AlexNet 阅读总结

15331416 赵寒旭

content

注：文中标记为橙色的均为有疑问的部分

Abstract

1）target
1.2 million high-resolution images -> 1000 different classes
2）methods
60 million parameters
65000 neurons
5 convolutional layers (some of which are followed by max-pooling layers)
3 fully-connected layers with a final 1000-way softmax
Make training faster :
used non-saturating neurons
a very efficient GPU implementation of the convolution operation
Reduce overfitting in the fully-connected layers:
a recently-developed regularization method called “dropout”
also entered a variant of this model
3）result
top-5 test error rate :15.3%

1. Introduction

1.1 To improve machine learning methods’ performance

To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting.

1）Datasets
collect labeled datasets with millions of images

2）CNNs

The immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don’t have. Convolutional neural networks (CNNs) constitute one such class of models.（物体识别任务极高的复杂度意味着即使拥有 ImageNet 这么大的数据集，这个问题也很难被具体化。所以 我们的模型也需要大量先验知识去补全所有缺失数据。卷积神经网络（CNNs）就是一种这样的模型。）

3）GPU

Current GPUs, paired with a highly-optimized（高度优化的） implementation of 2D convolution, are powerful enough to facilitate the training of interestingly-large CNNs, and recent datasets such as ImageNet contain enough labeled examples to train such models without severe overfitting.

1.2 Contributions of this paper

1）trained one of the largest convolutional neural networks to date on the subsets of ImageNet used in the ILSVRC-2010 and ILSVRC-2012 competitions and achieved by far the best results ever reported on thesedatasets.

2）wrote a highly-optimized GPU implementation of 2D convolution and all the other

operations inherent in training convolutional neural networks
3）Our network contains a number of new and unusual features which improve its performance and reduce its training time.（Section 3）
4）used several effective techniques for preventing overfitting（Section 4）

2. The Dataset

2.1 Data Sources（数据来源）

ImageNet is a dataset of over 15 million labeled high---resolution images belonging to roughly 22,000 categories. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.

2.2 Data processing（数据处理）

Our system requires a constant input dimensionality, we down-sampled the images to a fixed resolution of 256×256.

1）rescaled the image such that the shorter side was of length 256

2）cropped out the central 256×256 patch from the resulting image

We did not pre-process the images in any other way, except for subtracting the mean activity over the training set from each pixel. So we trained our network on the (centered) raw RGB values of the pixels.

3. The Architecture

3.1 ReLU Nonlinearity（ReLU 非线性）

The standard way to model a neuron's output f as a function of its input x is with $f(x) = \tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$. In terms of training time with gradient descent（梯度下降）, these saturating nonlinearities（饱和非线性）are much slower than the non-saturating nonlinearity $f(x) = \max(0, x)$.

We refer to neurons with this nonlinearity as Rectified Linear Units (ReLUs)（修正线性单元）. Deep convolutional neural networks with ReLUs train several times faster than their equivalents with tanh units.

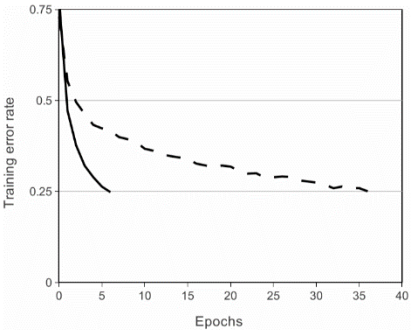
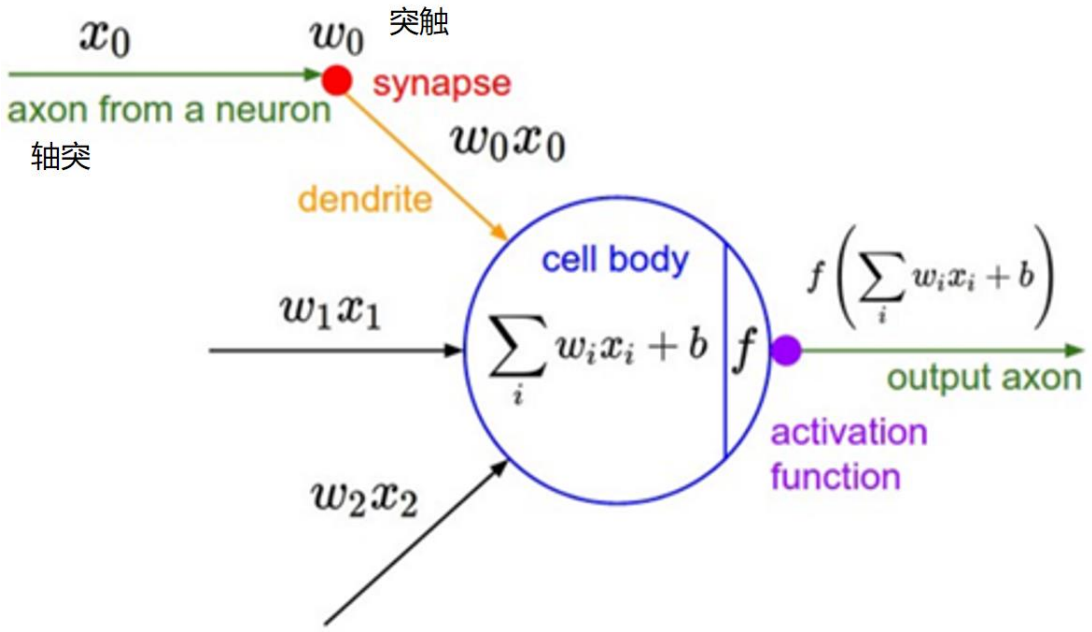


Figure 1 A four-layer convolutional neural network with ReLUs **(solid line)** reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons **(dashed line)**.

在一些之前传统的机器学习过程中我们一般采用 Simoid 和 tanh 函数来作为激活函数，但是这两个函数在做梯度下降的过程中（减小训练误差），速度比较慢，也就是迭代次数会比较多。在复杂的神经网络中，传统的激活函数效率太低，不切合实际。所以我们一般会在网络中采用 ReLu 函数，使学习周期大大缩短，提高速度和效率。

激活函数的相关概念



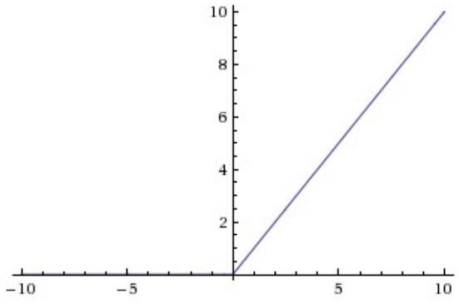
激活函数的饱和与不饱和

Definitions

- f is non-saturating iff $(|\lim_{z\to-\infty} f(z)| = +\infty) \vee |\lim_{z\to+\infty} f(z)| = +\infty)$
- f is saturating iff f is not non-saturating.

These definitions are not specific to convolutional neural networks.

ReLU 作为激活函数的优点：



- 相比起 Sigmoid 和 tanh，ReLU 在 SGD 中能够快速收敛。
- Sigmoid 和 tanh 涉及了很多很 expensive 的操作（比如指数），ReLU 可以更加简单的实现。
- 有效缓解了梯度消失的问题。
- 在没有无监督预训练的时候也能有较好的表现。

Neuron	MNIST	CIFAR10	NISTP	NORB
With **unsupervised pre-training**				
Rectifier	**1.20%**	**49.96%**	**32.86%**	**16.46%**
Tanh	**1.16%**	**50.79%**	35.89%	17.66%
Softplus	**1.17%**	**49.52%**	**33.27%**	19.19%
Without **unsupervised pre-training**				
Rectifier	**1.43%**	**50.86%**	**32.64%**	**16.40%**
Tanh	1.57%	52.62%	36.46%	19.29%
Softplus	1.77%	53.20%	35.48%	17.68%

- 提供了神经网络的稀疏表达能力。

3.2 Training on Multiple GPUs（多 GPU 上的训练）

A single GPU has only 3GB of memory, which limits the maximum size of the networks that can be trained on it, so we spread the net across two GPUs.
The parallelization scheme: puts half of the kernels (or neurons) on each GPU, with one additional trick: the GPUs communicate only in certain layers. This means that, for example, the kernels of layer 3 take input from all kernel maps（内核映射） in layer 2. However, kernels in layer 4 take input only from those kernel maps in layer 3 which reside on the same GPU.
Choosing the pattern of connectivity is a problem for cross-validation, but this allows us to precisely tune the amount of communication until it is an acceptable fraction of the amount of computation.

3.3 Local Response Normalization（局部响应归一化）

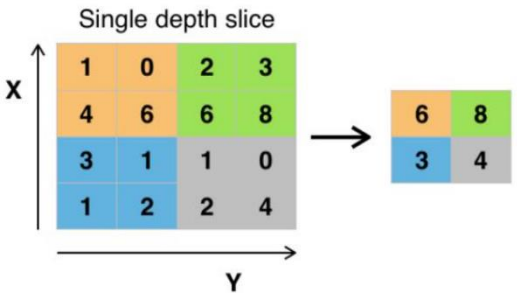
ReLUs 不需要对输入做归一化来避免饱和，但是局部归一化有助于模型的泛化，将模型 top1 和 top5 的错误率分别降低了 1.4%和 1.2%。(若使用传统的激活函数，input 不进行归一化的话，激活后的值都会进入平坦区，会使隐层的输出全部趋同。)
参考博客：
http://blog.csdn.net/yangdashi888/article/details/77918311

$$b_{x,y}^{i}=a_{x,y}^{i}/\left(k+\alpha\sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)}(a_{x,y}^{j})^{2}\right)^{\beta}$$

$a_{x,y}^{i}$：输入的(x,y)位置做第 i 次卷积并通过 relu 单元的结果
n：相同位置第 i 次前后附近的 n 次卷积
$b_{x,y}^{i}$：对应的归一化响应结果
N：总的卷积次数
这个局部响应归一化是模拟生物神经元的侧抑制作用——当前神经元的作用受附近神经元作用的抑制。在归一化阶段仅仅除以相同位置附近前后几次的神经元响应的平方，而不是 N 次结果响应的平方。
$k=2$，$n=5$，$\alpha=10^{-4}$，$\beta=0.75$ 都是预设参数。

3.4 Overlapping Pooling（重叠池化）

一般的池化层（也称下采样层）是不重叠的，以最大池化（max-pooling）为例，采用一个2×2的过滤器和一个同样长度的步长，应用到卷积层的输出上，通过过滤器计算每个子区域的卷积结果，按步长进行采样得到池化层的输出结果（如右图所示）。经过池化层，输入的空间维度大幅减小了（长宽改变，深度未变），降低了计算成本且能够控制过拟合。



AlexNet 使用的池化层是可重叠的，池化时每次移动的步长 s 小于池化的边长 z。

实验表示使用s = 2，z = 3重叠池化的效果比s = z = 2的传统池化效果要好，在产生相同维度的输出时分别将 top-1 和 top-5 的错误率降低了 0.4%和 0.5%。观察还发现，采用有重叠的池化能让模型更难过拟合。

3.5 Overall Architecture（整体结构）

CNN 的整体结构：8 个加权的层（前 5 个是卷积层，后 3 个是全连接层），最后一个全连接层输出一个 1000 维的 softmax 来表达对于 1000 个类别的预测。

Our network maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average across training cases of the log-probability of the correct label under the prediction distribution.

第 2、4、5 个卷积层的内核只与前一层与自己同在一个 GPU 上的内核映射相连接。第三层的内核与全部的第二层内核映射相连接，全连接层的神经元与上层神经元全都有连接。响应归一化层在第一、二个卷积层后面，最大值池化层跟在响应归一化层和第五个卷积层后面。ReLU 被应用在每个卷积层和全连接层。

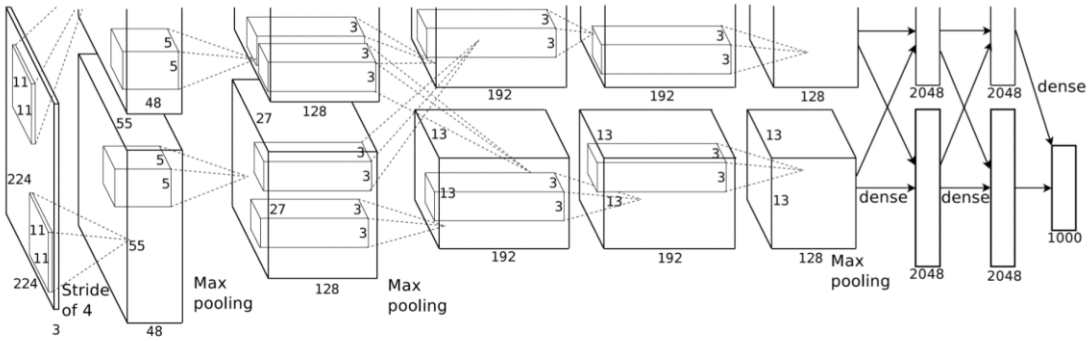


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

第一个卷积层的输入是 224 × 224 × 3（其实被预处理为227 × 227 × 3的图像，然后用 96 个 11 × 11 × 3 的步长为 4 像素的卷积核去过滤（步长是相邻神经元感知区域中心之间的距离）。

第二个卷积层将第一个卷积层的输出作为输入（响应归一化并池化），然后用 256 个 5 × 5 × 48 的内核进行过滤。

第三、四、五层卷积层前后相连，之间没有池化层和归一化层。

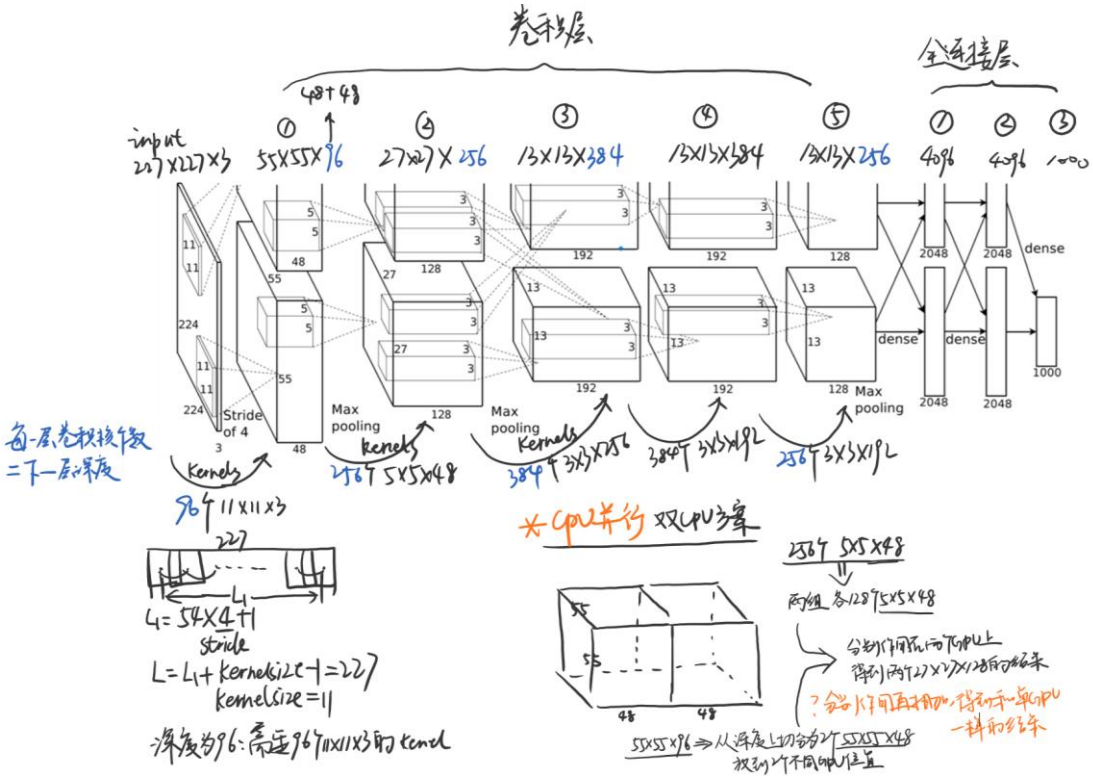
第三个卷积层有 384 个3 × 3 × 256 的卷积核。

第四个卷积层有 384 个 3 × 3 × 192 的卷积核，

第五个卷积层有 256 个 3 × 3 × 192 的卷积核。

每个全连接层各有 4096 个神经元。

注：尚未完成全连接层的具体理解



4. Reducing Overfitting

我们的这个网络有 6 千万个参数，尽管 ILSVRC 的 1000 个类可以仅用 10 比特来表示，但这远不足以让 我们学习这么大的一个网络。因此我们需要一些手段来处理过拟合问题。下面我们介绍两种基本的减少过拟合方法。

4.1 Data Augmentation（数据增强）

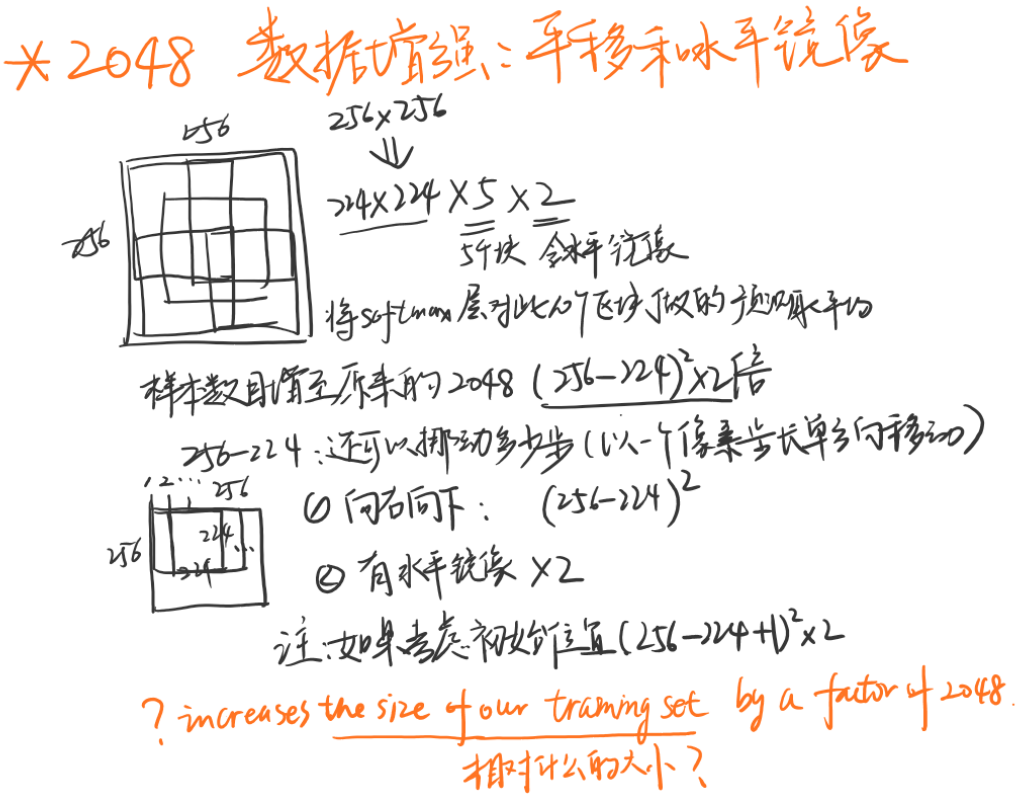
在数据集上减少过拟合的最简单的方法就是增加数据集的多样性，在样本固定的情况下，我们应该对现有的数据进行处理，使其具备更强的多样性。

可以利用 label-perserving transformations 放大数据集，我们采取两种不同形式的放大方法，它们都允许仅对原图做少量计算的情况下产生变形的新图，所以变形后的新图无需存储在硬盘中。

变形的新图由 Python 在 CPU 上计算产生，与此同时，GPU 仍在计算其他的之前批次的图片。所以这种放大数据集的方式是很高效很节省计算资源的。

1）generating image translations and horizontal reflections（图像平移和水平镜像）

从 256X256 的图片中随机抽取 224X224 的区块（及其水平镜像），在这些抽取得到的区块上训练神经网络。This increases the size of our training set by a factor of 2048, though the resulting training examples are, of course, highly interdependent.（样本数目增至原来的 2048 即$(256-224)^2 \times 2$ 倍）。如果不采用这种方法，网络会出现严重的过拟合。测试过程中网络会抽取 5 个（4 角和中间）224X224 的区块及其水平镜像进行预测，然后将 softmax 层对这十个区块做出的预测取平均。



2）altering the intensities of the RGB channels in training images（RGB 通道的亮度改变）

我们对训练集中的每张图片都做主成分分析（PCA），得到对应的特征向量$\{p_1, p_2, p_3\}$和特征值$\{\lambda_1, \lambda_2, \lambda_3\}$。训练过程中，每当这张样本被重复一次，我们就对其每一个像素按照公式$I'_{xy} = I_{xy} + \{p_1, p_2, p_3\}\{\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3\}^T$进行变换，得到一张“新”样本做下一轮的训练。每次到这张样本时，都需要用均值为 0，标准差为 0.1 的高斯随机数发生器产生新的α_i值。

通过这种方法我们可以获得富有多样性的样本。

4.2 Dropout（掉线）

降低测试错误的一种有效方法是联立多种不同模型的预测结果，但这种方法对于大型神经网络来说似乎太昂贵了，需要好几天去训练。然而，有一种非常高效的模型联立方法，只需要在训练过程中消耗一到两个因子。这种新近研究出来的技术叫做“DROPOUT”，它会以 50%的概率将每个隐藏层神经元置零。以这种方法被置零的神经元不再参与前馈和 BP 过程。

所以每次一个输入进来之后，这个神经网都会被置于不同的结构，但所有这些结构共享同一套参数。这种技术降低了神经元间相互适应的复杂性，因为每个神经元都不可能依赖其他特定某个神经元的表现。

在测试中，我们使用所有的神经元，但是把它们的输出乘以 0.5，这是一种对大量 dropout 网络产生的预测分布的几何均值的合理近似。我们在图 2 中的前两个全连接层使用 dropout。否则，我们的网络会表现出严重的过拟合。dropout 大概会让达到收敛所需要的迭代次数翻倍。

简单总结：

一些节点为了避免过拟合的需要而暂时停止更新。

每轮迭代时，网络中每个隐含层的输出节点都有一半的可能性被置为 0，使其掉线不参与此轮迭代，包括前向传播与反向传播。如此一来，我们每一轮的训练都是在不同架构的网络下进行的，并且还实现权值的共享，减少了神经节点的共适应复杂度，因为一个节点并不依赖于其它特定节点们的作用。

5. Details of learning

采用随机梯度下降法进行训练，每个批次有 128 个样本，动量momentum = 0.9，权重衰减weight decay = 0.0005。

Weight decay: a regularizer（正则项）, and reduces the model's training error.

权重 w 的更新规则：

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

i是迭代的轮数，表示是第几次的迭代；

v 是更新动量；

ϵ是学习率；

$\left\langle \frac{\partial L}{\partial \omega}|_{\omega_i} \right\rangle_{D_i}$ 是训练D_i（第i个 batch）时，目标函数关于ω的方向导数在ω_i的值。

每层网络的权值（weights）都用标准差为 0.01 的零均值高斯分布来初始化，2，4，5 卷积层的偏执（biases）和全连接层的偏置都用 1 来初始化，其余的偏置都用 0 初始化。（有一定的训练加速作用）

对所有层都采用相同的学习率，获得这个值的启发式方法是：每次都对当前的学习率除以 10，直到 validation（验证）error rate 不再改善为止。学习率被初始化为 0.01，在验证三次之后停止了更新。

6. Results

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

Alex 将全连接层最后一个隐含层的特征数据拿出来做对比，发现这 4096 维数据的欧氏距离比较接近的 Raw 输入数据都相当接近。这充分表明 AlexNet 的特征提取能力的有效性，说明深度学习有足够的能力找到多类数据的规律。