

课程介绍

图像分类与机器学习基础

- 数据驱动算法
  - 最邻近算法
  - k最近邻分类器
- 线性分类
  - 评分函数 (score function)
  - 损失函数 (loss function)
  - 理解线性分类器
    - W是所有分类器的组合
    - 将线性分类器看做模板匹配
    - 将图像看做高维空间的点
  - 偏置项和权重合并
  - 图像数据预处理
    - 归一化 (normalization)
  - 失效的情形

损失函数与最优化

- 线性分类：损失函数
  - 损失函数的概念
- 多类支持向量机损失
  - 数据损失 (data loss)
  - 正则化损失 (regularization loss)
  - Softmax分类器损失
  - Softmax 和 SVM 比较
- 优化
  - 损失函数可视化
  - 优化策略
    - 目标：找到能够最小化损失函数值的权重 公式。
    - 随机搜索
    - 随机本地搜索
    - 跟随梯度
- 梯度计算
  - 数值梯度法
    - 简单但缓慢
  - 分析梯度法
    - 快速但容易出错，使用微分实现
  - 普通梯度下降
- 梯度下降
  - 小批量梯度下降
    - 更快地收敛，并以此来进行更频繁的参数更新
  - 随机梯度下降SGD
    - 每次使用1个样本来计算梯度

神经网络与反向传播

- 反向传播算法
  - 标量形式反向传播
  - 向量形式反向传播
  - 计算图
- 神经网络简介
  - 神经网络算法介绍
  - 神经网络与真实的神经对比
  - 常用的激活函数
    - Sigmoid
      - 数学公式： $\sigma(x) = \frac{1}{1+e^{-x}}$
      - 图形
    - Leaky ReLU
      - 数学公式： $\max(0.1x, x)$
      - 图形
    - tanh
      - 数学公式： $\tanh(x)$
      - 图形
    - ReLU
      - 数学公式： $\max(0, x)$
      - 图形
    - Maxout
      - 数学公式： $\max(w_1^T x + b_1, w_2^T x + b_2)$
      - 图形
    - ELU
      - 数学公式： $\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$
      - 图形
- 神经网络结构
  - 全连接层 (fully-connected layer)
  - 图形

卷积神经网络

- 卷积神经网络的历史
  - 卷积网络的产生过程
- 卷积神经网络详述
  - 常规神经网络
    - 每个隐层都是由若干的神经元组成，每个神经元都与前一层中的所有神经元连接，但是在一个隐层中，神经元相互独立不进行任何连接。
    - 效率不高，且参数量大，可能会导致网络过拟合
  - 卷积神经网络
    - 各层中的神经元都是3维的：宽度、高度和深度
- 卷积神经网络的结构
  - 全连接层
  - 卷积层
    - 卷积层是构建卷积神经网络的核心层，它的参数是由一些可学习的滤波器 (filter) 集合构成的
    - 局部连接
      - 每个神经元和原图像只在一个小区域进行全连接
      - 局部连接的空间大小叫做神经元的感受野
    - 输出数据尺寸
      - 输出数据体的尺寸由三个超参数控制：深度 (depth)，步长 (stride) 和零填充 (zero-padding)
      - 深度：滤波器的数量
      - 步长：滤波器每次移动跨越的像素数量
      - 零填充：在图像的边界外填充零像素点
    - 参数共享机制
      - 在卷积层中使用参数共享是用来控制参数的数量
  - 池化层
    - 连续的卷积层之间会周期性地插入一个池化层，作用是逐渐降低数据体的空间 (宽、高) 尺寸，减少网络中参数的数量，使得计算资源消耗变少，也能有效控制过拟合
  - 归一化层
  - 全连接层
- 卷积神经网络经典案例
  - LeNet
  - AlexNet
  - ZF Net
  - GoogLeNet
  - VGGNet
  - ResNet

在全连接层或者卷积层，输入数据与权重相乘后累加的结果送给一个非线性函数，即激活函数

- Sigmoid函数
  - 数学公式： $\sigma(x) = 1/(1 + e^{-x})$
  - 求导公式： $\frac{d\sigma(x)}{dx} = (1 - \sigma(x)) \sigma(x)$  (F0小于0)
  - 缺点①：Sigmoid 函数饱和时使梯度消失
  - 缺点②：Sigmoid 函数的输出不是零中心的
  - 缺点③：指数型计算量比较大。
  - 数学公式： $\sigma(x) = \frac{1}{1+e^{-x}}$
  - 图形
- tanh函数
  - 将实数值压缩到-1,1之间
  - 数学公式： $\tanh(x) = 2\sigma(2x) - 1$
  - 图形
- ReLU 函数
  - ReLU 只有负半轴会饱和，节省计算资源，不含指数运算，只对一个矩阵进行阈值计算，更符合生物学观念；加速随机梯度下降的收敛。
  - 数学公式： $f(x) = \max(0, x)$
  - 图形
- Leaky ReLU
  - x<0 时给出一个很小的梯度值
  - 数学公式： $f(x) = 1(x < 0)(\alpha x) + 1(x > 0)(x)$
  - 图形
- ELU (指数线性单元)
  - 介于 ReLU 和 Leaky ReLU 之间，有负饱和的问题，但是对噪声有较强的鲁棒性
  - 数学公式： $f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(\exp(x) - 1) & \text{otherwise} \end{cases}$
  - 图形
- Maxout
  - 是对 ReLU 和 leaky ReLU 的一般化归纳
  - 公式： $\max(w_1^T x + b_1, w_2^T x + b_2)$

神经网络训练技巧

- 数据预处理
  - 减均值 (Mean Subtraction)
    - 对数据中每个独立特征减去平均值，在每个维度上都把数据的中心都迁移到原点
  - 归一化 (Normalization)
    - 将数据的所有维度都归一化，使其数值范围都近似相等
  - 主成分分析 (PCA)
    - 先对数据进行零中心化处理，然后计算协方差矩阵，它展示了数据中的相关性结构
  - 白化 (Whitening)
    - 白化操作的输入是特征基准上的数据，然后对每个维度除以其特征值来对数值范围进行归一化
- 权重初始化
  - 全零初始化
  - 小随机数初始化
  - Xavier/He初始化 (校准方差)
  - 稀疏初始化
- 批量归一化
- 层归一化
- 卷积神经网络中归一化
- 监控学习过程
  - 监控学习过程的步骤
    - 数据预处理，减均值
    - 选择网络结构
    - 合理性检查
    - 梯度检查
    - 正式训练，数值跟踪，特征可视化
  - 训练过程中的数值跟踪
    - 跟踪损失函数
    - 跟踪训练集和验证集准确率
    - 跟踪权重更新比例
    - 第一层可视化
- 超参数调优
- 更好的优化 (参数更新)
  - 批梯度下降 (BGD)
  - 随机梯度下降 (SGD)
  - 动量更新
  - Nesterov动量
  - 自适应梯度算法
  - 均方根支柱算法
  - 自适应-动量优化
  - 学习率退火
- 正则化
  - 正则化强度是控制神经网络过拟合的好方法
  - 正则化方法
    - L2正则化
    - L1正则化
    - 随机失活 (Dropout)
  - 随机失活 (Dropout)
- 迁移学习 (Transfer Learning)
- 模型集成 (Model Ensembles)

经典CNN架构

- AlexNet
  - 2012年，共有5个卷积层、3个池化层、2个归一化层和三个全连接层
  - 图形
- VGG
  - 子主题 2
  - VGG有16层和19层两种。卷积核只使用 公式，步长为 公式，pad为 公式；池化区域 公式，步长为2
  - 图形
- GoogLeNet
- ResNet
  - ResNet通过使用多个有参层来学习输入与输入输出之间的残差映射，而非像一般CNN网络 (如 AlexNet/VGG等) 那样使用有参层来直接学习输入输出之间的底层映射
  - 图形

深度学习框架

- 深度学习硬件
  - CPU
    - CPU一般有多个核心，每个核心速度都很快都可以独立工作，可同时进行多个进程，内存与系统共享，完成序列任务时很有用。图上CPU的运行速度是每秒约 540 GFLOPs 浮点运算，使用 32 位浮点数 (注：一个 GFLOPs (gigaFLOPs) 等于每秒十亿 (公式) 次的浮点运算
  - GPU
    - GPU (Graphics Processing Unit) 是图形处理单元 (又称显卡)，在物理尺寸上就比 CPU (Central Processing Unit) 大得多，有自己的冷却系统。最初用于渲染计算机图形，尤其是游戏。在深度学习上选择 NVIDIA (英伟达) 的显卡，如果使用AMD的显卡会遇到很多问题。
  - TPU
    - TPU (Tensor Processing Units) 是专用的深度学习硬件。
- 深度学习框架
  - TensorFlow
  - PyTorch