

Utmost IV 옵티마이저 소개

Utmost IV는 현재 파라미터 추출을 위해 6개의 옵티마이저를 포함하고 있습니다. 옵티마이저를 선택하는 것은 종종 어려울 수 있습니다. 여기서 옵티마이저를 검토하여 적절하게 선택하기 위한 몇 가지 방법을 제시합니다.

최적화 작업은 일반적으로 로컬 및 글로벌의 두 범주로 나뉩니다. 로컬 최적화는 비용 함수에 최소값이 하나 있다고 가정합니다. 이 최소값을 찾는 것은 비교적 간단합니다. 일부 초기 파라미터 값부터 시작하여 옵티마이저가 검색 방향을 연속적으로 찾고, 검색 방향에 따라 최저 비용 함수를 갖는 지점을 찾습니다. 올바른 검색 방향은 초기 파라미터 값과 옵티마이저가 방향을 계산하는 방법에 따라 달리 선택합니다. 반대로, 글로벌 최적화 문제는 비용 함수에 여러 개의 최소값이 있을 수 있다고 가정하고, 비용 함수에서 가장 작은 값을 갖는 최소값인 글로벌 최소값을 찾으려고 시도합니다. 글로벌 최적화는 수많은 가능성이 존재하므로 훨씬 어렵습니다. 이를 위해, 글로벌 옵티마이저는 일반적으로 가능성을 생성할 때 무작위 (또는 체계적) 방법을 사용합니다.

최적화 작업 사이의 차이점은 비용 함수가 1차, 때로는 2차 도함수를 가진다고 가정하는지 여부입니다. 최소값을 찾는 데 도함수를 사용하면, 특히 로컬 옵티마이저에서 최적화 시간을 상당히 단축할 수 있습니다.

도함수는 종종 수치적으로 계산됩니다. 그러나 실제로는, 모델 파라미터의 미분이 존재하지 않거나 계산이 어렵습니다. 또한, SmartSpice와 같은 시뮬레이터에 의존하여 비용 함수를 계산하는 최적화 작업의 경우, 시뮬레이터에서 발생하는 수치적 노이즈 때문에 수치적 도함수를 계산하기 어렵거나 불가능합니다. 수치적 노이즈는 시뮬레이터가 일부

내부 알고리즘 파라미터(예: 단계 크기, 반복 횟수 등)를 사용하여 수치적 솔루션에 도달한 결과입니다. 모델 파라미터를 조금만 변경하면 알고리즘 파라미터가 이러한 변화에 적응할 수 있습니다. 알고리즘 파라미터를 수정하면 “노이즈”가 일부 추가되어, 최적화에 필요한 수치적 도함수의 정확한 계산이 복잡하게 되거나 어렵게 됩니다.

이제 Utmost IV에 포함된 옵티마이저를 살펴봅니다: 표 1에 각 옵티마이저의 주요 특성이 요약되어 있습니다.

Levenberg-Marquardt (LM) [1], [2] 알고리즘과 Hooke-Jeeves (HJ) [3] 알고리즘을 사용하는 두 가지 로컬 옵티마이저가 있습니다. LM은 1차 도함수를 사용하는 로컬 최적화 알고리즘입니다. 1차 미분 행렬인 Jacobian 행렬이 비선형 모델에서 흔히 볼 수 있는 단수에 가까운 최적화일 때 특히 유용합니다. 도함수를 사용하기 때문에 상당히 빠를 수 있습니다. 모델이 간단하거나 최적화할 파라미터의 수가 적을 때, LM 옵티마이저가 우수할 수 있습니다. 다만, LM은 수치적 노이즈를 실제로 계산하기 어렵거나 불가능한 도함수가 필요하다는 단점이 있습니다. 도함수가 필요 없는 HJ 옵티마이저가 LM의 대안이 될 수 있습니다. HJ는 도함수를 사용하는 대신, 비용 함수의 감소에 더 유리한 검색 방향 패턴을 찾는 패턴 검색 알고리즘 클래스에 속합니다.

Optimizer	Local/Global	Uses derivatives	Stochastic
Levenberg-Marquardt	Local	Yes	No
Hooke-Jeeves	Local	No	No
Simulated Annealing	Global	No	Yes
Parallel Tempering	Global	No	Yes
Genetic Algorithm	Global	No	Yes
Differential Evolutions	Global	No	Yes

표 1. Utmost IV 옵티마이저 비교.

Utmost IV의 다른 네 가지 옵티마이저는 글로벌 옵티마이저로서, 검사할 파라미터 값을 생성할 때 모두 무작위 (또는 추계적) 방법을 사용합니다. 추계성에 의해 동일한 옵티마이저에서 동일한 최적화를 반복하면, 결과적으로 다른 최소값을 찾을 수 있습니다. 첫 번째 시도가 실패하더라도 두 번째 시도가 성공할 수 있으므로 이는 무작위의 장점입니다.

SA (Simulated Annealing) 알고리즘[4]은 특정 온도에서 평형 상태에 있는 물리적 시스템을 모방합니다. 비용 함수는 에너지를 대체하며, "온도"는 최적화에서 제어 파라미터일 뿐입니다. "온도"가 높을수록 비용이 많이 드는 파라미터 값을 찾을 가능성이 높습니다. 이는 보다 광범위한 검색을 할 수 있으므로 최적화의 초기 단계에 유리합니다. 최적화 진행에 따라 시스템은 안정되고, 검색은 글로벌 최소값의 주변으로 초점을 맞추게 됩니다.

SA는 강력한 알고리즘으로, 파라미터가 많은 고도의 비선형 모델에 매우 강점이 있습니다. 다만, SA는 때때로 느릴 수 있다는 단점이 있습니다. PT(Parallel Tempering) 알고리즘[5]은 SA와 유사하지만 점진적으로 온도를 낮추는 대신, 여러 시스템 사본을 상이한 온도로 유지합니다. 가끔 이 복사본을 교환하는데, 이는 로컬 최소값으로부터 시스템을 분리하여 융합하는 데 유용합니다. PT는 더 간단한 SA 알고리즘을 능가할 수 있는 로컬 최소값들이 많을 때 특히 유용합니다.

마지막 두 개의 옵티마이저는 진화 알고리즘에 속합니다. 물리적 시스템을 모방하는 SA, PT와 달리, 진화 알고리즘은 생물학적 시스템의 진화를 모방합니다. 이 알고리즘 클래스에는 최적화 문제에 대한 여러 솔루션("염색체")을 포함하는 "개체군"이 있습니다. 각 "세대"마다 이 개체군은 선택의 과정을 거칩니다. 여기서, (비용 함수와 밀접하게 관련있는) 불량한 "적합성" 때문에 구성원 중 일부가 탈락하고, 살아남은 개체는 "짝"이 되어 "변이" 과정을 거칩니다. 짝을 이룰 때, 선택된 두 구성원 (부모)은 하나 이상의 "자식"으로 결합합니다. 이렇게 하면 글로벌 최적화는 일부 로컬 최소값에 고착되지 않고, 계속해서 글로벌 최적에 대해 검색합니다. 변이에서, 각 구성원의 개별 요소가 수정됩니다. 최적화 과정에서, 변이를 통해 최소값에 가까운 해답이 해당 최소값에 계속 수렴합니다. 진화 알고리즘은 추계적입니다. 선택, 짝 및 변이는 모두 그들이 모델링하는 생물학적 시스템처럼 무작위 선택을 포함합니다.

Utmost IV에서 진화 알고리즘 클래스는 GA(Genetic Algorithm)와 DE(Differential Evolution)입니다. 최초의 유전 알고리즘이 처음으로 발명되었을 때[6], 생물학적 유사성에 기인하여 일련의 비트 또는 상이한 이산적 양을 표현하였습니다. 이는 이산(비교) 최적화에 더 적합합니다. 보다 최근에[7], 연구자들은 짝과 변이 연산자의 표현을 실제적 값을 갖는 양으로 확장하였습니다.

Utmost IV의 GA 알고리즘은 몇 가지 최신 개발 사항을 통합합니다.

유전 알고리즘과 달리, Differential Evolution[8]은 실제 값을 표현하기 위해 개발되었으므로 모델 파라미터를 최적화하는 데 적합합니다. 다른 진화 알고리즘처럼, 각 세대는 여러 가지 솔루션을 포함합니다. DE와 GA의 주요 차이점은 각 세대의 구성원을 선택하고 수정하는 방식입니다. 짝과 변이를 하는 GA와 달리, DE는 무작위로 선택한 두 개 이상의 구성원 간의 차를 우선 계산한 다음, 무작위로 선택한 다른 구성원에 추계적으로 차를 더합니다.

DE, GA는 매우 빠른 글로벌 옵티마이저가 될 수 있습니다. 그러나 속도를 빠르게 하려면 여러 알고리즘 파라미터의 튜닝이 필요할 수 있습니다. DE는 GA보다 알고리즘 파라미터에 덜 의존적입니다.

참조

- [1] Levenberg, K A method for the solution of Certain Non-linear Problems in Least Squares, *Quart. Appl. Math.* 2, 164-168, 1944.
- [2] Marquard, D An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM J. Appl. Math.* 11, 431-441, 1963.
- [3] R. Hooke, T. A. Jeeves, Direct Search Solution of Numerical and Statistical Problems, *Journal of ACM*, 8, pages 212-229, 1961.4
- [4] S. Kirkpatrick, Gelatt C. D., Vecchi M. P. Optimization by Simulated Annealing, *Science*, 220 (671-680), 1983.
- [5] C. J. Geyer, in *Computing Science and Statistics Proceedings of the 23rd Symposium on the Interface*, American Statistical Association, New York, 1991, p. 156.
- [6] J. Holland, *Adaptation In Natural and Artificial Systems*, University of Michigan Press, 1975.
- [7] A.H. Wright, Genetic Algorithms for Real Parameter Optimization, in G. Rawlins (Ed.), *Foundations of Genetic Algorithms, First Workshop on the Foundations of Genetic Algorithms and Classifier Systems*, 1991. Z. Michalewicz, G. Nazhiyath, M. Michalewicz, A Note on the Usefulness of Geometrical Crossover for Numerical Optimization Problems, *Proceedings of the 5th Annual Conference on Evolutionary Programming*, San Diego, CA, 29 February-3 March, MIT Press, Cambridge, MA, 1996.
- [8] Storn, K. Price, Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, Kluwer Academic Publishers, 11(341-359), 1997; K. Francken, G.E. Gielen, M. S. J. Steyaert, An Efficient, Fully Parasitic-aware Power Amplifier Design Optimization Tool, *IEEE Trans. On Circuits and Systems*, 52, 8, 1526-1534, 2005.