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

Optimizer

Trong machine learning hay cụ thể hơn là deep learning, nhiệm vụ của các optimizer hay optimizer algorithms (các thuật toán tối ưu) là những thuật toán giúp điều chỉnh các parameters (weights, biases, learning rates, ...) của model trong suốt quá trình training để làm sao giải thiểu được loss function. Những optimizers thường lặp đi lặp lại việc cập nhật các weights và biases.

Các optimizers có mục tiêu chung là phải giúp cho việc training càng nhanh càng tốt và cho ra các parameters sao cho dự đoán kết quả chính xác.

Một trong những vấn đề trong việc lựa chọn các optimizers là làm thế nào để có thể chọn đúng optimizer cho vấn đề hay ứng dụng cho việc gì đó? Vì trong thực tế, bộ dữ liệu có thể chứa hàng triệu features, hàng triệu instances, chỉ với một lần lặp để cập nhật các parameters cũng đã tốn một khoảng thời gian đáng kể.

Vì để tránh việc các áp dụng các optimizers không hiệu quả, sau đây tôi xin trình bày về những hiểu biết về các optimizers mà tôi tìm hiểu được, như Gradient Descent, Stochastic Gradient Descent, Stochastic Gradient descent with momentum, Mini-Batch Gradient Descent, Adagrad, RMSProp, AdaDelta, và Adam. Và tiếp theo đó là những so sánh các optimizers nhằm phần nào lựu chọn được optimizer phù hợp cho vấn đề của mình.

Những thành phần chung và quan trọng đối với một optimizer:

Epoch: Số lần lặp mà thuật toán chạy với toàn bộ training dataset.

Sample: Từng hàng trong một dataset.

Batch: Biểu thị số lượng samples được lấy cho mỗi lần cập nhật các parameters của model.

Learning rate: một parameter cung cấp cho model một scale (có thể hiểu là bước nhảy) dành cho việc cập nhật weights và biases.

Cost Function/ Loss Function: hàm thể hiện độ sai lệch giữa giá trị dự đoán và giá trị thật sự.

Weights và biases: các parameters của model, các parameters này sẽ liên tục được cập nhật, để cuối cùng có thể giúp model dự đoán chính xác.

Gradient Descent

Gradient Descent là một trong những optimizer cơ bản nhất. Nó được áp dụng cho cả hai vấn đề là regression và classification. Ngoài ra, cả backpropagation trong neural network có sử dụng gradient descent.

Gradient descent chủ yếu là đạo hàm từng phần các parameters gồm weights và bias trong loss function và sau đó cập nhật các parameters này sao cho loss function có giá trị nhỏ nhất (đạo hàm từng phần của loss function tiến tới giá trị là 0).

Trong đó:

|  |  |
| --- | --- |
|  | : tâp hợp các parameters (weights và bias) |
|  | : loss function |
|  | : đạo hàm của loss function tại một bất kì |
|  | : learning rate, gia tốc cho việc tìm global minimum |

Ví dụ loss function trong linear regression:

Trong đó:

|  |  |
| --- | --- |
|  | : điểm dữ liệu thứ |
|  | : target thứ |
|  | : tổng số lượng điểm dữ liệu |

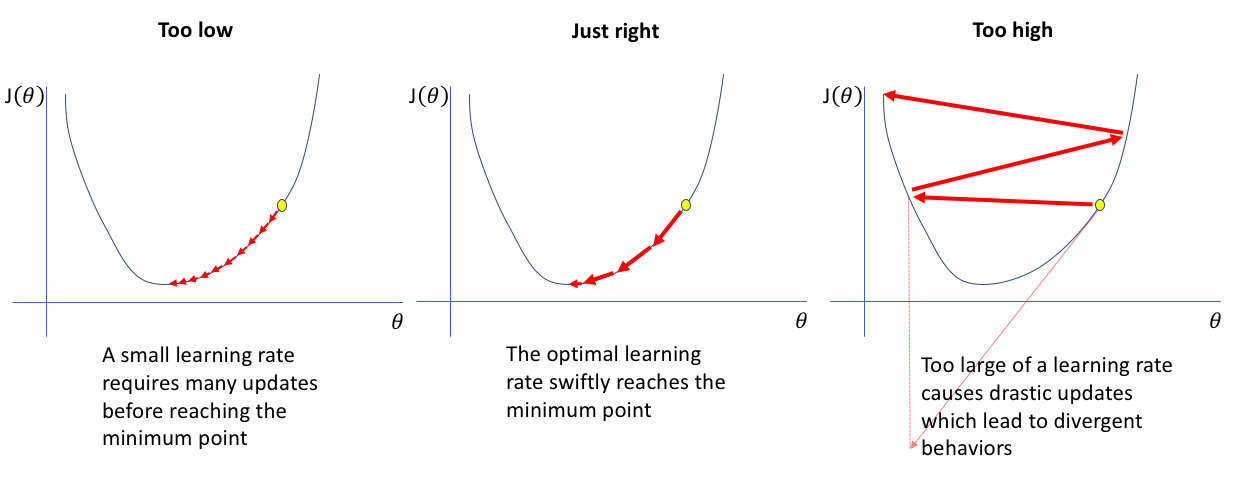
Thuật toán sẽ ngừng lại khi loss function có giá trị xấp xĩ bằng là 0 hoặc đã hết số lần lặp đã cho (dành trong việc cài đặt thuật toán).

Ưu điểm:

Thuật toán gradient descent cơ bản dễ hiểu.

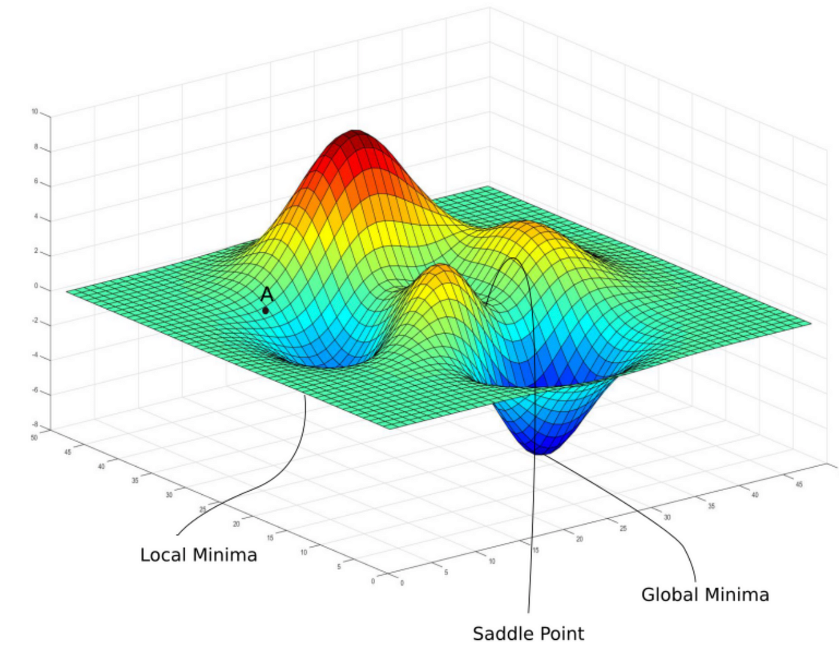
Nhược điểm:

Gradient Descent còn nhiều hạn chế như phụ thuộc vào khởi tạo ban đầu, learning rate không phù hợp như quá lớn chẳng hạn sẽ làm cho parameter dao động xung quanh hoặc tán xạ ra khỏi parameter tối ưu nhất.



Nguồn: <https://www.jeremyjordan.me/nn-learning-rate/>

Việc loss function có nhiều điểm local minimum cũng ảnh hưỡng đến việc tối ưu các parameters.



Nguồn: <https://medium.com/analytics-vidhya/journey-of-gradient-descent-from-local-to-global-c851eba3d367>

Stochastic Gradient Descent

Một tr