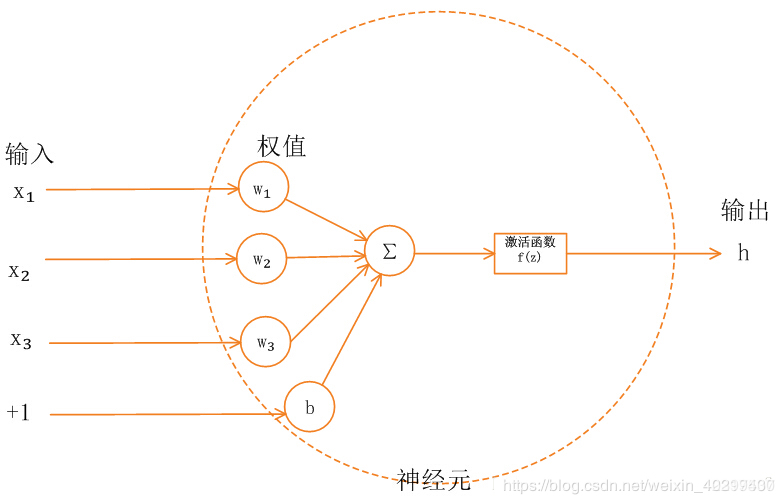
* 前馈神经网络、网络层数、输入层、隐藏层、输出层、隐藏单元、激活函数的概念。
* 感知机相关；定义简单的几层网络（激活函数sigmoid），递归使用链式法则来实现反向传播。
* 激活函数的种类以及各自的提出背景、优缺点。（和线性模型对比，线性模型的局限性，去线性化）
* 深度学习中的正则化（参数范数惩罚：L1正则化、L2正则化；数据集增强；噪声添加；early stop；**Dropout层**）、正则化的介绍。
* 深度模型中的优化：参数初始化策略；自适应学习率算法（梯度下降、AdaGrad、RMSProp、Adam；优化算法的选择）；**batch norm层**（提出背景、解决什么问题、层在训练和测试阶段的计算公式）；layer norm层。
* FastText的原理。
* 利用FastText模型进行文本分类。

前馈神经网络是一种最简单的神经网络，各神经元分层排列。每个神经元只与前一层的神经元相连。接收前一层的输出，并输出给下一层．各层间没有反馈。

前馈神经网络(feedforward neural network)是种比较简单的神经网络，只有输入层input layer (黄)、隐藏层hidden layer (绿)、输出层output layer (红)



如上图，神经网络神经元中，上层节点的输入值从通过加权求和后，到输出下层节点之前，还被作用了一个函数，这个函数就是激活函数(activation function)，作用是提供网络的非线性建模能力。

每个神经元中有三个组成部分：权重(weight)矩阵W，偏置(bias)向量b，以及激活函数(activation function) g

激活函数

所谓激活函数（Activation Function），就是在人工神经网络的神经元上运行的函数，负责将神经元的输入映射到输出端。

激活函数的主要作用是提供网络的非线性建模能力。如果没有激活函数，那么该网络仅能够表达线性映射，此时即便有再多的隐藏层，其整个网络跟单层神经网络也是等价的。因此也可以认为，只有加入了激活函数之后，深度神经网络才具备了分层的非线性映射学习能力。 那么激活函数应该具有什么样的性质呢？

可微性： 当优化方法是基于梯度的时候，这个性质是必须的。

单调性： 当激活函数是单调的时候，单层网络能够保证是凸函数。

输出值的范围： 当激活函数输出值是 有限 的时候，基于梯度的优化方法会更加 稳定，因为特征的表示受有限权值的影响更显著;当激活函数的输出是 无限 的时候，模型的训练会更加高效，不过在这种情况小，一般需要更小的learning rate。

正则化的形式很简单，是在目标函数后额外附加一项，使其影响目标函数最优点的选取。这种方法叫做正则化方法。

正则化可以解决过拟合问题。

从贝叶斯论者的角度来说，我们知道，深度学习使用神经网络将样本空间映射到特征的概率分布空间。当映射函数可能存在的空间太大时，我们通过学习方法得到的映射函数可能并不满足现实世界的一些规则，所以在泛化，即推广到未训练过的现实世界真实存在的数据集（验证集）时，得到的结果会变差，这叫做“Variance”大，说明模型过拟合了。

正则化的作用

正则化可以解决过拟合问题。

深度模型中的优化

随机梯度下降 (SGD)：基本训练方法，超参数：学习率

动量 (momentum)：动量算法主要有两个作用：

1、解决随机梯度下降算法梯度的高方差问题，使摆动不至于太剧烈：增加动量项，可以近似认为增加了梯度的采样的样本数(最近时间的梯度会有比较大的权重)，根据σ/n−−√σ/n可知,方差减小

2、加大了步长，提高了收敛速度：每一次梯度都包含正确的梯度方向和方差引起的摆动，增加动量，相当于将之前多个梯度叠加，增加了共同方向(期望梯度方向)，因此，相等与增大了步长。

自适应学习率算法

学习率是神经网络中难以设置的超参数之一，对模型的性能有显著的影响，因此需要自适应的学习率算法，更好的学习率参数一方面可以加速收敛，一方面可以减小训练误差，本文主要介绍：AdaGrad 、RMSProp、Adam

AdaGrad :Wt+1=Wt−ηt+1√1t+1∑ti=0(gi)2√dWt=Wt−η∑ti=0(gi)2√dWtWt+1=Wt−ηt+11t+1∑i=0t(gi)2dWt=Wt−η∑i=0t(gi)2dWt（求和和开根号都是元素级别的）

AdaGrad 自适应的为每个参数提供自适应的权重，如果某个梯度历史积累具有较大的梯度（调整已经比较大），那么给一个小的学习率（除以一个更大的值），相反，如果某个梯度历史积累相对较小（调整不是很大），则给比较大的学习率，净效果是在较为平缓的方向取得比较大的进步,有助于逃离高原平台。但是从训练开始积累平方和可能会导致有效学习率过早过过量的减小。

RMSProp

RMSProp 修改了AdaGrad梯度累加的方式，采用指数加权移动平均（丢弃时间过久的历史），多了一个超参数ββ,用以控制移动平均的长度范围。

**Adam**

Adam可以认为是将动量算法和RMSProp结合起来使用,将动量替换RMSProp中的梯度，同时Adam对动量和指数加权均分都做了偏差修正，因此更加鲁棒，超参数*β*1β1 (momentum) *β*2β2 (RMSProp)。

优化策略与元算法

**Batch norm(批标准化)**

Batch norm（批标准化）可以有效解决covariate shift 的问题，并有轻微的正则化的效果，可以让大型神经网络训练速度加快很多倍，同时收敛后的分类准确率也可以得到大幅提高。

batch\_norm还有几个细节需要说明：

1、每层训练时都要做标准化处理，因此不需要在每层假设偏置项b,因此，每一层需要学习的参数包括W,ββ,γγ

2、batch\_norm 通常要采用mini\_batch 方法进行训练，因此，利用每一个batch进行训练时，计算的均值和方差，都是该mini\_batch 的均值和方差，因此与全部数据的均值和偏差有差别，正式这些差别，使得，batch\_norm 有轻微的规则化的作用。

3、测试时，需要对每一个样本进行逐一处理 单个数据的均值和方差没有意义，需要单独计算均值和方差，理论上可以在整个数据上进行统计均值和方差，实际操作中，在训练时使用指数加权平均算法得到(只需要记住上一次加权，内存要求非常小)，这个值就是该隐藏层z均值、方差的估计。

利用Fast-Text进行中文文本分类

参考https://blog.csdn.net/weixin\_43977375/article/details/90200837