Dự án Cuối kỳ

Môn: Nhập môn Học Máy

**Bài 1 (3 điểm): làm riêng từng người**

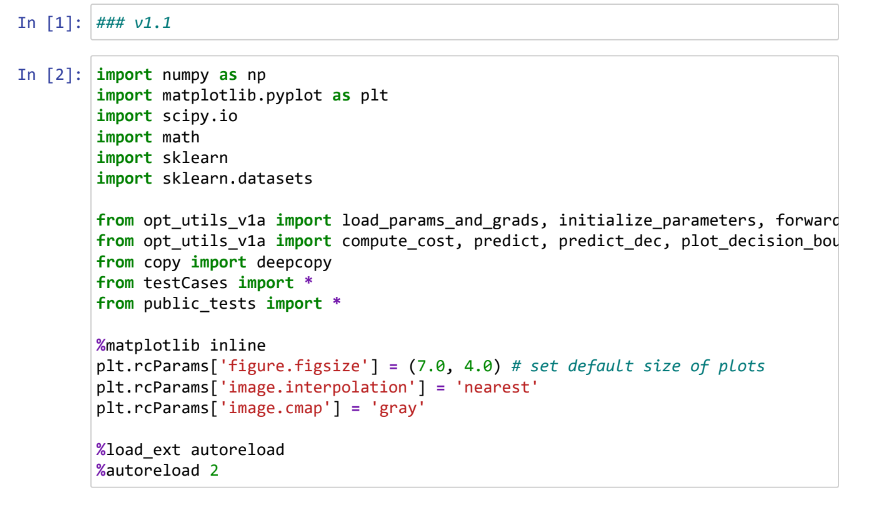
Trình bày một bài nghiên cứu, đánh giá của em về các vấn đề sau:

1. Tìm hiểu, so sánh các phương pháp Optimizer trong huấn luyện mô hình học máy;
   1. Optimizer là gì?

Optimizer còn được hiểu là “trình tối ưu hóa”, là một thành phần quan trọng trong quá trình training model maching leaning. Đảm nhận nhiệm vụ điều chỉnh các trọng số của mô hình để giảm thiểu loss funcion (mất hàm) – một bộ đo của sự chênh lệch giữa các giá trị dự đoán vả giá trị thực tế. Khi ta đào tạo mô hình học máy, mỗi lần đưa dữ liệu vào mô hìn, mô hình sẽ tạo ra dự đoán và sau đó ta sẽ so sánh các dự đoán này với các giá trị thực tế bằng cách sử dụng các loss function và đo lường mức độ mà mô hình khi đã training so với dữ liệu thực tế.

Optimizer có nhiệm vụ điều chỉnh các tham số (hoặc trọng số) của mô hình để giảm thiểu già trị của loss function. Quá trình này thường được thực hiện thông qua việc sử dụng các phương pháp tối ưu hóa như Gradient Descent (hạ gradient) và các biến thể của nó.

Một só Optimeizers phổ biến trong học máy là:Gradient Descent ( GD), Stochasis Gradient Descent ( SGD), Mini- batch Gradient Descent, Momentum, Nesterov Accelearated Gradient ( NAG), Adagrad, RMSprop, Adam, Adadelta, FTRL- Proximal.  
1.2 Phân tích các phương pháp Optimizer trong huấn luyện mô hình học máy

1.2.1 Packages  


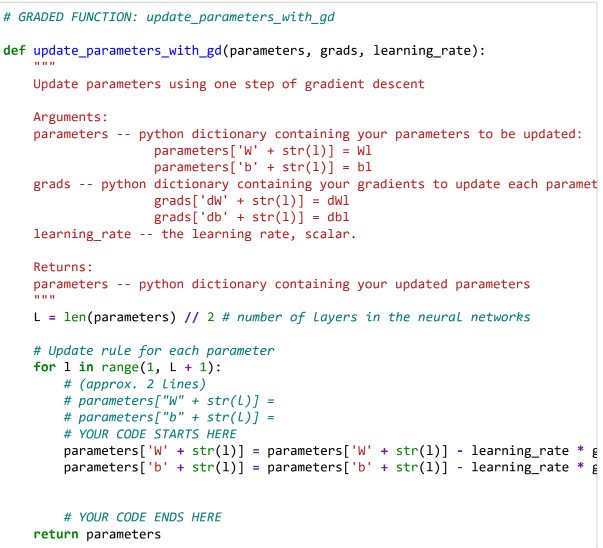
1.2.2 Gradient Descent

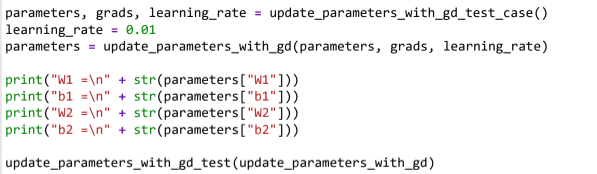
- Gradient descent (GD) Là một trong phương pháp tối ưu hóa đơn giản trong machine learning. Khi bạn thực hiện các bước gradient đối với tất cả các ví dụ trong mỗi bước, nó được gọi là Batch Gradient Descent.

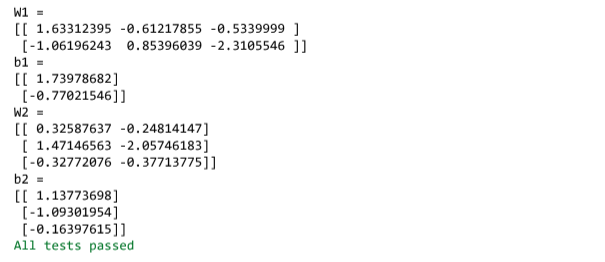
- Triển khai quy tắc cập nhật Gradient,for l = 1, . . . , L:



-Ở đây, L là số lớp và α là tốc độ học. Tất cả các tham số nên được lưu trữ trong







1.3 So sánh phương pháp Optimizer trong mô hình học máy.

|  |  |  |
| --- | --- | --- |
| Phương pháp | Ưu điểm | Nhược đi |
| Gradient Descent ( GD) | - Dễ hiểu và triển khai.  - Hoạt động tốt trên các mô hình đơn giản.  - Giải quyết vấn đề tối ưu model neural network | - Có thể chậm và không hiệu quả trên các mô hình lớn hoặc phức tạp.  - Phụ thuộc vào điểm khởi tạo ban đầu và learning rate.  - Tốc độ học quá lớn hoặc quá nhỏ có thể ảnh hưởng đến quá trình training. |
| Stochasis Gradient Descent ( SGD) | - Phù hợp cho dữ liệu lớn và phức tạp.  - GIảm độ độ phức tạp tính toán.  - Cập nhập trọng số sau mỗi mẫu dữ liệu, thích hợp cho dữ liệu lớn.  - Cập nhập tham số nhanh chóng. | - Có thể dao động nhiều hơn do tính ngẫu nhiên.  - Đô chính xác có thể bị giảm vì sự ngẫu nhiên trong việc chọn mẫu. |
| Mini- batch Gradient Descent | - Kết hợp lợi ích của GD và SGD  - Phù hợp với nhiều loại dữ liệu | - Cần điều chỉnh kích thước mini- batch |
| Momentum | - Vượt qua Local Mometum  - Giảm độ dao động và giúp vượt qua các đồng lõi  - Thuật toán giảm thiểu đảo hàm và giúp cập nhật nhanh chóng | - Cần chọn thêm hyperparameter. |
| Nesterov Accelearated Gradient ( NAG) | - Cải thiện trên Momentum bằng cách ước lượng gradient trước khi thực hiện cập nhật | - Cần điều chỉnh thêm hyperparameter |
| Adagrad | - Điều chỉnh tỷ học tự động cho từng tham số  - Tự điều chỉnh learning rate.  - Không cần điều chỉnh thủ công giảm công sức và độ rủi ro. | - Tăng dần learning rate có thể dẫn đến việc giảm độ chính xác. |
| RMSprop | - Giải quyết vấn đề tốc độ học giảm dần.  - Cải thiện Adagrad bằng cách giảm độ chậm của learning rate | - Cân điều chỉnh thêm hyperparameter.  - Khả năng cao rơi vào Local Minimum  - Có thể dẫn đến tốc độ học cực kỳ nhỏ và làm cho quá trình training trở nên đóng băng |
| Adam | - Kết hợp Momemtum và RMSprop, hiệu quả trên nhiều loại dữ liệu. | - Cần điều chỉnh thêm hyperparameter.  - Phưc tạp hơn so với ác thuật toán đơn giản như GD và SGD |
| Adadelta | - Không yêu cầu thiết lập learning rate ban đầu | - Cần điều chỉnh hyperparameter |
| FTRL- Proximal | - Phù hợp cho các mô hình với lượng lớn các tính năng | - Cần điều chỉnh hyper pameter |

1. Tìm hiểu về Continual Learning và Test Production khi xây dựng một giải pháp học máy để giải quyết một bài toán nào đó.

**Bài 2: (7 điểm): làm chung trong nhóm**

Đưa ra một bài toán dự đoán có thể giải quyết bằng học máy (machine leanring) với các yêu cầu sau:

* Số Feature/Attribute gồm nhiều kiểu: categorial và numerical;
* Dữ liệu phải chưa được học, thực tập trên lớp và trong bài tập về nhà;

1. Phân tích thống kê trên dữ liệu, vẽ các đồ thị để hiểu bài toán, hiểu dữ liệu. Tìm hiểu các đặc trưng và đánh gía vai trò của các đặc trưng đối với mục tiêu bài toán;
2. Ứng dụng các mô hình học máy cơ bản để giải quyết bài toán, bao gồm cả các mô hình thuộc Ensemble Learing;
3. Sử dụng Feed Forward Neural Network và Reccurent Neural Network (hoặc mô thuộc loại này) để giải quyết bài toán;
4. Áp dụng các kỹ thuật tránh Overfiting trên các mô hình của câu (2) và câu (3) để giải quyết bài toán;
5. Sau khi huấn luyện xong mô hình thì muốn cải thiện độ chính xác, ta sẽ làm gì để giải quyết nó? Phân tích các trường hợp sai, đề ra giải pháp và thực hiện nó, sau đó đánh giá xem có cải tiến so với trước không.

Lưu ý: Tạo project trên Github để làm bài này, chứa code và data;

Nộp file .ipynp và data. Trong file .ipynp ghi rõ tên các thành viên trong nhóm.