VietAI

Bài 8: Các kỹ thuật huấn luyện Neural Networks

Tuần 4B

Neural Network

Outline

1. Gradient descent và một số biến thể
2. Một số vấn đề khi huấn luyện neural network

Neural Network

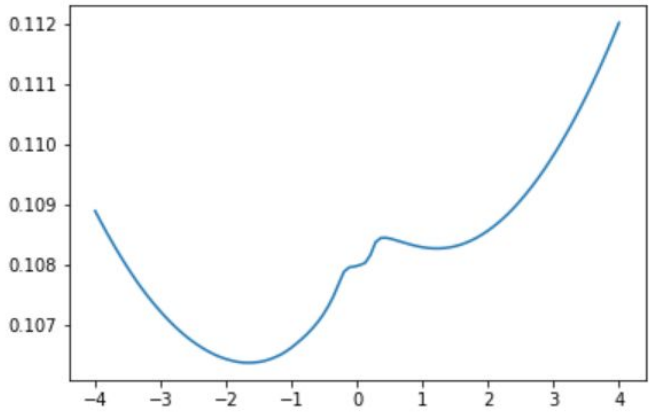
Non-convexity

- Với neural network có 1 hoặc nhiều lớp ẩn (hidden layers), làm Cost thường sẽ không còn là một hàm convex
- Ta phải thay đổi thuật toán gradient descent thế nào để có thể tối thiểu hóa một hàm không convex?

```
a = sess.run(W)  # a weight matrix
a
```

```
array([[ 0.355169  ,  0.1662432 , -1.009055  ,  0.972674  ,  0.3268592 ,
        -0.20735765, -0.1601009 , -0.08534323,  1.8651795 ,  1.3178728 ]
      dtype=float32)
```

```
w_values = np.linspace(start=-4.0, stop=4.0, num=100)
L = []
for w in w_values:
    amod = a.copy(); amod[0,5]=w
    sess.run(W.assign(amod))
    L.append(sess.run(loss, feed_dict={x:xdat.reshape([m,1]), ytrue:ydat
sess.run(W.assign(a))
plt.plot(w_values, L); plt.show()  #weight values vs loss
```



Neural Network

Gradient Descent và một số biến thể

- Sau khi có được đạo hàm cho các tham số, ta cần cập nhật lại chúng:
  - Giải thuật cập nhật kinh điển nhất là Gradient descent
- Ngoài Gradient descent tiêu chuẩn, ra còn có các giải thuật khác:
  - Mini-batch / Stochastic gradient descent - SGD
  - SGD với quán tính (Momentum)
  - Các phương pháp tự động điều chỉnh gradient (adaptive gradient methods)

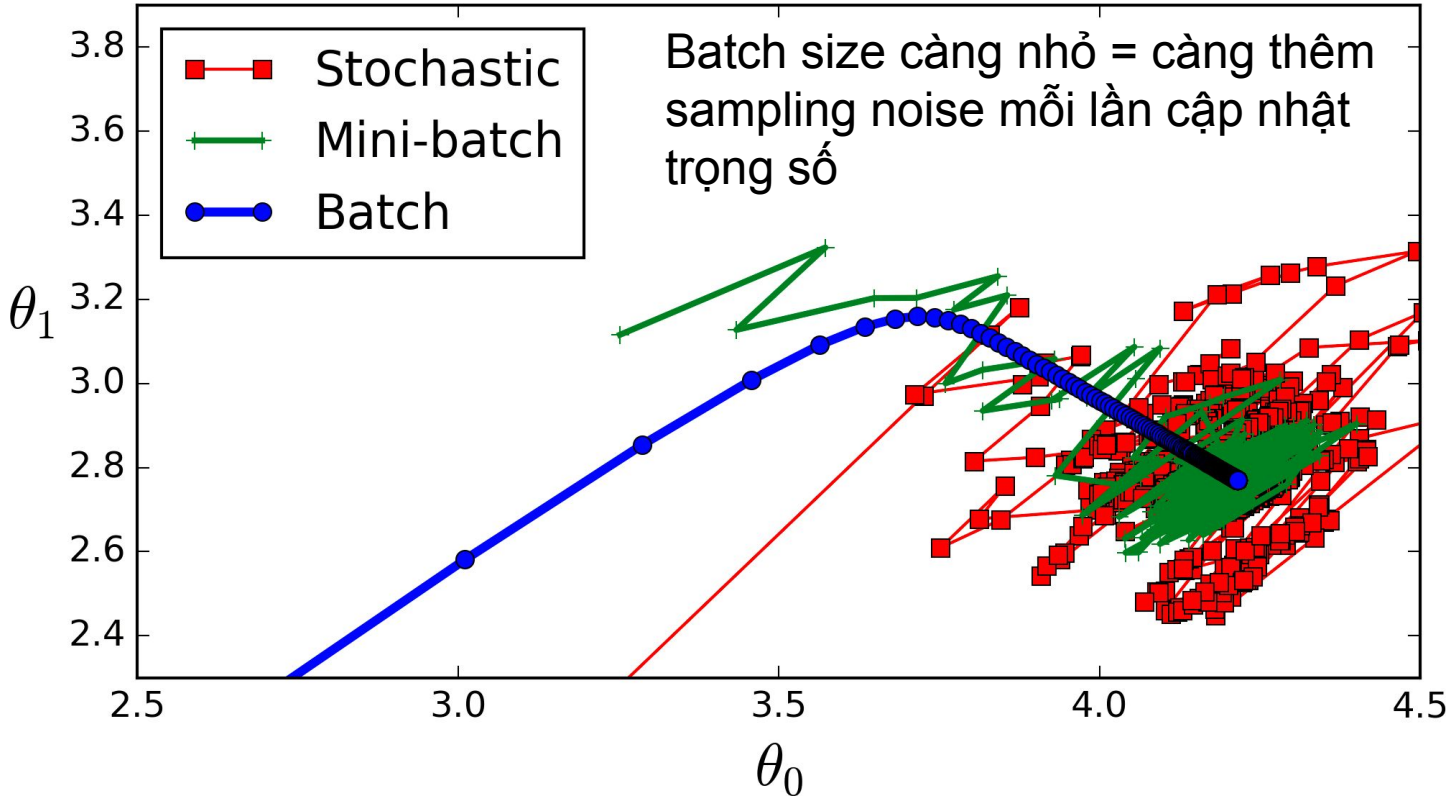
Neural Network

Gradient Descent và một số biến thể

- Các biến thể liên quan đến số lượng dữ liệu:
  - Batch gradient descent: mỗi vòng lặp, dùng tất cả dữ liệu để tính gradient
  - Stochastic gradient descent (SGD): mỗi vòng lặp, dùng 1 mẫu dữ liệu để tính gradient
  - Mini-batch gradient descent: mỗi vòng lặp, dùng một số lượng nhỏ *k* mẫu dữ liệu để tính gradient. Con số *k* được gọi là batch size.
- (Trong thuật ngữ hiện đại, Stochastic gradient descent và Mini-batch gradient descent là như nhau: mỗi vòng lặp dùng *k* mẫu, *k* có thể là 1.)

Neural Network

Gradient Descent và một số biến thể



Neural Network

Stochastic Gradient Descent

Các bước của SGD / Mini-batch gradient descent:

1. Trộn đều (shuffle) dữ liệu
2. Lặp lại tới khi *hài lòng*:
   a. Lấy ra k mẫu cho mini-batch tiếp theo $(x^{(i_1)}, y^{(i_1)}), \ldots, (x^{(i_k)}, y^{(i_k)})$
   b. Lan truyền thuận và tính hàm lỗi *J* của mini-batch trên
   c. Lan truyền ngược, tính gradient ∇_θ
   d. Cập nhật tham số: $\theta := \theta - \alpha \nabla_\theta$

Neural Network

Stochastic Gradient Descent với quán tính

Các bước của SGD / Mini-batch gradient descent với quán tính:

1. Trộn ngẫu nhiên (shuffle) dữ liệu
2. Khởi tạo ma trận quán tính $v := 0 * \theta$
3. Lặp lại tới khi *hài lòng*:
   a. Lấy ra k mẫu cho mini-batch tiếp theo $(x^{(i_1)}, y^{(i_1)}), \ldots, (x^{(i_k)}, y^{(i_k)})$
   b. Lan truyền thuận và tính hàm lỗi *J* của mini-batch trên
   c. Lan truyền ngược, tính gradient $\nabla_\theta J$
   d. Cập nhật quán tính: $v := \mu v + \nabla_\theta J$ *(phiên bản giản lược)*
   e. Cập nhật tham số: $\theta := \theta - \alpha v$

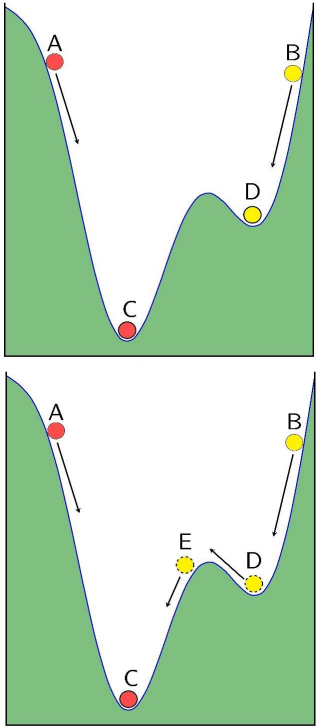
Lưu ý hệ số quán tính: $0 \leq \mu < 1$

Neural Network

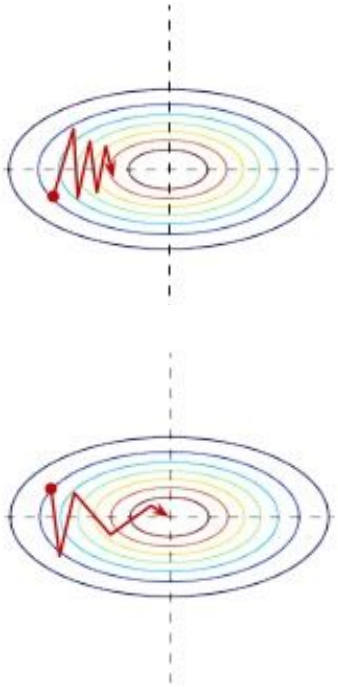
Stochastic Gradient Descent với quán tính

Không dùng quán tính

Có sử dụng quán tính



Minh họa thuật toán khi chỉ có 1 tham số cần tối ưu



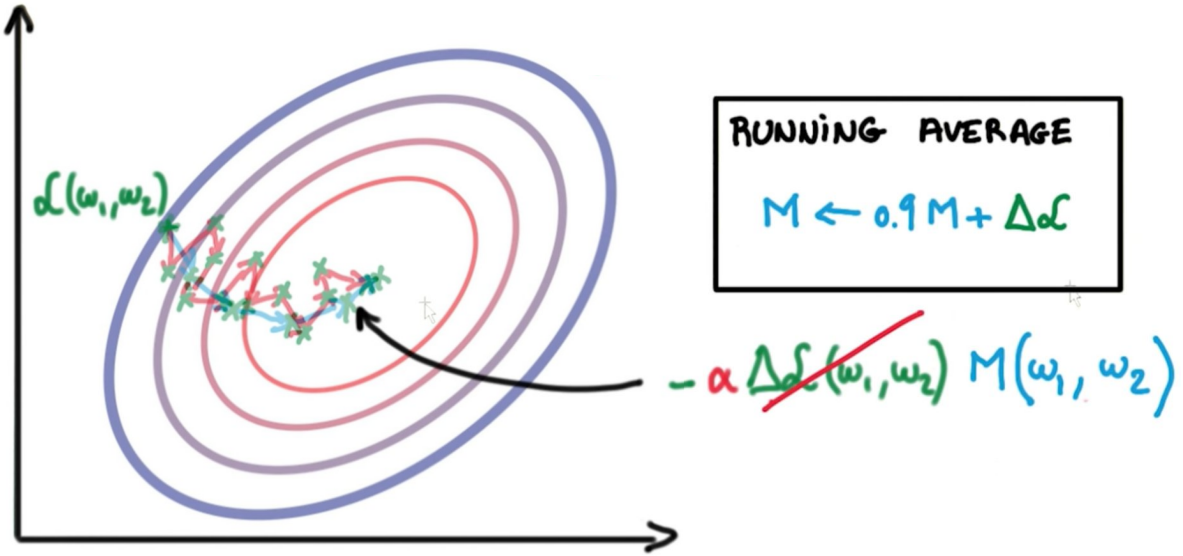
Minh họa thuật toán khi có 2 tham số cần tối ưu

Source: https://machinelearningcoban.com

Neural Network

Stochastic Gradient Descent với quán tính

Quán tính tại mỗi thời điểm là trung bình chuyển động tính trên $\approx \frac{1}{1-\mu}$ iteration trước đó



Source: Udacity's Deep Learning course

Neural Network

Stochastic Gradient Descent và một số biến thể khác

Tóm tắt các biến thể phổ biến:

- SGD thông thường:

$$\theta := \theta - \alpha \nabla_\theta$$

- SGD với quán tính:

$$v := \mu v + \nabla_\theta$$
$$\theta := \theta - \alpha v$$

Biến thể: Nesterov Momentum

Learning rate nên được điều chỉnh giảm dần khi số vòng lặp đủ lớn

Neural Network

Stochastic Gradient Descent và một số biến thể khác

Tóm tắt các biến thể phổ biến:

- SGD thông thường:

$$\theta := \theta - \alpha \nabla_\theta$$

- SGD với quán tính:

$$v := \mu v + \nabla_\theta$$
$$\theta := \theta - \alpha v$$

Biến thể: Nesterov Momentum

Learning rate nên được điều chỉnh giảm dần khi số vòng lặp đủ lớn

- AdaGrad: (adaptive gradient)

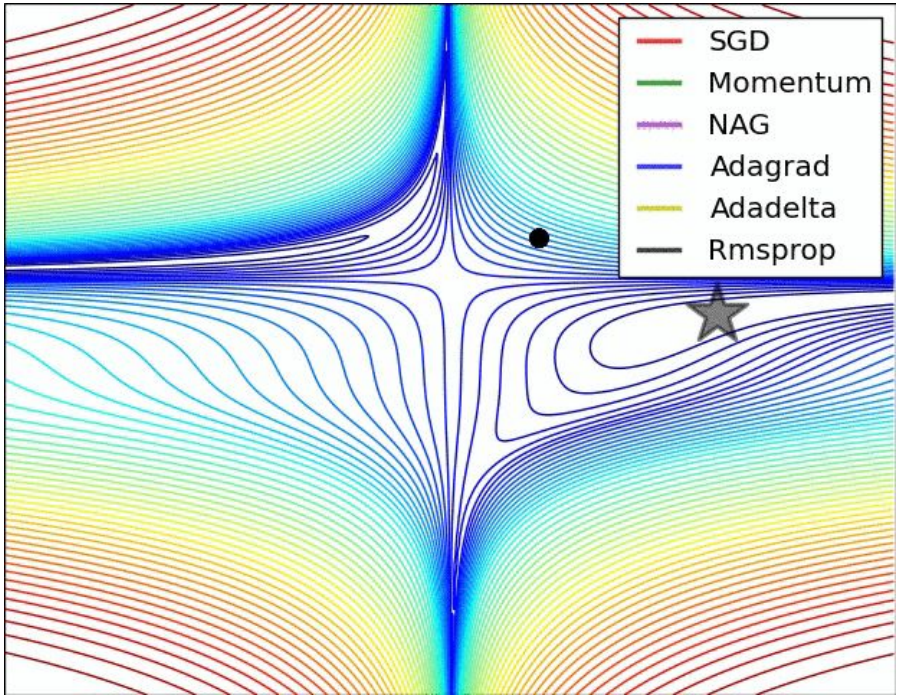
$$G := G + (\nabla_\theta)^2$$
$$\theta := \theta - (\alpha / \sqrt{G + \epsilon}) \nabla_\theta$$

Learning rate được tự động giảm dần khi số vòng lặp tăng lên

- RMSprop - “improved” AdaGrad
- ADAM = RMSprop + Momentum

Neural Network

Stochastic Gradient Descent và một số biến thể khác



Source: http://ruder.io/optimizing-gradient-descent/

Neural Network

Một số vấn đề khi huấn luyện neural network

- Overfitting
- Underfitting
- Dying ReLU
- Khởi tạo tham số giống nhau
- Gradient vanishing
- Hàm lỗi tăng dần đều
- Hàm lỗi giảm rất chậm
- Hàm lỗi không ổn định (tăng giảm thất thường)

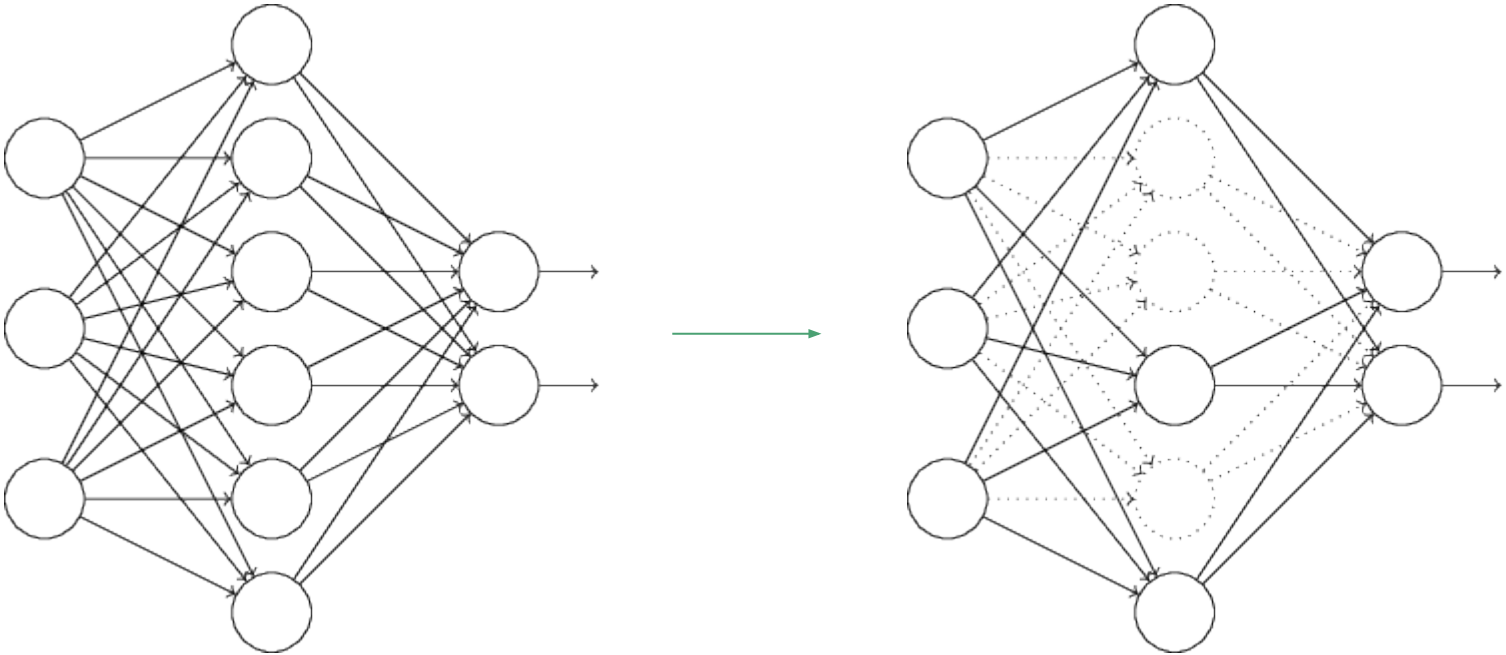
Neural Network

Một số vấn đề khi huấn luyện neural network

- Overfitting
  - Giảm số tham số.
  - Giảm số lớp.
  - Thêm L2 regularization cho các tham số.
  - Sử dụng giải thuật dropout.

Dropout

Trực quan



Improving the way neural networks learn - Michael Nielsen

Dropout

Diễn giải

"This technique reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons. It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons."

ImageNet Classification with Deep Convolutional Neural Networks, by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton (2012)

Neural Network

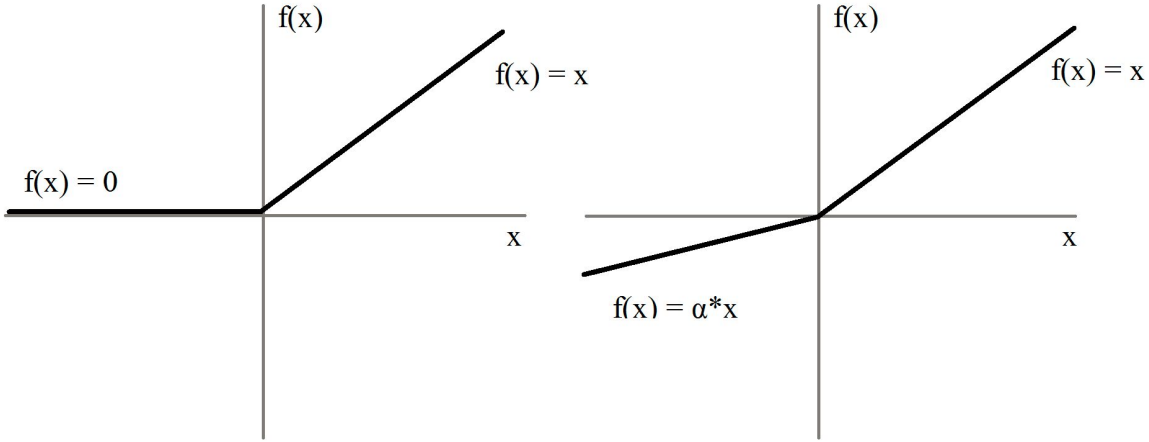
Một số vấn đề khi huấn luyện neural network

- Underfitting
  - Tăng số tham số bằng cách tăng số lớp hoặc số nút ẩn
  - Giảm hệ số L2 hoặc không dùng L2
  - Tập giá trị của đầu ra (output range) bất khả thi (ví dụ: đầu ra của mạng dùng activation là sigmoid nhưng bài toán cần làm lại là regression với output target có giá trị > 1)

Neural Network

Một số vấn đề khi huấn luyện neural network

- Dying ReLU
  - Xảy ra khi toàn bộ tất cả các nút ẩn rơi vào số âm.
  - Cách giải quyết: dùng phiên bản Leaky ReLU.



Neural Network

Một số vấn đề khi huấn luyện neural network

- Khởi tạo tham số bằng nhau
  - Các tham số sẽ có đạo hàm giống nhau trong bước lan truyền ngược.
  - Việc cập nhật cũng sẽ giống hệt nhau cho các bộ tham số.

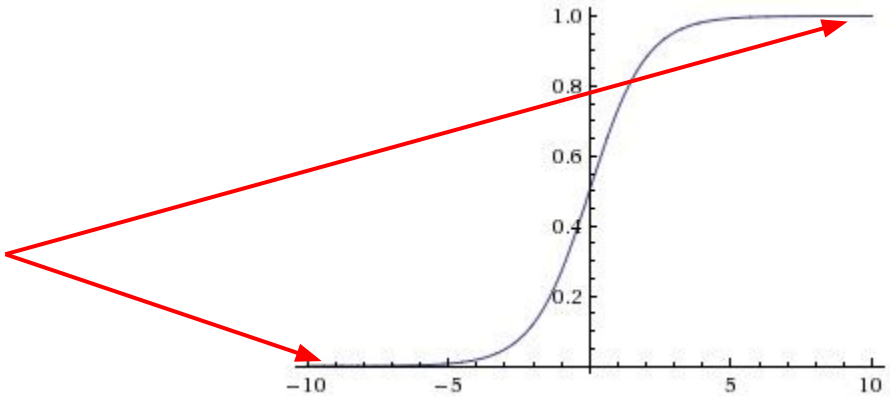
=> Khởi tạo tham số ngẫu nhiên để tránh lỗi này (symmetry breaking).

Neural Network

Một số vấn đề khi huấn luyện neural network

- Gradient vanishing
  - Thường gặp khi sử dụng nhiều lớp với hàm kích hoạt là sigmoid hoặc tanh.

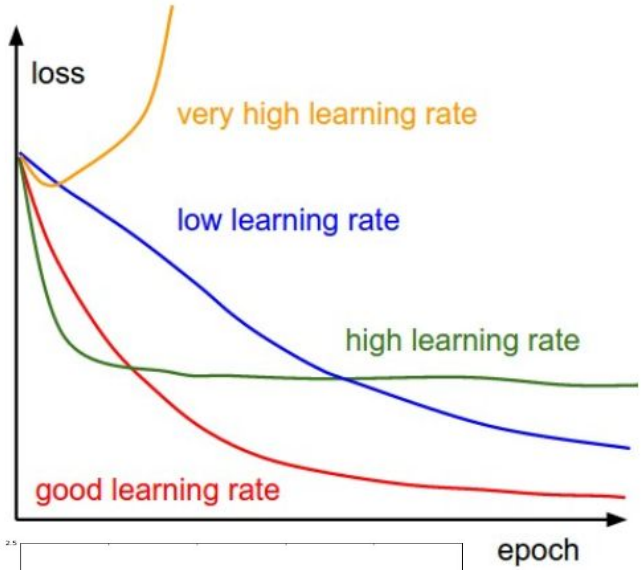
Đạo hàm rất nhỏ

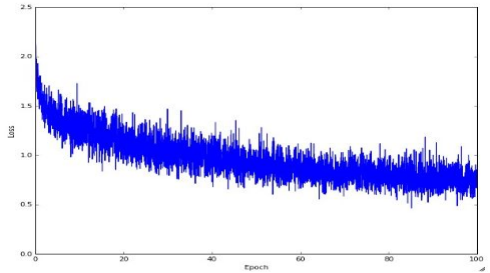


Neural Network

Một số vấn đề khi huấn luyện neural network

- Hàm lỗi tăng dần đều
  - Hệ số học hoặc hệ số quán tính quá lớn.
  - Khi tính đạo hàm bị ngược dấu ở một giai đoạn nào đó.
- Hàm lỗi giảm rất chậm:
  - Hệ số học quá bé.
  - Sử dụng batch gradient descent để xấp xỉ hàm quá phức tạp.
- Hàm lỗi không ổn định (tăng giảm thất thường)
  - Hệ số học vẫn còn lớn.
  - Cách khắc phục: giảm hệ số học sau một số vòng lặp nhất định.





Source: StanfordU's CS231n

Tài liệu tham khảo

1. https://cs231n.github.io/neural-networks-3
2. *Improving the way neural networks learn* - Michael Nielsen
   http://neuralnetworksanddeeplearning.com/chap3.html
3. *An overview of gradient descent optimization algorithms* - Sebastian Ruder
   http://ruder.io/optimizing-gradient-descent/