Tensorflow 2.0

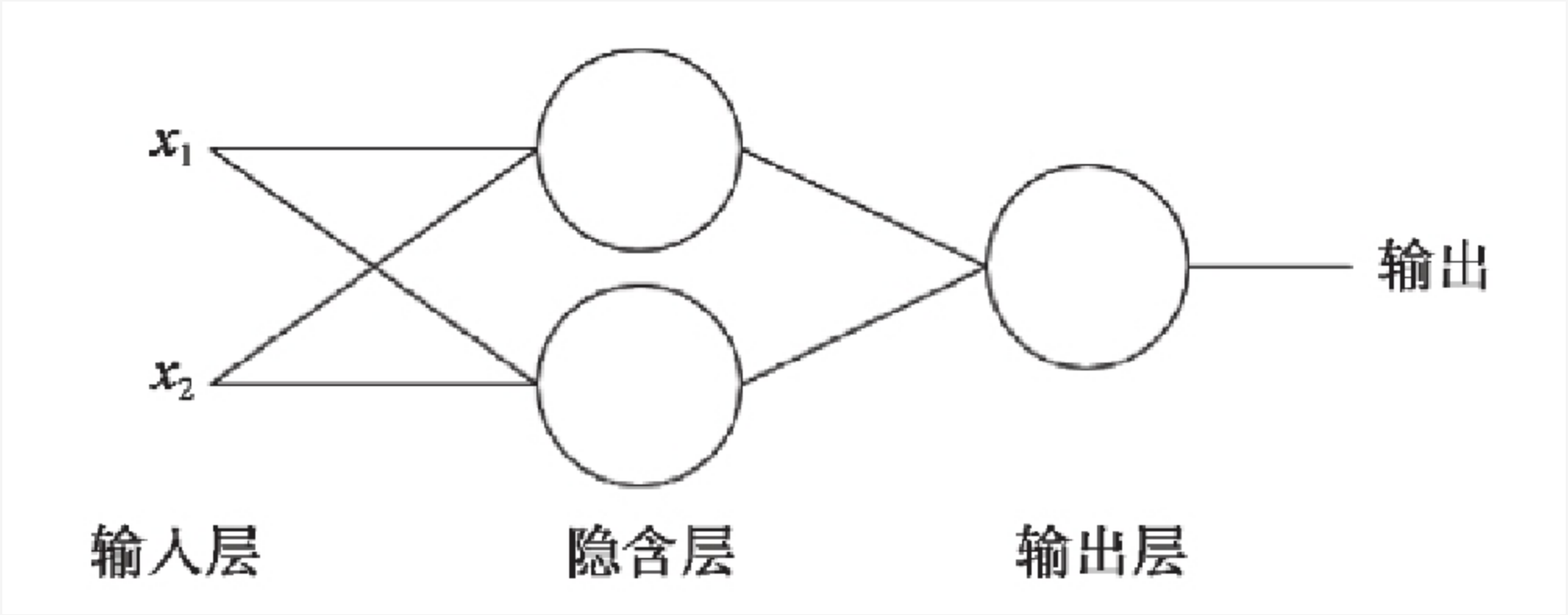
简明实战教程

讲师：日月光华

优化函数、学习速率与反向传播算法

讲师：日月光华　tf2.0 答疑群：738790253

多层感知器



梯度下降法

梯度下降法是一种致力于找到函数极值点的算法。

前面介绍过，所谓“学习”便是改进模型参数，以便通过大量训练步骤将损失最小化。有了这个概念，将梯度下降法应用于寻找损失函数的极值点便构成了依据输入数据的模型学习。

梯度下降法

梯度的输出是一个由若干偏导数构成的向量，它的每个分量对应于函数对输入向量的相应分量的偏导：

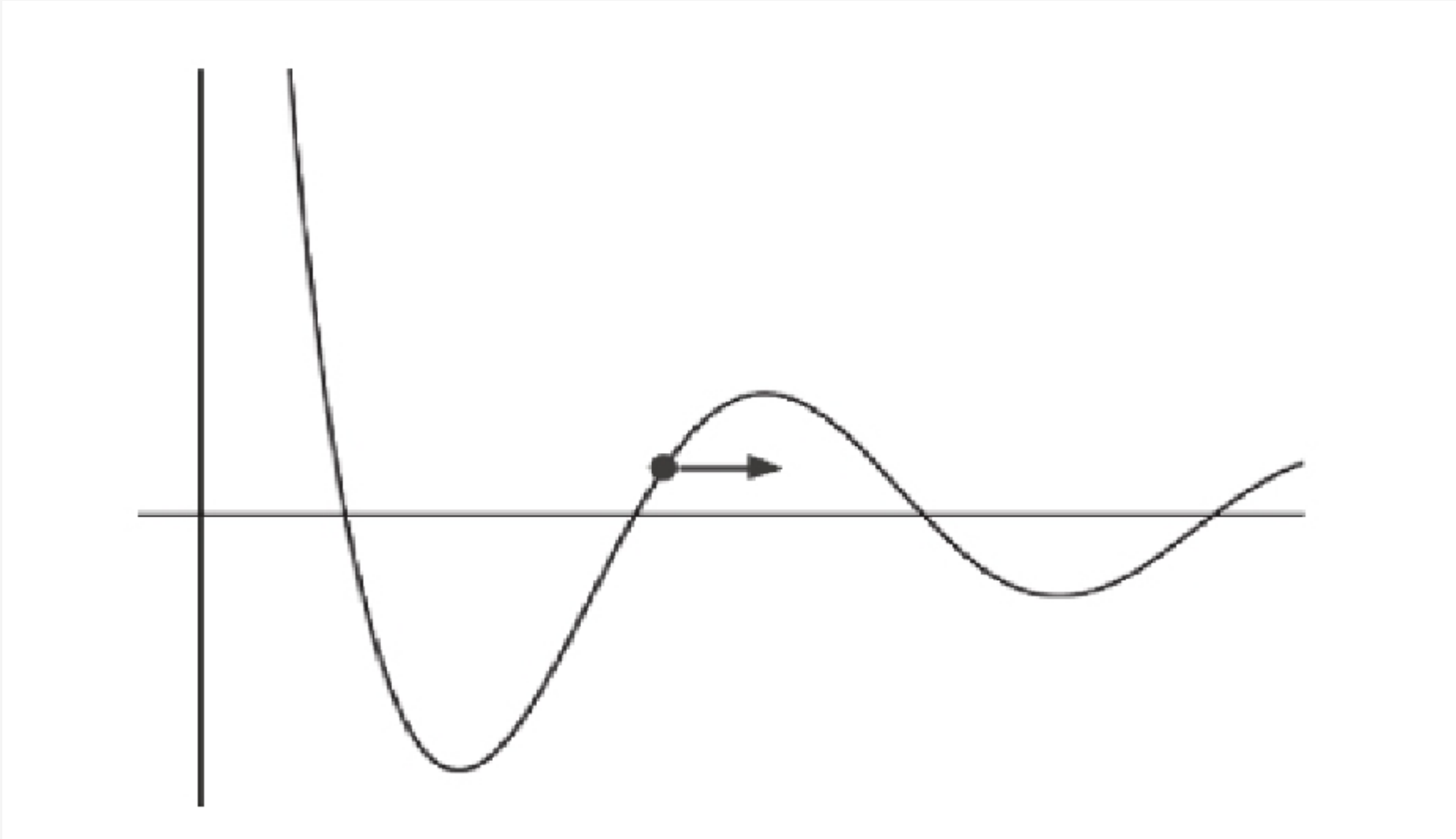
$$\nabla \equiv \left(\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \cdots, \frac{\partial}{\partial w_N} \right)^{\mathrm{T}}$$

梯度下降法

梯度的输出向量表明了在每个位置损失函数增长最快的方向，

可将它视为表示了在函数的每个位置向哪个方向移动函数值可以增长。

梯度下降法



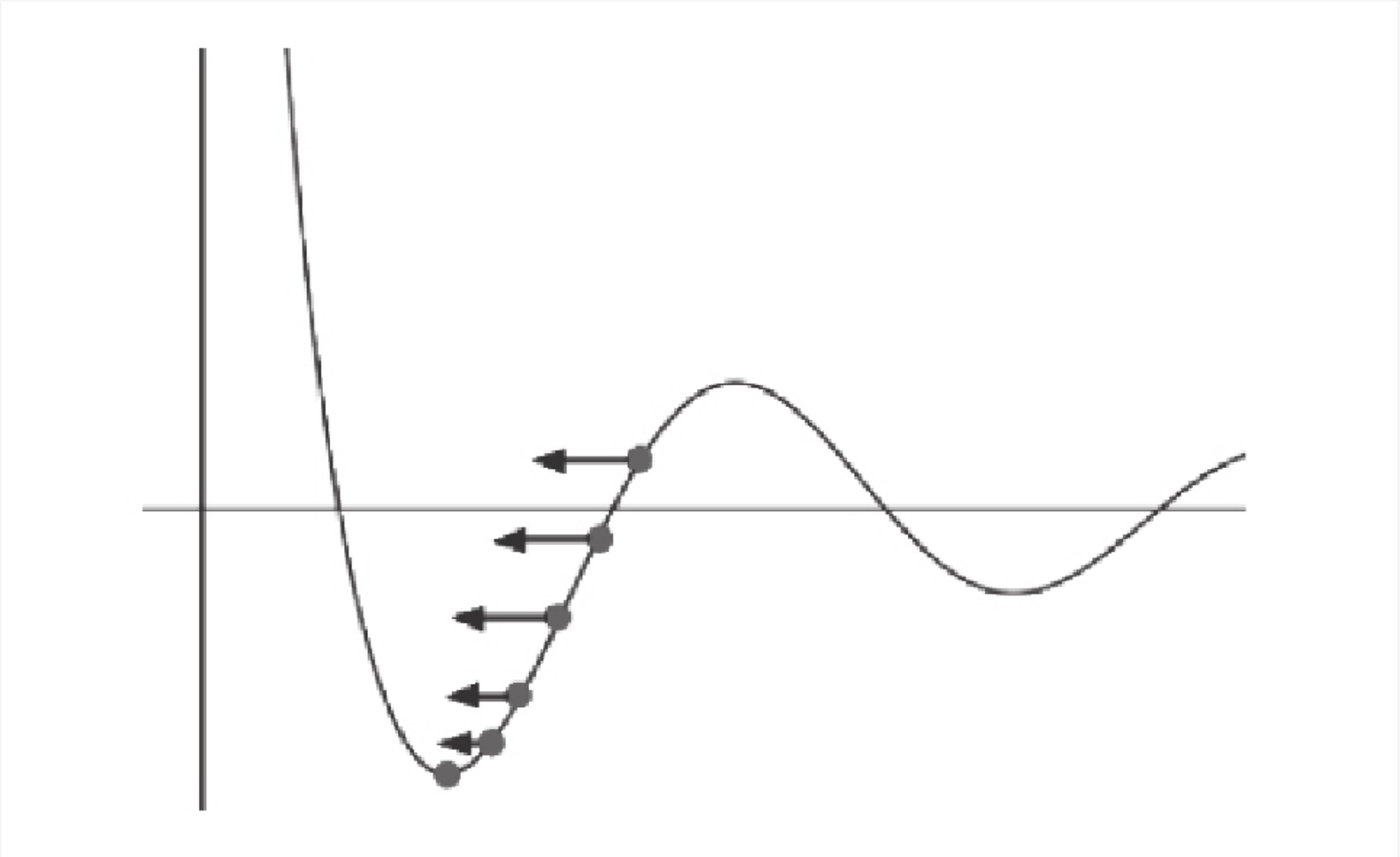
梯度下降法

曲线对应于损失函数。点表示权值的当前值，即现在所在的位置。

梯度用箭头表示，表明为了增加损失，需要向右移动。此外，箭头的长度概念化地表示了如果在对应的方向移动，

函数值能够增长多少。如果向着梯度的反方向移动，则损失函数的值会相应减小。

梯度下降法



梯度下降法

沿着损失函数减小的方向移动，并再次计算梯度值，并重复上述过程，直至梯度的模为0，将到达损失函数的极小值点。

这正是我们的目标

学习速率

梯度就是表明损失函数相对参数的变化率

对梯度进行缩放的参数被称为学习速率（learning rate）

学习速率

学习速率是一种超参数或对模型的一种手工可配置的设置

需要为它指定正确的值。如果学习速率太小，则找到损失函数极小值点时可能需要许多轮迭代；如果太大，则算法可能会“跳过”极小值点并且因周期性的“跳跃”而永远无法找到极小值点。

学习速率

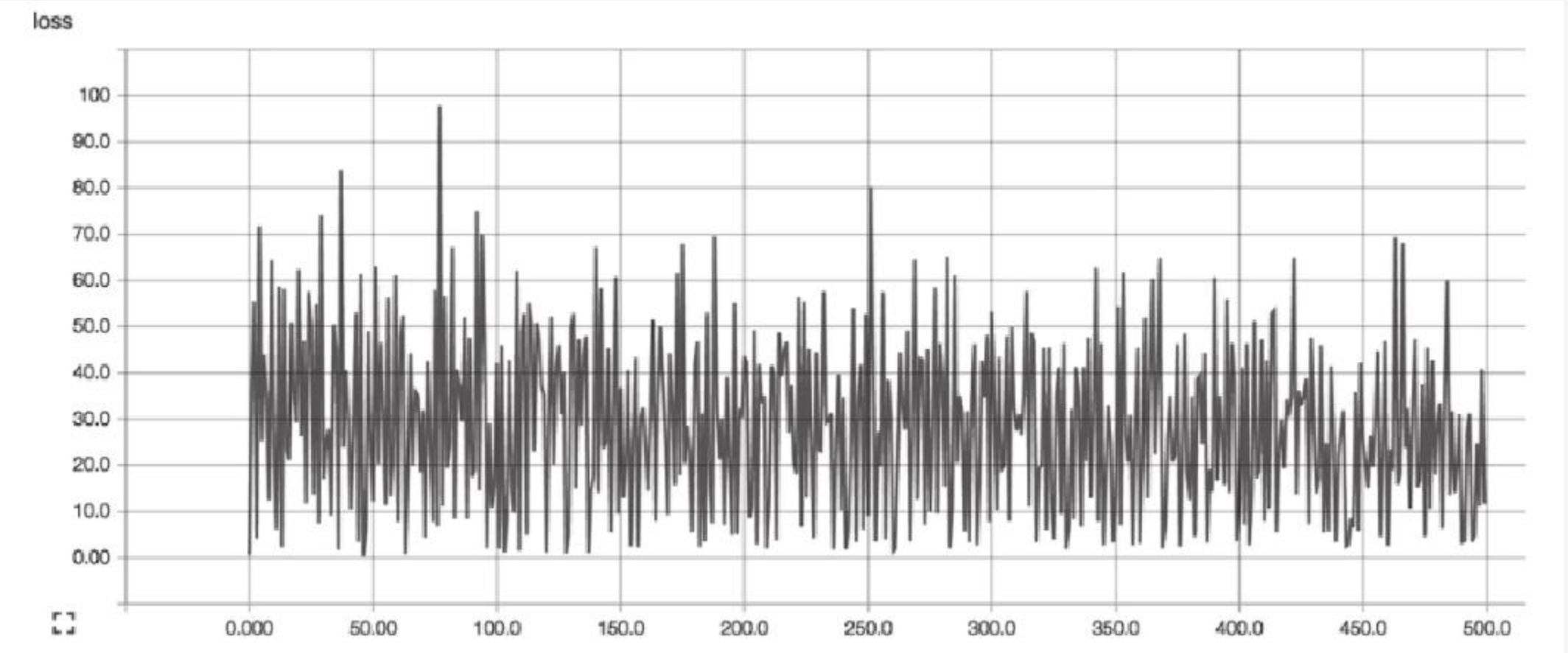
在具体实践中，可通过查看损失函数值随时间的变化曲线，来判断学习速率的选取是合适的。

学习速率

合适的学习速率，损失函数随时间下降，直到一个底部

不合适的学习速率，损失函数可能会发生震荡

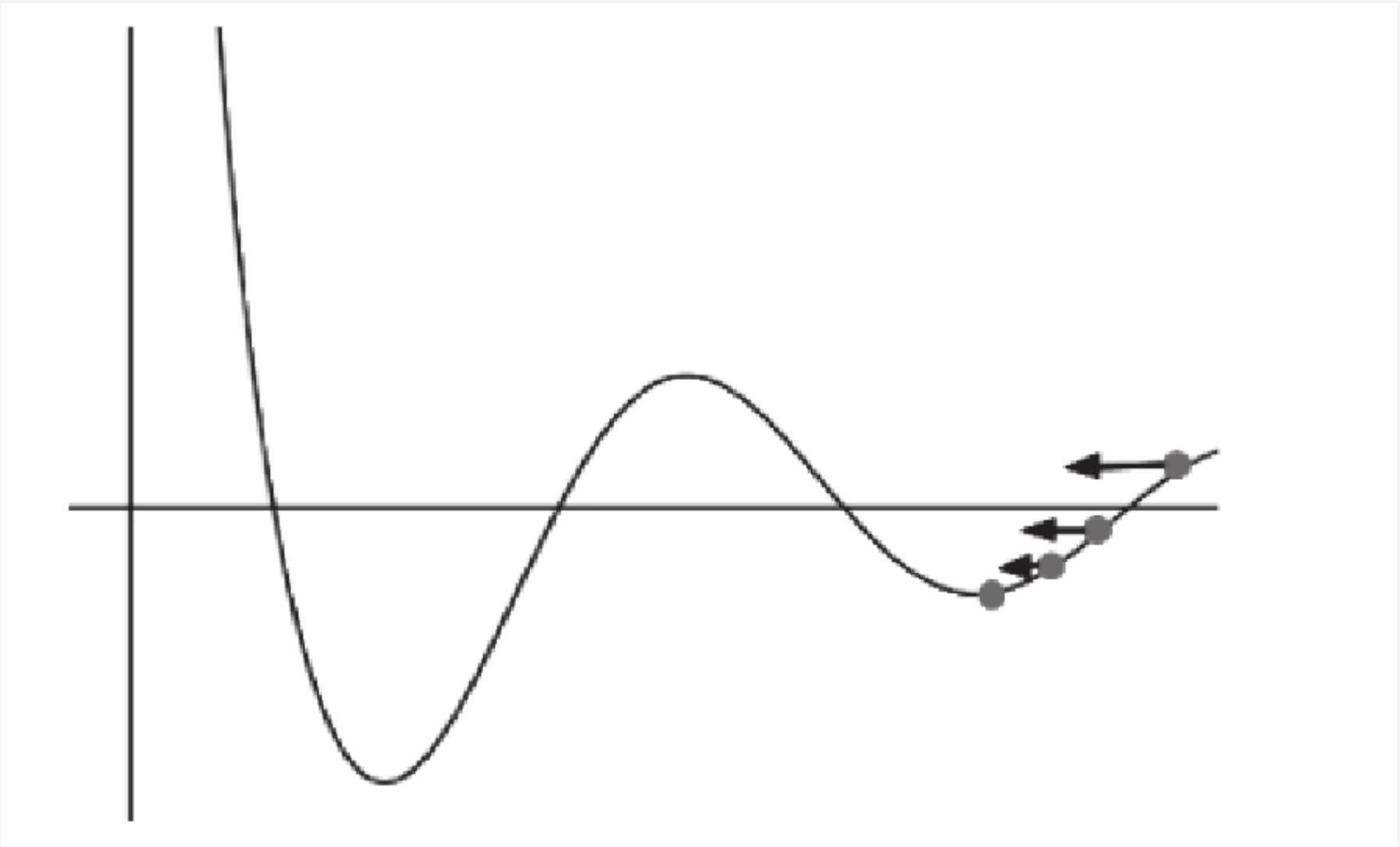
学习速率


loss
100
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.00
0.000
50.00
100.0
150.0
200.0
250.0
300.0
350.0
400.0
450.0
500.0

学习速率选取原则

在调整学习速率时，既需要使其足够小，保证不至于发生超调，也要保证它足够大，以使损失函数能够尽快下降，从而可通过较少次数的迭代更快地完成学习

局部极值点



局部极值点

可通过将权值随机初始化来改善局部极值的问题。

权值的初值使用随机值，可以增加从靠近全局最优点附近开始下降的机会。

反向传播算法

反向传播算法是一种高效计算数据流图中梯度的技术

每一层的导数都是后一层的导数与前一层输出之积，这正是链式法则的奇妙之处，误差反向传播算法利用的正是这一特点。

反向传播算法

前馈时，从输入开始，逐一计算每个隐含层的输出，直到输出层。

然后开始计算导数，并从输出层经各隐含层逐一反向传播。为了减少计算量，还需对所有已完成计算的元素进行复用。这便是反向传播算法名称的由来。

常见的优化函数

优化器 (optimizer) 是编译 模型的所需的两个参数之一。你可以先实例化一个优化器对象，然后将它传入model.compile()， 或者你可以通过名称来调用优化器。在后一种情况下，将使用优化器的默认参数。

常见的优化函数

SGD：随机梯度下降优化器

随机梯度下降优化器SGD和min-batch是同一个意思，抽取m个小批量（独立同分布）样本，通过计算他们平梯度均值。

SGD参数

lr: float >= 0. 学习率。

momentum: float >= 0. 参数，用于加速 SGD 在相关方向上前进，并抑制震荡。

decay: float >= 0. 每次参数更新后学习率衰减值。

nesterov: boolean. 是否使用 Nesterov 动量。

常见的优化函数

RMSprop：经验上，RMSProp被证明有效且实用的深度学习网络优化算法.

RMSProp增加了一个衰减系数来控制历史信息的获取多少，RMSProp会对学习率进行衰减。

常见的优化函数

建议使用优化器的默认参数 （除了学习率 lr，它可以被自由调节）

这个优化器通常是训练循环神经网络RNN的不错选择。

RMSprop

lr: float >= 0. 学习率。

rho: float >= 0. RMSProp梯度平方的移动均值的衰减率.

epsilon: float >= 0. 模糊因子. 若为 None, 默认为 K.epsilon()。

decay: float >= 0. 每次参数更新后学习率衰减值。

常见的优化函数

Adam优化器:

1. Adam算法可以看做是修正后的Momentum+RMSProp算法.

2. Adam通常被认为对超参数的选择相当鲁棒

3. 学习率建议为0.001

常见的优化函数

Adam 是一种可以替代传统随机梯度下降过程的一阶优化算法，它能基于训练数据迭代地更新神经网络权重。

Adam 通过计算梯度的一阶矩估计和二阶矩估计而为不同的参数设计独立的自适应性学习率

Adam常见参数

lr: float >= 0. 学习率。

beta_1: float, 0 < beta < 1. 通常接近于 1。

beta_2: float, 0 < beta < 1. 通常接近于 1。

decay: float >= 0. 每次参数更新后学习率衰减值。

谢谢大家

讲师：日月光华　　tf2.0 答疑群：738790253