# 神经网络的架构

## 神经网络

* 神经网络（nn），有建立在它之上的许多类型。卷积神经网络，循环神经网络等等，常用于视觉图像处理领域的是卷积神域网络（cnn）。
* LeNet：是一种的用于处理手写数字识别的经典卷积神经网络，他的问世，写数字识别准确率得到大大的提高，取得了巨大的成功。

## 层次

* 卷积层：用多层卷积和对图像进行卷积运算。提取图像特征。
* 池化层：分为均值池化和最大池化。在保留图像特征的同时，极大的减少了计算量and防止过拟合现象。
* 全连接层：传统的神经网络架构。它的每一层中的每一个感知机都与前一层每一个感知机相连。由于他这样的特性会使得他的输出会与所有的数据相关，卷积池化解决了这样的问题。
* 激活函数：Sigmoid；ReLU：解决了梯度过小消失的问题；softmax：解决了归一化问题。
* 感知机：由一堆线性参数w和偏置参数b，将输入它的所有参数与w相乘再加上参数b，再由激活函数激活输出。

## 损失函数

* 损失函数（loss function）就是用来度量模型的预测值f(x)与真实值Y的差异程度的运算函数。
* 最小二乘法
* 极大似然估计法&交叉熵法：计算上相同，前者由概率论导出；后者由信息论导出。

## 训练

大量数据进入神经网络计算输出到损失函数，损失函数与标签进行比较计算出损失，再由反向传播与梯度下降，调整神经网络各个参数，直至损失越来越小。每次下降由步长控制。可设计计算法控制步长，防止局部最优，提高训练效率。

# 神经网络的搭建的流程

## 数据准备

* 数据集：神经网络训练需要大量的数据，要满足丰富（场景、明暗）、复杂（遮挡）、随机（角度）等要求。网络上由许多公共数据集。
* 数据增强：由程序随机对数据进行各种处理，可以大大增强数据量。
* 标注：数据集还需要标注，对于图像分类来说，标注就是他是哪一个分类。

## 模型搭建

设计针对任务的神经网络模型。有多少层，各层怎样安排等等。在计算速度和精度之间取得平衡。Python可以采用动态图的写法。设计损失函数计算方法以及后续返向传播等。还需要控制学习的步长。

## 训练

将大量的数据输入神经网络进行训练，使准确率越来越高。当最后训练模型准确率达到可接受程度，训练结束。

# 优化方法

* 如果数据比较稀疏，那么像SGD，NAG以及Momentum的方法往往会表现的比较差，这是因为对于模型中的不同参数，他们均使用相同的学习率，这会导致那些应该更新快的参数更新的慢，而应该更新慢的有时候又会因为数据的原因的变得快。因此，对于稀疏的数据更应该使用Adaptive方法（Adagrad、AdaDelta、Adam）。同样，对于一些深度神经网咯或者非常复杂的神经网络，使用Adam或者其他的自适应（Adaptive）的方法也能够更快的收敛。（<https://blog.csdn.net/autocyz/article/details/83114245>）

## 模型的设计原则

在Inception系列的第三篇论文里总结了四条CNN设计的四条原则。

**1. 避免表示瓶颈，特别是在网络的浅层。**

一个前向网络每层表示的尺寸应该是从输入到输出逐渐变小的。以下图为例，按照左边第一种的方式进行下采样，将会出现表示瓶颈。为了避免这个问题，提出了第四个图所示的结构来进行降采样。

**2. 高维度的表示很容易在网络中处理，增加激活函数的次数会更容易解析特征，也会使网络训练的更快。**

**3. 可以在较低维的嵌入上进行空间聚合，而不会损失很多表示能力。**

例如，在执行更分散（例如3×3）的卷积之前，可以在空间聚集之前（浅层）减小输入表示的尺寸，而不会出现严重的不利影响。

我们假设这样做的原因是，如果在空间聚合环境中（中高层）使用输出，则相邻单元之间的强相关性会导致在尺寸缩减期间信息损失少得多。鉴于这些信号应易于压缩，因此减小尺寸甚至可以促进更快的学习。

**4. 平衡网络的宽度和深度。**

通过平衡每个阶段的滤波器数量和网络深度，可以达到网络的最佳性能。增加网络的宽度和深度可以有助于提高网络质量。但是，如果并行增加两者，则可以达到恒定计算量的最佳改进。因此，应在网络的深度和宽度之间以平衡的方式分配计算预算。

**5. 此外再补充一条关于池化的使用。**

在网络的特征提取部分，使用最大池化。在分类部分，使用平均池化。

## 设计网络需要注意的问题

* 除去第一层会采用大卷积核一般也不会超过7\*7，其它层均使用3\*3或者1\*1卷积核
* 学会巧妙的使用1\*1的卷积核来进行通道降维和升维
* 使用batchnorm层，加快收敛速度（在tensorflow，注意设置batchnorm参数，在caffe中，bn层要和scale配合使用）
* 网络通常为block堆叠，每一个block都是一个最小重复单元（由多个卷积组成）
* 通常在进行下采样的时候会进行 channel数量的翻倍
* 在shuffleNet V2中，作者也给出CNN 网络结构设计的准则来帮助神经网络可以更高效。下面先贴出文中观点：
  1. 输入通道数与输出通道数保持相等可以最小化内存访问成本（memory access cost,MAC）。
  2. 分组卷积中使用过多的分组数会增加内存访问成本（MAC)
  3. 网络结构太复杂（分支和基本单元过多）会降低网络的并行程度
  4. Element-wise 的操作消耗也不可忽略（包括ReLU，Tensor的相加，偏置的相加等等操作）
* 主干网络优先考虑resnet+FPN网络结构作为baseline在进行额外的优化。
* <https://zhuanlan.zhihu.com/p/66500659>