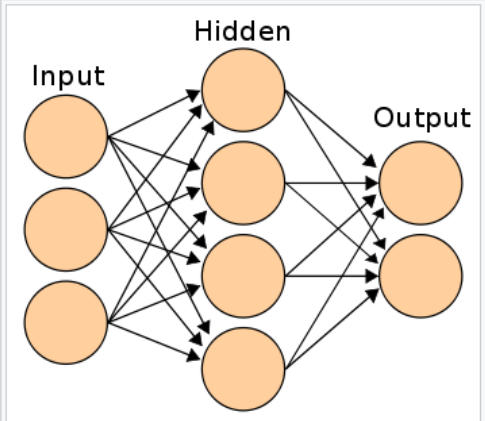
Dưới góc nhìn của trí tuệ nhân tạo, động lực chính học máy là nhu cầu thu nhận tri thức (knowledge acquisition). Thật vậy, trong nhiều trường hợp ta cần kiến thức chuyên gia là khan hiếm (không đủ chuyên gia ngồi phân loại lừa đảo thẻ tín dụng của tất cả giao dịch hàng ngày) hoặc chậm vì một số nhiệm vụ cần đưa ra quyết định nhanh chóng dựa trên xử lý dữ liệu khổng lồ (trong mua bán chứng khoán phải quyết định trong vài khoảng khắc của giây chẳng hạn) và thiếu ổn định thì buộc phải cần đến máy tính. Ngoài ra, đại đa số dữ liệu sinh ra ngày nay chỉ phù hợp cho máy đọc (computer readable) tiềm tàng nguồn kiến thức quan trọng. Máy học nghiên cứu cách thức để mô hình hóa bài toán cho phép máy tính tự động hiểu, xử lý và học từ dữ liệu để thực thi nhiệm vụ được giao cũng như cách đánh giá giúp tăng tính hiệu quả.

Học máy bao gồm nhiều hướng nghiên cứu hết sức đa dạng. Một hướng nghiên cứu của nó là mô hình hóa các hàm mật độ xác suất điều kiện: hồi quy và phân loại bằng mạng neural nhân tạo (Artificial Neural Network).

Đối với bài toán đánh giá tựa sách mà chúng tôi đang đề cập, ANN là phương pháp mà chúng tôi hướng tới và sẽ trình bày cụ thể ở phần dưới đây.

III. Đánh giá tựa sách bằng mô hình ANN

1. Đôi nét về ANN



Một mạng nơ-ron là một nhóm các nút nối với nhau, gần giống như mạng khổng lồ các nơ-ron trong não người.

Mạng neurals nhân tạo hay thường gọi ngắn gọn là mạng neurals là một mô hình toán học hay mô hình tính toán được xây dựng dựa trên các mạng neurals sinh học. Nó gồm có một nhóm các neurals nhân tạo (nút) nối với nhau, và xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị mới tại các nút (cách tiếp cận connectionism đối với tính toán). Trong nhiều trường hợp, mạng neurals nhân tạo là một hệ thống thích ứng (*adaptive system*) tự thay đổi cấu trúc của mình dựa trên các thông tin bên ngoài hay bên trong chảy qua mạng trong quá trình học.

Trong thực tế sử dụng, nhiều mạng neurals là các công cụ mô hình hóa dữ liệu thống kê phi tuyến. Chúng có thể được dùng để mô hình hóa các mối quan hệ phức tạp giữa dữ liệu vào và kết quả hoặc để tìm kiếm các dạng/mẫu trong dữ liệu.

2. Sử dụng ANN để huấn luyện mô hình học

- Thuật toán:

- Stochastic Gradient Descent (SGD)
- Backpropagation Algorithm (thuật toán lan truyền ngược).

- Hàm kích hoạt:

- Sigmoid function (đối với tầng ẩn)
- Softmax function (đối với tầng output)

- Hàm chi phí: Mean Negative Log Likelihood

- Hàm tính độ lỗi: Mean Binary Error (MBE)

- Regularization: Early Stopping/Weight Decay

2.1. Cấu trúc mạng neurals cho bài toán

Dữ liệu thu thập cho bài toán gồm 29705 mẫu, 10 thuộc tính. Sau quá trình tiền xử lý, chúng tôi loại bỏ 3 thuộc tính, chuẩn hóa các thuộc tính còn lại về dạng thích hợp. Như vậy, mạng neurals trong bài toán có dạng như sau:

- Input layers: 6 neurals
- Hidden layers: 1 lớp ẩn gồm 50 neurals
- Output layers: 5 neurals (phân thành 5 lớp từ một thuộc tính được chọn làm nhãn).

(Quá trình tiền xử lý chúng tôi sẽ nêu cụ thể ở phần sau).

2.2. Các bước huấn luyện

Dùng thuật toán SGD (và backpropagation):

- Khởi tạo ma trận trọng số W (các trọng số có giá trị nhỏ, thường có mean = 0, std = 1)
- Lặp cho tới khi thỏa điều kiện dừng (xét hết mẫu):
    - Xáo trộn thứ tự của các mẫu huấn luyện

- Với mỗi minibatch mb (kích thước B):
  - Với mỗi mẫu (x, y) ϵ mb :

    Forward : $x \rightarrow a^{(1)} \rightarrow a^{(2)} \rightarrow \ldots \rightarrow a^{(L)}$

    Dùng hàm sigmoid để tính output của các tầng ẩn

    $a_i = 1/(1+e^{-z})$ với $z = w_i^T x$

    Dùng hàm sofmax để tính output của tầng cuối cùng

$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}, \quad \forall i = 1, 2, \ldots, C$$

    (tầng output).

    Backward : Tính độ lỗi từ tầng output đến tầng ẩn đầu tiên

$$\delta^{(1)} \leftarrow \cdots \leftarrow \delta^{(L-1)} \leftarrow \delta^{(L)}$$

$$\delta_i^{(l)} = \frac{\partial e}{\partial s_i^{(l)}} = \theta'\left(s_i^{(l)}\right) \sum_{k=1}^{d^{(l+1)}} W_{ik}^{(l+1)} \delta_k^{(l+1)}$$

    Tính gradient: $\nabla_W e(h(x), y)$: $\frac{\partial e}{\partial W_{ji}^{(l)}} = a_j^{(l-1)} \delta_i^{(l)}$

  - Cập nhật trọng số

$$W \leftarrow W - \alpha * \frac{1}{B} \sum_{(x,y) \in mb} \nabla_W e(h(x), y)$$

<u>Sử dụng regularization trong huấn luyện</u>

- Weight decay: tại mỗi bước cập nhật trọng số W,

  bổ sung: $\mathbf{W \leftarrow W - 2\eta\lambda W}$.
- Early stopping: theo dõi và cập nhật độ lỗi tốt nhất (nhỏ nhất) MBE trên tập validation. Kết quả bộ trọng số trả về từ hàm huấn luyện tương ứng với độ lỗi tốt nhất MBE trên tập validation.

IV. Chương trình thực thi

Ngôn ngữ: Python 2.7

1. Chương trình thu thập dữ liệu

- Tên chương trình: **ThuThapDuLieuWeb.ipynb**

- Mục đích: thu thập dữ liệu cho các tập training, validation, test

- Nguồn: https://tiki.vn/nha-sach-tiki

- Cấu trúc chương trình:

- Thư viện sử dụng: requests, re, pandas, BeautifulSoup (from bs4)
- Danh sách các url được dùng để lấy thông tin
- Chương trình đọc dữ liệu từ các url
- Chuẩn hóa dữ liệu (tiền xử lý giai đoạn thu thập dữ liệu)
- Ghi dữ liệu vào file

=> Output của chương trình là các tập dữ liệu training set (20000 mẫu), validation set(4705 mẫu), test set (5000 mẫu). Mỗi mẫu gồm 10 thuộc tính (id, title, topic, author, nxb, date, rating, rating_Count, price, reviews).

2. Chương trình huấn luyện

- Tên chương trình: **DACK_DanhGiaTuaSach.ipynb**

- Mục đích:

- Thực hiện huấn luyện đối với dữ liệu thuộc training set.
- Kiểm tra độ lỗi trong quá trình huấn luyện đối với validation set.
- Kiểm tra kết quả huấn luyện cuối cùng với test set.

- Cấu trúc chương trình:

- Tiền xử lý dữ liệu:
  - Loại bỏ cột ‘id’, ‘title’, ‘reviews’.
  - Chuẩn hóa dữ liệu các cột có datatypes dạng chuỗi về dạng numeric.
  - Phân lớp dữ liệu thuộc tính rating: 5 lớp

    rating ϵ [0.0, 1.0): rating = 0

rating ϵ [1.0, 2.0): rating = 1

rating ϵ [2.0, 3.0): rating = 2

rating ϵ [3.0, 4.0): rating = 3

rating $\geq$ 4.0: rating = 4

- Huấn luyện dữ liệu:
  - Các hàm sigmoid(), softmax(), compute_nnet_outputs() dùng để tính output của Neural Network.
  - Hàm train_nnet(): huấn luyện mạng neurals
    ⇨ Kết quả trả về :
    - Bộ trọng số của mạng neurals Ws
    - Độ lỗi trên training set và validation set
- Kiểm tra kết quả huấn luyện:
  - Kiểm tra độ lỗi trên training set và validation set
  - Kiểm tra hiệu suất trên test set

V. Thực nghiệm

Các tham số mặc định:

- Số lớp ẩn: 1

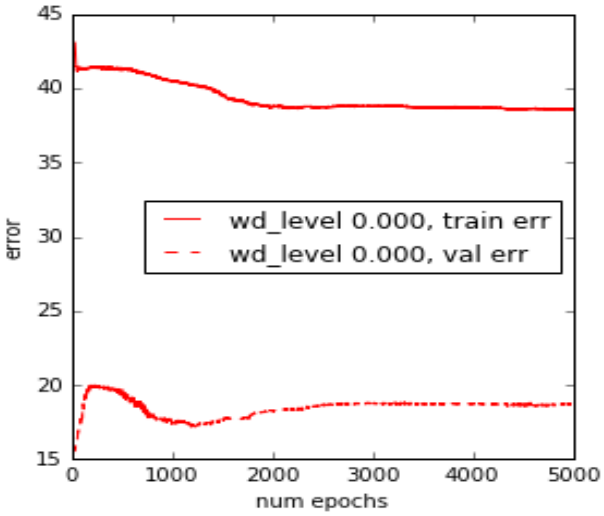
- Số neural trên lớp ẩn: 50

- Kích thước minibatch: 40

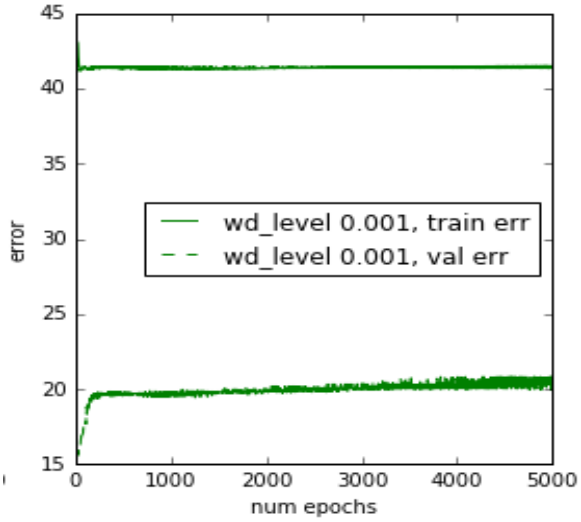
- Learning_rate = 0.0005 (riêng đối với early stopping: learning_rate = 0.1)

- Epoch = 5000 (trừ trường hợp early stopping: max_epoch = 1000000)

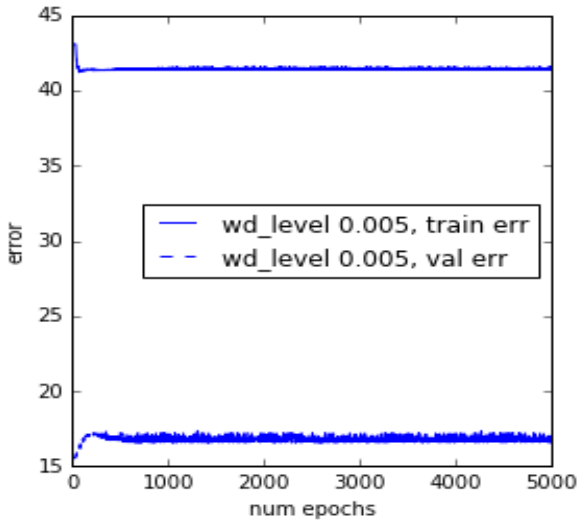
Error			**train_err**	**val_err**	**test_err**
Non-regularization			38.575%	18.661%	
Regularization	Weight Decay	wd = 0.001	41.450%	20.191%	
		wd = 0.005	41.430%	16.642%	
	Early Stopping	max_patience = 3000	18.465%	7.97%	11.32%



H.1.1. Non-regularization



H.1.2. Weight Decay (wd_level = 0.001)



H.1.1. Weight Decay (wd_level = 0.005)

Đối với trường hợp non-regularization (H.1.1), sau những biến động mạnh trong 2000 epoch đầu, từ khoảng epoch 2000 về sau, độ lỗi error giảm dần chậm ở cả train_err và val_err và dao động lần lượt là 38% và 18%.

Trong khi đó, với regularization là weight decay.

- Ở trường hợp thứ nhất (H.1.2.), wd_level = 0.001, độ lỗi ở train_err sau khi giảm mạnh ở vài epoch đầu thì có vẽ chững lại xuyên suốt các epoch về sau, còn val_err tăng dần chậm theo epoch.

- Ở trường hợp thứ hai (H.1.3), wd_level = 0.005, sau vài biến động mạnh ở khoảng 200 epoch đầu, độ lỗi train_err và val_err giảm dần chậm theo số epoch.

Với regularization là early stopping, độ lỗi biến động mạnh trong suốt quá trình huấn luyện và đạt kết quả tốt nhất sau khoảng 3000 epoch với độ lỗi train_err và val_err đạt giá trị thấp (lần lượt là 18.465% và 7.97%).

Như vậy từ kết quả thu được, dễ thấy hàm huấn luyện với non-regularization cho kết quả kém khả quan hơn việc sử dụng regularization. Mặt khác, trong 2 regulization được thử nghiệm, early stopping cho kết quả khả quan hơn cả.

(Kết quả test_err chúng tôi chỉ kiểm tra với mô hình huấn luyện cho kết quả tốt nhất, ở đây là mô hình sử dụng early stopping).

VI. Tài liệu tham khảo

1. Slide và bài tập môn học Khoa học dữ liệu
2. https://docs.scipy.org/doc/numpy-dev/user/quickstart.html
3. https://tiki.vn/nha-sach-tiki (nguồn thu thập dữ liệu)
4. https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_m%C3%A1y
5. https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_n%C6%A1-ron_nh%C3%A2n_t%E1%BA%A1o