<https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html>

<https://www.bilibili.com/video/BV1Y7411d7Ys?p=1>

1. P1, overview; human intelligence; make decision; infer; prediction; supervised learning, training; 穷举法, 贪心法, 分治法, 动态规划; AI > machine learning > representation learning > deep learning; rule-based system, hand-designed program; classic machine learning, hand-designed features; representation learning, 希望自动提取feature; curse of dimension; manifold; classification, clustering, regression, dimension reduction; disadvantage of SVM; 哺乳动物大脑里视觉中枢是分层的; 感知机; back propagation; 计算图; pytorch; 动态图; (2021-3-6)
2. P2, 线性模型; dataset, model, training, inferring; 训练集, 测试集; dev set; starts with a random guess; 观察loss是否收敛; DL可视化很重要; (2021-3-6)
3. P3, 梯度下降算法; 数据规模大, 不能用穷举法; 凸函数方便用分治法找最优点; 避免鞍点(高维空间很容易避免鞍点); 线性回归的梯度下降法; 常见的是随即梯度下降(SGD); 批量随机梯度下降batch; (2021-3-7)
4. P4, 反向传播; 计算图; 线性图层多层少没有区别; 链式求导法则; (1) 创造计算图; (2) local gradient; (3) give gradient from successive node; (4) use chain rule to compute backward gradient; computational graph of linear model; tensor in pytorch; tensor contain data and grad; 直接使用张量是在构建计算图, 不是在做乘法运算; 一做backward计算图就释放了, 下次重新创建; 取data不涉及建立计算图, 只是纯数值的操作; item也是为了防止产生计算图; 梯度清零必须显示表示; (2021-3-7)
5. P5, 用pytorch实现线性回归; nn.Module; (1) prepare dataset; (2) design model using class inherit from nn.Module, computing yhat; init and forward (前馈时候执行的计算) function are necessary; 定义forward以后反向传播可以自动实现; super那句所有模型基本一样, just do it; nn就是neural network; \*args可以放好多个参数, 转成元组; \*\*kwargs则是转成词典; forward执行call的功能; 实例化的对象可以callable; 只要涉及计算图必然继承自nn.Module; (3) construct loss and optimizer; optimizer不会构建计算图; SGD是一个类; model.parameters()自动取出所有权重; (4) training cycle (forward, backward, update); 梯度归零, 反向传播, 更新; (2021-3-7)
6. P6, 逻辑回归; 逻辑回归是分类; torchvision包含了常见的数据集; regression vs classification; 二分类问题; 映射到[0, 1]; 饱和函数sigmoid; loss function for binary classification; 不是比较度量空间的差异, 而是比较分布的差异; 用交叉熵(越小越好); F来自函数包; (2021-3-8)
7. P7, 处理多维特征的输入; row is sample, column is feature; 转换成矩阵可以利用GPU的并行能力提高运行速度; 矩阵是空间维数变换的函数; 学习能力太强会过拟合; important skills, 读文档, 基本架构理解; prepare dataset; define model; construct loss and optimizer; training cycle; (2021-3-9)
8. P8, 加载数据集; dataset是为了支持索引(下标操作), data\_loader是为了拿出一个mini batch(一组操作); epoch, one forward pass and one backward pass of all training examples; batch size, the number of training examples in one forward backward pass; iteration, number of passes, each pass containing batch size examples; shuffle就是把数据集打乱; utility就是工具; 结构化数据基本可以一次性读到内存中; 使用多线程时加上if \_name\_ == ‘\_main\_’; enumerate可以得知当前是第几次迭代; 嵌套循环是为了利用mini batch; (2021-3-10)
9. P9, 多分类问题; softmax layer; 交叉熵损失; 越吻合损失越小, 损失越小越好; CrossEntropyLoss = LogSoftmax + NLLLoss; 黑白图像是单通道, RGB图像是三通道; 提取特征, 小波变换比FFT更好; CNN是自动特征提取; (2021-3-11)
10. P10, 卷积神经网络基础篇; 全连接层, 每一个输入都参与下一层输出的计算; subsampling是为了减少数据量; 卷积+下采样=特征提取器; 栅格图像; 矢量图像; 现场画的; 对图像块做卷积, 图象块滑动遍历, 再把结果拼到一起; 和卷积核做数乘; 每个通道的卷积核各不相同, 再把所有通道的卷积结果相加; 卷积后, 3通道变成1通道; 输入通道, 输出通道, 卷积核尺寸; padding填充0, 保持图像大小不变; stride是步长; max pooling; move model to gpu; (2021-3-11)
11. P11, 卷积神经网络高级篇; GoogleNet; 减少代码冗余, 函数/类; inception块; 找到最优路径(策略); 1\*1卷积; 信息融合; 减少运算量; 训练神经网络不是越多越好, 太多可能过拟合; 层数过多可能过拟合, 也可能梯度消失; residual net解决梯度消失; densely connected CNN, 各种跨层连接; 花书《深度学习》; 阅读pytorch文档; 复现经典工作(读代码, 写代码); 扩充视野; (2021-3-11)
12. P12, 循环神经网络基础篇; RNN专门处理序列; RNN cell是同一个线性层; 依次算出隐藏层就是RNN; 绿色的是一个线性层, 黄色的是一个线性层, 红色的是一个线性层; change index to one-hot vector; 58:56
13. -