Natural Evolution Strategies (N E S)

论文中函数自己理解

$\pi(z|\theta)$概率密度函数

$f(z)$ 适应度函数（fitness function）打分衡量模型好坏的？z是样本

$J(\theta)$ 是$f(z)$ 适应度函数的期望 有$J(\theta) = E_\theta[f(z)] = \int f(z)\pi(z|\theta)dz$

目标 $\max_{\delta\theta} J(\theta + \delta\theta)$

$D(\theta + \delta\theta||\theta)$是KL散度 _论文里用Fisher Information Matrix 表示 KL散度

plain gradient方向：$\nabla_\theta J$

natural gradient 方向：$\tilde{\nabla}_\theta J = F^{-1}\nabla_\theta J(\theta)$

自然选择策略, 是一种黑箱式优化算法. 所谓黑箱式优化算法, 是说我们只需要告诉计算机"什么解法是好的", "什么解法是坏的". 具体来说, 我们需要有一个函数, 该函数能够针对一个解法, 返回一个该解法"多好"的程度的值. 除此之外, 我们不需要操心怎么样找到好的解法. 这一切交给计算机做就好.

目前主流的黑箱优化算法包括神经网络, 进化算法等. 传统算法难以解决的问题, 比如旅行推销员问题(Traveling salesman problem), 在这些新式算法的协助下, 得到了相当令人满意的答案.

Wirestra等人提出了将进化算法和神经网络中的梯度下降思路结合在一起的想法. 传统的进化算法包含突变和重组这两个步骤. 我们通过这两个步骤, 期待找到更好的解法. 然而, 突变和重组是完全随机的. 多数情况下, 他们会导致和当前解法相比更差的解法. 因此, 我们想引入梯度下降(gradient descent)或梯度上升(gradient ascent)的思想, 从而使得突变总是能够朝着更好的解法迈进.

换句话说, 我们用梯度下降替代了突变和重组步骤.

【使用梯度让进化算法向更优的方向进化，替代之前的随机突变重组】

链接： https://echenshe.com/class/ea/1-01-intro.html

1. 进化策略ES和遗传算法GA的异同

遗传算法 (后面简称 GA) 和 ES 真 TM 差不多. 导致很多朋友学习的时候, 都傻傻分不清. 不过我具体的列出来, 方便看清楚.

- 选好父母进行繁殖 (`GA`); 先繁殖, 选好的孩子 (`ES`)
- 通常用二进制编码 DNA (`GA`); 通常 DNA 就是实数, 比如 1.221 (`ES`)
- 通过随机让 1 变成 0 这样变异 DNA (`GA`); 通过正态分布(Normal distribution)变异 DNA (`ES`)

具体来说, 传统的 GA 的 DNA 形式是这样:

DNA=`11010010`

而传统的 ES DNA 形式分两种, 它有两条 DNA. 一个 DNA 是控制数值的, 第二个 DNA 是控制这个数值的变异强度. 比如一个问题有4个变量. 那一个 DNA 中就有4个位置存放这4个变量的值 (这就是我们要得到的答案值). 第二个 DNA 中就存放4个变量的变动幅度值.

DNA 1=`1.23, -0.13, 2.35, 112.5` 可以理解为4个正态分布的4个平均值.

DNA 2=`0.1, 2.44, 5.112, 2.144` 可以理解为4个正态分布的4个标准差.

所以这两条 DNA 都需要被 `crossover` 和 `mutate`.

2. 启发式算法

链接：[优化二——启发式算法 - 知乎 (zhihu.com)](zhihu.com)

这类算法算是算法在实际应用过程中的技术性算法，因为在实际计算过程中，很有可能出现如下状况：第一：无确定的目标函数；第二：无法找到真实的最优解；第三：计算代价非常大，现有的计算机技术都无法解决此类问题。

所以人们就找到另一种情况，使得在可接受的计算成本内去搜寻最好的解，但不一定能保证所得的是可行解和最优解，甚至在多数情况下，无法阐述所得解同最优解的近似程度。通熟点讲就是求出来的解可能不是最好的，只能说是相对较好的，但是这个相对程度就不敢保证了。虽然是这样，但是在工程应用上来说已经非常好了。

启发式算法是相对于最优化算法提出的，是基于直观或者经验构造的算法，在可接受的开销（时间和空间）内给出待解决组合优化问题的一个可行解。

算法举例：模拟退火（Simulated Annealing,SA）、信赖域算法、遗传算法（GA）、粒子群算法、人工神经网络算法（ANN）、禁忌搜索等等

【感受：有点贪心的感觉，找局部最优】

STORY

启发式算法蕴含着许多人生哲学，它虽不是数学方法，其思想更类似于人类解决问题的思想和一些人生中总结的道理，值得好好体会。最后用网上一段描述局部搜索的例子来作为总结：

为了找出地球上最高的山，一群有志气的兔子们开始想办法。

· 兔子朝着比现在高的地方跳去。他们找到了不远处的最高山峰。但是这座山不一定是珠穆朗玛峰。这就是Iterative Improvement，它不能保证局部最优值就是全局最优值。

·兔子喝醉了。他随机地跳了很长时间。这期间，它可能走向高处，也可能踏入平地。但是，他渐渐清醒了并朝最高方向跳去。这就是模拟退火。

·兔子们知道一个兔的力量是渺小的。他们互相转告着，哪里的山已经找过，并且找过的每一座山他们都留下一只兔子做记号。他们制定了下一步去哪里寻找的策略。这就是禁忌搜索。

兔子们吃了失忆药片，并被发射到太空，然后随机落到了地球上的某些地方。他们不知道自己的使命是什么。但是，如果你过几年就杀死一部分海拔低的兔子，多产的兔子们自己就会找到珠穆朗玛峰。这就是遗传算法。

链接：https://www.zhihu.com/question/316175486/answer/1326928711

《智能优化算法及其**MATAB**实例》，电子工业出版社

在电子、通信、计算机、自动化、机器人、经济学和管理学等众多学科中，不断地出现了许多复杂的组合优化问题。面对这些大型的优化问题，传统的优化方法（如牛顿法、单纯形法等）需要遍历整个搜索空间，无法在短时间内完成搜索，且容易产生搜索的“组合爆炸”。例如，许多工程优化问题，往往需要在复杂而庞大的搜索空间中寻找最优解或者准最优解。鉴于实际工程问题的复杂性、非线性、约束性以及建模困难等诸多特点，寻求高效的优化算法已成为相关学科的主要研究内容之一。

受到人类智能、生物群体社会性或自然现象规律的启发，人们发明了很多智能优化算法来解决上述复杂优化问题，主要包括：模仿自然界生物进化机制的遗传算法；通过群体内个体间的合作与竞争来优化搜索的差分进化算法；模拟生物免疫系统学习和认知功能的免疫算法；模拟蚂蚁集体寻径行为的蚁群算法；模拟鸟群和鱼群群体行为的粒子群算法；源于固体物质退火过程的模拟退火算法；模拟人类智力记忆过程的禁忌搜索算法；模拟动物神经网络行为特征的神经网络算法；等等。这些算法有个共同点，即都是通过模拟或揭示某些自然界的现象和过程或生物群体的智能行为而得到发展；在优化领域称它们为智能优化算法，它们具有简单、通用、便于并行处理等特点。

---

3. 贪心算法 启发式算法 近似算法 区别

近似算法是一个大类。所有对于有确切最优解但是并不能保证得到最优解的算法都可以称之为近似算法。

贪心算法不一定就是近似算法，如果可以证明决策既不受之前决策的影响，也不影响后续决策，则贪心算法就是确定的最优解算法。除此之外都不可以保证贪心算法是最优的。（这里有个很有意思的事情是，某种意义上动态规划方法就是将原问题变形为贪心可解的问题，并且是可以证明贪出来的是最优解，因此对于每个状态只需要保留最优决策）

启发式算法则是一个折中。我们知道对于大多数问题而言纯粹的遍历所有状态总比贪心来的慢，但是贪心得到确切解的前提很苛刻，因此很有可能会得到很烂的解。启发式算法则是通过一个启发函数对下一阶段状态进行粗略评估，这个评估是预计结果（并不确信）；在遍历状态的过程中，我们优先去看可能得到好结果的情况，并且如果发现一个状态的理想估计比当前确实遍历过的解中最好的要差则可以放弃，这就叫启发式算法，这里我们叫确定性启发式算法或者启发式搜索。然而启发式搜索的本质仍旧是遍历所有状态，要保证确定最优解，它仍旧是个遍历，耗时仍然有可能很大。这个情况下加入一条限制：如果计算时间已经到达可承受的极限，则放弃剩下部分的搜索直接以当前找到的最好结果为结果，这样的启发式算法自然是非确定性的，于是可以算作近似算法。

---

4. Fisher Informatica Matrix

链接：https://www.zhihu.com/question/26561604/answer/33275982 writed by Detian Deng

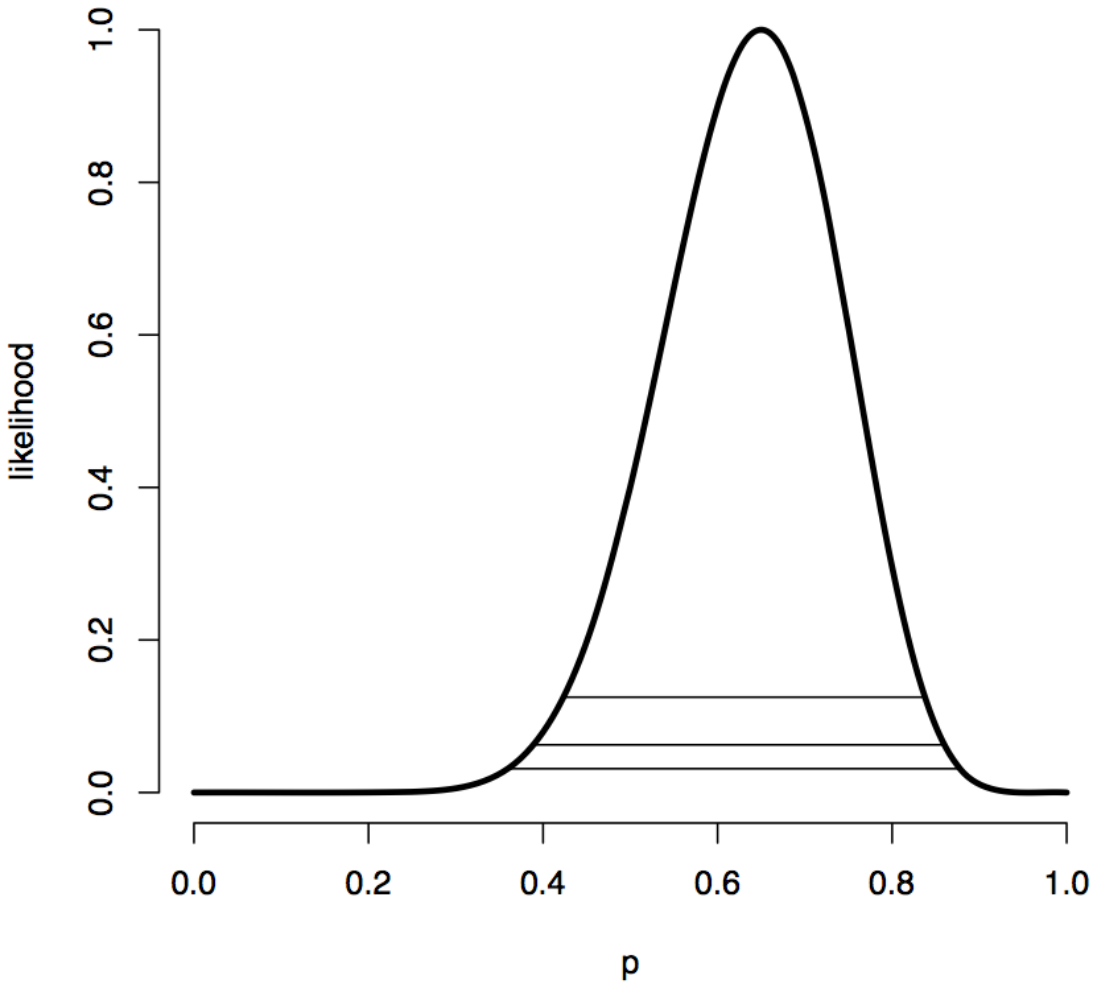
https://math.stackexchange.com/questions/265917/intuitive-explanation-of-a-definition-of-the-fisher-information

Fisher 信息矩阵（Fisher Information Matrix）的定义，正是这个 score function 的二阶矩（second moment）：$\boldsymbol{I(\theta) = E\left[S(X;\theta)^2\right]}$， 其中$S(X;\theta) = \sum_{i=1}^{n} \frac{\partial \log f(X_i;\theta)}{\partial \theta}$

二阶矩就是 随机变量 平方 的期望

Fisher Information的第一条数学意义：就是用来估计**MLE**的方程的方差。它的直观表述就是，随着收集的数据越来越多，这个方差由于是一个Independent sum的形式，也就变的越来越大，也就象征着得到的信息越来越多

Fisher Information的第二条数学意义：**log likelihood**在参数真实值处的负二阶导数的期望。对于这样的一个log likelihood function，它越平而宽，就代表我们对于参数估计的能力越差，它高而窄，就代表我们对于参数估计的能力越好，也就是信息量越大。而这个log likelihood在参数真实值处的负二阶导数，就反应了这个log likelihood在顶点处的弯曲程度，弯曲程度越大，整个log likelihood的形状就越偏向于高而窄，也就代表掌握的信息越多。



Fisher Information的第三条数学意义可以证明MLE的渐进分布的方差是$I^{-1}(\theta)$,即$Var(\hat{\theta}_{MLE}) = I^{-1}(\theta)$直观解释就是，Fisher Information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。

总结：Fisher Information Matrix越大，代表的信息就越多

---

5. NES VS CMA-ES

链接： https://zhuanlan.zhihu.com/p/97120383

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 的情况下仍然可以评估 f(x)，而且得到确定性的结果。我们认为随机变量 x 的概率分布 pθ(x) 是函数 f(x) 优化问题的一个较优的解，θ 是分布 pθ(x) 的参数。目标是找到 θ 的最优设置。

假设初始值为 θ，我们可以通过循环进行下面的三个步骤持续更新 θ：

1. 生成一个【 如何生成是个问题 】样本的种群 D={(xi,f(xi)}，其中 xi~pθ(x)。
2. 估计 D 中样本的「适应度」。
3. 根据适应度或排序，选择最优的个体子集，并使用它们来更新 θ。

5.1 关于 协方差矩阵自适应演化策略（CMA-ES）

标准差 σ 决定了探索的程度：当 σ 越大时，我们就可以在更大的搜索空间中对后代种群进行采样。在简单高斯演化策略中，σ(t+1) 与 σ(t) 密切相关，因此算法不能在需要时（即置信度改变时）迅速调整探索空间。

虽然不理解，但是 （CMA-ES）通过使用协方差矩阵 C 跟踪分布上得到的样本两两之间的依赖关系，解决了这一问题。

以简单的高斯演化策略为例，之前的分布是

$\theta = (\mu, \sigma), \quad p_\theta(x) \sim \mathcal{N}(\mu, \sigma^2 I) \sim \mu + \sigma\mathcal{N}(0, I)$

现在变成

$\theta = (\mu, \sigma, C), \quad p_\theta(x) \sim \mathcal{N}(\mu, \sigma^2 C) \sim \mu + \sigma\mathcal{N}(0, C)$

其中，σ 控制分布的整体尺度，我们通常称之为「步长」。

看不懂啊 =（

6. NES补充

在参数的搜索分布上进行优化，并将分布朝着自然梯度所指向的高适应度方向移动

问题：给定一个参数为 θ 的目标函数 J(θ)，我们的目标是找到最优的 θ，从而最大化目标函数的值

朴素梯度：会以当前的 θ 为起点，在很小的一段欧氏距离内找到最「陡峭」的方向，同时我们会对参数空间施加一些距离的限制。

自然梯度：用到了参数为 θ, pθ(x)（在 NES 的原始论文中被称为「搜索分布」)的概率分布空间。它在分布空间中的一小步内寻找最「陡峭」（变化最快）的方向，其中距离由 **KL 散度**来度量。在这种限制条件下，我们保证了每一步更新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。

自然梯度中涉及到的Fisher Information Matrix可以理解成一个表示

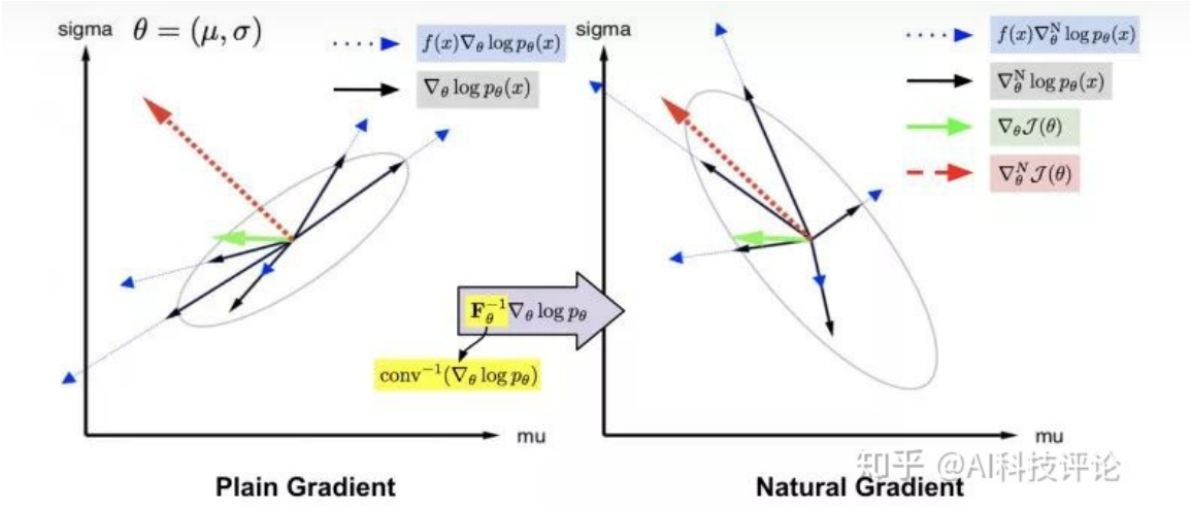


图 4：右侧的自然梯度样本（黑色实箭头）是左侧的朴素梯度样本（黑色实箭头）乘以其协方差的逆的结果。这样一来，可以用较小的权重惩罚具有高不确定性的梯度方向（由与其它样本的高协方差表示）。因此，合成的自然梯度（红色虚箭头）比原始的自然梯度（绿色虚箭头）更加可信。

除了自然梯度，NES 还采用了一些重要的启发式方法让算法的性能更加鲁棒。

- NES 应用了基于排序的适应度塑造（Rank-Based Fitness Shaping）算法，即使用适应度值单调递增的排序结果，而不是直接使用 f(x)。它也可以是对「效用函数」进行排序的函数，我们将其视为 NES 的一个自由参数。
- NES 采用了适应性采样（Adaptation Sampling）在运行时调整超参数。当进行 θ→θ′ 的变换时，我们使用曼-惠特尼 U 检验（ [Mann-Whitney U-test）对比从分布 pθ 上采样得到的样本以及从分布 pθ′ 上采样得到的样本。如果出现正或负符号，则目标超参数将减少或增加一个乘法常数。请注意，样本 xi′~pθ′(x) 的得分使用了重要性采样权重 wi′=pθ(x)/pθ′(x)。

我们在**BBOB基准**测试上的结果表明，NES算法在各种各样的黑盒优化问题上表现良好。我们已经证明了特定变体的优势和局限性，并因此建立了**NES框架的通用性和灵活性**。对重尾分布和可分离分布的实验证明了该方法在高维域上的可行性。在训练双极平衡神经控制器这一艰巨任务上，我们获得了最好的报告结果。

我们介绍的各种技术。普通搜索梯度存在过早收敛和缺乏尺度不变性的问题（见第2.2节）。因此，我们使用自然梯度代替，这将使NES成为一种可行的优化方法。为了提高性能和鲁棒性，我们引入了几种新技术。**Fitness shaping**使NES算法对拟合函数的保序变换保持不变，从而提高了鲁棒性。自适应采样在线调整学习率，从而在标准基准上产生高性能的结果。最后，指数参数化对于保持正定协方差矩阵至关重要，而自然坐标系的使用保证了计算的可行性。

实验：NES适用于一般的可参数化分布。在本文中，我们对两种变体进行了实验研究，根据不同问题类别的特殊性质进行了调整。我们使用一个完整的多正态分布证明了**xNES**变体的威力，该分布在标准基准的规范套件上，在任意平移和旋转下都是不变的。此外，我们还表明，在困难的非马尔可夫双极平衡任务中，协方差矩阵对对角线参数化（SNES）的限制允许缩放到非常高的维度。

然而，在实际场景中，我们无法事先知道最佳网络规模，因此谨慎的选择在于高估所需的规模。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，我们发现（图8，右图）SNES正显示出这种行为。随着维数的增加，SNES在函数求值数量上的表现也优于SNES，这表明基准不是病态的，不需要使用完整的协方差矩阵。我们推测，这是大多数神经进化问题的共同特性，这些问题有足够的权重来显示冗余的全局最优解（其中一些问题可以在不考虑所有参数协方差的情况下找到）。例如，Schaul等人（2011年）在Lennard-Jones原子簇基准（维数大于200）上的其他SNES结果