An Odyssey in Hamilton-Jacobi Equations

Hopf-Lax Formula, Numerical Algorithms, and Link to Deep Learning

Yixuan Wang

Peking University

December 30, 2017

Section 1

Mathematical Background

Definition

Hamilton-Jacobi Equation

$$\frac{\partial \varphi}{\partial t} + H(x, p, t) = 0 \qquad in\ \mathbb{R}^d \times (0, \infty) \tag{1}$$

$$\varphi(x, 0) = g(x) \qquad in\ \mathbb{R}^d \tag{2}$$

where $x \in \mathbb{R}^d$ denotes the state coordinate and $t \in \mathbb{R}$ denotes the time coordinate; $H : \mathbb{R}^d \times \mathbb{R}^d \times (0, \infty) \to \mathbb{R}$ is a prescribed function called the Hamiltonian; $\varphi := \varphi(x, t) : \mathbb{R}^d \times (0, \infty) \to \mathbb{R}$ is our target solution for the Hamilton-Jacobi Equation; $p := \nabla_x \varphi$ denotes the gradient vector with respect to x; $g(x)$ is given as the initial data.

Viscosity Solution
Motivation

We assume the Hamiltonian has the form of $H(p,x)$.
The original Hamilton-Jacobi equation can often be a fully nonlinear first-order PDE, so it is difficult to tackle. In the method of vanishing viscosity, we introduce a second-order term for regularization, converting it into a semilinear parabolic PDE as follows

$$\frac{\partial \varphi}{\partial t} + H(p,x) - \varepsilon \Delta_x \varphi = 0 \qquad in\ \mathbb{R}^d \times (0,\infty) \tag{3}$$

where $\Delta_x \varphi$ denotes the Laplacian with respect to x, ε is a constant and we denote the solution by φ^ε. As $\varepsilon \to 0$, we hope φ^ε would converge to our weak solution φ, or at least in terms of a subsequence as Arzela-Ascoli theorem would imply.

Viscosity Solution

Formulation

Viscosity Solution

Assume H, g are continuous. A bounded function u, which is uniformly continuous for each $T > 0$ in $\mathbb{R}^d \times [0, T]$, is a viscosity solution provided that: (1) $u(x, 0) = g(x)\ in\ \mathbb{R}^d$; (2) for all $v \in C^\infty(\mathbb{R}^d \times (0, \infty))$, if $u - v$ has a local maximum at (x_0, t_0), then $v_t(x_0, t_0) + H(\nabla_x v(x_0, t_0), x_0) \leq 0$; (3) $v_t(x_0, t_0) + H(\nabla_x v(x_0, t_0), x_0) \geq 0$ for a local minimum at (x_0, t_0).

It can be verified that if u is constructed using the method of vanishing viscosity, it indeed satisfies the previous condition.

Viscosity Solution

Consistency

1. A classical solution is clearly a viscosity solution.

Viscosity Solution

Consistency

1. A classical solution is clearly a viscosity solution.
2. If a viscosity solution u is differentiable at (x_0, t_0), then $u_t(x_0, t_0) + H(\nabla_x u(x_0, t_0), x_0) = 0$.

Viscosity Solution

Uniqueness

Thm (Uniqueness)

Suppose H enjoys the Lipschitz continuity

$$\begin{cases} |H(p,x)-H(q,x)| \leq C|p-q|, \\ |H(p,x)-H(p,y)| \leq C|x-y|(1+|p|). \end{cases} \tag{4}$$

Then there is at most one viscosity solution for the Hamilton-Jacobi equation

$$\begin{cases} \frac{\partial\varphi}{\partial t} + H(p,x) = 0 \qquad in\ \mathbb{R}^d \times (0,T], \\ \varphi(x,0) = g(x) \qquad in\ \mathbb{R}^d. \end{cases} \tag{5}$$

Intro to Control Theory

We have the following optimal control problem

$$\begin{cases} \dot{x}(t) = f(x(t), \alpha(t)) \qquad t > 0, \\ x(0) = x^0. \end{cases} \tag{6}$$

where α ($\alpha(t) \in A$) denotes a control from an admissible set $\mathcal{A}$; x denotes the response to the control according to our ODE. Now we wish to maximize the following payoff functional

$$P[\alpha(\cdot)] := \int_0^T r(x(t), \alpha(t))dt + g(x(T)) \tag{7}$$

where r and g are given as the running payoff and the terminal payoff respectively; the terminal time $T > 0$ is given as well.

Dynamic Programming

Motivation

When evaluating the integral

$$\int_0^\infty \frac{\sin x}{x} dx$$

We define

$$I(\alpha) := \int_0^\infty e^{-\alpha x} \frac{\sin x}{x} dx$$

Now since we can compute

$$I'(\alpha) = -\frac{1}{\alpha^2 + 1}$$

We can get that our integral equals $\frac{\pi}{2}$ since $I(\infty)$ equals 0.

Dynamic Programming

Perspective

Embed the optimal control problem into a larger family of similar problems, namely we vary the initial state and time of the controlled dynamics

$$\begin{cases} \dot{x}(t) = f(x(t), \alpha(t)) & s < t \leq T, \\ x(s) = x. \end{cases} \tag{8}$$

with the target payoff functional

$$P_{x,s}[\alpha(\cdot)] := \int_s^T r(x(t), \alpha(t))dt + g(x(T)) \tag{9}$$

Now we define the value function v to be the greatest payoff starting at a given state and time

$$v(x,t) := \sup_{\alpha(\cdot)\in\mathcal{A}} P_{x,t}[\alpha(\cdot)] \tag{10}$$

Note that $v(x,T) = g(x)$.

Dynamic Programming

Property of the Value function

Thm (Optimality Conditions)

For each $h > 0$ $s.t$ $t + h \leq T$, *we have*

$$v(x,t) = \sup_{\alpha(\cdot)\in\mathcal{A}} \{\int_t^{t+h} r(x(s),\alpha(s))ds + v(x(t+h),t+h)\} \tag{11}$$

where $x(\cdot)$ *solves the ODE for the control* $\alpha(\cdot)$.

Dynamic Programming

Derivation of the Equation

Thm (Hamilton-Jacobi-Bellman Equation)

Assume the value function v is C^1. Then v solves the PDE

$$v_t(x,t) + \max_{a \in A} \{f(x,a) \cdot \nabla_x v(x,t) + r(x,a)\} = 0 \tag{12}$$

Now we can define the Hamiltonian

$$H(x,p) := \max_{a \in A} H(x,p,a) := \max_{a \in A} \{f(x,a) \cdot p + r(x,a)\} \tag{13}$$

Dynamic Programming

Application

Now we discuss how to solve the optimal control problem using the idea of dynamic programming.

1. Firstly we solve the Hamilton-Jacobi-Bellman equation to compute value function v.

Dynamic Programming

Application

Now we discuss how to solve the optimal control problem using the idea of dynamic programming.

1. Firstly we solve the Hamilton-Jacobi-Bellman equation to compute value function v.
2. Then we use the value function to design α at each point, according to $\alpha(x,t) = a$, for the maximum in Hamiltonian to be attained.

Dynamic Programming

Application

Now we discuss how to solve the optimal control problem using the idea of dynamic programming.

1. Firstly we solve the Hamilton-Jacobi-Bellman equation to compute value function v.
2. Then we use the value function to design α at each point, according to $\alpha(x,t) = a$, for the maximum in Hamiltonian to be attained.
3. Next we solve the control system of ODE, now that α can be expressed as a function of x and t.

Dynamic Programming

Application

Now we discuss how to solve the optimal control problem using the idea of dynamic programming.

1. Firstly we solve the Hamilton-Jacobi-Bellman equation to compute value function v.
2. Then we use the value function to design α at each point, according to $\alpha(x, t) = a$, for the maximum in Hamiltonian to be attained.
3. Next we solve the control system of ODE, now that α can be expressed as a function of x and t.
4. Finally we define the feedback control $\alpha(t) := \alpha(x(t), t)$.

Pontryagin Maximum Principle

Statement of the Theorem

Thm (Pontryagin Maximum Principle)

Assume $\alpha(\cdot)$ is optimal for our control problem, and $x(\cdot)$ is the corresponding trajectory. Then there exists a function $p : [0, T] \to \mathbb{R}^d$ such that

$$\begin{cases} (ODE) & \dot{x}(t) = \nabla_p H(x(t), p(t), \alpha(t)), \\ (I) & x(0) = x^0, \\ (ADJ) & \dot{p}(t) = -\nabla_x H(x(t), p(t), \alpha(t)), \\ (T) & p(T) = \nabla g(x(T)), \\ (C) & \textit{the mapping} \quad t \mapsto H(x(t), p(t), \alpha(t)) \quad \textit{is constant}, \\ (M) & H(x(t), p(t), \alpha(t)) = \max_{a \in A} H(x(t), p(t), a). \end{cases} \tag{14}$$

Pontryagin Maximum Principle

Connection with Dynamic Programming

If the value function defined in the dynamic programming process is C^2, then the costate $p(\cdot)$ in the Pontryagin Maximum Principle is given by

$$p(s) = \nabla_x v(x(s), s) \qquad (t \leq s \leq T) \tag{15}$$

Pontryagin Maximum Principle

Application

Now we discuss how to solve the optimal control problem using the Pontryagin Maximum Principle.

1. We write out those PDE equations and solve $x(\cdot)$, $\alpha(\cdot)$, $p(\cdot)$ simultaneously.
2. We often utilize the maximum equation (M) to compute the control $\alpha(\cdot)$.

Miscellaneous Complements

1. We can utilize dynamic programming method to derive Hamilton-Jacobi equations for solving differential games.

Miscellaneous Complements

1. We can utilize dynamic programming method to derive Hamilton-Jacobi equations for solving differential games.
2. Without assuming value function $\in C^1$, we still have that v is the unique viscosity solution to our Hamilton-Jacobi-Bellman Equation, provided that g, r, f are bounded and Lipschitz continuous. In this way, we can formally obtain our value function by solving the PDE, since viscosity solution is unique.

Section 2

Hopf-Lax Type Formula

Characteristic Equations

General Review

ColoredThm (Structure of Characteristic ODE)

For a nonlinear first-order PDE $F(Du, u, x) = 0$, where F is smooth, we have the following equations

$$\begin{cases} (a)\ \dot{p}(s) = -D_x F(p(s), z(s), x(s)) - D_z F(p(s), z(s), x(s)) p(s), \\ (b)\ \dot{z}(s) = D_p F(p(s), z(s), x(s)) \cdot p(s), \\ (c)\ \dot{x}(s) = D_p F(p(s), z(s), x(s)). \end{cases} \tag{16}$$

Assume a C^2 function u solves the original PDE, and $\dot{x}(\cdot)$ solves the ODE (c), where $p(\cdot) := Du(x(\cdot))$, $z(\cdot) := u(x(\cdot))$, then $p(\cdot)$ solves (a) and $z(\cdot)$ solves (b).

Characteristic Equations

For Hamilton-Jacobi Equation

For our Hamilton-Jacobi Equation $u_t + H(Du, x) = 0$, the characteristic equations become

$$\begin{cases} \dot{x} = D_pH(p,x), \\ \dot{p} = -D_xH(p,x), \\ \dot{z} = D_pH(p,x) \cdot p + H(p,x). \end{cases} \tag{17}$$

Legendre Transform

Definition

Convex Conjugate

For a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$, we define its convex conjugate f^* $: \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ as follows

$$f^*(p) := \sup\{\langle p, x\rangle - f(x) | x \in \mathbb{R}^d\} \tag{18}$$

f^* is also called the Legendre-Fenchel transformation of f.

If L is convex, and $\lim_{|v|\to\infty} \frac{L(v)}{|v|} = +\infty$. Then for $H = L^*$, H is convex, and $\lim_{|p|\to\infty} \frac{H(p)}{|p|} = +\infty$. Moreover, $L = H^*$.

Legendre Transform

For Hamiltonian

Suppose now Hamiltonian $H = H(Du)$, the mapping H is convex, and $\lim_{|p|\to\infty} \frac{H(p)}{|p|} = +\infty$. $L = H^*$.
We notice that $L^*(p) = p \cdot v - L(v)$ implies the derivative of $r.h.s.$ $w.r.t.$ v equals 0, since it reaches a maximum at v. Hence $DL(v) = p$.
Now we have that the following equations are equivalent

$$\begin{cases} v = DH(p), \\ p = DL(v), \\ p \cdot v = L(v) + H(p). \end{cases} \tag{19}$$

Hopf-Lax Formula

Formulation

Now let's return to our problem of Hamilton-Jacobi equation. The method of characteristics implies $\dot{x} = DH(p)$, and $\dot{z} = DH(p) \cdot p - H(p)$. As a result, $\dot{z} = L(\dot{x})$, which provides a clue for our following definition.

Hopf-Lax Formula

Assume additionally that the initial data $g(\cdot)$ is Lipschitz continuous, we define $u(x,t)$ as follows

$$u(x,t) := \inf\{\int_0^t L(\dot{w}(s))ds + g(w(0)) | w(t) = x\} \tag{20}$$

Hopf-Lax Formula

Properties

1. u is Lipschitz continuous, thus differentiable $a.e.$.

Hopf-Lax Formula

Properties

1. u is Lipschitz continuous, thus differentiable $a.e.$.
2. $u(x, 0) = g(x)$.

Hopf-Lax Formula

Properties

1. u is Lipschitz continuous, thus differentiable $a.e.$.
2. $u(x, 0) = g(x)$.
3. $u(x, t) = \min_{y \in \mathbb{R}^d}\{tL(\frac{x-y}{t}) + g(y)\}$.

Hopf-Lax Formula

Properties

1. u is Lipschitz continuous, thus differentiable $a.e.$.
2. $u(x,0) = g(x)$.
3. $u(x,t) = \min_{y\in\mathbb{R}^d}\{tL(\frac{x-y}{t}) + g(y)\}$.
4. $u(x,t) = \min_{y\in\mathbb{R}^d}\{(t-s)L(\frac{x-y}{t-s}) + u(y,s)\} \qquad for\ 0 \le s < t$.

Hopf-Lax Formula

Properties

1. u is Lipschitz continuous, thus differentiable $a.e.$.
2. $u(x, 0) = g(x)$.
3. $u(x, t) = \min_{y \in \mathbb{R}^d}\{tL(\frac{x-y}{t}) + g(y)\}$.
4. $u(x, t) = \min_{y \in \mathbb{R}^d}\{(t-s)L(\frac{x-y}{t-s}) + u(y, s)\} \qquad for\ 0 \leq s < t.$
5. u satisfies the Hamilton-Jacobi equation at points where it is differentiable.

Hopf-Lax Formula

Uniqueness

The uniqueness of a weak solution can be guaranteed if we impose a semi-concavity restriction, which is indeed satisfied by our Hopf-Lax formula provided that g is semiconcave or H is uniformly convex.

Hopf-Lax Formula

Uniqueness

Thm (Semi-concavity)

If there exists a constant C*, s.t* $g(x+z) - 2g(x) + g(x-z) \le c|z|^2$,
then $u(x+z,t) - 2u(x,t) + u(x-z,t) \le c|z|^2$,
Or, if there exists a constant $\theta > 0$*, s.t.*
$\sum_{i,j=1}^{n} H_{p_i p_j}(p)\xi_i\xi_j \ge \theta|\xi|^2 \qquad for\ all\ p, \xi \in \mathbb{R}^d$,
then $u(x+z,t) - 2u(x,t) + u(x-z,t) \le \frac{1}{\theta t}|z|^2$.

Hopf-Lax Formula

Uniqueness

Thm (Semi-concavity)

If u solves the initial value problem $a.e.$, and $u(x+z,t) - 2u(x,t) + u(x-z,t) \leq C(1+\frac{1}{t})|z|^2$ for some constant C, then u is unique.

Additional Comment

1. Hopf-Lax formula gives us a viscosity solution.

Additional Comment

1. Hopf-Lax formula gives us a viscosity solution.
2. We can view Hopf-Lax formula as a special case of dynamic programming.

Time-dependent Case

Formulation

Consider the Hamilton-Jacobi equation

$$\frac{\partial \varphi}{\partial t} + H(p,t) = 0 \qquad in\ \mathbb{R}^d \times (0,T) \tag{21}$$

$$\varphi(x,T) = g(x) \qquad in\ \mathbb{R}^d \tag{22}$$

where g is convex.

Note that we are considering Hamilton-Jacobi equation given the terminal state for the sake of consistency with reference materials, reversing it back in time would give us a Hamilton-Jacobi equation we originally considered.

Time-dependent Case

Assumption

Define $S = \{s \in \mathbb{R}^d : |s| = 1\}$,
$$B_+ = \{(s, r) \in \mathbb{R}^d \times \mathbb{R} : |s|^2 + r^2 \leq 1, r > 0\}.$$
We have the following assumptions on Hamiltonian

1. H is continuous in $t \in (0, T)$ for every $s \in \mathbb{R}^d$.

Time-dependent Case

Assumption

Define $S = \{s \in \mathbb{R}^d : |s| = 1\}$,

$$B_+ = \{(s, r) \in \mathbb{R}^d \times \mathbb{R} : |s|^2 + r^2 \leq 1, r > 0\}.$$

We have the following assumptions on Hamiltonian

1. H is continuous in $t \in (0, T)$ for every $s \in \mathbb{R}^d$.
2. H is summable on $(0, T)$ for every $s \in \mathbb{R}^d$.

Time-dependent Case

Assumption

Define $S = \{s \in \mathbb{R}^d : |s| = 1\}$,
$B_+ = \{(s, r) \in \mathbb{R}^d \times \mathbb{R} : |s|^2 + r^2 \le 1, r > 0\}$.

We have the following assumptions on Hamiltonian

1. H is continuous in $t \in (0, T)$ for every $s \in \mathbb{R}^d$.
2. H is summable on $(0, T)$ for every $s \in \mathbb{R}^d$.
3. For all $(t, s) \in (0, T) \times S$, $\lim_{r \to 0^+} rH(\frac{s}{r}, t) = H_0(s, t)$ exists and $H_0(s, \cdot)$ is continuous on $(0, T)$ for every $s \in S$.

Time-dependent Case

Assumption

Define $S = \{s \in \mathbb{R}^d : |s| = 1\}$,
$B_+ = \{(s, r) \in \mathbb{R}^d \times \mathbb{R} : |s|^2 + r^2 \leq 1, r > 0\}$.
We have the following assumptions on Hamiltonian

1. H is continuous in $t \in (0, T)$ for every $s \in \mathbb{R}^d$.
2. H is summable on $(0, T)$ for every $s \in \mathbb{R}^d$.
3. For all $(t, s) \in (0, T) \times S$, $\lim_{r \to 0^+} rH(\frac{s}{r}, t) = H_0(s, t)$ exists and $H_0(s, \cdot)$ is continuous on $(0, T)$ for every $s \in S$.
4. For all $t \in (0, T)$, $(s_1, r_1), (s_2, r_2) \in B_+$, and for some constant L, $|r_1 H(\frac{s_1}{r_1}, t) - r_2 H(\frac{s_2}{r_2}, t)| \leq L(|s_1 - s_2|^2 + (r_1 - r_2)^2)^{\frac{1}{2}}$.

Time-dependent Case

Hopf-Lax Formula

We give the following theorem when the Hamiltonian is dependent on time t, without formally state the exact definition of minimax solutions.

Thm (Hopf-Lax Formula in Time-dependent Case)

For mild assumptions on terminal data g and Hamiltonian H, we have

$$v(x,t) = \sup_{s \in \mathbb{R}^d} \{\langle s, x\rangle + \int_t^T H(s,r)dr - g^*(s)\} \tag{23}$$

the formula above gives a minimax solution.

Namely, $v(x,t) = \left(g^*(s) - \int_t^T H(s,r)dr\right)^*(x)$.

Time-dependent Case

Further assumptions

The "mild assumptions" in the statement of the theorem could be

1. $H(\cdot, t)$ is convex (or concave) in s for every t.

Time-dependent Case

Further assumptions

The "mild assumptions" in the statement of the theorem could be

1. $H(\cdot, t)$ is convex (or concave) in s for every t.
2. The maximizer in our formula for v is unique for all (s, t). As is a special case when g is an affine function.

Function-dependent Case

Formulation

Consider the Hamilton-Jacobi equation

$$\frac{\partial V}{\partial t} + H(t, V, p) = 0 \qquad in\ \mathbb{R}^d \times (0, T) \tag{24}$$

$$V(T, x) = \varphi(x) \qquad in\ \mathbb{R}^d \tag{25}$$

We would not go into the details of the Hopf-Lax formula for this type of equation, see the reference material for more information.

Additional Comment

1. minimax solutions and viscosity solutions are in fact equivalent.

Additional Comment

1. minimax solutions and viscosity solutions are in fact equivalent.
2. The above Hopf-Lax Formulas are natural generalizations of the prototype.

Section 3

Algorithms for Numerical Computation

Comments in Advance

1. Let's look at the optimality condition. In this situation, the conjecture is a natural generalization of Hopf-Lax formula using the idea of dynamic programming.

Comments in Advance

1. Let's look at the optimality condition. In this situation, the conjecture is a natural generalization of Hopf-Lax formula using the idea of dynamic programming.
2. The formulas coincide with the time-dependent Hopf-Lax formula when the Hamiltonian is independent of the current state.

Lax Type Conjecture

Minimization principle (Lax Formula) when $H(x,p,t)$ is smooth and convex w.r.t. p and possibly under some further mild assumptions:

$$\varphi(x,t) = \min_{v\in\mathbb{R}^d}\left\{g(\gamma(v,0)) + \int_0^t \{\langle p(v,s), \partial_p H(\gamma(v,s),p(v,s),s)\rangle - H(\gamma(v,s),p(v,\overset{s}{x}),s)\}\,ds : \begin{array}{c} \dot\gamma(v,s) = \partial_p H(\gamma(v,s),p(v,s),s), \\ \dot p(v,s) = -\partial_x H(\gamma(v,s),p(v,s),s), \\ \gamma(v,t) = x,\ p(v,t) = v \end{array}\right\}$$

and its discrete approximation given a small δ,

$$\varphi(x,t) \approx \min_{v\in\mathbb{R}^d}\left\{g(x_0(v)) + \delta\sum_{n=1}^{N-1}\{\langle p_n(v), \partial_p H(x_n(v),p_n(v),t_n)\rangle - H(x_n(v),p_n(v),t_n)\} : \begin{array}{c} x_{n+1}(v) - x_n(v) = \delta\partial_p H(x_n(v),p_n(v),t_n), \\ p_{n-1}(v) - p_n(v) = \delta\partial_x H(x_n(v),p_n(v),t_n), \\ x_N = x,\ p_N = v \end{array}\right\}$$

Lax Type Conjecture

Allowing a more general case when H is non-smooth w.r.t. p, we postulate the following minimization principle. In what follows, we denote $\partial_x^- f(x)$ as the (regularized) subdifferential of f for a given f.

Conjecture 3.1. *When $H(x,p,t)$ is smooth w.r.t. x and convex w.r.t. p (and perhaps under some other mild conditions on $H(x,p,t)$ and $g(p)$), the viscosity solution to (2.1)-(2.2) can be represented as*

$$\varphi(x,t) = \inf_{v\in\mathbb{R}^d} \inf_{\gamma\in C^\infty} \Big\{ g(\gamma(v,0)) + \int_0^t \{\langle p(v,s), \dot\gamma(v,s)\rangle - H(\gamma(v,s), p(v,x), s)\}\, ds : \\ \left.\begin{array}{c} \dot\gamma(v,s) \in \partial_p^- H(\gamma(v,s), p(v,s), s), \\ \dot p(v,s) = -\partial_x H(\gamma(v,s), p(v,s), s), \\ \gamma(v,t) = x, p(v,t) = v \end{array}\right\} \tag{3.1}$$

for small time t. In here we always use the convention that the infrimum of an empty set is minus infinity, $\inf\emptyset = -\infty$. If furthermroe that ϕ is differentiable at a neighbourhood of (x,t), the minimum argument in the above formula shall coincide with $\partial_x\varphi(x,t)$.

Hopf Type Conjecture

Maximization principle (Hopf Formula) when $H(x,p,t)$ is smooth and $g(p)$ is convex w.r.t. p and possibly under some further mild assumptions:

$$\varphi(x,t) = \sup_{v\in\mathbb{R}^d}\left\{\langle x,v\rangle - g^*(p(v,0)) - \int_0^t \left\{H(\gamma(v,s),p(v,s),s) - \langle\partial_x H(\gamma(v,s),p(v,s),s),\gamma(v,s)\rangle\right\}ds : \begin{array}{c}\dot{\gamma}(v,s) = \partial_p H(\gamma(v,s),p(v,s),s),\\ \dot{p}(v,s) = -\partial_x H(\gamma(v,s),p(v,s),s),\\ \gamma(v,t)=x,\ p(v,t)=v\end{array}\right\}$$

and its discrete approximation given a small δ

$$\varphi(x,t) \approx \max_{v\in\mathbb{R}^d}\left\{\langle x_N,v_N\rangle - g^*(p_0(v)) - \delta\sum_{n=1}^{N-1}\{H(x_n(v),p_n(v),t_n) - \langle x_n(v),\partial_x H(x_n(v),p_n(v),t_n)\rangle\} : \begin{array}{c}x_{n+1}-x_n = \delta\partial_p H(x_n(v),p_n(v),t_n),\\ p_{n-1}-p_n = \delta\partial_x H(x_n(v),p_n(v),t_n),\\ x_N = x,\ p_N = v\end{array}\right\}$$

Hopf Type Conjecture

Conjecture 3.2. *When $H(x,p,t)$ is smooth w.r.t. x, and $g(p)$ is convex w.r.t. p (and perhaps under some other mild conditions on $H(x,p,t)$ and $g(p)$), the viscosity solution to (2.1)-(2.2) can be represented as*

$$\varphi(x,t) = -\inf_{v\in\mathbb{R}^d}\sup_{\gamma\in C^\infty}\left\{g^*(p(v,0)) + \int_0^t \left\{H(\gamma(v,s),p(v,s),s) + \langle \dot{p}(v,s),\gamma(v,s)\rangle\right\}ds - \langle x,v\rangle : \left.\begin{array}{c}\dot{\gamma}(v,s)\in\partial_p^+ H(\gamma(v,s),p(v,s),s),\\ \dot{p}(v,s) = -\partial_x H(\gamma(v,s),p(v,s),s),\\ \gamma(v,t)=x,\ ,p(v,t)=v\\ p(v,0)\in\partial_y^- g(\gamma(v,0))\end{array}\right\}\right. \tag{3.4}$$

for small time t (at least) such that the differential $\partial_v p(0)$ is a non-singular matrix. Such a mild conditon might be some convexity assumption of $H(x,p,t)$ w.r.t. the convex hull of the set of minimizers in the variable p. (see [30] for predicting this technical assumption) . In here we again always use the convention that the infrimum of an empty set is minus infinity $\inf\emptyset = -\infty$. If furthermroe that ϕ is differentiable at a neighbourhood of (x,t), the maximum argument in the above formula shall coincide with $\partial_x\varphi(x,t)$.

Objective function

We wish to minimize the following functions (Lax and Hopf type conjecture respectively)

$$\mathcal{F}^1_{x,t}(v) := g(\gamma(v,0)) + \int_0^t \{\langle p(v,s), \partial_p H(\gamma(v,s), p(v,s), s)\rangle - H(\gamma(v,s), p(v,s), s)\}\, ds$$

$$\mathcal{G}_{x,t}(v) := g^*(p(v,0)) + \int_0^t \Big\{ H(\gamma(v,s), p(v,s), s) - \langle \partial_x H(\gamma(v,s), p(v,s), s), \gamma(v,s)\rangle \Big\} ds - \langle x, v\rangle$$

subject to the following restriction

$$\begin{cases} \dot\gamma(v,s) = \partial_p H(\gamma(v,s), p(v,s), s), \\ \dot p(v,s) = -\partial_x H(\gamma(v,s), p(v,s), s), \\ \gamma(v,t) = x, \\ p(v,t) = v \end{cases}$$

Optimization method

We perform the following method of gradient descent

Algorithm 1. *Take an initial guess of the Lipschitz constant L, and set count $:= 0$. Initialize $j_1 := 1$ and a parameter $\alpha := 1/L$. For $k = 1, ..., M$, do:*

1:

$$\begin{cases} v_i^{k+1} = v_i^k - \alpha\, \partial_i \mathcal{G}_{x,t}(v^k) & \text{if } i = j_k, \\ v_i^{k+1} = v_i^k & \text{otherwise.} \end{cases}$$

2:

$$j_{k+1} := j_k + 1.$$

If $j_{k+1} = d + 1$, then reset $j_{k+1} = 1$.

3: If $|v_i^{k+1} - v_i^k| > \varepsilon$, then set count $:= 0$. If $k = M$, then reset $k := 0$ and set $\alpha := \alpha/2$, (i.e. let $L := 2L$.)

4: If $|v^{k+1} - v^k| < \varepsilon$, set count $:=$ count $+ 1$.

5: If count $= d$, stop.

Return $v_{\text{final}} = v^{k+1}$.

where the gradient could be taken by numerical differentiation, and the ODE could be solved numerically.

Optimization method for ordinary Hopf-Lax Formula

For our Hopf-Lax formula

$$\varphi(x,t) = -\min_{v\in\mathbb{R}^d}\{tH(v) + J^*(v) - \langle v, x\rangle\} \tag{26}$$

We can use the following ADMM algorithm for optimization

For $n = 1, 2, ...,$ do the following:

Step 1:

$$w^{k+1} \in \operatorname{argmin}_{w\in\mathbb{R}^d}\left\{tH(w) + \frac{\rho}{2}\|\lambda^k - v^k + w\|^2\right\},$$

Step 2:

$$v^{k+1} = \operatorname{argmin}_{v\in\mathbb{R}^d}\left\{J^*(v) - \langle x, v\rangle + \frac{\rho}{2}\|\lambda^k - v + w^{k+1}\|^2\right\},$$

Step 3:

$$\lambda^{k+1} = \lambda^k - v^{k+1} + w^{k+1}.$$

Section 4

Connection with Deep Learning

Intuition

- Since the ultimate goal of machine learning is to create a class of functions that can represent the data with desired accuracy, our aim is to approximate a target function with minimum loss.
- In this perspective, we view deep learning and convolutional neural networks as discrete dynamic systems.
- We can use continuous dynamic systems to approximate the data label.

Formulation

The essential task of supervised learning is to approximate some function $F : \mathbb{X} \to \mathbb{Y}$ which maps inputs (e.g images, time-series) to labels. We are given a collection of sample pairs (x_i, y_i).

Consider the system of ODEs

$$\begin{cases} \dot{X}_t^i = f(t, X_t^i, \theta_t) & 0 \leq t \leq T, \\ X_0^i = x^i. \end{cases} \tag{27}$$

where θ represents the control parameters and f is chosen as part of a machine learning model. Our output data is a deterministic transformation of the terminal state, namely $g(X_T^i)$ for some fixed g.

Formulation

We aim at minimizing the loss function. Assume a loss function $\phi : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}$ is given (where we often choose the distance between two points in a certain norm). Define $\phi_i(\cdot) \coloneqq \phi(g(\cdot), y^i)$. Then the supervised learning problem becomes

$$\min_{\theta}\{\sum \phi_i(X_T^i) + \int_0^T L(\theta_t)dt\} \tag{28}$$

where L is a running cost, or the regularizer.

Optimization Algorithms

We utilize the Pontryagin Maximum Principle and thus devise the algorithm in the following way

Algorithm 1 Basic MSA

1: Initialize: $\theta^0 \in \mathcal{U}$
2: **for** $k = 0$ to #Iterations **do**
3: Solve $\dot{X}_t^{\theta^k} = f(t, X_t^{\theta^k}, \theta_t^k), \quad X_0^{\theta^k} = x$
4: Solve $\dot{P}_t^{\theta^k} = -\nabla_x H(t, X_t^{\theta^k}, P_t^{\theta^k}, \theta_t^k), \quad P_T^{\theta^k} = -\nabla\Phi(X_T^{\theta^k})$
5: Set $\theta_t^{k+1} = \arg\max_{\theta\in\Theta} H(t, X_t^{\theta^k}, P_t^{\theta^k}, \theta)$ for each $t \in [0, T]$

This algorithm is called the method of successive approximations.

Optimization Algorithms

Connection with Gradient Descent

The basic MSA above can not guarantee the convergence property (Step 5 may incur too much error in the Hamilton dynamics when replacing θ^k with θ^{k+1}), so we invoke a extended MSA algorithm, based on the extended version of Pontryagin Maximum Principle.
After discretization, we obtain a E-MSA formula for the discrete-time scenario (which is relevant to the optimization of deep residual network), and if we replace the maximization step with a gradient ascent step, this method is equivalent to gradient descent with back-propagation.

Optimization Algorithms

Advantages

1. Rigorous error estimates and convergence results can be established.
2. The algorithm enjoys fast initial descent of loss function and ease for parallelization.
3. When applying PMP method, the gradient $w.r.t$ the trainable parameters is not needed, so we can apply it even when the parameters are not differentiable.
4. Optimization is performed at each layer separately, and propagation is independent of optimization.

Additional Comments

1. The renowned deep residual network could be viewed as a discretization of the dynamic system, because the outputs for adjacent layers have the following connection $z_{l+1} = z_l + \mathbb{F}(z_l, W_l)$, where W_i are weights to be trained of each layer.

Additional Comments

1. The renowned deep residual network could be viewed as a discretization of the dynamic system, because the outputs for adjacent layers have the following connection $z_{l+1} = z_l + \mathbb{F}(z_l, W_l)$, where W_i are weights to be trained of each layer.
2. Neural networks have the advantage of easy change of dimensionality at each layer. The dynamic system has to be split to accomplish so.

Additional Comments

1. The renowned deep residual network could be viewed as a discretization of the dynamic system, because the outputs for adjacent layers have the following connection $z_{l+1} = z_l + \mathbb{F}(z_l, W_l)$, where W_i are weights to be trained of each layer.
2. Neural networks have the advantage of easy change of dimensionality at each layer. The dynamic system has to be split to accomplish so.
3. In order to solve the control equations, it would be time efficient to accomplish this via solving back in time, since in this way we can solve for different times in a parallel fashion, which resembles the idea of back-propagation.

Section 5

Acknowledgements

Reference : Books and Techreports

Lawrence C Evans. *An introduction to mathematical optimal control theory*. Lecture Notes, University of California, Department of Mathematics, Berkeley, 2005.

Lawrence C. Evans. *Partial differential equations*. Providence, R.I.: American Mathematical Society, 2010.

Yat T Chow et al. *Algorithm for Overcoming the Curse of Dimensionality for Certain Non-convex Hamilton-Jacobi Equations, Projections and Differential Games*. Tech. rep. University of California, Los Angeles Los Angeles United States, 2016.

Reference : Articles

- Yat Tin Chow et al. "Algorithm for Overcoming the Curse of Dimensionality for State-dependent Hamilton-Jacobi equations". In: *arXiv preprint arXiv:1704.02524* (2017).
- Qianxiao Li et al. "Maximum Principle Based Algorithms for Deep Learning". In: *arXiv preprint arXiv:1710.09513* (2017).
- IV Rublev. "Generalized Hopf formulas for the nonautonomous Hamilton–Jacobi equation". In: *Computational Mathematics and Modeling* 11.4 (2000), pp. 391–400.
- E Weinan. "A Proposal on Machine Learning via Dynamical Systems". In: *Communications in Mathematics and Statistics* 5.1 (2017), pp. 1–11.

Thanks!