PyTorch深度学习实践知识点总结整理 (by A.L.)

> [PyTorch深度学习实践]

01 Linear Model

- 线性模型形如以下，其中w, b均从random (initial) guess开始取值，在给定范围内进行搜索。

$$y = x * w \text{或} y = x * w + b$$

- 将给定数据集输入线性模型后，输出预测值以及损失函数 (loss function) 以评估真实值和预测值之间的误差

---

02 Gradient Descent

- 损失函数 (loss function) 沿梯度 (gradient) 方向下降最快
- 以线性模型为例，梯度 gradient = ∂cost / ∂w，update如下

$$w := w - \alpha * (\partial cost / \partial w)$$

-

	Gradient Descent	Stochastic Gradient Descent (SGD)
特点	对所有样本的梯度求均值更新权重	对每一个样本的梯度进行更新；可一定程度解决DL中的鞍点问题
学习器性能	低	**高**
时间复杂度	**低**（可并行计算）	高（两个样本之间的GD不能并行，存在依赖）

  - 折衷方案：(Mini -) Batch SGD

---

03 Back Propagation

- 反向传播的本质是用**链式法则**求出目标函数关于每一层参数的梯度 (《深度强化学习》P16)
- 计算图 (computational graph) : 通过 参数 (e.g. w1, w2, b)、输入样本 (e.g. x1, x2) 、算子 (neuron, e.g. +, -, *) 沿图前向 (forward) 计算各项 (经过算子计算的结果)及其对上一项（参数或上一算子计算结果）的偏导数，和损失函数；经由各项偏导反向 (backward) 利用链式法则计算损失函数对参数的梯度，以待后续更新参数
- PyTorch中的Tensor (类) 存储Data (参数, e.g. w) 以及关于损失函数的梯度Grad (e.g. ∂lost / ∂w)
- 调用损失函数loss构建计算图，利用 .backward() 计算设置 requires_grad = True 的参数的梯度，使用SGD更新参数。注意执行.backward()后构建的计算图会被释放。

---

04 Linear Regression with PyTorch

1. 准备数据集
2. 利用继承自nn.Module的类 (class) 设计网络模型
3. 利用PyTorch API构建 (选择) 损失函数 (loss) 及优化器 (optimizer)
4. 训练：前馈 (计算y^, loss) --> (梯度清零后) 反馈 (计算梯度) --> 更新参数

---

05 Logistic Regression

- logisitc regression 主要用于**分类 (classification)**，预测属于每一个分类的概率 [0,1]
- logistic function: $\sigma(x) = 1 / 1+e^{-x}$; PyTorch中约定俗成logistic function为sigmoid function
- loss function for binary classification: `loss = - ( ylogy^ + (1-y)log(1-y^) )`
  - y=1时, `loss = -ylogy^`
  - y=-1时, `loss = -log(1-y^)`

---

06 Multiple Dimension Input

- dataset中通常一行称为一个**样本 (sample)**，一列称为一个**特征 (feature)**
- ```
  self.linear = torch.nn.Linear(8, 1)
  ```
  - 8: 8D, size (number of features) of each input sample
  - 1: 1D, size of each output sample
- 神经网络本质上是寻找一种非线性的空间变换函数（e.g. 8D --> 1D，引入激活函数）
- 变换的维度和层数，决定了网络的复杂程度。过程中到底如何变化，即为超参数搜索

  ```

07 Dataset and Dataloader

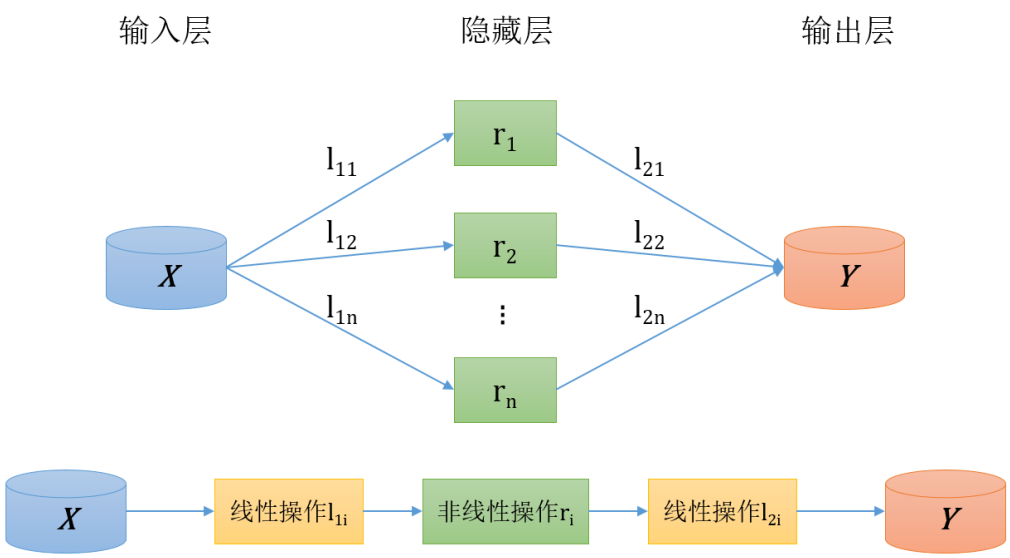
> 参考: 深度学习实践 加载数据集

1. `DataSet` 是抽象类 (abstract class)，不能实例化对象，主要是用于构造我们的数据集
2. 需要 `mini_batch` 就需要 `import DataSet, DataLoader`
3. 继承DataSet的类需要重写 `__init__()`，`__getitem__()`，`__len__()`，分别是为了加载数据集，获取数据索引，获取数据总量
4. DataLoader对数据集先打乱 (shuffle)，然后划分成mini-batch
5. 在训练模块加入：`if __name__ == '__main__':`

补充1：神经网络基础

> 参考：五分钟秒懂神经网络原理，机器学习入门教程

神经网络 (neural network) 主要用于完成一种离散的**非线性**拟合的任务



补充2：卷积神经网络

> 参考：《深度强化学习》P12

- 卷积神经网络 (convolutional neural network, CNN) 的输入是矩阵或三阶张量，CNN从该张量提取特征，并输出提取的特征向量。
- 图片通常是矩阵（灰度图片）和三阶张量（彩色图片），可以用CNN从中提取特征，然后用一个或多个全连接层做分类或回归