1. Normalization

Các phương pháp Normalization có thể giúp mô hình huấn luyện nhanh và kết quả tốt

Chuẩn hóa dữ liệu mỗi feature sẽ giữ được sự đóng góp của mọi feature trong quá trình huấn luyện trong trường hợp giá trị một số features cao hơn nhiều so với các feature còn lại. Với cách làm này sẽ làm mô hình unbiased

Làm giảm **Internal Covariate Shift**. Việc mô hình càng sâu sẽ có nhiều layer cùng với đó là có nhiều hàm kích hoạt, nó sẽ làm biến đổi đi phân phối của dữ liệu. Do đó chúng ta cần chuẩn hóa lại nó để có được sự đồng bộ phân phối của dữ liệu trong quá trình huấn luyện

Theo tác giả, nó được định nghĩa như sau: "Chúng tôi định nghĩa Internal Covariate Shift là sự thay đổi phân phối của các hàm kích hoạt trong mô hình do sự thay đổi của các tham số mô hình trong quá trình huấn luyện."

Batch Normallization có thể làm bề mặt hàm loss trở nên mịn màng hơn (i.e. nó giới hạn độ lớn của gradients chặt chẽ hơn nhiều ). Đại khái là việc tối ưu hàm mục tiêu sẽ trở nên dễ dàng và nhanh chóng hơn.

Việc tối ưu sẽ trở nên nhanh hơn bởi vì normalization không cho phép việc bùng nổ (explode) weights và nó giới hạn chúng trong một vùng nhất định

Một lợi ích nữa nằm ngoài mục đích của normalization là nó giúp mô hình Regularization (chỉ một ít, không đáng kể)

1. Tầm quan trọng

Giả sử chúng ta có bài toán cho vay trong ngân hàng, bạn có các feature (đặc trưng) là tiền lương và số năm kinh nghiệm.

Vấn đề không có gì đáng lo cho đến khi số tiền lương được tính theo đơn vị hàng triệu trong khi số năm chỉ vỏn vẹn hàng chục hoặc đôi khi là chưa tới.

Việc sử dụng 1 dữ liệu như vậy sẽ ảnh hướng đến quá trình huấn luyện khi mà feature năm kinh nghiệm ít tác động nhiều đến kết quả hơn là tiền lương, trọng số liên quan đến các features sẽ khác nhau rất nhiều vì giá trị đầu vào của các features biểu diễn ở các cùng khác nhau. Để thích nghi với sự khác biệt giữa các features thì một số trọng số sẽ lớn và một số trọng số lại nhỏ. Nếu ta có trọng số lớn thì sẽ ảnh hưởng đến việc cập nhật nó trong quá trình lan truyền ngược cũng sẽ lớn và ngược lại. Vì sự phân bố không đồng đều của các trọng số nên dẫn tới việc thuật toán sẽ bị dao đồng trong vùng tối ưu trước khi nó tìm đến vị trí cực tiểu toàn cục.

Để tránh việc thuật toán học dành quá nhiều thời gian dao động khi tối ưu, ta normalize các features đầu vào về cùng tỉ lệ, phân phối. Từ đó thì dữ liệu đầu vào sẽ tỉ lệ và trọng số liên quan đến chúng cũng sẽ cùng tỉ lệ. Điều này giúp mô hình có thể học nhanh hơn.

Sau khi normalied dữ liệu, khoảng dữ liệu khác nhau của 2 feature được đưa về cùng một khoảng nào đó mà cân bằng nhất ví dụ như lượt bỏ hàng chục số 0 của số tiền lương để về hàng chục chẳng hạn

1. BN

Mục tiêu của phương pháp này chính là việc muốn chuẩn hóa các feature (đầu ra của mỗi layer sau khi đi qua các activation) về trạng thái **zero-mean** với độ lệch chuẩn

**Non zero mean** là hiện tượng dữ liệu không phân bố quanh giá trị 0, mà dữ liệu có phần nhiều giá trị lớn hơn 0, hoặc nhỏ hơn 0

Kết hợp với vấn đề high variance khiến dữ liệu trở nên có nhiều thành phần rất lớn hoặc rất nhỏ

Vấn đề này rất phổ biến khi training các mạng nơ ron với số layer sâu.

Việc feature không phân phối trong những khoảng ổn định sẽ có ảnh hưởng đến quá trình tối ưu của mạng

Giả sử như một công thức tính layer đơn giản là y = (Wx + b) thì đạo hàm của ytheo w có dạng: *dy*  = *dWx*

Do đó nếu *x* mang các giá trị thay đổi không ổn định dẫn đến đạo hàm sẽ có thể bị quá lớn, hoặc quá nhỏ dẫn đến việc learning model không được ổn định

**Batch normalization** có thể giúp chúng ta tránh được hiện tượng giá trị của *x* rơi vào **khoảng bão hòa** sau khi đi qua các hàm kích hoạt phi tuyển

Điều này giúp cho các weights mà khi không dùng **BN** có thể sẽ không bao giờ được học thì nay lại được học bình thường. Điều này giúp chúng ta làm giảm đi sự phụ thuộc vào giá trị khởi tạo của các tham số.

Batch Normalization còn có vai trò như một dạng của **regularization** giúp cho việc giảm thiểu overfiting.

Sử dụng batch normalization, chúng ta sẽ không cần phải sử dụng quá nhiều dropput và điều này rất có ý nghĩa vì chúng ta sẽ không cần phải lo lắng vì bị mất quá nhiều thông tin khi dropout weigths của mạng.

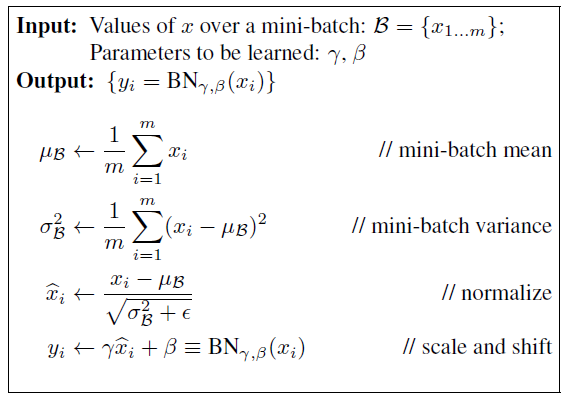
Tuy nhiên vẫn nên sử dụng kết hợp cả hai kĩ thuật này

Cách tiếp cận này chủ yếu liên quan đến **Internal Covariate Shift** (ICS)

Để cải thiện việc huấn luyện mô hình, điều quan trọng là phải giảm ICS bằng cách kiểm soát means và variances của dữ liệu đầu vào các layer, nôm na là điều chỉnh phân phối của dữ liệu cho đồng bộ trên toàn mô hình

Batch normalization là một phương thức chuẩn hóa các hàm kích hoạt trong mạng qua một mini-batch theo kích thước được định nghĩa trước đó

Với mỗi feature, batch normalization tính toán trung vị và phương sai của feature đó trong một mini-batch. Sau đó, nó trừ đi giá trị trung bình và chia cho độ lệch chuẩn của mini-batch đó. Công thức sẽ được biểu diễn như sau:



(1) shift data về mean *β =* 0, scale variance về *γ* = 1

(2) shift data về mean *β* và scale variance về *γ*

*γ* và *β* là do mạng học được.

Như vậy quá trình này giống kiểu shift và scale dữ liệu về mean 0, variance 1 là bắt buộc, còn sau đó mạng tự học shift và scale về một phân bố nào đó.

Việc này giúp cho mạng giúp cho việc học của mạng tốt hơn, cụ thể:

* Có những hàm có đạo hàm tốt tại khoảng [-1, 1] vì thế bước (1) sẽ có ích, bước (2) chỉ cần weight scale *α* học được là 1 và weight shift *β* là 0
* Có những hàm lại tốt lại thuộc khoảng khác [-1, 1]. Vì thế không phải lúc này shift và scale data về [-1, 1] như ở bước (1) là không tốt, do vậy bước (2) model sẽ học shift và scale đến khoảng nào đó tốt cho đạo hàm.

Ở đây đạo hàm tốt nghĩa là đạo hàm của hàm loss có giá trị không quá nhỏ để dẫn tới hiện tượng Vanish gradient, không quá to để dẫn tới explode gradient

1. **Các vấn đề của BN**

**Variable batch size**

Nếu barch size là 1, thì phương sai sẽ bằng 0, lúc này thì batch norm không hoạt động được

Nếu ta cho mini-batch nhỏ thì nó sẽ thành nhiễu và ảnh hưởng đến quá trình huấn luyện mô hình

Trong RNN, các kích hoạt lặp lại của mỗi bước thời gian sẽ có một câu chuyện khác nhau để kể (tức là số liệu thống kê). Điều này có nghĩa là chúng ta phải phù hợp với các layer batch norm riêng biệt cho mỗi time-step. Điều này làm cho mô hình phức tạp hơn và tốn không gian hơn vì nó buộc chúng ta phải lưu trữ số liệu thống kê cho từng bước thời gian trong quá trình đào tạo.

**Các lợi ích của batch normalization**

Làm giảm internal covariate shift (ICS) và tăng tốc độ huấn luyện cho mô hình deep learning.

Cách tiếp cận này làm giảm sự phụ thuộc của gradients vào tỉ lệ của các tham số hoặc giá trị ban đầu của chúng, dẫn đến learning rate cao hơn mà không có nguy cơ phân kỳ.  
Batch normalization giúp bạn có thể sử dụng các chế độ phi tuyến bão hòa bằng cách ngăn mạng khỏi bị kẹt trong các chế độ bão hòa

1. **Sử dụng**

Trong các mô hình CNN như Alexnet, VGG16, VGG19 thì BN là một layer được sử dụng khá nhiều được áp dụng ngay sau conv layer và thường ở những vị trí đầu tiên của mô hình để đạt hiểu quả cao nhất dể chuẩn hoá dữ liệu ở các layer the batch về phân phối chuẩn để quá trình GD hội tụ nhanh hơn

Mô hình LeNet có BN và không có BN

<https://phamdinhkhanh.github.io/2019/12/02/DeepLearningLayer.html#21-time-distributed>

<https://viblo.asia/p/ai-interview-12-cau-hoi-phong-van-deep-learning-sieu-hay-khong-the-bo-qua-LzD5djvEZjY>

https://viblo.asia/p/normalization-and-normalization-techniques-in-deep-learning-QpmleJyn5rd