BDMI-课程编号-01510243

SaturnLab

iCenter 清华大学iCenter

大数据与机器智能

机器智能-深度学习2

清华大学iCenter

导学

简介

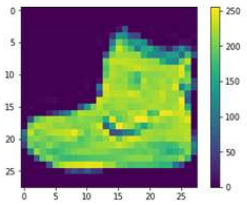
- 上一节
  - 人工神经元和神经元网络
  - 模拟布尔运算
  - 解决异或问题
- 这节课内容
  - 用神经网络做什么？怎么做？
  - 神经网络的用途是什么？
  - 如何训练一个神经网络？

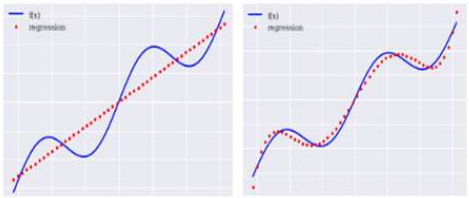
目录

- 神经网络的用途
- 神经网络的训练
  - 损失函数
  - 反向传播
  - 权重更新
- 神经网络的张量表示

神经网络的用途

- 神经网络是一个很好的数学模型，属于**机器学习**的**监督学习方法**
- 多层的神经网络又称为深度学习，主要用于解决两类任务：
- 分类任务：通过对数据集的学习，对新数值进行集合分类
  - **对应的分类方法（Classification）**
- 预测任务：通过对数据集的学习，预测新的数值
  - **对应的回归方法（Regression）**





用途示例：分类任务-判断性别

- 根据身体特征的数据集，进行一个判断性别的简单任务

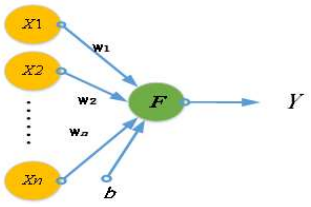
- 数据集：记录全班60人的身高\体重的身体特征，以及对应的性别
  - 比如，
  - X1：头发长度、
  - X2：身高、
  - X3：体重
  - … 等身体指标

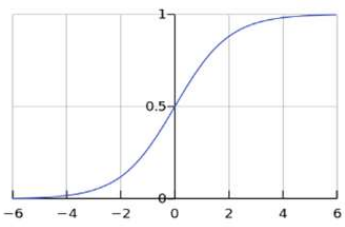
想一想，练一练？

- （判断性别）-简单分类
- 数据集：
  - 请每个同学（匿名）给出自身数据(身高、体重、性别)，为一份样本
  - 所有同学的样本集组成本班的数据集
- 设计一个分类器，进行分类？
  - 提示可以用人工神经元的方法

人工神经元进行二元分类

- 具体方法采用逻辑斯提模型（logistic regression），输出Y：性别的概率
  - 取40人的数据，输入每个人的身体特征：X1，X2，X3，…，
  - 权重：W1，W2，W3，…，表示不同特征对判断结果的贡献率
- 如何找到合适的权重？经验调参法





想一想

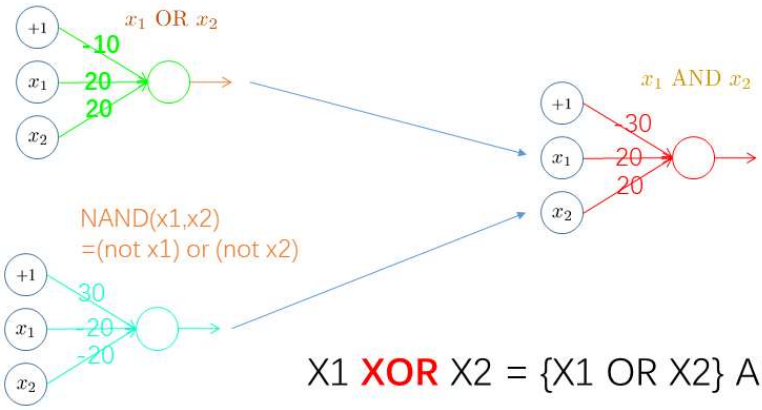
- 单个人工神经元-判断性别-二元分类
- 如何进行**手动**调整权重参数?
- 判断输入与结果是正相关或负相关，确定权重W是正还是负。
- 判断输入与结果的重要程度，越重要，权重越大。

多层神经网络

Deep

两层网络结构-解决XOR问题

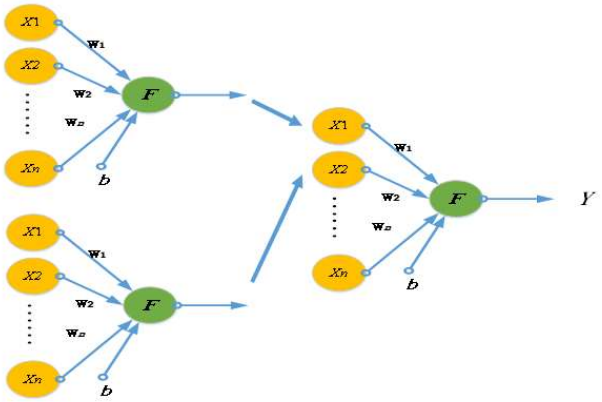
- 两层网络，一层神经单元的输出，作为另一层神经元的输入



X1 **XOR** X2 = {X1 OR X2} AND {NAND{X1,X2}}

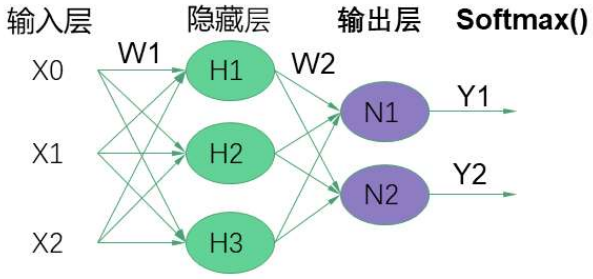
想一想

- 用二层网络，进行二元分类？
- 如何确定这些网络参数？



多层神经网络

- 前馈网络（Feedforward NN），举例：
  - 多层全连接网络（FCN）、多层感知机（MLP）、多个密集层网络（Dense）
- 示例网络：有3个输入，2个输出；输入层、隐藏层，输出层；
- 隐藏层、输入层，共有5个神经元。



Softmax处理

- 输出层的Softmax 处理，计算出一个概率分布向量:
- 所有输出的数值是正的，所有分量之和为 1。

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

numpy_activation_function.ipynb

想一想，练一练

- 用python代码描述一下Softmax函数
- 提示可以用Numpy

- 输入为[1,2,3,4,5,6,7,8,9],
- softmax输出为[0.0,0.001,0.002,0.004,0.012,0.031,0.086,0.233,0.632]

分对数logit

- 分对数（logit）模型
- 把区间(0,1)内的数值，变换到区间(-∞,+∞)

- 与sigmoid函数互为反函数

$$\sigma(x) = \frac{1}{1+\exp(-x)}.$$

$$\forall x \in (0,1), \sigma^{-1}(x) = \log\left(\frac{x}{1-x}\right)$$

想一想，练一练

- 用python代码描述一下logit函数
- 提示1：可以用Numpy的函数
- 提示2：sigmoid函数的反函数

- 输入：x=0.9
- 输出：logit

遇到了问题：神经网络的层数

- 网络层数越多
  - 一方面，神经网络的表达能力增强了
  - 另一方面，神经网络的权重数量增加了

- 如何确定这些权重的优化参数呢?

- 有没有自动化的方法
  - 答案是：BP方法

神经网络的训练

- 如何求W1, W2?

- 一般的流程:
  - 1. 确定损失函数（Loss）
  - 2. 权重初始化（Initialization）
  - 3. 反向传播（Back Propagation）
  - 4. 权重修正(weights Adjusting)

损失函数

从优化的角度

神经网络方法

- 神经网络方法属于机器学习的监督学习方法（supervised learning）
- 监督学习特点是根据训练数据集（data set）进行学习，然后推断新的实例。
- 训练数据集由样本（sample）集组成，每个样本上都有相应的标签（label），用来指导训练过程。

神经网络的输出与标签的差异

- 采用带标签的训练样本对神经网络进行学习，确定网络的权重参数
  - 数据集：(x_1,y_1'), (x_2,y_2'), (x_3,y_3'), …,
- 网络输出
  - 实际输出：(x_1, y_1), (x_2, y_2), (x_3, y_3), …,
  - 其中：Y为实际输出
- 实际输出Y与预期输出Y'的差异：
  - Y'为预期输出（标签）
  - 预测输出Y与标签Y'：(y_1, y_1'), (y_2, y_2'), (y_3, y_3'), …,

定义损失函数

- 引入度量函数：度量y与y'差异，计算出差异值Delta；
- 度量函数D(y, y')：
  - 举例，如绝对值求和：$\min\{(|y_1-y_1'|+ |y_2-y_2'|+ |y_3-y_3'|+\cdots)\}$
  - 平方和 /均方差，交叉熵等
- 训练的目标，通过调整网络的内部权重，使得在训练数据（样本）输入到网络后，网络的实际输出与预期输出（即标签）之间差异值Delta的最小化过程。

损失函数的度量函数

- 深度网络的预期输出结果会与所对应的实际标签之间的存在差别。
- 用**度量函数**来表示这种差异，称为损失函数(Loss)或成本函数(Cost)。
- 损失函数的的度量函数的具体方式：
  - 对于回归任务，通过**均方误差**的公式，来计算损失。
  - 对于分类任务，通过**交叉熵**的公式，来计算损失(Loss)。
- 训练过程是损失函数的最小化过程，转化为最优化问题。

$$H_{y'}(y) = -\sum_i y_i' \log(y_i)$$

其中y'表示训练样本对应的标签，y表示神经网络的输出。

均方差-回归任务的损失度量函数

- **方差（variance）**是各个数据和平均数之差的 平方和 的平均数。
- **标准差（standard deviation）**是方差的**平方根**。标准差也称为均方差。

- **均方误差（mean squared error, MSE）**是各个数据偏离真实值差值的平方和的平均数，也就是误差平方和的平均数。
- 均方误差的开方叫**均方根误差（Root Mean Squared Error, RMSE）**
- 注意：均方误差不同于均方差

想一想，练一练

- 何为均方差?

- 用python代码描述一下，然后计算一下：

- 输入一组数，计算均方差。
- [72、94、79、83、65、81、73、67、85、82]

交叉熵-分类问题的损失的度量函数

- 交叉熵是负对数似然损失函数，主要用于分类任务
- 交叉熵损失函数表示如下：
  - 其中**y'**表示训练样本对应的标签，**y**表示神经网络的输出。

$$H_{y'}(y) = -\sum_i y'_i \log(y_i)$$

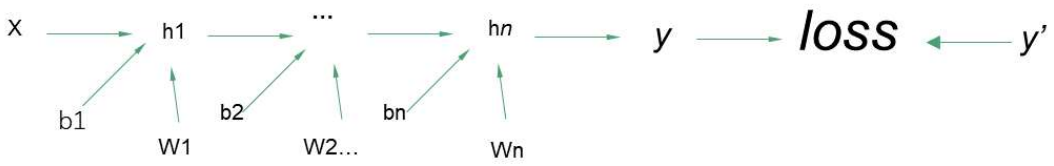
- 比如，MNIST手写数字的分类，0的向量[1,0,0,0,0,0,0,0,0,0]，网络输出（softmax）的概率向量是[0.9，0.05,0.05,0,0,0,0,0,0,0]

想一想，练一练

- 何为交叉熵?
- 用Python代码描述一下
- 输入两组数组，计算一下交叉熵。
- 向量1=[1,0,0,0,0,0,0,0,0,0],
- 向量2=[0.9, 0.05,0.05,0,0,0,0,0,0,0]

损失函数的计算过程

- 观察：损失函数是权重值的函数。
- $y=h_n(\cdots h_3(h_2(W_2*h_1(W_1*X+b_1)+b_2)+\cdots+b_n)$;
- loss=D(y, y')



Q.V. Le. A Tutorial on Deep Learning Lecture Notes 1-2, 2015.

反向传播算法

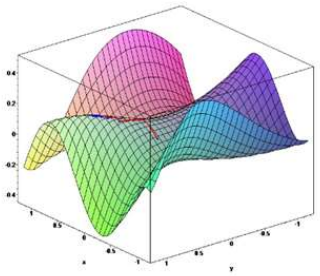
梯度计算

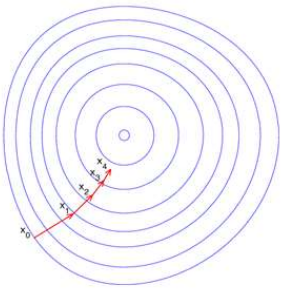
导数与梯度

- 回顾一下高数:
- 什么是导数（Derivative）？
  - 导数的定义：函数F(x)在某个点上变化趋势。
- 什么是梯度（Gradient）？
  - 梯度的定义：函数F(x)在某个点上增长变化率最大的方向，就是导数的方向。

梯度下降法（Gradient descent）

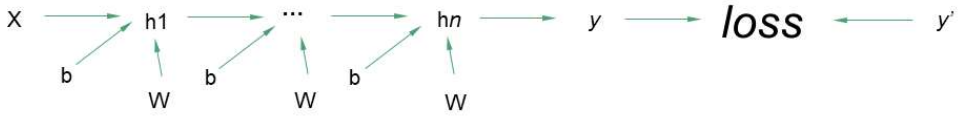
- 梯度下降法的原理：如果实值函数F(x)在点 a 处可微且有定义，那么函数F(x) 在a 点沿着梯度相反的方向下降最快。
- 梯度下降法的过程：找到一个函数的局部极小值，必须向函数上当前点对应梯度的反方向，按照规定步长距离点，进行迭代搜索。
- 梯度下降法也称为最速下降法，是一个最优化算法。



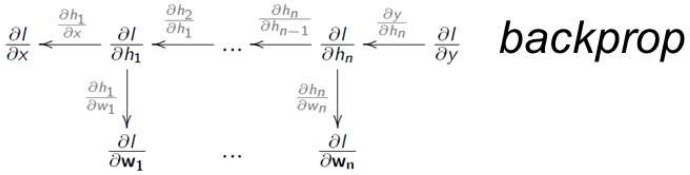


损失函数对权重的梯度计算-反向传播算法

- 一种更加抽象表示多层神经网络的结构，这种表示抽象了每层基本特征。



- 反向传播算法：
  - 根据损失函数的性质以及链式求导法则
  - 反向逐层计算损失函数对权重的梯度（各个偏导数）



想一想，练一练

- 如何进行求导运算？
- Python计算偏导数？

练习代码： 3_automatic_differentiation.ipynb

权重更新

随机梯度下降法

梯度算子-权重更新与学习率

- Loss记为函数J(θ)， 是权重数值θ的函数;
  - 在θ原值的基础上，沿着梯度方向的行动一个步长，得到一个新的θ‘数值，使得损失函数变化的最快
  - 不断调整θ数值，Loss随着θ取值的序列，Loss序列的数值不再有大的变化，就认为是收敛了。
- 步长又称为学习率（learning rate），一般为0.01
  - 学习率过大会引起震荡，学习率太小收敛太慢

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta).$$

随机梯度下降法-权重值更新

- **随机**梯度下降方法(stochastic gradient descent, SGD)是最常用的权重调节方法，通过权重的调整，最小化损失函数。
- 随机梯度下降法的核心是一个“随机样本”
- 随机梯度下降法步骤如下：
  - 步骤 1. 随机初始化每个神经元输入权重和偏差；
  - 步骤 2. 选取一个随机样本；
  - 步骤 3. 根据网络的输出结果，从最后一层开始后，逐层计算每层权重的偏导数；
  - 步骤 4. 逐层调整每层的权重，产生新的权重值。
  - 返回到步骤2，继续随机选取下一个样本。

神经网络训练实际过程

小批量训练

神经网络训练的实际过程

- 深度网络的训练将样本数据“分批训练”
  - 最简单的批量选择是使用整个数据集，又称为批量训练（batch training）
- 优点：
  - 批量训练的收敛性是有保证的。
- 缺点：
  - 更新模型参数时，需要遍历整个数据集

神经网络训练的实际-小批量训练(mini-batch)

- 小批量训练是批量训练和随机梯度下降法的一个折中
  - 整个训练集称为一个批次（batch），先将整个训练集分成多个大小相同的子集，每个子集称为一个迷你批次（mini-batch）。子集的大小由参数迷你批次大小（mini-batch_size）控制。
  - 每个批次的数据被依次送入网络进行训练。训练完一个迷你批次， 被称为一次迭代（iteration）。
  - 一个时代（epoch）是指训练集中所有训练样本都被送入网络，完成一次训练的过程。
- 其中每一步的模型参数的更新
  - 使用不止一个样本，这称为迷你批次（minibatch）。
  - 每次迭代所用的样本数目称为迷你批次大小（minibatch size）。
  - 当迷你批次大小为1时，就退化为普通的随机梯度下降。

神经网络的训练与推断

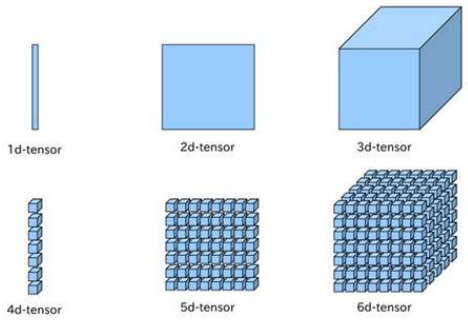
- 神经网络的训练（确定网络结构与权重）：
  - 训练目标：网络输出与预期输出（即标签）之间差异的度量函数，取得最小值（或极小值）
  - 训练过程：输入训练数据（样本）到人工神经网络，调整的每层的内部权重参数（梯度下降法等优化方法）
  - 参数固化（frozen）：固定每层内部权重模型参数，形成统一的模型参数网络
- 神经网络的应用-推断（Inference）：
  - 用**训练集上**得到模型参数，对训练**数据集外**的数据推断（预测）其可能的标签。

用张量表示神经网络

张量（Tensor）

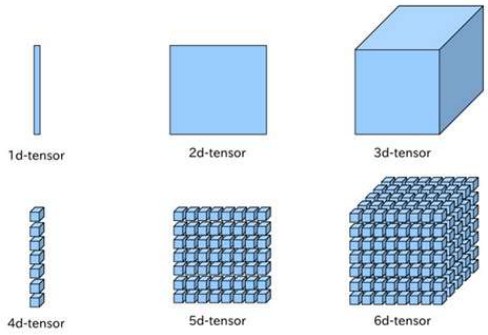
张量（Tensor）的概念

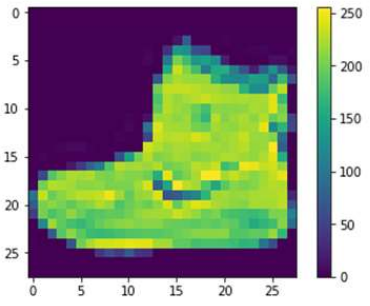
- 张量（Tensor）的实质是N维数组
  - 1维Tensor的形式是向量；
  - 2维Tensor的形式是矩阵；
  - 3维Tensor的形式是由矩阵组成的向量；
- … …
- 计算机处理离散的数值的组织方式



Tensor用（Numpy）表示

- Tensor的Numpy表示Ndarray
- 对于一个4*5*6的Tensor
  - rank : 3d
  - length: 4, 5, 6
  - shape: [4, 5, 6]
  - volume: 4*5*6=120
- 形象化例子：彩色图像，三维张量
  - 每个像素点可以用（行，列，颜色）来表示。
  - 每个像素点的位置（行，列）
  - 每个像素点的颜色用RGB表示





想一想，练一练

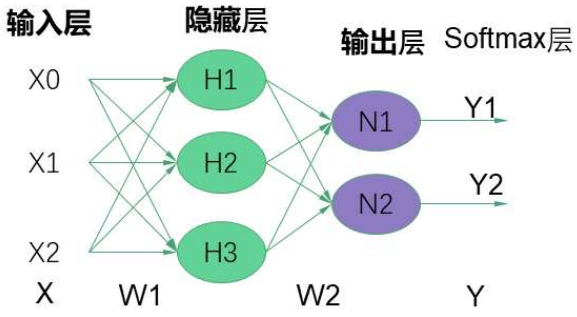
- 用Python代码描述一下：
- 提示：用Numpy库

- 一个4x4的二维张量
- 一个4x5x6的三维张量

- 请大家投稿一下

用张量表示神经网络

- 一个前馈网络结构（feedforward）的神经网络，又称为：
  - 多层全连接网络（FCN）、多层感知机（MLP）、密集网络（Dense）
- 实际例子：
  - 多层全连接网络有5个神经元
  - 有3个输入，2个输出，中间有1个隐藏层，有1个输出层。
- 张量表示：
  - X, W1, H, W2, N, Y均为张量
  - N=W2*(W1*X);　（张量运算）
  - Y=Softmax(N)；　（张量操作）
  - 其中W1，W2是待确定的张量

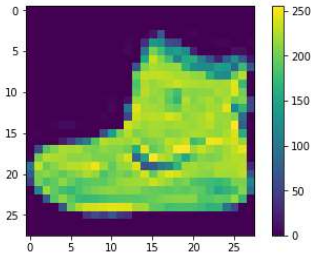


想一想，练一练

- 用Python代码描述一下：
  - 提示：用Numpy库
- 3层神经网络有5个神经元
  - 有3个输入，中间隐藏层有3个神经元，输出层有2个神经元
- 输入：X=[22, 35, 86]
  - 权重矩阵：W1，W2；用随机数填充 。
- 计算输出：Y
- 请大家投稿一下

用张量表示图像

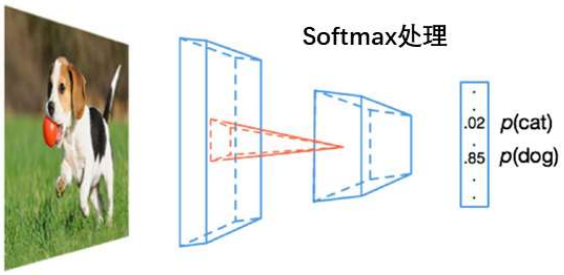
- 图像的张量表示
  - 图像是像素的二维张量
  - 每个像素也是张量
- 手写数字MNIST数据集
  - 灰度图像，0~255，对应8位2进制数字
  - 二值图像，黑白，0代表黑色，1代表白色
- 时尚MNIST数据集
  - 彩色图像（RGB)
  - 24位二进制数字
  - 红(Red)，绿 (Green)，蓝(Blue)， 对应的值域从0到255，对应8位2进制数字



用张量进行图像分类的示例

- 输入：图像，
- 输出：概率向量
- 输出层计算出概率分布，用Softmax 操作:
  - 输出值的所有分量之和为 1，所有输出的数值是正的。

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$



https://tensorflow.google.cn/api_docs/python/tf/keras/layers/Softmax?hl=en

扩展思考

- 用张量的角度看待这个世界，是一种数据思维的方式
  - 你认为有什么优点和缺点?
  - 存在什么样的问题呢?
- 请大家投稿一下!

扩展思考

- 请大家思考一下：
  - 神经网络的理论与实践是如何结合的？
- 请大家投稿一下：
  - 你认为神经网络的模型有什么优点和缺点？

谢谢指正！

扩展思考

- 如果你自己设计一个神经网络，你认为其中最大的技术挑战是什么呢?
  - 网络结构、权重的表示
  - 自动微分运算
  - 权重更新
- 请大家投稿一下!