Abstract

심층 신경 네트워크 모델의 새로운 유형을 소개한다. 숨겨진 계층의 이산 시퀀스를 탐색하는 대신, 우리는 신경망을 사용하여 hidden state의 파생물을 parameterize한다. 네트워크의 출력은 블랙박스 미분방정식을 사용하여 계산된다. 이러한 연속 심층 모델은 효율적인 메모리 사용이 가능하고, 각 입력에 대한 평가 전략을 채택하며 속도와 수치 정밀도를 명시적으로 교환할 수 있다. 우리는 연속 심층 잔류 네트워크 및 연속 시간 잠재 변수 모델에서 이러한 속성을 입증한다. 또한 데이터 차원을 분할하거나 정렬하지 않고 최대우도로 훈련할 수 있는 생성 모델인 연속 정규화 흐름을 구성합니다. training을 위해 내부 동작에 접근하지 않고 ODE를 통해 확장 가능한 역전파 방법을 보여준다. 이를 통해 더 큰 모델 내에서 ODE를 종단간 교육할 수 있다.

1. Introduction

Residual network, Recurrent network decoder, normalizing flow와 같은 모델들은 hidden state로의 변환 sequence를 구성함으로써 복잡한 변환을 수행한다.

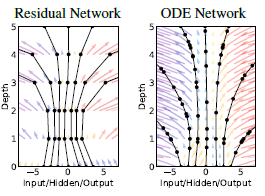
(이 때, 이고, 이다.)

이러한 반복적인 업데이트는 연속적인 변환의 오일러 이산화로 볼 수 있다.

(Euler discretization of a continuous transformation)

더 많은 레이어를 추가하고 더 작은 step을 수행하면 어떻게 될까? 극한에 도달하면, 우리는 Neural network에 의해 지정된 보통 미분 방정식(ODE)을 사용하여 숨겨진 단위의 연속 역학을 parameterize한다.

입력층 h(0)에서부터 시작해서, 우리는 어떤 시간 T에 대한 ODE 문제의 초기값 문제를 출력층 h(T)로 해결할 수 있다.

이 값은 블랙박스 미분 방정식 Solver에 의해 계산될 수 있으며, 이는 필요한 경우 원하는 정확도로 솔루션을 결정하기 위해 숨겨진 단위 역학 f를 평가한다. 그림 1은 이 두 가지 접근방식을 대조한다.

ODE solver를 이용한 모델 정의와 평가는 아래와 같은 이점을 갖는다

1. Memory efficiency
2. Adaptive computation
3. Scalable and invertible normalizing flows
4. Continuous time-series model
5. Reverse-mode automatic differentiation of ODE solutions

연속-깊이 신경망의 훈련에 있어 주요한 기술적 어려움은 ODE solver를 이용한 역전파를 수행하는 것이다. Forward pass를 통한 Differentiation은 간단하지만, 메모리 소비가 많고 수치적 오류를 발생시킬 수 있다.

우리는 ODE solver를 black box로 가정하고, adjoint sensitivity method를 이용해 gradient를 계산한다. 이 접근은 두 번째 증강된 ODE를 시간의 역순으로 해결해 gradient를 계산하며, 또한 모든 ODE solver에 적용 가능하다.

ODE solver의 결과인 스칼라 값 손실 함수 L()의 최적화가 필요하다.

L의 최적화를 위해, θ와 관련한 gradient가 필요하다. 첫 번째 단계는 각 순간의 숨겨진 상태인 z(t)에 따라 손실의 gradient가 어떻게 달라지는지를 결정하는 것이다. Gradient의 변화는 로 설명할 수 있다. 그 역학은 또 다른 ODE에 의해 주어지는데, 이는 체인 규칙의 순간 아날로그로 볼 수 있다.

(식 4)

우리는 다른 ODE solver로부터 를 계산할 수 있다. 해당 solver는 반드시 backward 방향으로 동작해야 하며, 초기값 로 시작해야 한다. 한 가지 문제는 해당 ODE를 풀기 위해서는 전체 궤도를 따라 z(t)의 알려진 값이 필요하다는 것이다. 그러나 최종 값 h(t\_1)에서 시작하여 인접 값과 함께 시간 역순으로 간단히 h(t)를 계산할 수 있다.

Parameter θ에 대한 gradient를 계산하려면 z(t)와 a(t) 모두에 따라 달라지는 세 번째 적분을 평가해야 한다.

(식 5)

Vector-Jacobian 곱인 식 4의 와 식 5의 는 f를 평가하는 것과 비슷한 시간 비용으로, 자동 미분에 의해 효율적으로 평가될 수 있다. Z와 a, 그리고 를 풀기위한 모든 적분은 하나의 ODE solver에 의해 계산될 수 있는데, 이 ODE solver는 original state와 인접 상태 및 다른 편미분들을 단일 벡터로 합친다. Algorithm 1은 필요한 역학을 구성하고 ODE solver를 호출하여 모든 gradient를 한 번에 계산하는 방법을 보여준다.

|  |
| --- |
| Algoritm 1 : Reverse-mode derivative of an ODE initial value problem |
| Input : dynamics parameter θ start time t\_0, stop time t\_1final state z(t\_1), loss gradient  : initial augmented state 정의  Def aug\_dynamics([z(t), a(t), ∙], t, θ) : augmented state의 dynamics 정의  Return : Vector-Jacobian product 계산  : reverse-time ODE 계산  Return : Gradient 반환 |

대부분의 ODE solver는 상태 z(t)를 여러 번 출력하는 옵션이 있다. Loss가 이러한 초기 값에 종속되었을 때, reverse-mode 미분은 각 연이은 출력 쌍 사이에 하나 씩 구분된 solve의 sequence로 나누어져야한다. 각 관측치에 대해, 수반행렬은 대응하는 편미분 의 방향으로 조정되어야 한다.