# Batch normalization in Neural Networks 神经网络中的批处理规范化

原文链接：  
原文链接：

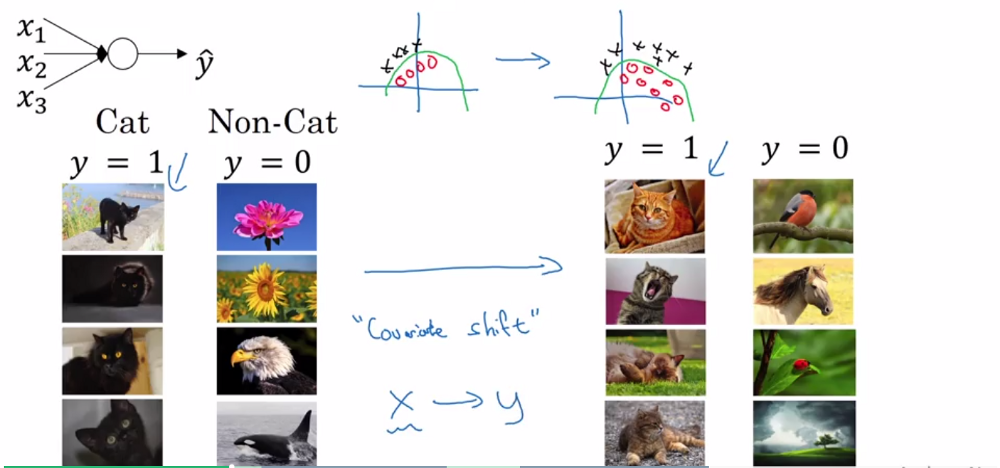
This article explains batch normalization in a simple way. I wrote this article after what I learned from Fast.ai and deeplearning.ai. I will start with why we need it, how it works, then how to include it in pre-trained networks such as VGG.  
本文以一种简单的方式解释批处理规范化。我是根据从Fast.ai和deeplearning.ai中学到的东西写这篇文章的。我将从我们为什么需要它开始，它如何工作，然后如何将它包含在预先训练的网络中，如VGG。

#### Why do we use batch normalization? 为什么要使用批处理规范化？

We normalize the input layer by adjusting and scaling the activations. For example, when we have features from 0 to 1 and some from 1 to 1000, we should normalize them to speed up learning. If the input layer is benefiting from it, why not do the same thing also for the values in the hidden layers, that are changing all the time, and get 10 times or more improvement in the training speed.  
我们通过调整和缩放激活来规范化输入层。例如，当特性从0到1，有些特性从1到1000时，我们应该将它们规范化，以加快学习速度。如果输入层从中受益，为什么不对隐藏层中的值做同样的事情，这些值一直在变化，并且在训练速度上得到10倍或更多的提高。

Batch normalization reduces the amount by what the hidden unit values shift around (covariance shift). To explain covariance shift, let’s have a deep network on cat detection. We train our data on only black cats’ images. So, if we now try to apply this network to data with colored cats, it is obvious; we’re not going to do well. The training set and the prediction set are both cats’ images but they differ a little bit. In other words, if an algorithm learned some X to Y mapping, and if the distribution of X changes, then we might need to retrain the learning algorithm by trying to align the distribution of X with the distribution of Y. ( Deeplearning.ai: Why Does Batch Norm Work? ())  
批处理规范化通过隐藏的单位值的移位（协方差移位）来减少量。为了解释协方差偏移，让我们在cat检测上有一个深入的网络。我们只在黑猫的图片上训练数据。所以，如果我们现在尝试将这个网络应用于有色猫科动物的数据，这是显而易见的；我们不会做得很好。训练集和预测集都是猫的图像，但它们有一点不同。换言之，如果一个算法学习了一些X到Y的映射，并且X的分布发生了变化，那么我们可能需要通过尝试将X的分布与Y的分布对齐来重新训练学习算法（Deeplearning.ai:为什么批范数有效？())

Also, batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers.  
此外，批处理规范化允许网络的每一层独立于其他层学习。



img  
img公司

Deeplearning.ai: Why Does Batch Norm Work? ()  
深度学习：为什么批量规范有效？()

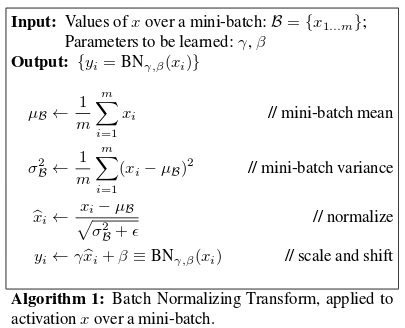
* We can use higher learning rates because batch normalization makes sure that there’s no activation that’s gone really high or really low. And by that, things that previously couldn’t get to train, it will start to train.  
  我们可以使用更高的学习率，因为批量规范化可以确保没有真正高或真正低的激活。到了那个时候，以前不能训练的东西，就会开始训练。
* It reduces overfitting because it has a slight regularization effects. Similar to dropout, it adds some noise to each hidden layer’s activations. Therefore, if we use batch normalization, we will use less dropout, which is a good thing because we are not going to lose a lot of information. However, we should not depend only on batch normalization for regularization; we should better use it together with dropout.  
  它减少了过度拟合，因为它有轻微的正则化效果。与dropout类似，它为每个隐藏层的激活添加一些噪波。因此，如果我们使用批处理规范化，我们将使用更少的辍学率，这是一件好事，因为我们不会丢失很多信息。然而，我们不应该仅仅依赖于正则化的批处理规范化；我们应该更好地将其与辍学一起使用。

#### How does batch normalization work? 批量规范化是如何工作的？

To increase the stability of a neural network, batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.  
为了提高神经网络的稳定性，批处理规范化通过减去批处理平均值并除以批处理标准差来规范化先前激活层的输出。

However, after this shift/scale of activation outputs by some randomly initialized parameters, the weights in the next layer are no longer optimal. SGD ( Stochastic gradient descent) undoes this normalization if it’s a way for it to minimize the loss function.  
然而，在通过一些随机初始化的参数对激活输出进行这种移位/缩放之后，下一层中的权重不再是最优的。如果随机梯度下降（SGD）是一种使损失函数最小化的方法，则它将取消这种规范化。

Consequently, batch normalization adds two trainable parameters to each layer, so the normalized output is multiplied by a “standard deviation” parameter (gamma) and add a “mean” parameter (beta). In other words, batch normalization lets SGD do the denormalization by changing only these two weights for each activation, instead of losing the stability of the network by changing all the weights.  
因此，批处理规范化向每个层添加两个可训练参数，因此规范化输出乘以“标准偏差”参数（gamma）并添加“平均”参数（beta）。换言之，批处理规范化允许SGD对每个激活仅改变这两个权重来进行非规范化，而不是通过改变所有权重来失去网络的稳定性。



img  
img公司

[From the original batch-norm paper](https://arxiv.org/pdf/1502.03167v3.pdf)

#### Batch normalization and pre-trained networks like VGG: 批量规范化和预先训练的网络，如VGG：

VGG doesn’t have a batch norm layer in it because batch normalization didn’t exist before VGG. If we train it with it from the start, the pre-trained weight will benefit from the normalization of the activations. So adding a batch norm layer actually improves ImageNet, which is cool. You can add it to dense layers, and also to convolutional layers.  
由于没有批处理正常化，所以它没有批处理规范层。如果我们从一开始就用它训练，训练前的体重将从激活的正常化中受益。所以添加一个批处理规范层实际上改善了ImageNet，这很酷。可以将其添加到密集层，也可以添加到卷积层。

If we insert a batch norm in a pre-trained network, it will change the pre-trained weights, because it will subtract the mean and divide by the standard deviation for the activation layers and we don’t want that to happen because we need those pre-trained weights to stay the same. So, what we need to do is to insert a batch norm layer and figure out gamma and beta in order to undo the outputs change.  
如果我们在预先训练的网络中插入一个批范数，它将改变预先训练的权重，因为它将减去平均值，除以激活层的标准差，我们不希望发生这种情况，因为我们需要这些预先训练的权重保持不变。因此，我们需要做的是插入一个批处理规范层并计算出gamma和beta，以便撤消输出更改。

To summarize everything, you can think about batch normalization as doing preprocessing at every layer of the network.

#### References:

* [Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](https://arxiv.org/pdf/1502.03167v3.pdf)
* [Fast.ai: lesson 5](http://course.fast.ai/lessons/lesson5.html)
* Deeplearning.ai: Why Does Batch Norm Work? ([C2W3L06](https://www.youtube.com/watch?v=nUUqwaxLnWs))