The Final Project of Optimization–NES

162040217 谢善杰

南京航空航天大学　计算机科学与技术学院/人工智能学院

日期：2022 年 5 月 31 日

1. 文章要处理的问题是什么, 这个问题为什么重要?

现实世界中的优化问题往往不会像期末考试题那样理想，它们往往都是过于困难且复杂的，无法直接建模，就更不用提函数凸否等等性质，同时这类问题普遍的存在于日常生活中的很多领域，从健康和科学应用到航空设计与控制，所以处理这类问题是具有现实需求的。

本文介绍的自然进化策略（NES）就是用来解决此类问题，它是一种黑盒式的优化算法，我们只需要告诉计算机最终的解法好坏与否，无需关心中间的过程是怎样的。如果没有这类算法，问题的解决只能依靠该领域的专家以巨大的成本手工制作解决方案，而且效果往往是非常差劲的。

2. 此问题存在什么难点? 现有的其他人/其他方法是怎样处理这些难点的? 这些方法有什么优缺点?

黑盒优化算法已经发展很长时间了，也产生了许多令人熟知的算法，第一类经典的黑盒优化算法是受到经典优化方法启发，像单纯形法、拟牛顿法、模拟退火、遗传算法等等。另外一类比较新颖的黑盒优化算法就是进化策略，其旨在应对高维连续值域，较高的性能和较好的处理结果让此类方法一直都处在一个比较活跃的研究领域。虽然进化策略在黑箱优化方面已被证明是有效的，但分析程序的实际动态结果却很困难。

3. 本文使用什么方法来处理问题的难点? 这样处理有什么优点?

本文介绍的 NES 的思想其实与进化策略很是相像，都是选择好的下一代进行迭代更新，但是 NES 还结合了神经网络中的梯度下降思路，因为在传统的进化算法中，涉及到突变和重组两个步骤，而这两个步骤是完全随机的，也就是说我们很可能找到比当前解法更差的解法，这是我们不愿看到的，而通过引入梯度上升的思想，保证我们的突变能够向更好的方向前进。此外，NES 还将“总体”表示为一个参数化分布，将原问题转换为一个概率分布上的优化问题，这样做便于我们用样本来估计总体以达到问题优化的目的。

本文用到的自然梯度又与普通的梯度下降不同。首先我们先抽象一下我们的目标，假设我们想要优化一个函数 $f(x)$，而且无法直接计算梯度。但是，我们在给定任意 x 的情况下是能够确切地评估 $f(x)$ 的好坏。另外，我们认为随机变量 x 的概率分布 $p_\theta(x)$ 是函数 $f(x)$ 优化问题的一个较优的解，向量 θ 是分布 $p_\theta(x)$ 的参数。因此我们的目标就是找到一个能使 $f(x)$ 较优的参数解 θ。

普通的梯度：以当前的 θ 为起点，在很小的一段欧式距离内, 针对参数空间中的 θ 找下降最快的方向（也可以形象地说是最陡峭的方向）。

自然梯度：在参数为 θ 的概率分布空间中，以 KL 散度[1] 作为距离的度量，寻找最陡峭的下降方向。

3.1. 普通梯度的弊端

我们通常会遇到多维度问题，但是对于普通梯度法，即便在处理一维的问题，严重的短板就显现出来了，以一维高斯分布为例，我们需要优化的参数 $\theta = (\mu, \sigma)$, 通过简单的推导可得到，$\Delta\mu \propto \frac{1}{\sigma}$ 和 $\Delta\sigma \propto \frac{1}{\sigma}$。

更新时如图（1）所示，通过分析，我们的目标是让 μ 和 σ 都尽可能接近于 0，问题就产生了：刚开始更新时，σ 的取值比较大，远大于 1，这样会使得参数 θ 的更新缓慢进行，当然这不是问题的重点，因为我们可以通过增大学习率，来提高更新的步伐，问题出现在当更新进行一段时间后，$\sigma \ll 1$，这就会使得之后的参数更新过大，非常容易一次更新就使得参数变的非常差（就像图中（3）到（1）的更新效果），同样我们可以通过调小学习率（增加样本数量也可以）来缓解这种更新不稳定的情况，但是只能算得上是缓解，终究达不到避免此类弊端的地步，因此就需要从源头上下手，这边自然引出我们的自然梯度。

相比普通梯度，自然梯度的改变是将处理对象从简单的参数空间转换到概率分布空间中，使用 KL 散度替换欧式距离，这样做可以解除普通梯度更新时对特定参数的依赖，使更新相对于所使用的特定参数保持不变。观察自然梯度优化的目标函数：

$$\max_{\delta\theta} J(\theta + \delta\theta) \approx J(\theta) + \delta\theta^T \nabla_\theta J,$$
$$s.t. D(\theta + \delta\theta || \theta) = \epsilon$$

其中，$J(\theta)$ 是适应度函数的期函数，$D(\theta + \delta\theta||\theta)$ 是 KL 散度。

普通梯度的方向是 $\nabla_\theta J$，自然梯度的方向是 $\tilde{\nabla}_\theta J = F^{-1}\nabla_\theta J(\theta)$，其中 F 是 Fisher Information 矩阵。对于 Fisher Information 矩阵[2] 我们知道 F 是 $s(\theta)$ 的协方差矩阵（其中 $s(\theta) = \nabla_\theta \log p(x|\theta)$ 是用来评估参数 θ 好坏的函数）直观理解就是，Fisher Information 矩阵反映了我们对参数估计的准确度，它越大，对参数估计的准确度越高，即代表了越多的信息。同时 F 也是 KL 散度的 Hessian 矩阵，即 $\mathrm{F} = \mathrm{H}_{\mathrm{KL}[p(x|\theta) \,\|\, p(x|\theta')]}$，可以看到 F 的作用其实是对 KL 散度的曲率的分析衡量，这样保证了每一步更新 $\tilde{\nabla}_\theta J$ 都是沿着分布的流形以恒定的速率移动，不会因为其曲率而减速。

我们比较普通梯度和自然梯度的更新方向，容易发现自然梯度多了个 F^{-1}，这个 F 是与参数估计的准确度、KL 散度的 Hessian 矩阵相关的，涉及到方差分量的更新，方差（不确定性）高的方向

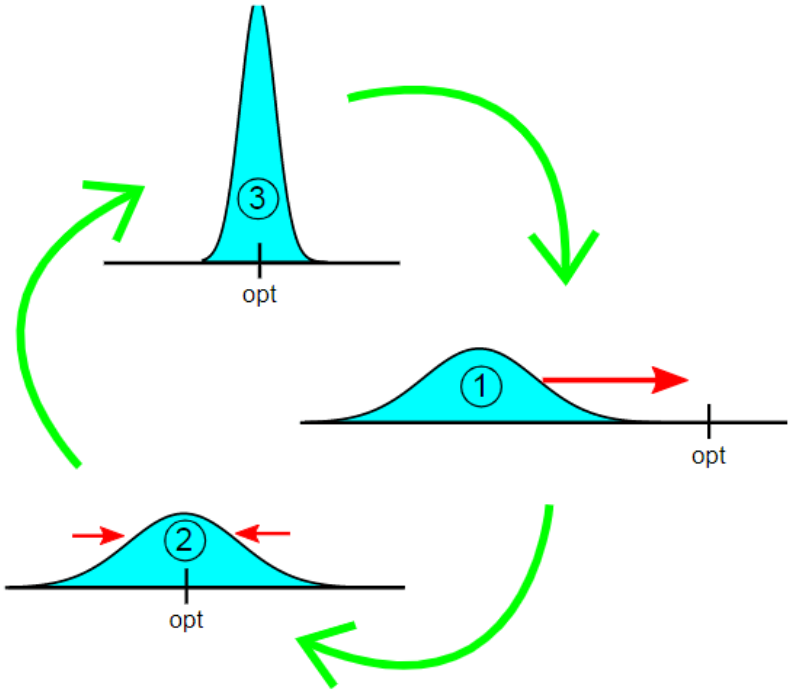


图 1: 一维高斯分布

会受到惩罚，因此会缩小，而方差（不确定性）低的分量会增加，这使得（红色虚线）自然梯度比（绿色）普通梯度具有更可靠的更新方向，如图（2）所示。

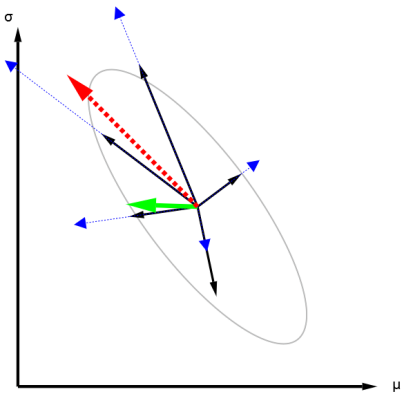


图 2: 普通梯度更新 VS 自然梯度更新

3.2. 性能和鲁棒性

本文除了介绍自然梯度来替换普通梯度外，还使用了一些启发式算法提高 NES 的性能和鲁棒性。

(1) 基于排序的适应度塑造（Fitness shaping）算法，使 NES 算法在适应度函数的单调递增变换下保持不变，从而提高了鲁棒性。注意这里不是直接使用适应度函数 $f(x)$，而是用效用函数将其替代，从而将效用函数看作为 NES 的一个自由参数。

(2) 自适应采样（Adaptation sampling）算法来在线调整学习率 η，从而在标准基准上产生高性能的结果。这是一个普遍且稳健的算法，它能在能够取得好的进步的时候，采取一个比较大的 η，否则就用一个小的 η，很谨慎地前进。

(3) 指数参数化，对于维护协方差矩阵正定至关重要，使用局部自然坐标保证计算的可行性。

3.3.xNES 与 CMA-ES 联系（便于后续实验对比）

原始协方差矩阵适应进化策略（CMA-ES）与 xNES 是密切相关的，在一阶泰勒近似下，指数化的 xNES 更新与 CMA-ES 的秩 μ 更新一致。CMA-ES 实际上遵循一个近似的自然梯度，可以说，CMA-ES 可以被视为 NES 家族的一员。

两种算法虽然密切相关，但是也存在很多不同，尽管都使用了三个参数来参数化搜索分布，但是对参数的更新有所差异，对于均值 σ 的更新是相同的，但是对于另外两个参数的更新则不同，xNES 是进行解耦更新，即分开更新，而 CMA-ES 则采用耦合更新。而针对更为复杂的协方差矩阵，相对于 xNES 在局部自然坐标中进行指数参数化，CMA-ES 是在全局线性坐标中表示的。xNES 和 CMA-ES 之间的另一个区别是 xNES 中指数参数化更新，这导致对于协方差矩阵，xNES 会乘法更新方程，与 CMA-ES 的加法更新相反。

另外，CMA-ES 使用成熟的进化路径技术[3] 来消除多代的随机效应，这项技术在处理小规模问题时尤其有用，因此默认在这两种算法中都存在，有利于改善算法的稳定性，同时我们要承认进化

路径技术会让 CMA-ES 算法变得复杂。

4. 实验效果如何？从实验效果上来说, 相对其他方法有什么优缺点?

实验将 xNES 和 xNES-as（as 表示使用自适应采样）以 BIPOP-CMA-ES（2009 年 BBOB 的获胜者）为参考，在黑盒优化基准测试的全部基准函数上进行测试比较，BIPOP-CMA-ES 在大多数含噪音的基准函数上表现的更好的多，在无噪音函数上，xNES、xNES-as 和 BIPOP-CMA-ES 表现的差不多。

同时，在某些特定的情况下，xNES 和 xNES-as 也是有优势的。例如：在维数等于 5 或 20 时，无论有无噪音，xNES 的表现都是优于 BIPOP-CMA-ES 的，这表明，假设函数评估的预算超过维数的 100 倍，xNES 是最佳算法之一。对于 xNES-as 算法，实验发现，在函数 f_{115} 和有预算限制下的函数 f_{18}, f_{118}, f_{119} 中，xNES-as 是优于 BIPOP-CMA-ES 的。

实验还分析了可分离 NES 在神经进化上的表现，在困难的非马尔可夫双极平衡任务中，SNES 很好地处理了这种高维、多模态的问题。但有一点值得提的是，在实际场景中，我们无法事先知道最佳网络规模，因此，我们非常需要一种能够根据问题维度进行适当扩展的算法，而正巧我们发现 SNES 可以满足我们的需求，随着维数的增加，SNES 在众多评估函数上表现也是优于 xNES 的，事实上，针对神经进化问题，xNES 和 SNES 的结果都是最先进的，比之前最好的算法高出一倍。

最终我们总结出，NES 适用于一般的可参数化分布，其相关变体能在某些特定问题中表现出优越的性能。

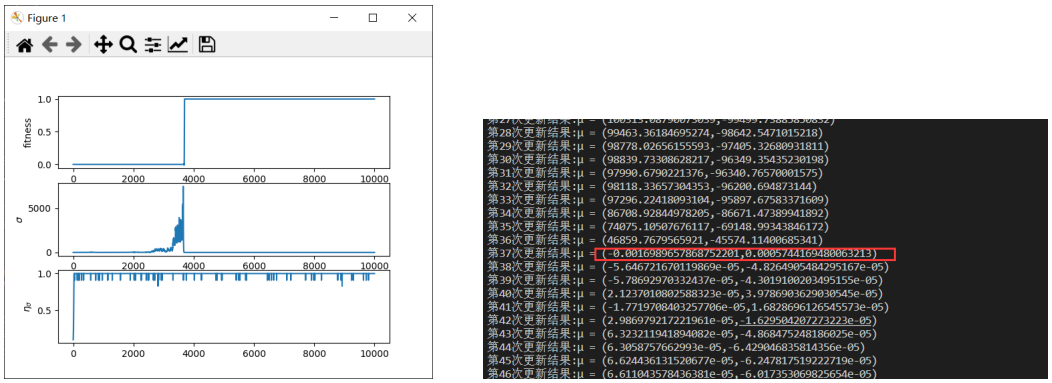
5. 代码链接

NES 代码实现

例子中，我们要找

$$f(x,y)=\frac{\sin(x^2+y^2)}{x^2+y^2}$$

的精确解 $\mu=(0,0)$, 即使在一个非常坏的初始值 (99999,-99999) 下，如图（3）所示，算法仍能在第 37 次迭代到最优点。



(a) Fig1 (b) Fig2

图 3: Result of xNES

参考文献

[1] PASCANU R, BENGIO Y. Revisiting natural gradient for deep networks[J]. arXiv preprint arXiv:1301.3584, 2013.

[2] KRISTIADI A. Fisher Information Matrix[EB/OL]. 2018, Mar 11. https://agustinus.kristia.de/techblog/2018/03/11/fisher-information/.

[3] LI Z, ZHANG Q. What does the evolution path learn in cma-es?[C]//International Conference on Parallel Problem Solving from Nature. [S.l.]: Springer, 2016: 751-760.