리뷰 - Learning to Learn without Grandient Descent by Grandient Descent

Created by 임 도형, last modified on Apr 22, 2018

개요

리뷰 - Learning to Learn by Gradient Descent by Gradient Descent에 이은 논문

이전 논문은 대상의 경사를 직접 사용했지만,

경사를 사용할 수 없는 문제에 대한 방법.

결론

DL의 hyper parameter 결정이 요 논문에서 말하는 black box problem의 대표적 예다.

활용할 수 있겠다.

특히 병렬 처리가 가능해서 더 유용.

방법 요약

전 논문에서는 f에서 직접 gradient를 구해서 optimizer를 학습.

여기서는 RNN 학습에 사용된 혹은 생성된 데이터를 Gaussian Process를 사용하여 함수를 구하고, 요 함수의 gradient로 optimizer를 학습

대상 f의 gradient를 사용하지 않는 것이 논문 제목의 'Without'의 의미.

방법 기타

학습 단계에서는 RNN optimizer를 학습시키기 위해 Gaussian Process가 생성한 많은 수의 미분가능한 함수를 사용.

두가지 RNN을 고려함. LSTM(Long-Short-Term Memory)와 DNC(Differentiable Neural Computer)

multi-armed bandit 문제의 AB 테스트(탐색과 활용(exploration and exploitation))를 사용하여 학습에 최대치를 둠.

개념/용어

black box optimization

x에 의한 f(x)를 볼 수 는 있지만, gradient를 알 수 없는(derivative free) 문제의 최소값 구하기

https://bbcomp.ini.rub.de/#background 참조.

time horizon

특정 처리가 끝날 것에 대한 시간적인 한계치

https://en.wikipedia.org/wiki/Time_horizon 참조.

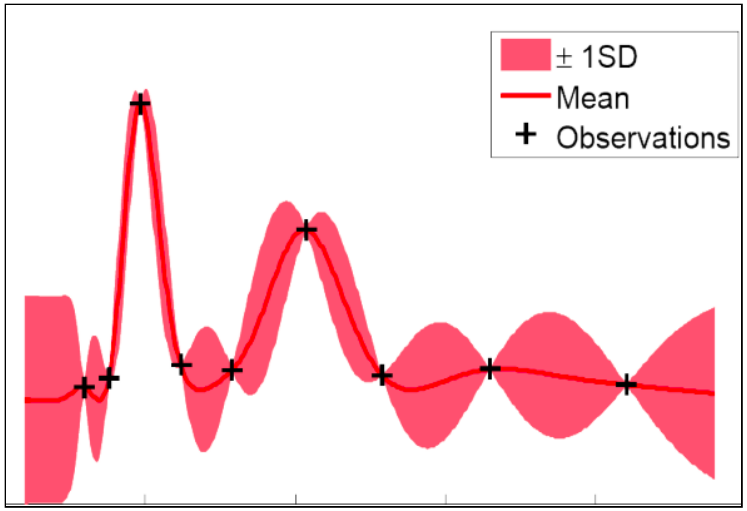
Gaussian Process

관측된 데이타 시계열 를 가지고 대상 함수를 재구성한다.

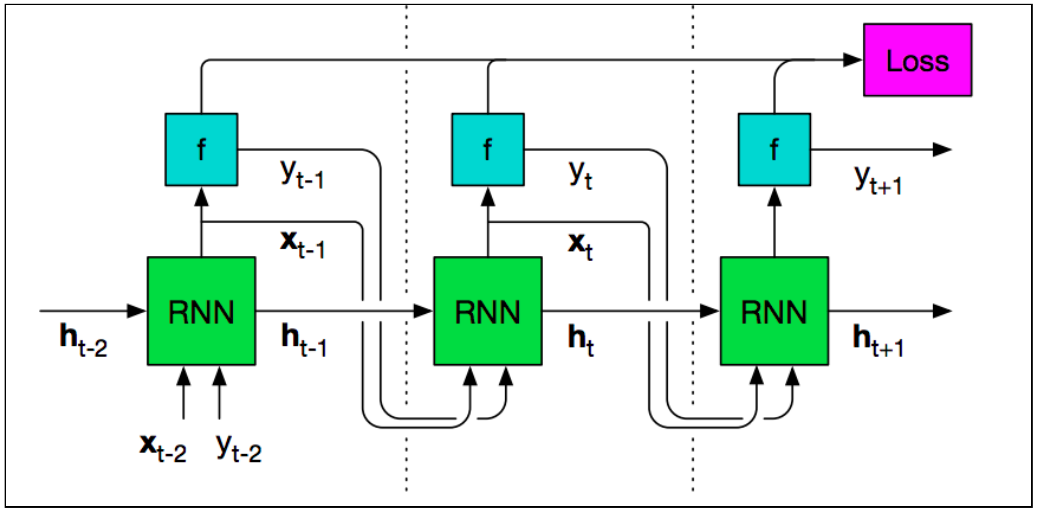
이 때 평균과 공분산으로 대상 함수를 정의할 수 있다.

Gaussian Process는 데이터에 노이즈가 있다고 전재한다.

이런 이유로 같은 데이터라 하더라도 이를 만족하는 복수개의 함수가 존재할 수 있다. 논문에서는 이점을 사용.`



학습 방법



$$\mathbf{h}_t, \mathbf{x}_t = \mathrm{RNN}_\theta(\mathbf{h}_{t-1}, \mathbf{x}_{t-1}, y_{t-1}),$$
$$y_t \sim p(y \mid \mathbf{x}_t) \ .$$

f는 우리가 풀려는 black box. 최소 결과를 내는 x를 찾는 것이 목적이다.

> ⚠ 이 방법으로 하면 f 최적화가 왜 가능한지 설명하고 있지 않다.

손실 함수

손실함수는 모든 스템의 결과의 평균.

$$L_{\text{sum}}(\theta) = \mathbb{E}_{f, y_{1:T-1}} \left[\sum_{t=1}^{T} f(\mathbf{x}_t) \right]$$

이 손실함수는 f의 최적화(f를 최소로하는 것)에 대한 것만 있고, optimizer의 탐색(exploration)을 위한 사항은 없다.

이를 위해 다음과 같은 Bayesian 최적화에에서 사용하는 EI(Expected Posterior Improvement)나 OI(Observed Improvement)를 적용.

$$L_{\text{EI}}(\theta) = -\mathbb{E}_{f, y_{1:T-1}} \left[\sum_{t=1}^{T} \text{EI}(\mathbf{x}_t \mid y_{1:t-1}) \right]$$

$$L_{\text{OI}}(\theta) = \mathbb{E}_{f, y_{1:T-1}} \left[\sum_{t=1}^{T} \min \left\{ f(\mathbf{x}_t) - \min_{i<t}(f(\mathbf{x}_i)), 0 \right\} \right]$$

함수 분포 사용

f의 기울기를 사용하지 못한다.

그래서 대신 미분할 수 있는 함수들의 분포를 사용.

그 함수들은 Gaussian Process로 구한다.

병렬 함수 계산

함수 계산을 병렬로 처리할 수 있다.

이는 Bayesian Optimizaiton의 기법덕분이라 한다.

Bayesian optimization은 데이터를 가지고 Gaussian Process에 의한 함수를 구할 때 사용된다.

복수의 함수를 구할 때 병렬적으로 구할 수 있다.

일반적 사용 가능

다양한 분양에 일반적으로 사용할 수 있다.

다음으로 실험 결과를 보임

- 합성함수 최적화
- 기존 벤치마크
- hyper parameter 튜닝

결론 요약

힘들게 해야 하는(heavily engineering)다른 기법에 비해 유한된 횟수(horizon)에 비슷한 성능.

사람손이 전혀 사용될 필요 없음. 시행착오 방법이나 hyber parameter 조정 필요 없이.

일반적인 사용 가능

다른 기법 보다 massively faster하다. 유한된 시간내에 학습해야 하는 것이 중요할 때 좋다. 특히 병렬 처리 된다.

Reference

- paper : https://arxiv.org/abs/1611.03824
- Black Box Optimization Competition : https://bbcomp.ini.rub.de
- Gaussian Process
  - A Tutorial on Gaussian Process : http://mlss2011.comp.nus.edu.sg/uploads/Site/lect1gp.pdf
  - An Introduction to Fitting Gaussian Process to Data : http://www.robots.ox.ac.uk/~seminars/seminars/Extra/2012_30_08_MichaelOsborne.pdf

No labels