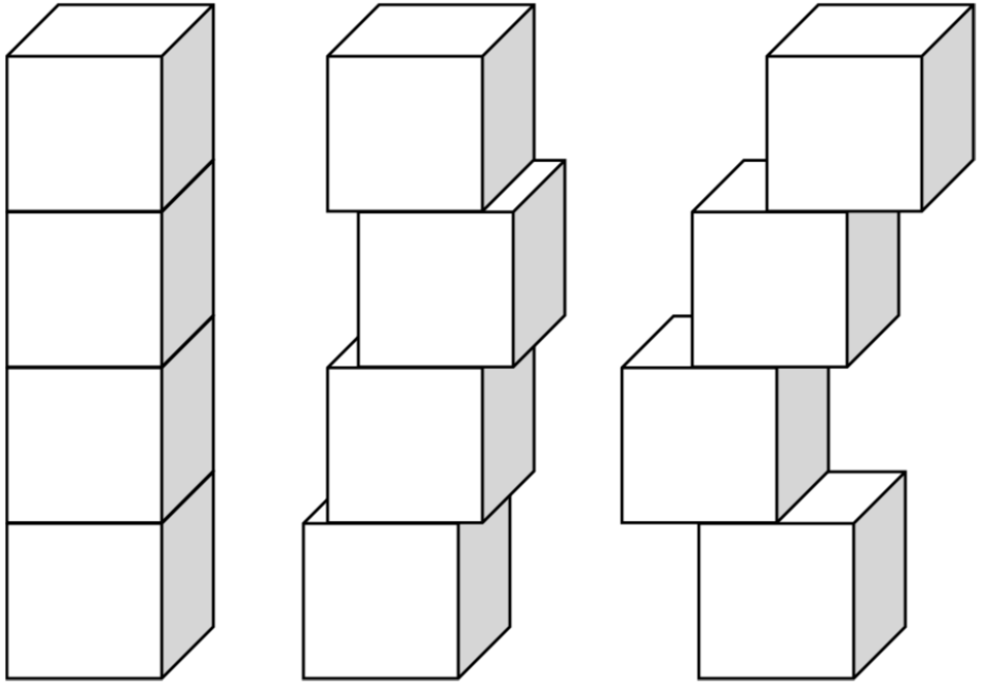
批次归一化 (Batch Normalization)的采用理由

Internal Covariate Shift （内部协变量偏移）

深度网络内部数据分布在训练过程中发生变化的现象



S. Ioffe, C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” . 2015.


Rose
(y=1)
Not Rose
(y=0)

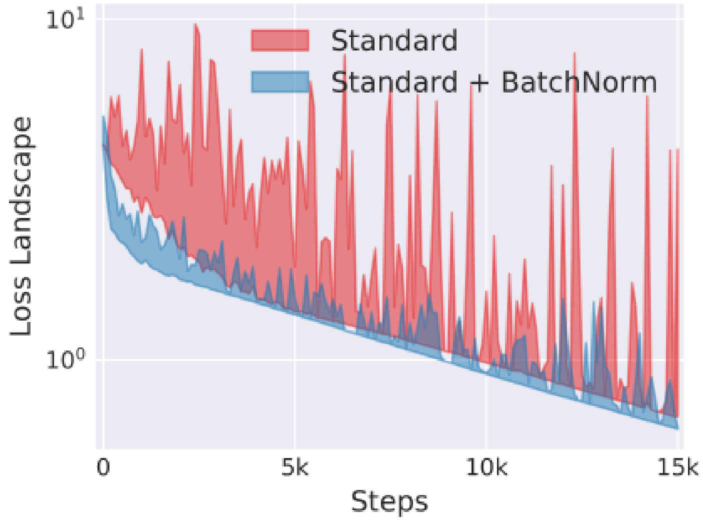

Rose
(y=1)
Not Rose
(y=0)

- This difference in distribution is called the covariate shift（协变量偏移）。输入层可通过样本随机化解决。
- 在神经网络中，每次在前一层中存在参数更新时，每个隐藏单元的输入分布都会发生变化。 这称为 Internal Covariate Shift（内部协变量偏移）。 这使得训练变慢并且需要非常小的学习率和良好的参数初始化 。

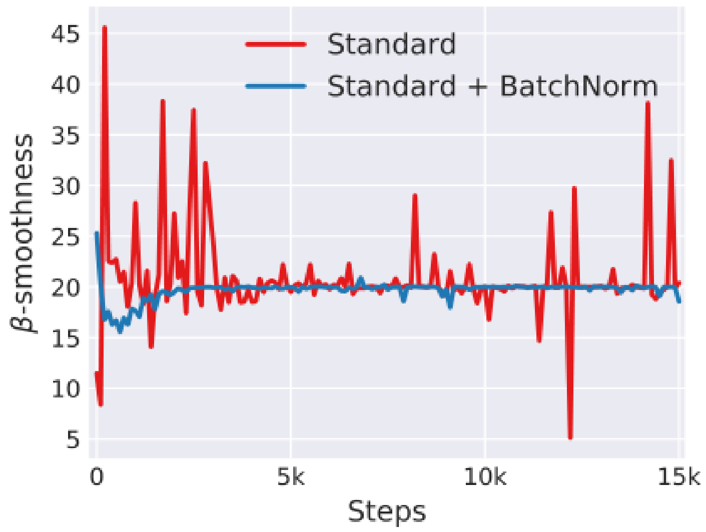
批次归一化 (Batch Normalization)的采用理由

论文：How Does Batch Normalization Help Optimization?
(No, It Is Not About Internal Covariate Shift) [2018]

优化地貌（optimization landscape ）更加平滑，使梯度更具预测性和稳定性，允许更快的训练。



(a) loss landscape



(b) “effective” β-smoothness

(c) gradient predictiveness

Analysis of the optimization landscape of VGG networks.

批次归一化 (Batch Normalization)

- 通常插入在卷积和全连接之后，在非线性处理前。 位置：卷积 → BN → ReLU
- 为使每一维成为标准高斯分布(均值为0，方差为1)，可应用 $\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$
- 为能工作在激活的非线性区，再进行缩放和移位（scale&shift）处理

The output of the previous layer in the neural net

The average of h

A regularization parameter

$$\mathrm{BN}(\mathbf{h};\gamma,\beta) = \beta + \gamma \underbrace{\frac{\mathbf{h} - \widehat{\mathbb{E}}(\mathbf{h})}{\sqrt{\widehat{\mathrm{Var}}(\mathbf{h}) + \epsilon}}}$$

The variance of h

Batch Normalize

We use these to scale and shift the (normalized) distribution. Both are parameters that are learned.

This piece basically normalizes h to have a mean and variance of 1

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

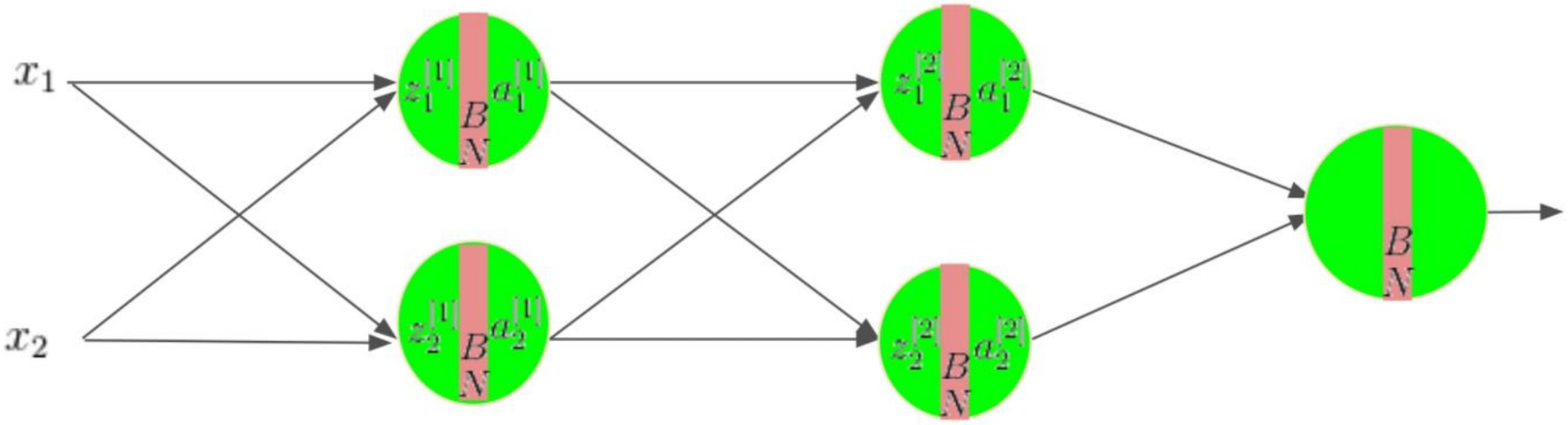
$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

在每个mini-batch中计算得到mini-batch的mean和variance来替代整体训练集的mean和variance



$$z_i^l = \left(w_i^l\right)^T a_i^{l-1} \quad \text{(BN 1.1)}$$

$$\mu_B^l = \frac{1}{m}\sum_{i=1}^{m} z_i^l \quad \text{(BN 1.2)}$$

$$\left(\sigma_B^2\right)^{(l)} = \frac{1}{m}\sum_{i=1}^{m}\left(z_i^l - \mu_B^l\right)^2 \quad \text{(BN 1.3)}$$

$$\hat{z}_i^l = \frac{z_i^l - \mu_B^l}{\sqrt{\left(\sigma_B^2\right)^{(l)} + \varepsilon}} \quad \text{(BN 1.4)}$$

.00001f

$$y_i^l = \gamma \hat{z}_i^l + \beta \quad \text{(BN 1.5)}$$

$$a_i^l = g\left(y_i^l\right) \quad \text{(BN 1.6)}$$

预测时，计算总体的均值和方差是不实际的，也是无法实现的，因为无法采样到所有样本。

在训练过程中的batch下的均值μ_B和方差σ_B，可以加以利用来估计总体

$$\mu = E(x) = E\left(\mu_B\right) = \frac{1}{K}\sum_{B}^{K} \mu_B$$

$$D(x) = \frac{m}{m-1} E\left(\sigma_A^2\right)$$

m为batch_size

虽然理论上是如此, Darknet里面用滚动平均值来算。

$\overline{x}_m$ 表示前m个数据的平均值

滚动平均：
$$
\begin{aligned}
\overline{x}_m &= \frac{x_1 + x_2 + \cdots + x_m}{m} \\
&= \frac{x_1 + x_2 + \cdots + x_{m-1} + x_m}{m} \\
&= \frac{(x_1 + x_2 + \cdots + x_{m-1})(m-1)}{m} + \frac{x_m}{m} \\
&= \frac{m-1}{m}\overline{x}_{m-1} + \frac{x_m}{m} \\
&= \left(1 - \frac{1}{m}\right)\overline{x}_{m-1} + \frac{1}{m}x_m
\end{aligned}
$$