CNN 补充资料

指导老师：刘胥影
助教：林淑霞

May 2021

1 卷积网络基本概念

1.1 卷积层的感受野

在处理图像这样的高维度输入时，让每个神经元都与前一层中的所有神经元进行全连接是不现实的。相反，让每个神经元只与输入数据的一个局部区域连接。该连接的空间大小叫做神经元的感受野（receptive field），它的尺寸是一个超参数（其实就是滤波器的空间尺寸）。在深度方向上，这个连接的大小总是和输入量的深度相等。空间维度（宽和高）与深度维度是不同的：连接在空间（宽高）上是局部的，但是在深度上总是和输入数据的深度一。

下图展现的卷积神经网络的一部分，其中的红色为输入数据，假设输入数据体尺寸为 $[32 \times 32 \times 3]$（比如 CIFAR-10 的 RGB 图像），如果感受野（或滤波器尺寸）是 5×5，那么卷积层中的每个神经元会有输入数据体中 $[5 \times 5 \times 3]$ 区域的权重，共 $5 \times 5 \times 3 = 75$ 个权重（还要加一个偏差参数）。需要注意的是这个连接在深度维度上的大小必须为 3，和输入数据体的深度一致。其中还有一点需要注意，对应一个感受野有 75 个权重，这 75 个权重是通过学习进行更新的，所以很大程度上这些权值之间是不相等（也就对于同一个卷积核，它对于与它连接的输入的每一层的权重都是独特的，不是同样的权重重复输入层层数那么多次就可以的）。在这里相当于前面的每一个层对应一个传统意义上的卷积模板，每一层与自己卷积模板做完卷积之后，再将各个层的结果加起来，再加上偏置，无论输入输入数据是多少层，一个卷积核就对应一个偏置。

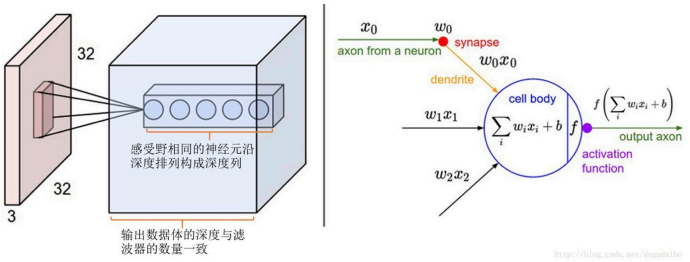

图 1: 感受野的连接尺寸及说明

1.2 归一化层

卷积神经网络的隐藏层输出特征图的均值的绝对值可能很大，方差也很大，这种特征需要做归一化才能进一步利用。归一化层是对隐藏层输出特征图的操作，目的是使特征图的均值与方差映射到可接受的范围，常用的归一化方法包括局部响应归一化（Local Response Normalization）与批归一化 Batch Normalization）。

LRN 层是利用通道维度之间的值进行归一化，与长、宽维度无关，其公式为:

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

其中 k,α,β,n 为人为设定的参数，$a_{x,y}^{i}$ 表示第 i 个通道下，第 (x, y) 位置的输入像素值，$b_{x,y}^{i}$ 表示相同位置的输出像素值。LRN 层对局部神经元的活动创建竞争机制，使得其中响应比较大的值变得相对更大, 并抑制其他反贵较小的神经元。LRN 层在 AlexNet 中提出，但是在最新的卷积神经网络设计中已经很少使用。

BN 层是利用样本的均值和方差进行归一化，与特征图三个维度无关，其公式为:

$$\hat{x}_{\mathrm{x,y}}^{c}=\frac{x_{x,y}^{c}-E\left[x_{x,y}^{c}\right]}{\sqrt{\mathrm{Var}\left(x_{x,y}^{c}\right)}}$$

式中 $x_{x,y}^{c}$ 表示第 c 个特征图在（x, y ）位置的像素值。归一化后，BN 层还需要进一步将归一化的特征重新映射到新的分布上

$$y^{c}=\gamma^{c}\hat{x}^{c}+\beta^{c}$$

式中 $\hat{x}^c$ 表示第 c 个特征图，γ^c, β^c 为学习的参数。

1.3 Dropout 层

常用在全连接层后，按一定概率随机使部分神经元失活，失活的神经元不参与前向传播。实验证明 Dropout 层可以有效缓解过拟合问题。

2 网络训练

2.1 Iteration & Batch size

Iteration 表示对神经网络做一次前向传播和反向传播的过程，目的是用迭代法，更新参数值，进行模型训练。深度学习任务中训练数据量很大，如果遍历一遍数据集计算损失函数再更新参数，计算开销很大，速度慢，因此深度学习中通常采用批梯度下降 (batch gradient descent) 的方法，即一次仅使用少量的样本计算损失函数更新参数，每次迭代使用的样本个数称为 batch size。

2.2 Dataloader

深度学习用到的数据量一般都比较大，无法一下子送入网络进行优化。在目前比较主流的深度学习框架 PyTorch 中，其定义了 Dataloader 类，用以加载数据。在 PyTorch 中，数据加载到模型的操作顺序是：

- 创建一个 Dataset 对象；
- 创建一个 DataLoader 对象；
- 循环这个 DataLoader 对象，将 image, label 加载到模型中进行训练。

DataLoader 是 PyTorch 中数据读取的一个重要接口，该接口定义在 dataloader.py 中。该接口的目的：将自定义的 Dataset 根据 batch size 大小、是否 shuffle 等封装成一个 Batch Size 大小的 Tensor，用于后面的训练。

2.3 Epoch

1 epoch 表示训练数据集中训练样本全部训练过 1 次，通常训练一个模型需要迭代多个 epoch。

3 参数初始化方法

3.1 均匀分布初始化

给定均匀分布的最大值和最小值，参数初始化按均匀分布进行采样得到。

3.2 高斯分布初始化

按照给定的均值和方差，得到对应的高斯分布，参数初始化值按高斯分布采样得到。

3.3 高斯分布初始化

按照给定的均值和方差，得到对应的高斯分布，对高斯分布进行采样，当采样值与均值的距离超过 2 个标准差，则舍弃从新采样。

3.4 Xavier 初始化

该初始化方法属于均匀分布初始化，但均匀分布的最大值和最小值由所在层的输入维度与输出维度决定，如下式，其中 n_i 为第 i 层的输入维度，设第 i 层卷积参数维度为 $C_{out} \times C_{in} \times K_h \times K_w$，则 $n_i = C_{in} \times K_h \times K_w, n_{i+1} = C_{out} \times K_h \times K_w$

$$w \sim \left[-\sqrt{\frac{6}{n_i + n_{i+1}}}, \sqrt{\frac{6}{n_i + n_{i+1}}} \right]$$

3.5 MSRA 初始化

该初始化方法属于高斯分布初始化，高斯分布的均值为 0，方差由输入维度决定，如下式，其中 n 表示输入维度。设某层卷积参数维度为 $C_{out} \times C_{in} \times K_h \times$

K_w，　则 $n = C_{in} \times K_h \times K_w$

$$w \sim G\left(0, \sqrt{\frac{2}{n}}\right)$$

4 经典网络结构

4.1 AlexNet

AlexNet 是 2012 年 ImageNet 项目的大规模视觉识别挑战（ILSVRC 中的胜出者。特点如下：

- AlexNet 网络由有 5 层卷积层，和 3 层全联接层构成，最后一个全联接层得到各类别概率。
- 使用 ReLU 作为激活层，代替 $tanh$。
- 使用 LRN (Local Response Normalization) 作为归一化函数。
- 使用了 3 个 maxpooling 层来做下采样，同时第一层卷积的卷积核大小设定为 11×11，滑动步长为 4，也有下采样的作用。
- 采用 Dropout 层来减少过拟合问题。

4.2 VGG

VGG 是 2014 年 ImageNet 分类任务的亚军，特点如下：

- VGG 的第一个卷积层没有使用 11×11 或 7×7 这种大卷积核，作者经过分析，发现由 3 个 3×3 的卷积级联在一起，可以得到和 7×7 相同的感受野，而 3 个 3×3 卷积的参数几乎为 1 个 7×7 卷积参数的一半。
- VGG 中去除了 LRN 归一化层，作者经过实验发现 LRN 层并不能提高精度，反而增加了计算量，需要更多的计算存储单元。
- VGG 首先训练浅网络结构，然后利用训练好的浅层网络，初始化深层网络，网络深度逐步加深。
- 在测试阶段，VGG 最后的三层 fc 层均被替换成相同参数量的 1×1 卷积层，这样可以保证输入图片的尺度可以变化，不需要局限在 224×224，最终输出的特征图直接做平均，即得到了最终的概率。

4.3 GoogLeNet

GoogLeNet 是 2014 年 ImageNet 分类任务的冠军，特点如下：

- Inception 结构有 4 个分支，包括 1×1 卷积，3×3 卷积，5×5 卷积，以及下采样分支，这种不同卷积核尺度的分支可以提供不同的感受野，最终各个分支的特征图级联在一起得到 Inception 结构的输出。进一步减少参数量。
- Inception 结构中每个卷积后都会经过 ReLU 激活。
- GoogLeNet 在 fc 层之前，采用 global average pooling 的方法，将特征图空间尺度压缩为 1×1，然后仅用 1 层 fc 结构，输出为各类别的概率。
- GoogLeNet 同样适用 dropout 层，减少过拟合问题。

4.4 ResNet

ResNet 是 2015 年 ImageNet 分类任务的冠军。特点如下：

- 提出了 shortcut 连接，和 BottleNeck 结构，成功保证深度越深的网络可以有越高的精度。
- 使用“卷积层-BN 层 -ReLU 层作为”基本网络单元。

4.5 DenseNet

DenseNet 进一步推广了 shortcut 结构，提出 DenseBlock 结构，每个中间层的特征图输出都会连接到后面的层的特征图，而 DenseNet 与 ResNet 最大的不同之处就是：ResNet 中两个分支采用加法的方式进行融合，而 DenseNet 中多个分支采用级联（在通道维度上拼接在一起）的方式融合。

4.6 SENet

SENet 是 2017 年 ImageNet 分类任务的冠军。利用 Squeeze 和 Excitation 两个关键操作来建模特征通道之间的相互依赖关系。通过学习的方式来自动获取到每个特征通道的重要程度，然后依照这个重要程度去提升有用的特征并抑制对当前任务用处不大的特征。

4.7 FCN

FCN: 全卷积网络（所有的层都是卷积层），是第一篇基于 CNN 的语义分割网络。FCN 将 AlexNet 结构中的全连接层转化成一个个的卷积层。在传统的 AlexNet 结构中，前 5 层是卷积层，第 6 层和第 7 层分别是一个长度为 4096 的一维向量，第 8 层是长度为 1000 的一维向量，分别对应 1000 个类别的概率。FCN 将这 3 层表示为卷积层，卷积核的大小 (通道数，宽，高) 分别为（4096,1,1）、（4096,1,1）、（1000,1,1）。为了得到像素级别的分割结果，对于最后一层的输出图像，需要进行 32 倍的上采样，以得到原图一样的大小。在 FCN 的基础上，研究人员做了进一步的改进，提出了 U-Net SegNet PSPNet DeepLab 等网络。

4.8 R-CNN

R-CNN 是区域卷积神经网络，是第一篇基于 CNN 的目标检测网络。R-CNN 的主要步骤包括：

(1) 利用选择性搜索 (SelectiveSearch) 算法在图像中从下到上提取 2000 个左右的可能包含物体的候选区域 (Region Proposal)；

(2) 因为取出的区域大小各自不同，所以需要将每个候选区域缩放（warp）成统一的 227×227 的大小并输入到 CNN，将 CNN 的 fc7 层的输出作为特征；(3) 将每个候选区域提取到的 CNN 特征输入到 SVM 进行分类；

(4) 使用 bounding box 回归器精细修正候选框位置。在 FCN 的基础上，研究人员做了进一步的改进，提出了 SPP-Net, Fast R-CNN, Faster R-CNN, YOLO，SSD 等网络。

4.9 基于传统机器学习算法的深度网络

受卷积神经网络的启发，将传统的机器学习算法扩展到多层得到的网络。比如将 DAISY 特征描述子扩展到多层得到的小波散射网络 (ScatNet)；将 PCA 扩展到多层得到的主成分分析网络 (PCANet)；将随机森林扩展到多层得到的深度森林 (Deep Forest)。