人工智能的概念：机器模拟人的意识和思维。

图灵：人工智能之父

人工智能助理：Assistant Cortana Siri Alexa 小爱同学

卷积神经网络：CNN

循环神经网络：RNN

机器学习：机器学习是一种统计学方法，计算机利用已有数据，得出某种模型。在利用此模型预测结果。

机器学习三要素：数据，算法，算力

深度学习（深层神经网络）：是机器学习的一种实现方法

机器学习的应用：对连续数据的预测；对离散数据进行分类

应用领域：计算机视觉，机器学习，语音识别，自然语言处理

ubuntu

anaconda安装

安装tensorflow:

pip install tensorflow

pip install tensorflow-gpu

更新Tensorflow:

pip uninstall tensorflow

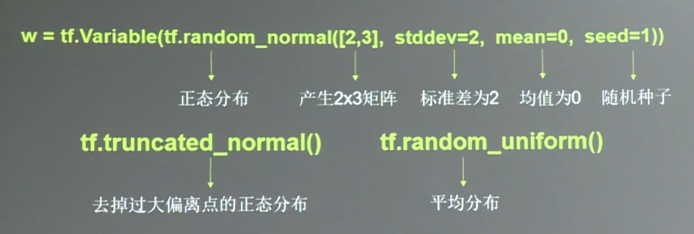
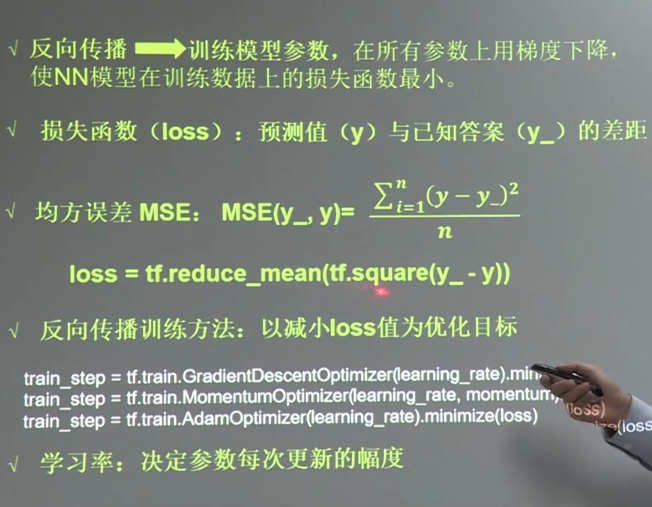
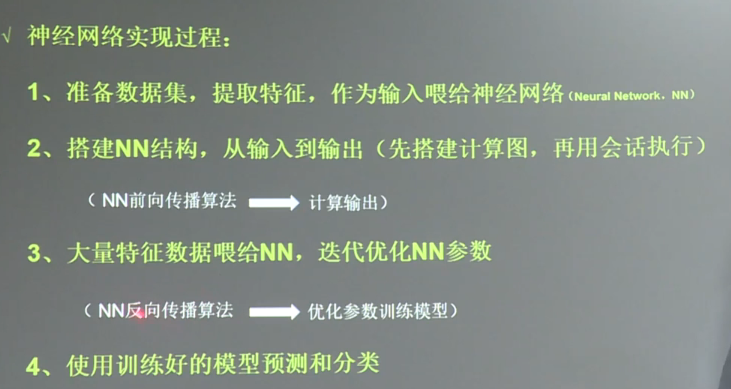
pip install tensorflow

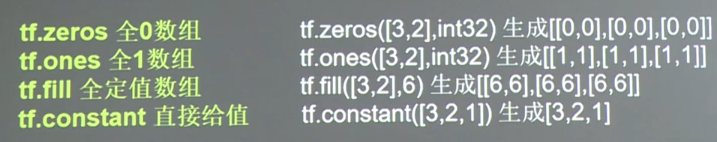
Tensorflow基本概念：

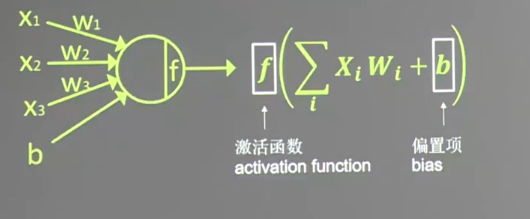
数据类型：tf.float32 tf .int32…

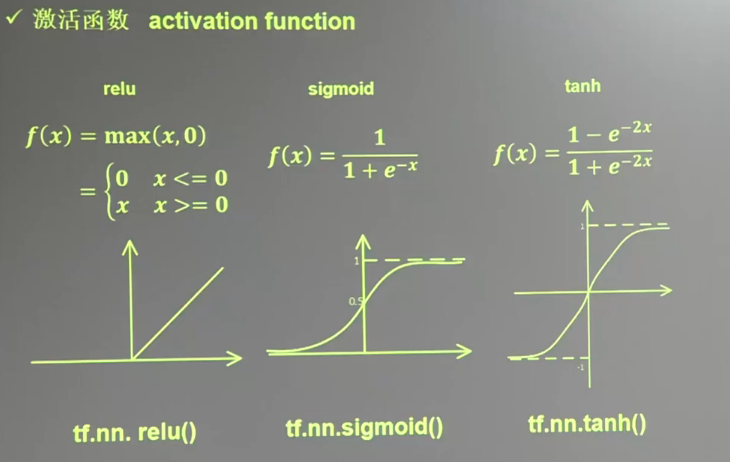
计算图：搭建神经网络的计算过程，只搭建不计算

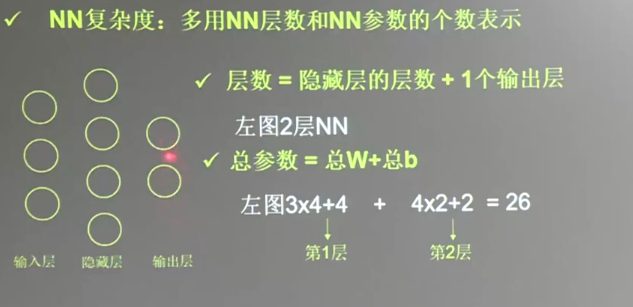
使用图（graphs）来表示计算任务

在被称之为会话（Session）的上下问（context）种执行图会话：执行计算图中的节点运算



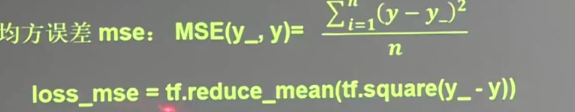
前向传播搭建网络结构，反向传播训练网络参数。

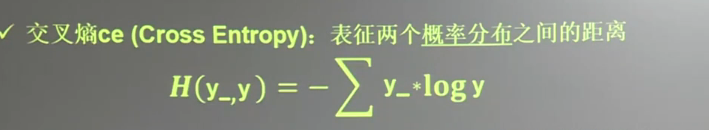
常用激活函数:tf.nn.relu();tf.nn.sigmoid();tf.nn.tanh();



损失函数：预测值与已知答案的差距：

神经网络优化目标：损失函数最小：-》mse（均方误差）； 自定义；ce(交叉熵)







softmax函数：，它将多个神经元的输出，映射到（0,1）区间内，可以看成概率来理解，从而来进行多分类！

学习率learning\_rate:每次参数更新的幅度

指数衰减学习率：指数衰减学习率是先使用较大的学习率来快速得到一个较优的解，然后随着迭代的继续,逐步减小学习率，使得模型在训练后期更加稳定。在训练神经网络时，需要设置学习率（learning rate）控制参数的更新速度，学习速率设置过小，会极大降低收敛速度，增加训练时间；学习率太大，可能导致参数在最优解两侧来回振荡。tf.train.exponential\_decay(

|  |
| --- |
| tf.train.exponential\_decay(  learning\_rate,  global\_step,  decay\_steps,  decay\_rate,  staircase=False, # 默认为False  name = None ) |

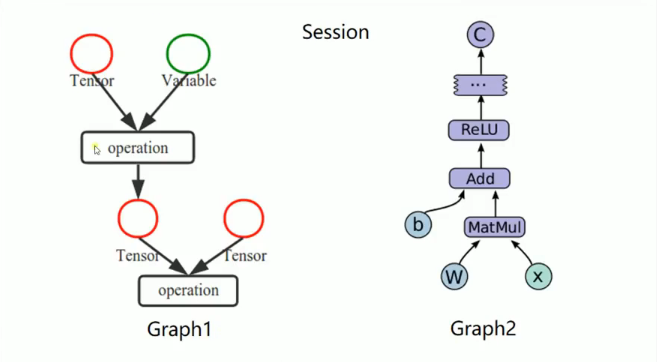
|  |
| --- |
| # 初始学习率 learning\_rate = 0.1 # 衰减系数 decay\_rate = 0.9 # decay\_steps控制衰减速度 # 如果decay\_steps大一些,(global\_step / decay\_steps)就会增长缓慢一些 # 从而指数衰减学习率decayed\_learning\_rate就会衰减得慢一些 # 否则学习率很快就会衰减为趋近于0 decay\_steps = 50 # 迭代轮数 global\_steps = 3000 |

learning\_rate是初始化的学习率，decayed\_learning\_rate 是随着global\_step 的递增而衰减。显然，当 global\_step为初值0时， 有下面等式：decayed\_learning\_rate=learning\_rate

decay\_steps用来控制衰减速度，如果decay\_steps 大一些, 就会增长缓慢一些。从而指数衰减学习率 decayed\_learnningrate就会衰减得慢一否则学习率很快就会衰减为趋近于0。

使用tensor表示数据

通过变量（Variable）维护状态

使用feed和fetch可以为任务的操作赋值或者从中获取数据

Tensorflow的使用：

创建启动视图：

|  |
| --- |
| import tensorflow as tf #创建一个常量op m1=tf.constant([[3,3]]) #创建一个常量op m2=tf.constant([[2],[3]]) #矩阵乘法op，把m1和m2传入 product=tf.matmul(m1,m2) print(product) #定义一个绘画，启动默认的图 sess=tf.Session() #调用sess的run方法执行矩阵乘法op #run(product)出发了图中的3个op result=sess.run(product) print(result) sess.close() with tf.Session() as sess:  result=sess.run(product)  print(result) |

变量的使用1：

|  |
| --- |
| import tensorflow as tf  #创建一个变量初始化为0 state=tf.Variable(0,name="counter") new\_value=tf.add(state,1)  update=tf.assign(state,new\_value)  init=tf.global\_variables\_initializer()  with tf.Session() as sess:  sess.run(init);  print(sess.run(state))  for \_ in range(5):  sess.run(update)  print(sess.run(state)) |

变量的使用2：

|  |
| --- |
| import tensorflow as tf X=tf.Variable([1,2]) a=tf.constant([3,3]) #增加一个减法op sub=tf.subtract(X,a) #增加一个加法op add=tf.add(X,sub) init=tf.global\_variables\_initializer() with tf.Session() as sess:  sess.run(init)  print(sess.run(sub))  print(sess.run(add)) |

Fetch And Feed:

|  |
| --- |
| import tensorflow as tf #Fetch 运行多个op [op1,op2] input1=tf.constant(3.0) input2=tf.constant(2.0) input3=tf.constant(5.0) add=tf.add(input1,input2) mul=tf.multiply(input3,add)  #Feed的概念 #定义一个占位符 input4=tf.placeholder(tf.float32) input5=tf.placeholder(tf.float32) output=tf.multiply(input4,input5) with tf.Session() as sess:  #运行多个op  result=sess.run([mul,add])  print(result)  #运行时传入值 以字典的形式传入  print(sess.run(output,feed\_dict={input4:[5.0],input5:[6.0]})) |

Tensorflow简单案例实现：

|  |
| --- |
| import tensorflow as tf import numpy as np #使用numpy生成100个随机点 x\_data=np.random.rand(100) y\_data=x\_data\*0.1+0.2 #构造一个线性模型： b=tf.Variable(0.) k=tf.Variable(0.) y=k\*x\_data+b  #二次代价函数 #tf.reduce\_mean 平均值 tf.square平方值 误差 loss=tf.reduce\_mean(tf.square(y\_data-y)) #定义一个梯度下降来进行训练的优化器 0.2的学习率 optimizer=tf.train.GradientDescentOptimizer(0.2) #定义最小化代价函数 train=optimizer.minimize(loss)  #初始化变量 init=tf.global\_variables\_initializer()  with tf.Session() as sess:  sess.run(init)  for \_ in range(201):  sess.run(train)  if \_%20==0:  print(\_,sess.run([k,b])) |

MNIST数据集：

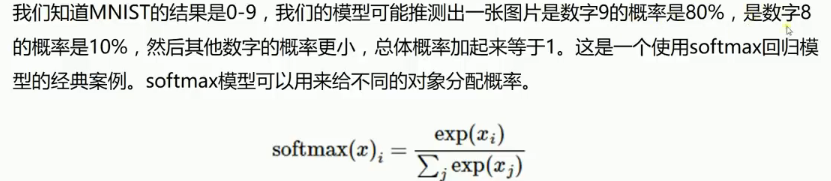
mnist.train.image=[60000,784]张量

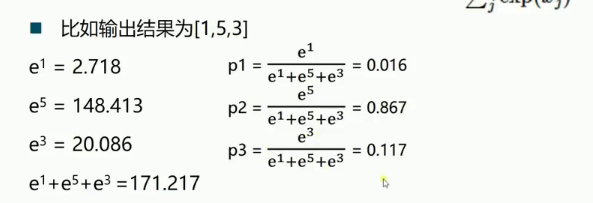
mnist.train.labels=[60000,10]

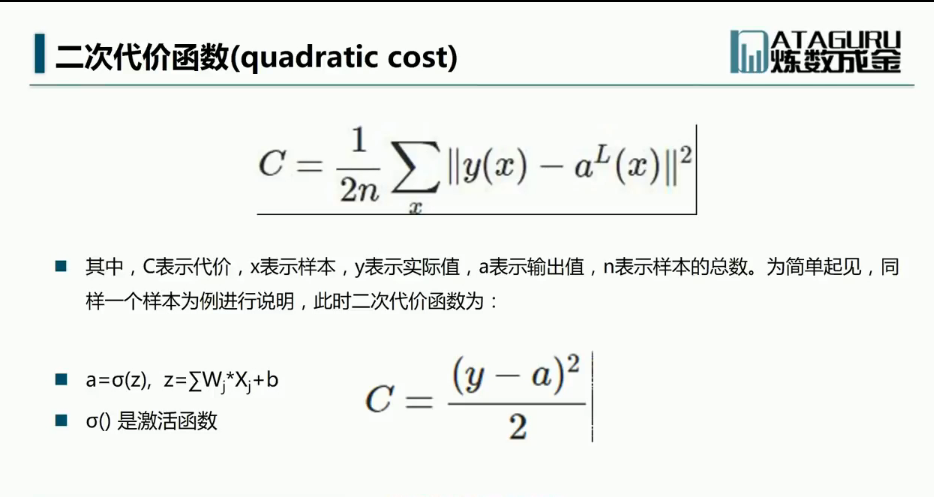
one-hot vectors 除某一位数字是1以外其他维度数字为0

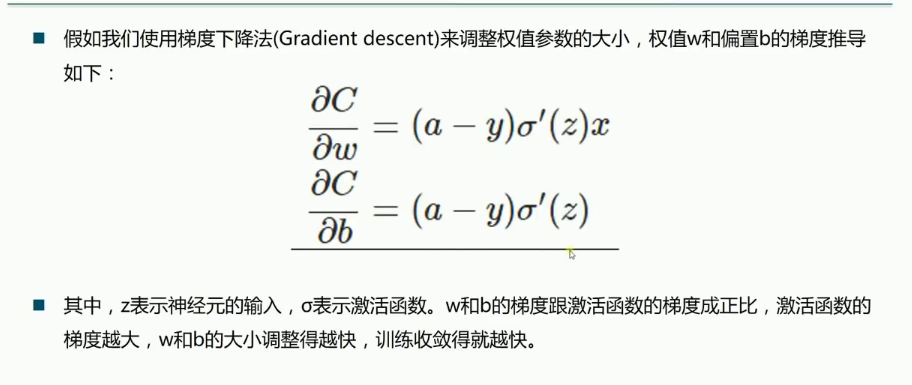
0 =[1,0,0,0,0,0,0,0,0,0]

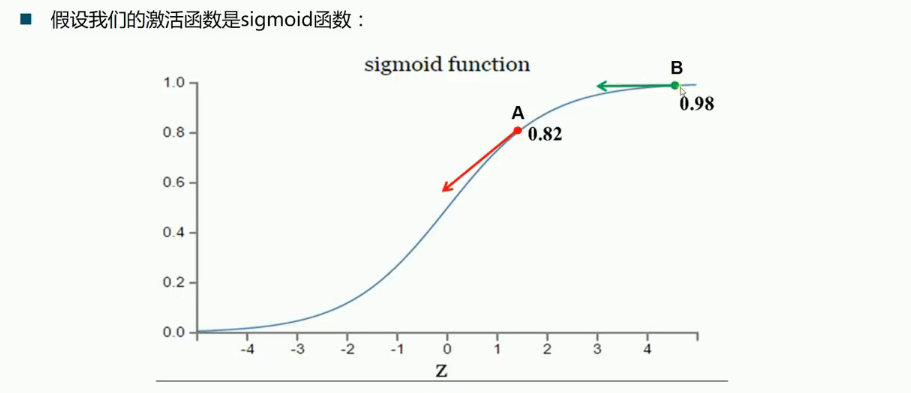
5 =[0,0,0,0,0,5,0,0,0,0]



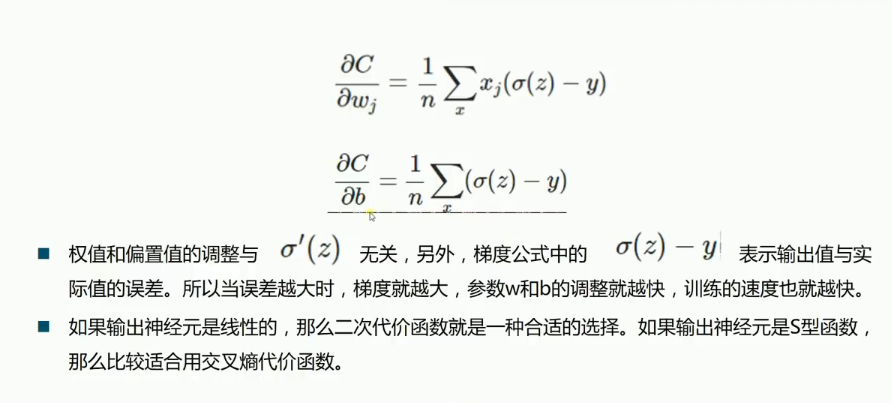
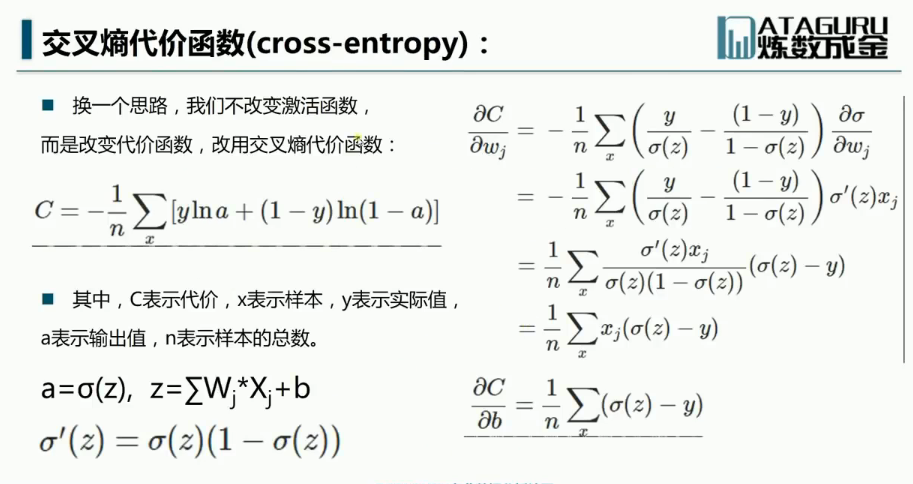


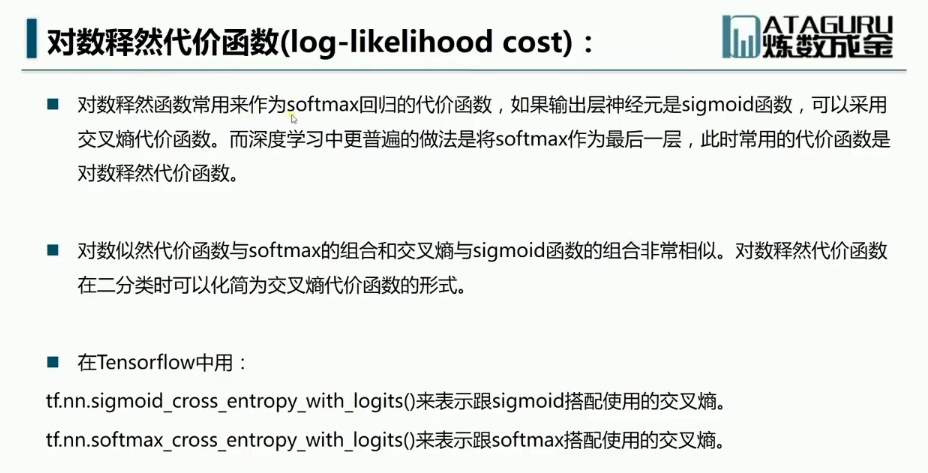


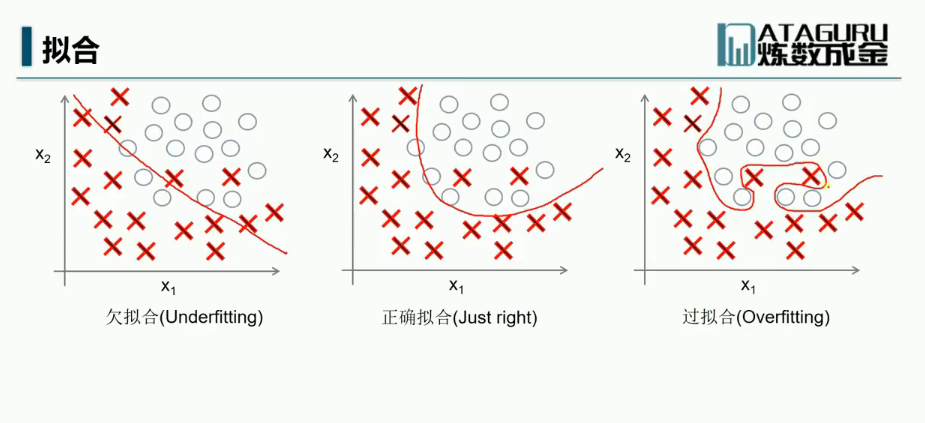
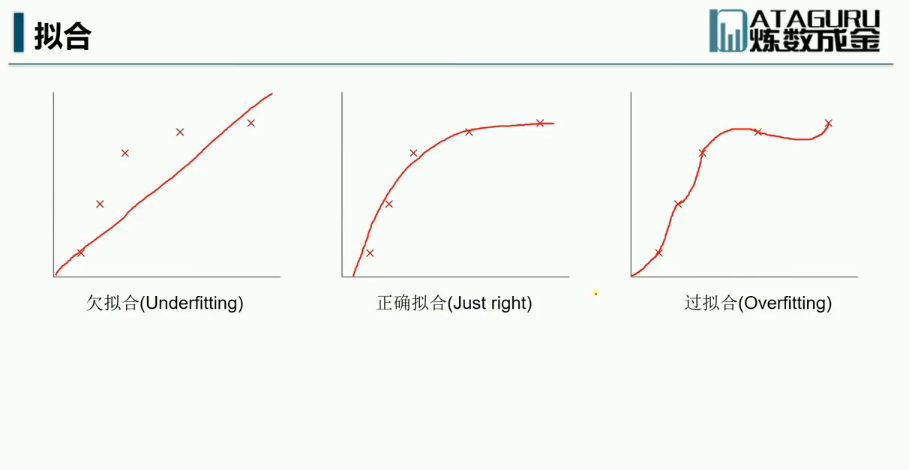


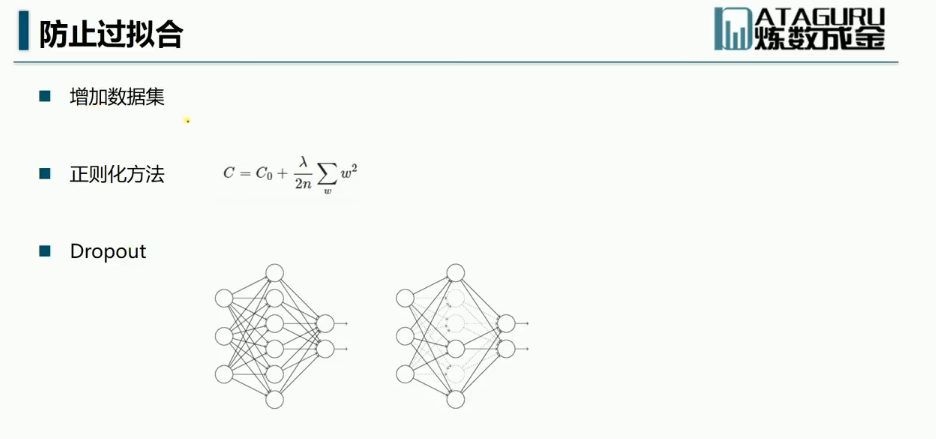


交叉熵：加快收敛速度









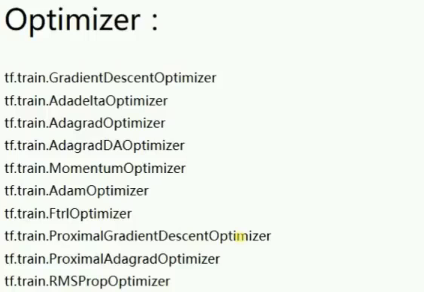
防止过拟合的方法;

1.增加数据集

2.正则化

3.Dropout

Tensorflow 优化器：



梯度下降：

标准梯度下降法;计算所有的样本汇总误差，然后根据总误差更新权值

随机梯度下降法：随机抽取一个样本计算误差，然后更新权值

批量梯度下降法：折中方法