# 2-4 梯度下降法及其衍生算法

王中雷

厦门大学王亚南经济研究院和经济学院, 2025

# 内容摘要

1. 梯度下降法

2. 衍生算法

3. 学习率衰减策略

# 回顾

## 步骤1．批量梯度下降法

- 随机初始化$\boldsymbol{\theta}^{(0)}$
- 基于当前模型参数$\boldsymbol{\theta}^{(t)}$，计算

$$\nabla \mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right)=\frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(t)}\right)=n^{-1}\sum_{i=1}^{n}\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(t)}\right)$$

  - $\mathcal{L}_i$：针对于第$i$ 个训练样本的损失函数

- 参数更新

$$\boldsymbol{\theta}^{(t+1)}=\boldsymbol{\theta}^{(t)}-\alpha\nabla \mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right)$$

- 回到步骤 2 直至收敛

# 回顾

1．缺点

- 计算效率低下，尤其当样本量$n$ 很大的时候

2．为什么不能只用部分训练样本更新参数呢?

# 小批量梯度下降法

1. 想法：每一步参数更新中，只使用一小部分训练样本
2. 将训练样本指标集划分成若干互斥的集合：$\{1, \ldots, n\} = S_1 \cup \cdots \cup S_k$
   - $|S_1| = \cdots = |S_{k-1}| = m$
   - $|S_k| \leq m$
   - $k = \lceil n/m \rceil$
   - $m$ 通常是 2 的指数次幂。例如，$m = 512$

# 小批量梯度下降法

1．对于第$t$ 步更新，模型参数的更新规则为

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha \nabla \mathcal{J}_t \left(\boldsymbol{\theta}^{(t)}\right)$$

$$\nabla \mathcal{J}_t \left(\boldsymbol{\theta}^{(t)}\right) = |S_{t\%k+1}|^{-1} \sum_{i \in S_{t\%k+1}} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{\theta}} (\boldsymbol{\theta}^{(t)})$$

- 只用$S_{t\%k+1}$ 中的样本点计算梯度

2．当全部样本被遍历完，我们称训练完成一个周期（epoch）

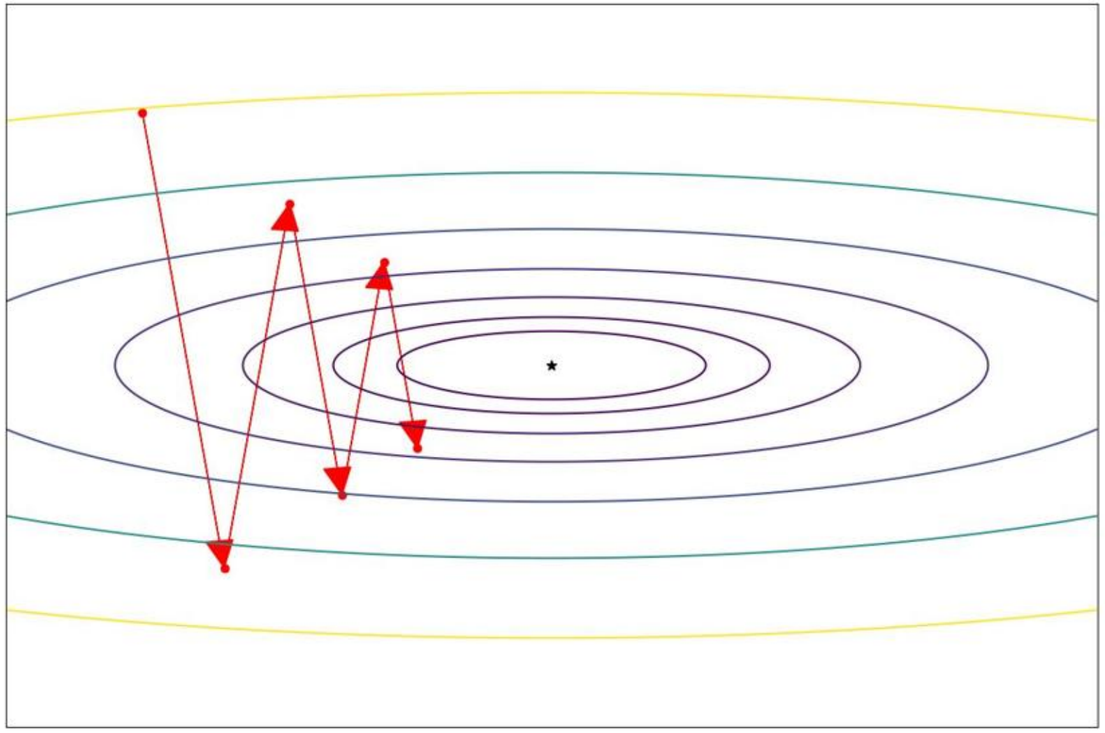
# 讨论

1. 两种特殊情况
   - $m = 1$：随机梯度下降法
   - $m = n$：批量梯度下降法
2. 当$m < n$ 时，小批量梯度下降法损失了计算精度
3. 然而，小批量梯度下降法却能够节省内存以及提高计算效率
4. 广泛用于深度学习模型的训练
5. 接下来，我们讨论几种更加有效的计算梯度的算法

# 问题背景

1．梯度展示的是代价函数值增加最快的方向

2．当代价函数的梯度“不稳定”时，梯度下降法的效率将降低



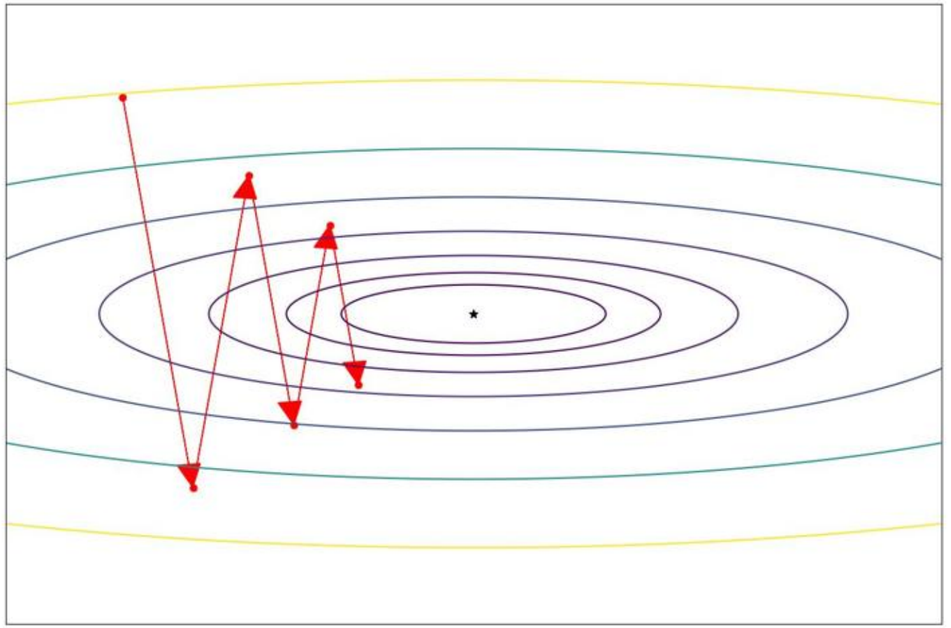
# 问题背景

1．我们希望用更加“稳定”的方向替代梯度

2．为了达到这个目标，我们考虑下面几种算法

- 动量法（Momentum）
- RMSprop
- Adam

# 动量法（Momentum）



1．我们希望达到如下效果

- 减小上-下方向的效应
- 增大左-右方向的效应

# 动量法（Momentum）

1．对过往梯度取均值?

- 将耗费大量内存用于存储过去的梯度
- 对于（带有海量参数的）深度学习模型而言，这么做几乎不可能

2．考虑EWMA (Exponential Weighted Moving Average)

- 原始序列：$\{s_i : i = 1, 2, \ldots\}$
- EWMA 序列：$\{v_i : i = 1, 2, \ldots\}$

$$v_i = \beta_1 v_{i-1} + (1 - \beta_1) s_i \quad (i = 1, 2, \ldots)$$

  - $v_0 = 0$
  - $\beta_1$ 控制我们多么重视 “动量” $v_{i-1}$ 的值

# 计算

1. 令$\boldsymbol{g}^{(t)}$ 表示第$t$ 步更新中，某参数的梯度值
   - 可以是$\mathrm{d}\boldsymbol{b}$ 或者$\mathrm{d}\boldsymbol{W}$ 在当前模型参数$\boldsymbol{\theta}^{(t)}$ 下的计算结果
   - 简便起见，我们忽略掉与迭代次数$t$ 相关的上角标
2. 通过以下方式得到 EWMA “梯度” $\boldsymbol{v}_b^{(t)}$

$$\boldsymbol{v}_b^{(t)} = \beta_1 \boldsymbol{v}_b^{(t-1)} + (1-\beta_1)\boldsymbol{g}^{(t)}$$

   - $\beta_1 = 0.9$：模型超参（通常不调参）
   - $\boldsymbol{v}_b^{(0)} = 0$：初始动量

# 计算

1. 一个事实

$$\frac{1}{(1-\beta_1)^{-1}}\sum_{i=1}^{\infty}\beta_1^i = 1$$

2. 更多细节

$$\begin{aligned}\boldsymbol{v}^{(t)} &= \beta_1\boldsymbol{v}^{(t-1)} + (1-\beta_1)\boldsymbol{g}^{(t)} \\ &= (1-\beta_1)\beta_1^{t-1}\boldsymbol{g}^{(1)} + (1-\beta_1)\beta_1^{t-2}\boldsymbol{g}^{(2)} + \cdots + (1-\beta_1)\boldsymbol{g}^{(t)} \\ &= \frac{1}{(1-\beta_1)^{-1}}\sum_{i=1}^{t}\beta_1^{n-i}\boldsymbol{g}^{(i)}\end{aligned}$$

- 当$t$ 较大时，以上结果可被视为求加权平均
- $(1-\beta_1)^{-1}$：可被视作 EMWA 的“有效样本量”

# 基于动量的梯度下降法

步骤1．随机初始化$\boldsymbol{\theta}^{(0)}$

步骤2．基于当前模型参数$\boldsymbol{\theta}^{(t)}$，计算$\mathrm{d}\boldsymbol{b}^{[1](t)}$

- 作为例子，我们仅展示模型参数$\boldsymbol{b}^{[1]}$ 的更新过程
- 相同的过程可用于更新其他模型参数

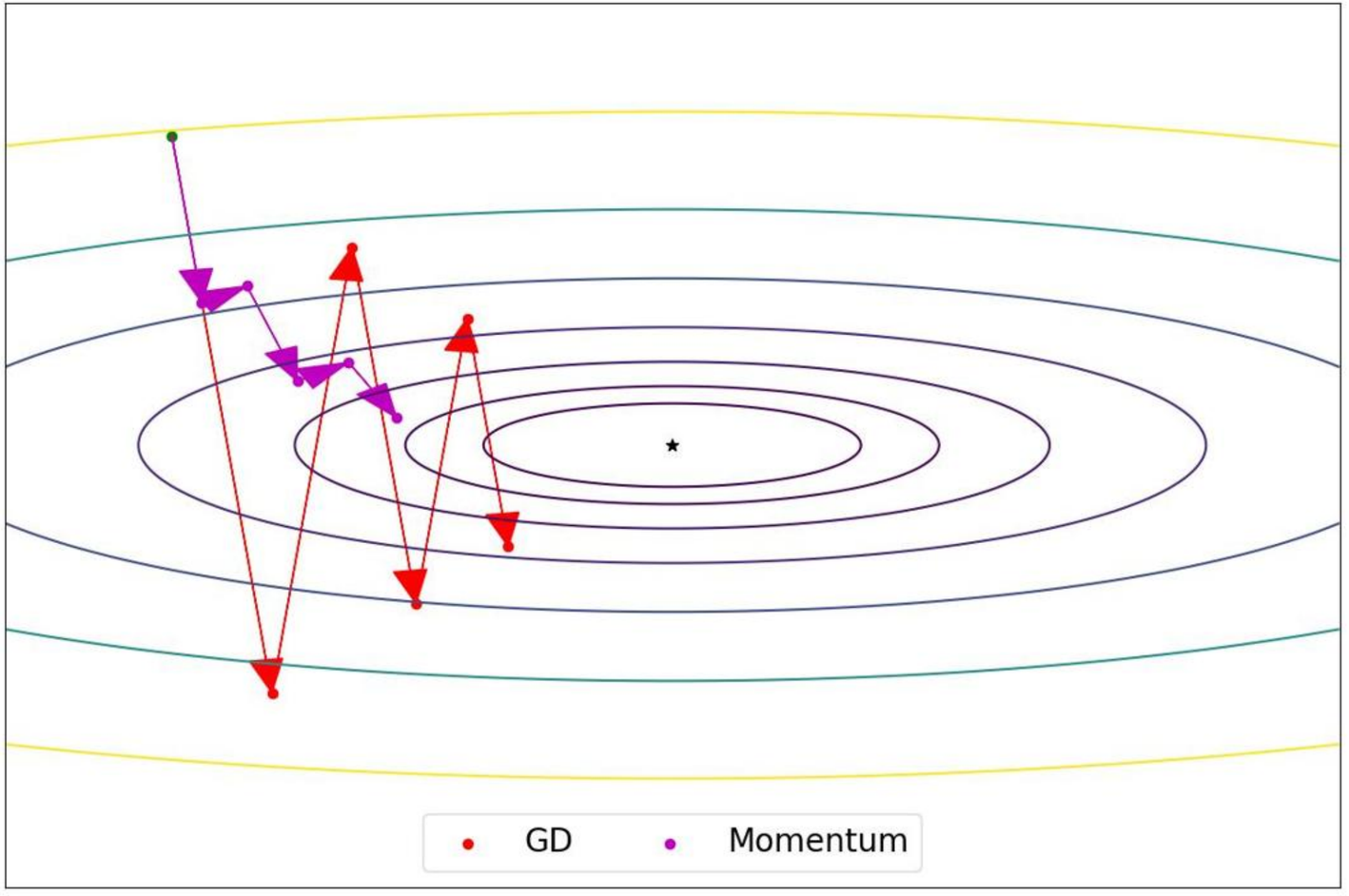
步骤3．更新模型参数

$$\boldsymbol{b}^{[1](t+1)} = \boldsymbol{b}^{[1](t)} - \alpha \boldsymbol{v}_b^{[1](t+1)}$$

- $\boldsymbol{v}_b^{[1](t+1)} = \beta_1 \boldsymbol{v}_b^{[1](t)} + (1-\beta_1)\mathrm{d}\boldsymbol{b}^{[1](t)}$
- $\boldsymbol{v}_b^{[1](0)} = 0$
- $\beta_1 = 0.9$ （默认值，通常不调参）

步骤4．回到步骤 2 直至收敛

# 比较



# 比较

1．动量的确降低了上-下方向的效应

2．然而，算法收敛速度较慢

3．一种可能的解决方法：不同方向用不同的学习率

- 较小的学习率用于上-下方向
- 较大的学习率用于左-右方向

# 基于RMSprop的梯度下降法

步骤1．随机初始化$\boldsymbol{\theta}^{(0)}$

步骤2．基于当前模型参数$\boldsymbol{\theta}^{(t)}$，计算$\mathrm{d}\boldsymbol{b}^{[1](t)}$

- 作为例子，我们考虑模型参数$\boldsymbol{b}^{[1]}$ 的更新过程

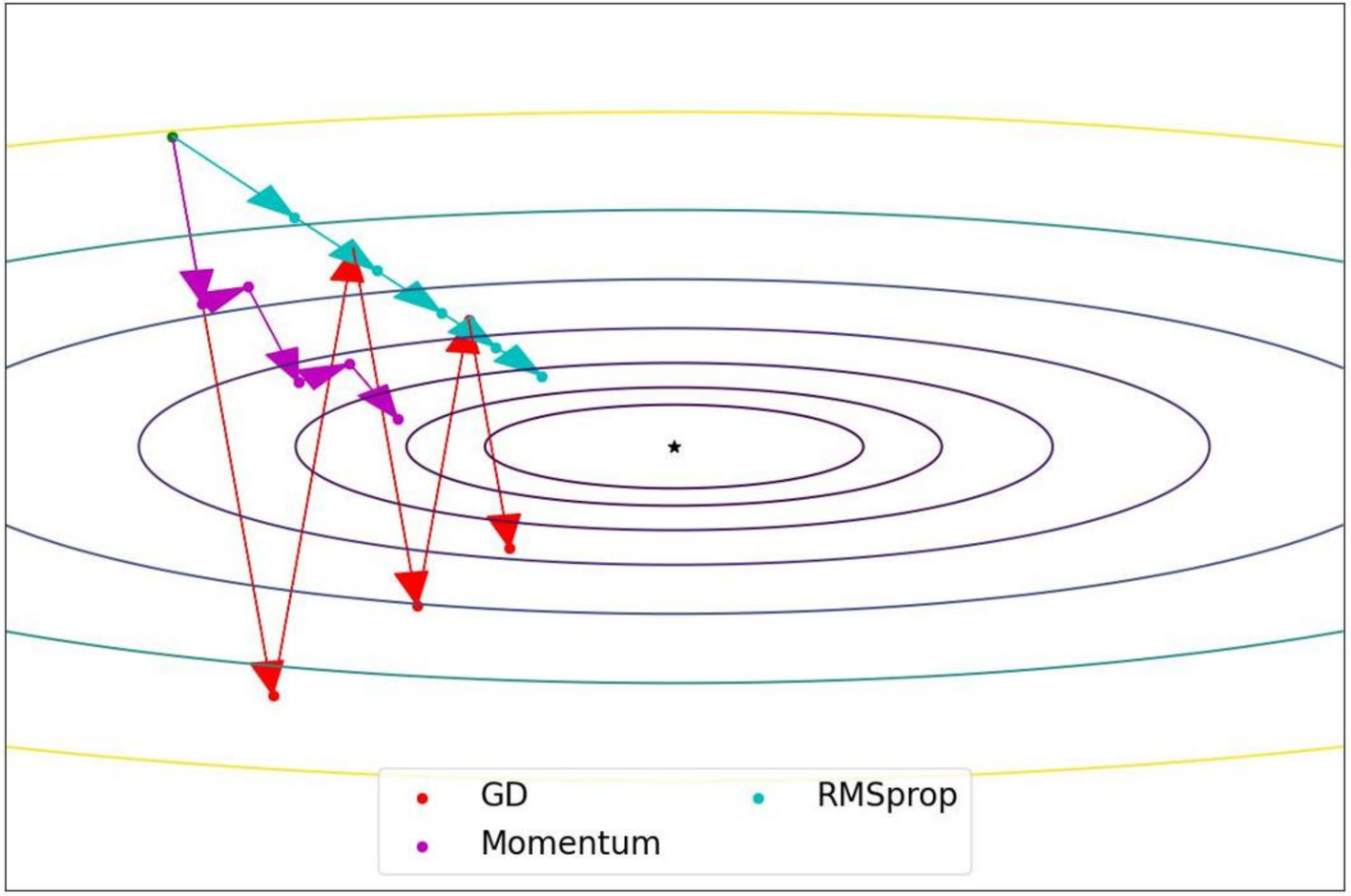
步骤3．更新参数

$$\boldsymbol{b}^{[1](t+1)} = \boldsymbol{b}^{[1](t)} - \frac{\alpha}{\sqrt{\epsilon + \boldsymbol{s}_b^{[1](t+1)}}} \mathrm{d}\boldsymbol{b}^{[1](t)}$$

- $\epsilon = 10^{-8}$ （默认值，通常不调参））
- $\boldsymbol{s}_b^{[1](t+1)} = \beta_2 \boldsymbol{s}_b^{[1](t)} + (1-\beta_2)\mathrm{d}\boldsymbol{b}^{[1](t)} \circ \mathrm{d}\boldsymbol{b}^{[1](t)}$
- $\boldsymbol{s}_b^{[1](0)} = 0$
- $\beta_2 = 0.99$ （默认值，通常不调参）

步骤4．回到步骤 2 直至收敛

# 比较



# 比较

1．RMSprop 对应的参数更新轨迹更加“平滑”

2．然而，收敛速度仍然不太理想

# 算法总结

1．传统的梯度下降法

$$\boldsymbol{b}^{[1](t+1)} = \boldsymbol{b}^{[1](t)} - \alpha \mathrm{d}\boldsymbol{b}^{[1](t)}$$

2．动量法只更改梯度部分

$$\boldsymbol{b}^{[1](t+1)} = \boldsymbol{b}^{[1](t)} - \alpha \boldsymbol{v}_b^{[1](t+1)}$$

3．RMSprop 只更改学习率部分

$$\boldsymbol{b}^{[1](t+1)} = \boldsymbol{b}^{[1](t)} - \frac{\alpha}{\sqrt{\epsilon + \boldsymbol{s}_b^{[1](t+1)}}} \mathrm{d}\boldsymbol{b}^{[1](t)}$$

4．为什么不同结合两种方法，同时更改两部分?

# Adam (Adaptive moment estimation)

步骤1. 随机初始化$\boldsymbol{\theta}^{(0)}$

步骤2. 基于当前模型参数$\boldsymbol{\theta}^{(t)}$, 计算$\mathrm{d}\boldsymbol{b}^{[1](t)}$

- 作为例子，我们考虑模型参数$\boldsymbol{b}^{[1]}$ 的更新过程

步骤3. 更新参数

$$\boldsymbol{b}^{[1](t+1)} = \boldsymbol{b}^{[1](t)} - \frac{\alpha}{\epsilon + \sqrt{\hat{\boldsymbol{s}}_b^{[1](t+1)}}} \hat{\boldsymbol{v}}_b^{[1](t+1)}$$

- 关于$\hat{\boldsymbol{s}}_b^{[1](t+1)}$ 以及$\hat{\boldsymbol{v}}_b^{[1](t+1)}$ 的解释请参见下一页
- $\epsilon = 10^{-8}$ （默认值，通常不调参）

步骤4. 回到步骤 2 直至收敛

# Adam (Adaptive moment estimation)

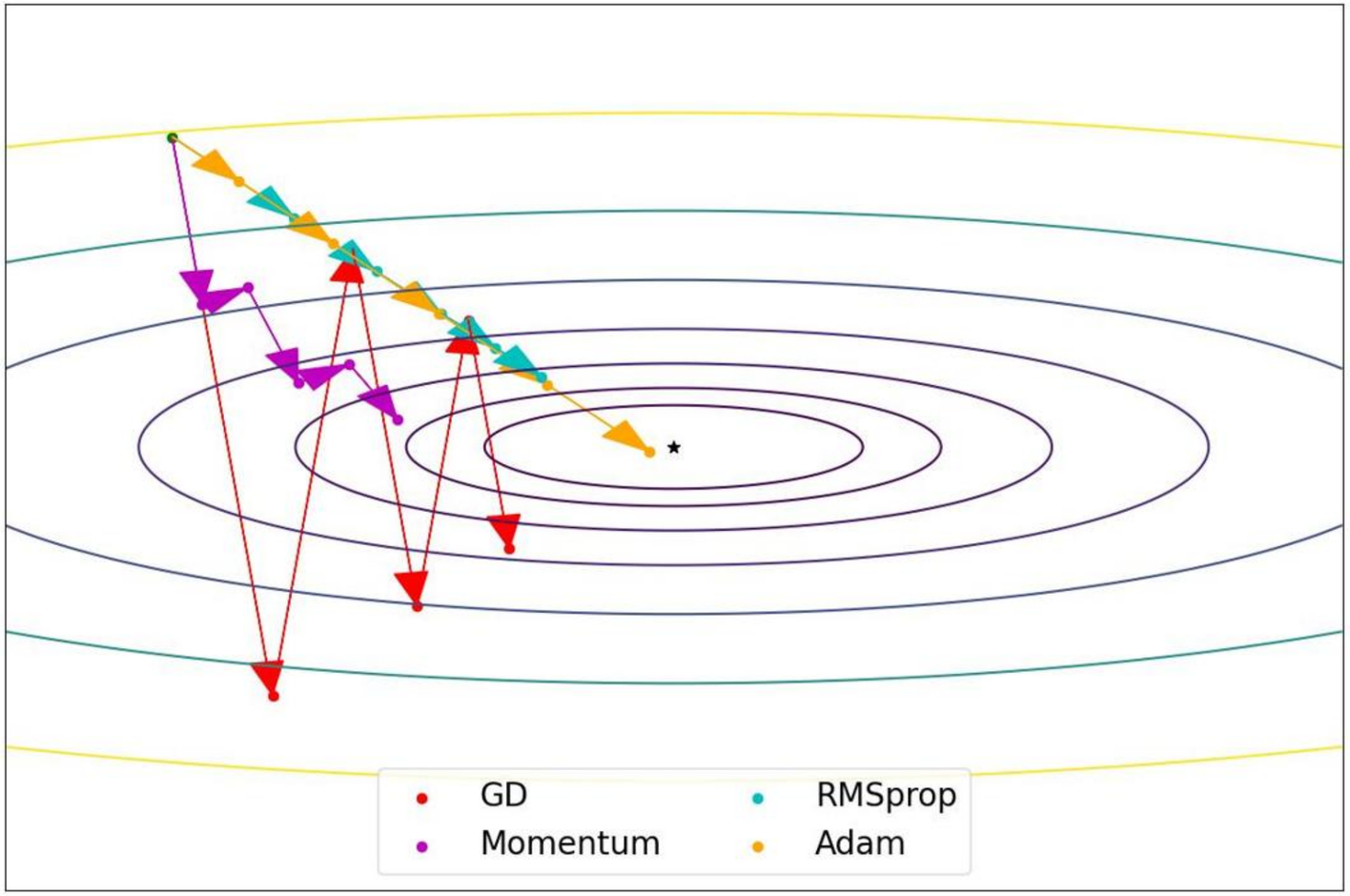
1. 关于$\hat{\boldsymbol{s}}_b^{[1](t+1)}$和$\hat{\boldsymbol{v}}_b^{[1](t+1)}$的计算细节

$$\boldsymbol{s}_b^{[1](t+1)} = \beta_2 \boldsymbol{s}_b^{[1](t)} + (1-\beta_2)\mathrm{d}\boldsymbol{b}^{[1](t)} \circ \mathrm{d}\boldsymbol{b}^{[1](t)} \quad \hat{\boldsymbol{s}}_b^{[1](t+1)} = \frac{\boldsymbol{s}_b^{[1](t+1)}}{1-\beta_2^{t+1}}$$

$$\boldsymbol{v}_b^{[1](t+1)} = \beta_1 \boldsymbol{v}_b^{[1](t)} + (1-\beta_1)\mathrm{d}\boldsymbol{b}^{[1](t)} \quad \hat{\boldsymbol{v}}_b^{[1](t+1)} = \frac{\boldsymbol{v}_b^{[1](t+1)}}{1-\beta_1^{t+1}}$$

- $\boldsymbol{v}_b^{[1](0)} = \boldsymbol{s}_b^{[1](0)} = 0$
- $\beta_1 = 0.9$ （默认值，通常不调参）
- $\beta_2 = 0.99$ （默认值，通常不调参）

# 算法对比



# 算法对比

1．Adam 表现最好

2．Adam 或者其变体通常用于深度模型参数估计

# 学习率衰减策略

1．直观

- 随着迭代的增加，估计值“应当”接近理论真值
- 我们不应当在每步更新中使用不变的学习率
  - [效率低] 较小的学习率通常具有较好的收敛性质，但我们通常需要多步迭代
  - [不稳定] 当迭代次数较大时，较大的学习率往往导致参数更新不太稳定
- 随着迭代次数的增加，我们应当以较为有效的方式降低学习率

# 学习率衰减策略

1. 记
   - $t$：迭代指标
   - $epoch$：周期指标
2. 有多种学习率衰减策略

$$\alpha_t = \frac{\alpha_0}{1+\gamma \cdot epoch}$$

$$\alpha_t = \frac{\alpha_0}{\sqrt{epoch}}$$

$$\alpha_t = 0.95^{epoch} \cdot \alpha_0$$

$$\alpha_t = \frac{1}{\sqrt{t}} \cdot \alpha_0$$

$$\alpha_t = 0.95^{t} \cdot \alpha_0$$

   - 例如，$\alpha_0 = 5 \times 10^{-3}$
   - $\gamma = 1$（默认值，通常不调参）