循环神经网络 RNN与LSTM

常用神经网络模型

- 卷积神经网络：CNN（Convolutional Neural Network），包括卷积层和池化层，常用和图像相关的操作，如图像分类等
- 循环神经网络：RNN(Recurrent Neural Network)，这种网络的内部状态可以展示动态时序行为，神经网络是一种节点定向连接成环的人工神经网络，常用和自然语言相关的操作，如在线翻译，聊天机器人
- LSTM(Long Short-Term Memory)是长短期记忆网络，是一种时间递归神经网络，适合于处理和预测时间序列中间隔和延迟相对较长的重要事件，是循环神经网络的变形，在RNN的基础上增加了时间记忆的概念

在NLP中我们主要是用RNN和LSTM

BP神经网络

分为三层，输入层，隐藏层，输出层

- 输入层：数据模型的输入，也就是我们要传入到模型中的数据，例如：文本、语音、图像等，比如实现聊天机器人要传入问答句
- 隐藏层：用于处理数据，并将处理的结果传递给输出层
- 输出层：经过隐藏层计算之后所输出的模型内容，例如我们用来判断一句话是正能量还是负能量，那么输出层就是输出一个相应的分类

  如输入“今天是个好日子‘，在隐藏层计算之后，通过输出层输出这句话是正能量还是负能量，正能量即今天就是一个好日子

BP神经网络处理过程

- 正向传播：即信息正向传播，向前传播，输入层——>隐含层——>输出层
  - 网络初始化：定义权重，数量信息等
  - 隐藏层的输出：权重求和，再利用聚合函数将权重输出
  - 输出层的输出：输出结果
- 误差计算：传播过程出现偏差，需要进行偏差计算
  - 得到的输出值和期望值有偏差，期望值-输出值
- 反向传播：即误差反向传播，得到误差后将误差往回传，知道误差有多少
  - 隐藏层到输出层：计算梯度和微分，更新权重
  - 输入层到隐藏层：计算误差后进行误差更新
- 偏置更新：知道误差后更新误差偏差
  - 隐藏层到输出层

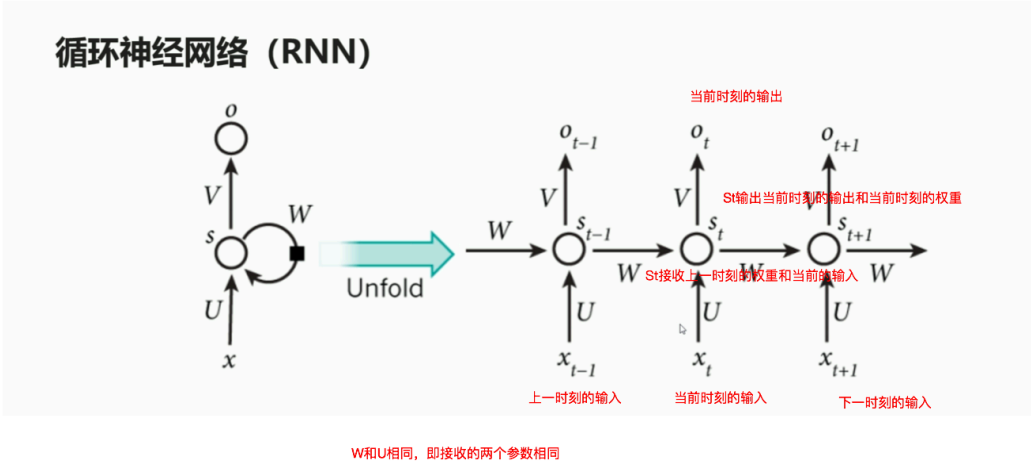
- 输入层到隐藏层

注意：BP神经网络

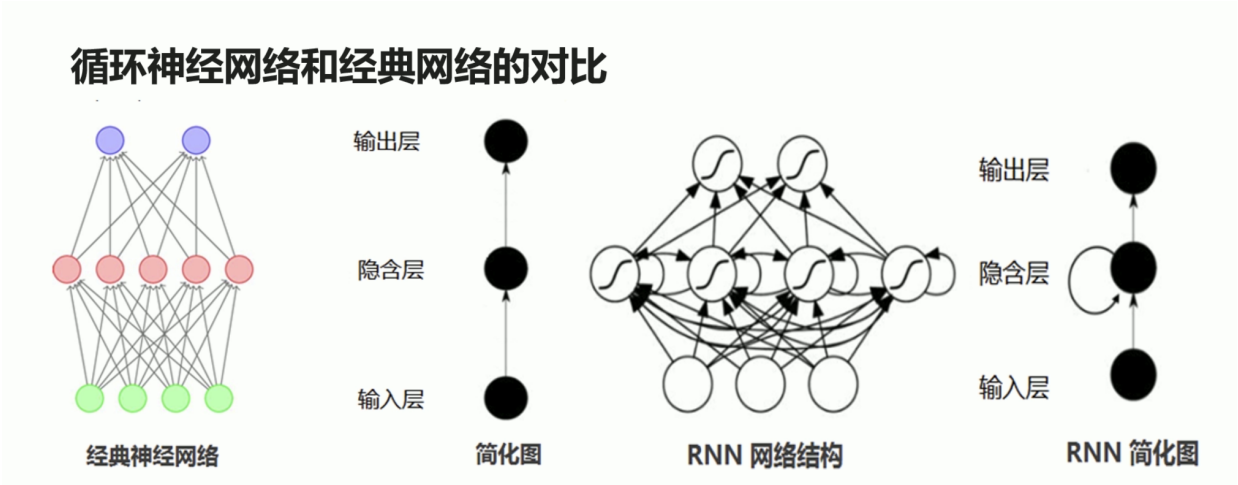
- 可以通过逐层信息传递到最后的输出
- 沿着一条直线计算，直到最后一层，求出计算结果
- 包含输入层、输出层、隐藏层，其目的是实现从输入到输出的映射
- 一般包含多层，并且层与层之间是全连接的，不存在同层和跨层连接，即第一层和第二层全连接，第二层和第三层全连接，不存在第一层和第三层的连接

循环神经网络RNN

有循环递归和递归参数



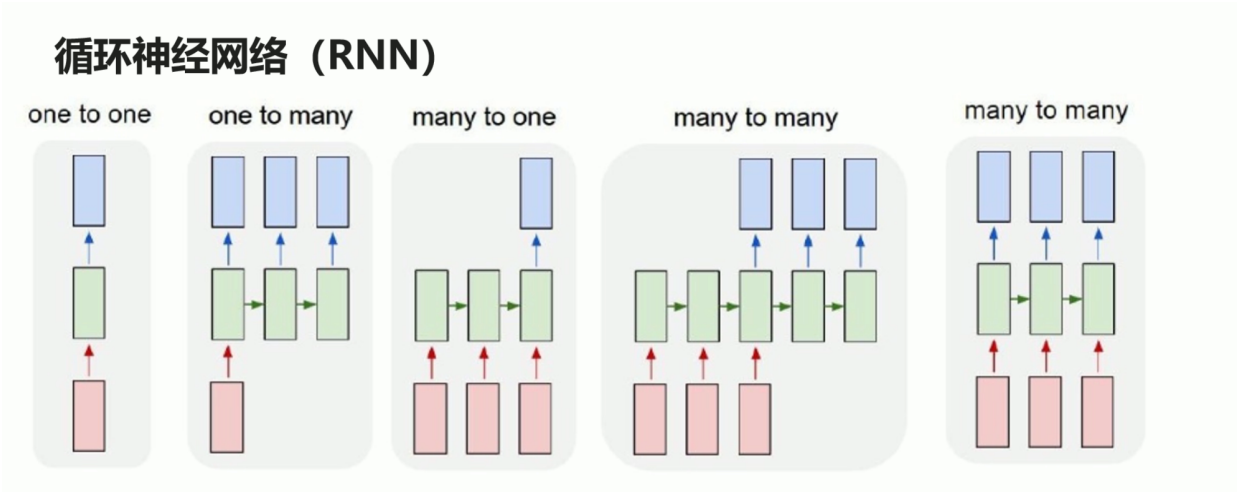
特性：记忆特性 、 接收两个参数 、 参数共享（确保每一步在做相同的事，w，u相同）



分类：

- one to one :一对一，一个维度输入一个维度输出
- one to many:一对多，如图片描述，音乐生成，输入音乐种类输出音乐序列
- many to one:多分类问题，多个类别样本对应样本输出
- many to many:输入输出不同维度，翻译

- many to many:输入输出相同维度，识别

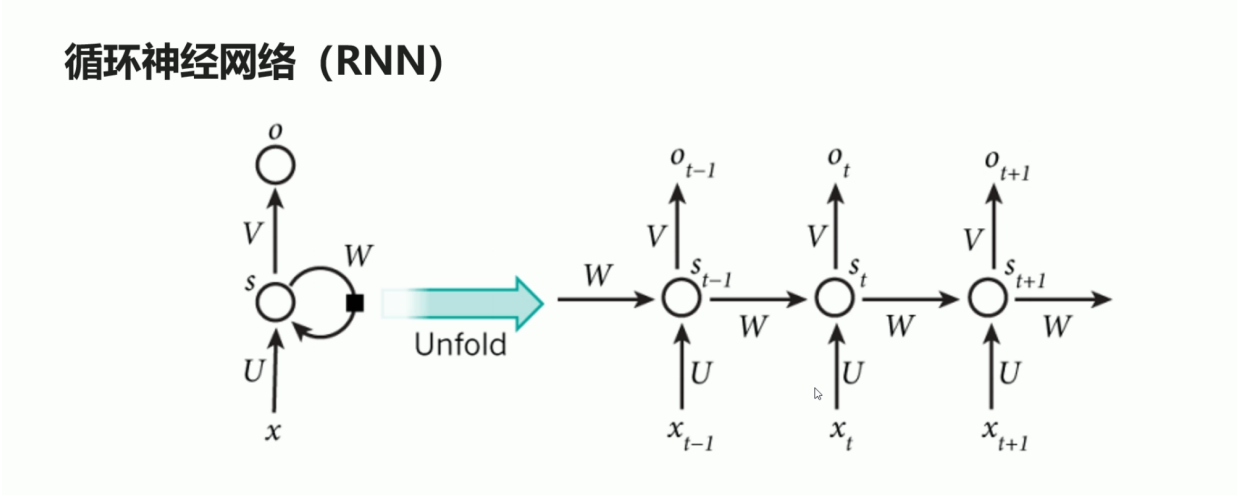


双向循环神经网络

- 从左到右和从右到左同时进行
- 从左到右参数共享
- 从右到左参数共享
- 两个方向参数不共享，参数独立
- 有四层：输入层，向前传播层，反向传播层，输出层
- 向前传播层和反向传播层是独立的

特点：

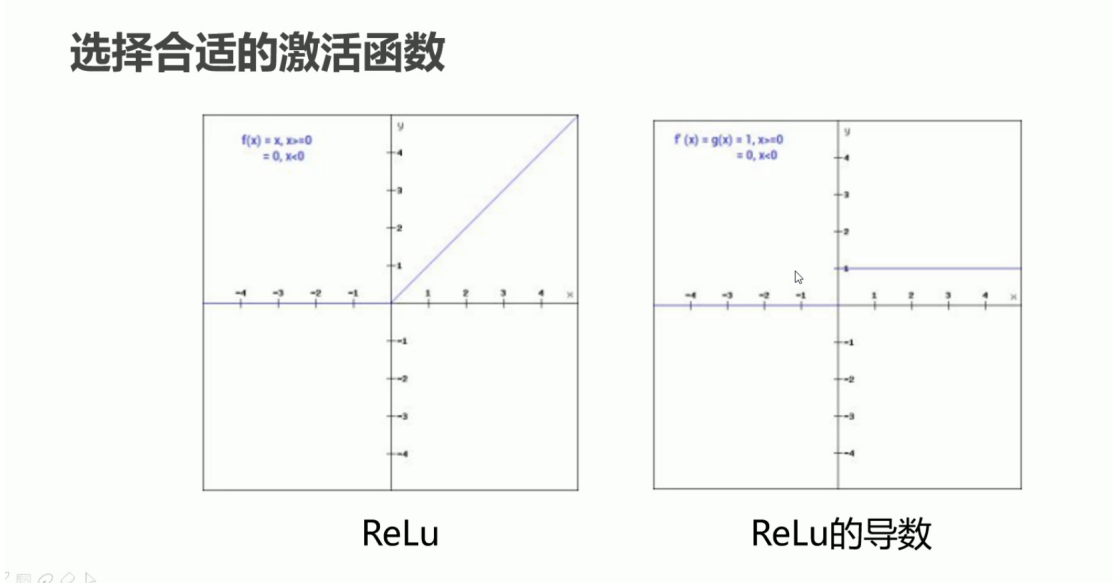
- 每个时刻有2个隐藏层
- 一个从左到右，一个从右到左
- 向前和向后传播参数独立，互不干扰



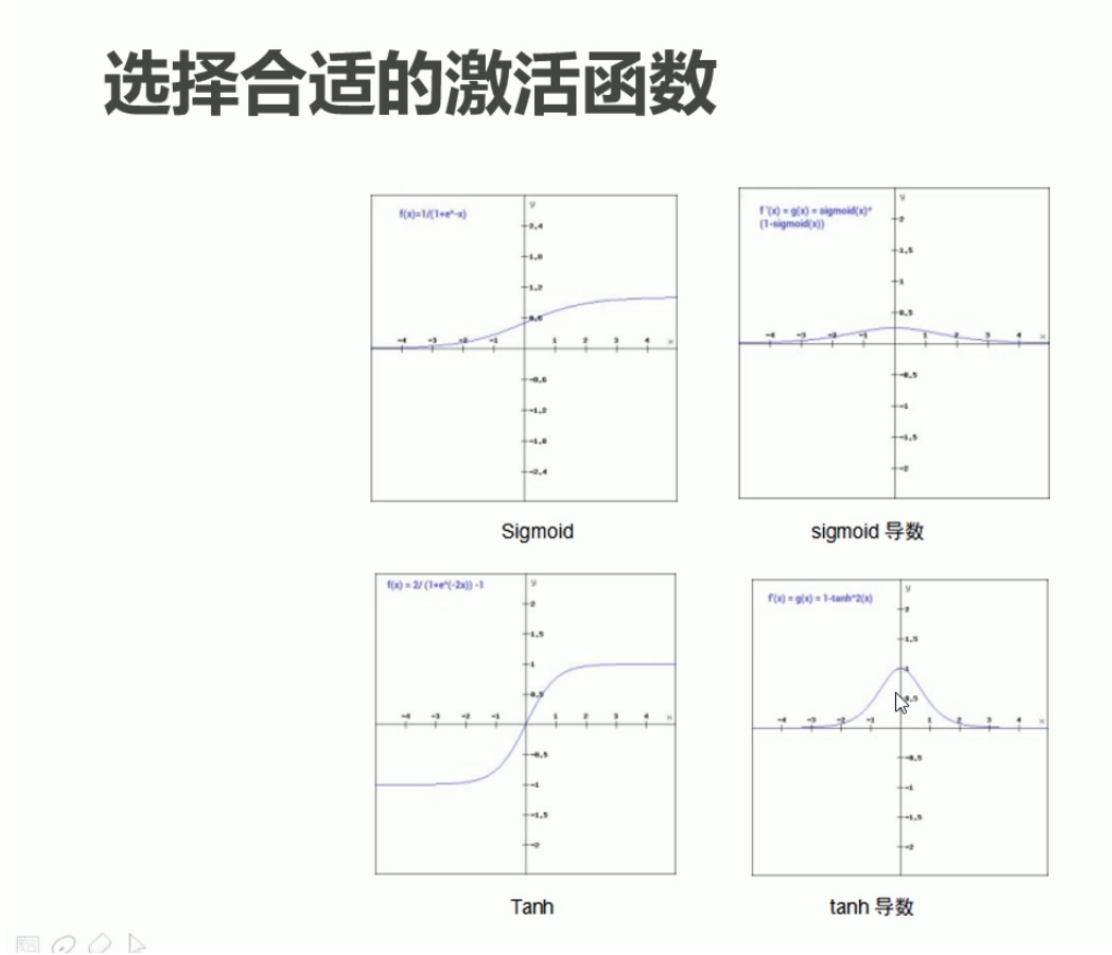
只有解决了梯度消失和梯度爆炸才能得到较准确的结果，w=1 的情况很少，想办法解决梯度消失和梯度爆炸

梯度消失和梯度爆炸的解决

- 选择合适的激活函数
  - 导数太小梯度消失，导数太大梯度爆炸
  - 这里是用ReLu函数，可以看出导数为1



  - 不建议使用sigmoid函数和Tanh函数，梯度太小，容易消失



- 选择合适的参数初始化方法
  - 不可以把所有参数都初始化为0，

选择合适的参数初始化方法

$$W^{[L]} = np.random.randn(shape^{[L]}) * 0.01$$

W^L 是第 L 层的权重参数， $shape$ 是第 L 层权重参数矩阵的形状。

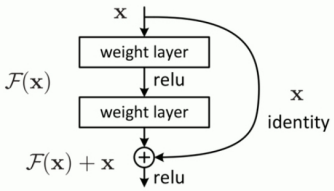
$$W^L = np.random.randn(shape^L) * np.sqrt(\frac{1}{n^{L-1}})$$

W^L 是第 L 层的权重参数， $shape$ 是第 L 层权重参数矩阵的形状， n^{L-1} 是 L-1 层的神经元数。

第一个乘以0.01容易导致梯度消散

- 使用权重参数正则化
- 使用BatchNormalization
  - 通过规范化操作将输出信号x规范化到均值为0、方差为1保证网络的稳定性
  - 可以加大神经网络训练的速度
  - 提高训练的稳定性
  - 缓解梯度爆炸和梯度消散的问题
- 使用残差结构

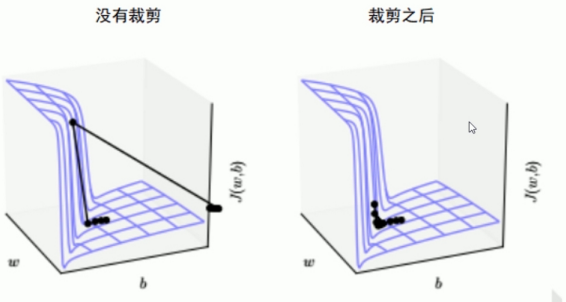
使用残差结构



- 极大地提高了神经网络的深度
- 很大程度上解决了梯度消散的问题
- 允许我们可以训练很深层的神经网络
- 残差结构可以看作解决梯度消散问题的最有效的、最重要的方法

- 使用梯度裁剪

使用梯度裁剪



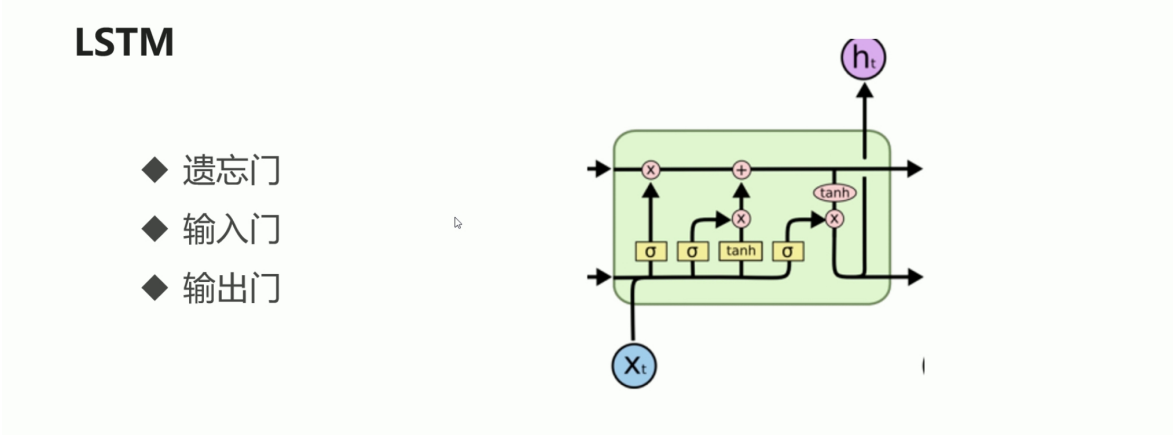
$$if \quad \|g\| > v$$

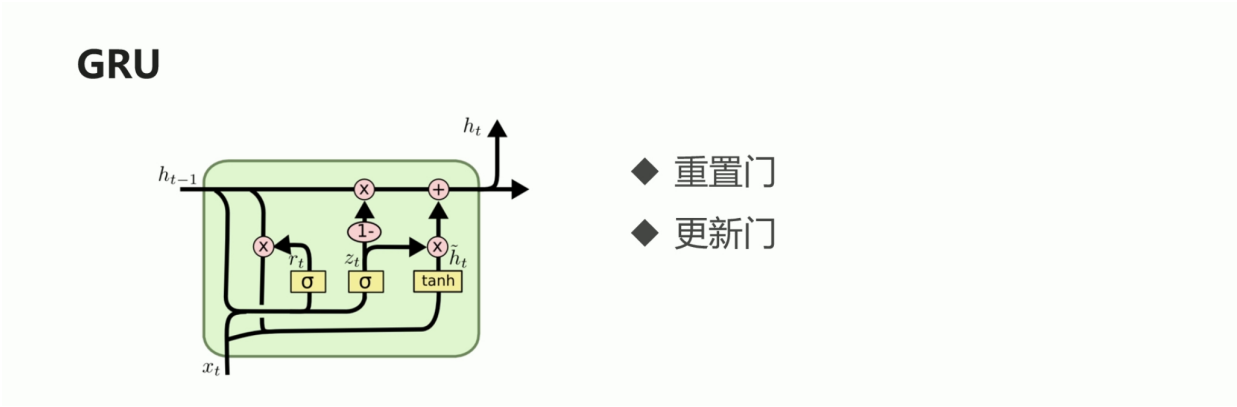
$$g \leftarrow \frac{gv}{\|g\|}$$

其中v是梯度范数的上界，g用来更新参数的梯度。

LSTM

- 遗忘门，上一时刻输入
- 输入门，当前时刻输出
- 输出门





学习：https://blog.csdn.net/qq_28743951/article/details/78974058