Intern Project

Intern Name

Abstract

Contents

1 Preliminary **2**

2 Second-order Algorithm **2**

1 Preliminary

2 Second-order Algorithm

Assume we define a field with $x_t = \alpha_t x_0 + \beta_t x_1$, we can manually calculate $\dot{x}_t$ and $\ddot{x}$, representing the first-order and second-order gradient, respectively.

Definition 2.1. *The loss function for second order method contain two parts. We define the first part which is trying to using $\dot{x}_t$, x_t and t to learn function $u_{1,t}$, thus the loss is*

$$L_{2,1,\theta_1} := \|\dot{x}_t - u_{1,\theta_1}(x_t, t)\|_2^2$$

Next, we define the second part which is trying to use $\ddot{x}_t, u_{1,\theta_1}(x_t, t)$, x_t and t to learn u_{2,θ_2} function, thus the loss is

$$L_{2,2,\theta_2,\theta_1} := \|\ddot{x}_t - u_{2,\theta_2}(u_{1,\theta_1}(x_t, t), x_t, t)\|_2^2$$

Overall the total loss is

$$L_{2,\theta} := L_{2,1,\theta_1,\theta_2} + L_{2,2,\theta_2}$$

Algorithm 1

procedure OurSecondOrderAlg()
  for each iteration **do**
    Random sample x_0 and time t, with target x_1
    $x_t \leftarrow \alpha_t \cdot x_0 + \sqrt{1 - \alpha_t^2} \cdot x_1$
    Compute gradient with respect to $L_{2,\theta}$ ▷ see Def. 2.1
  end for
  return u_1, u_2 ▷ Two network functions
end procedure
/* Below is an inference algorithm that only use first-order learner u_1*/
procedure Inf1(u_1)
  $x_0 \sim \mathcal{N}(0, 1)$
  Initial $x \leftarrow x_0$
  for t from 0 to 1 with step $\Delta t = 0.01$ **do**
    $x \leftarrow x + \Delta t \cdot u_1(x, t)$
  end for
  return x
end procedure

References

Algorithm 2

/* Option 1 */
procedure $\textsc{Inf12}(u_1, u_2)$
    $x_0 \sim \mathcal{N}(0, 1)$
    Initial $x \leftarrow x_0$
    for t from 0 to 1 with step $\Delta t_1 = 0.01$ **do**
        $y \sim \mathcal{N}(0, 1)$
        for t from 0 to 1 with step $\Delta t_2 = 0.01$ **do**
            $y \leftarrow y + \Delta t_2 \cdot u_2(y, x, t)$
        end for
        $x \leftarrow x + \Delta t_1 \cdot u_1(y, t)$
    end for
    return x
end procedure

Algorithm 3

/* Option 2 */
procedure $\textsc{InfXX}(u_1, u_2)$
    $x_0 \sim \mathcal{N}(0, 1)$
    Initial $x \leftarrow x_0$
    $t \leftarrow 0$
    $\Delta t \leftarrow 0$
    while $t < 1$ **do**
        $\Delta u_1 \leftarrow \Delta_t \cdot u_2(u_1(x, t), x, t)$
        $u_1 \leftarrow u_1(x, t) + \Delta u_1$
        $x \leftarrow x + \Delta t \cdot u_1$
    end while
    return x
end procedure

Algorithm 4

/* Option 3 */
procedure $\textsc{Inf12}(u_1, u_2)$
    $x_0 \sim \mathcal{N}(0, 1)$
    Initial $x \leftarrow x_0$
    for t from 0 to 1 with step $\Delta t = 0.01$ **do**
        $x \leftarrow x + \Delta t \cdot u_1(x, t) + \frac{(\Delta t)^2}{2} \cdot u_2(u_1(x, t), x, t)$
    end for
    return x
end procedure