Deep Reinforcement Learning: A Convex Optimization Approach

Ather Gattami ⟨ atherg@gmail.com ⟩

Abstract

In this paper, we consider reinforcement learning of nonlinear systems with continuous state and action spaces. We present an episodic learning algorithm, where we for each episode use convex optimization to find a two-layer neural network approximation of the optimal Q-function. The convex optimization approach guarantees that the weights calculated at each episode are optimal, with respect to the given sampled states and actions of the current episode. For stable nonlinear systems, we show that the algorithm converges and that the converging parameters of the trained neural network can be made arbitrarily close to the optimal neural network parameters. In particular, if the regularization parameter is ρ and the time horizon is T, then the parameters of the trained neural network converge to w, where the distance between w from the optimal parameters $w^\star$ is bounded by $\mathcal{O}(\rho T^{-1})$. That is, when the number of episodes goes to infinity, there exists a constant C such that

$$\|w - w^\star\| \leq C \cdot \frac{\rho}{T}.$$

In particular, our algorithm converges arbitrarily close to the optimal neural network parameters as the time horizon increases or as the regularization parameter decreases.

1. Introduction

1.1. Background

Deep Reinforcement Learning (RL) has been a cornerstone in most recent developments of Artificial Intelligence One example was defeating the highest ranked player in the ancient game Go (Silver et al., 2017), which in the 90's was considered a challenge that is hard to crack for the coming 100 years. Another example is the development of chat bots with Large Language Models, based on RL with Human Feedback.

Most of the recent practical progress is related to Markov Decision Processes (MDPs) with discrete state and/or action spaces. The case of continuous state and action spaces is hard in general due to the high complexity of the problem.

The optimal control in its full generality is given by a dynamical system

$$x_{t+1} = f(x_t, u_t)$$

where x_t represents the state and u_t the controller, at time step t. The goal of the controller is to minimize a certain criterion, given by

$$\sum_{t=1}^{T} c_t(x_t, u_t)$$

and a constraint on the control signal, $u \in \mathcal{U}$, typically given by a bound on its power, $\|u_t\|^2 \leq c_u$.

Bellman provided a general algorithm to solve the problem of optimal control based on dynamics programming, given by the well known Bellman equation

$$V_t(x) = \min_u \{c_t(x, u) + V_{t+1}(f(x, u))\}$$

The challenge with the above equation is what Bellman referred to as "The curse of dimensionality", where for most of the cases, the dynamic programming solution explodes exponentially in the dimensions of the state and action spaces, and the length of the time horizon. Therefore, most of the known approaches are approximate even for the case where the system parameters and the cost function are known.

1.2. Related Work

Nonlinear control theory is mainly considered with stabilizing nonlinear systems, relying on different approaches, see for instance (Khalil, 2002). The Bellman equation has been a standard tool for optimal control. For continuous state and action spaces, a straight forward approach is to discretize the state/action spaces and then use existing solutions for discrete MDPs such as different variants of Q-learning, see (Sutton & Barto, 2018) for an overview. However, the problem with this approach is that the higher resolution, the larger state and action spaces become. This will in turn increase the computational complexity of the problem. To tackle this computational complexity, methods relying on function approximations are used. A relatively simple approach from a complexity point of view

Algorithm 1 Episodic Learning with Convex Optimization

```
Input γ, ρ, R
Initialize u_1, ..., u_T
Sample D by running u_1, ..., u_T
Initialize w
Set w_1 = w
for episode k = 1, ..., K: do
    Observe x_1
    Set u_1 = arg min_u Q(x_1, u, w_k)
    for (t = 1, ..., T): do
        Apply u_t and observe x_{t+1}
        Set u_{t+1} = arg min_u Q(x_{t+1}, u, w_k)
        X_t = (1  x_t^T  u_t^T)
        y_t = c(x_t, u_t) + γQ(x_{t+1}, u_{t+1}, w_k)
    end for
    Solve (5) and obtain the solution w
    w_{k+1} ← w_k + α_k(w − w_k)
end for
```

is to use linear function approximation (Melo & Ribeiro, 2007), where convergence is shown, given certain conditions that could be too restrictive. Most recently, function approximation based on Neural Networks have been widely used, due to its success in the case of discrete state and action spaces. However, the current methods suffer from several drawbacks. First, there are no convergence guarantees when the Q function is approximated with a neural network. Second, even if the algorithms converge, it's not clear how far from the optimum they converge to. We refer the reader to (van Hasselt, 2012) For a more thorough literature review of Reinforcement Learning in continuous state and action spaces.

1.3. Contributions

Our main contribution is the introduction of Algorithm 1, where we episodically use convex optimization to find two-layer neural network approximation to the optimal Q-function. The convex optimization approach guarantees that the weights calculated at each episode are optimal, with respect to the given sampled states and actions of the current episodes. We show that the algorithm converges for stable nonlinear systems, and that the converging parameters of the trained neural network can be made arbitrarily close to the optimal neural network parameters. In particular, if the regularization parameter is ρ and the time horizon is T, then the algorithm parameters w distance from the optimal parameters $w^\star$ is bounded $\mathcal{O}(\rho T^{-1})$. That is, there is a constant C such that

$$\|w - w^\star\| \leq C \cdot \frac{\rho}{T}.$$

For instance, by decreasing the regularization parameter during the training phase and/or increasing the time horizon of the optimal control problem, we can get arbitrarily close to the optimal parameters. Finally, we provide experimental results that show the performance of the proposed algorithm with respect to a nonlinear dynamical system under power constraints on the control signal.

1.4. Notation

$\mathbb{N}$	The set of positive integers.
$\mathbb{R}$	The set of real numbers.
$\mathbf{P}_S(\ \cdot\ )$	$\mathbf{P}_S(x)$ is the projection of $x \in R^n$ on the space S.
A_i	A_i denotes ith i:th row of the matrix A.
$[A]_{ij}$	Denotes the element of the matrix A in position $(i.j)$
$A_{i,j}$	$A_{i,j} = [A]_{ij}$
$\|\cdot\|_F$	$\|A\|$ denotes the Frobenius norm of the matrix A.
$\|\cdot\|$	$\|A\|$ denotes the ∞-norm of the matrix A.
$(\ \cdot\ )_+$	For a vector $x \in \mathbb{R}$, $(x)_+ = v$, where $v_i = \max(x_i, 0)$.
w_-	Given a sample w_k, we have $w_- = w_{k-1}$.

2. Episodic Deep Reinforcement Learning with Convex Optimization

Consider a dynamical system given by

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Suppose that the convex cost function $c(x, u)$ is non-negative and that it's bounded by some constant $\bar{c} \geq c(x, u)$, for all stabilizing control signals u. The goal of the controller $u_t = \mu_t(x_t, x_{t-1}, ..., x_1)$ is to minimize the cost

$$\sum_{t=1}^{T} \gamma^t c(x_t, u_t)$$

where $\gamma \in (0, 1]$, subject to some power constraint on the control signal, $\|u_t\| \leq c_u$.

Bellman's equation gives the recursion

$$V(x) = \min_u \{c(x, u) + \gamma V(f(x, u))\} \tag{2}$$

where $V(x)$ is the value function. Alternatively, we can use the expression

$$Q(x, u) = c(x, u) + \gamma \min_{u_+} Q(x_+, u_+) \tag{3}$$

The Q function can be approximated by a neural network as in (Silver et al., 2017). We let the activation functions be given by the ReLU function

$$(x)_+ \triangleq \max(x, 0).$$

Now suppose Q is represented by a two layer neural network with layer parameters

$$\mathbf{w} = (w, w'), \;\; w \in \mathbb{R}^{d\times M}, \;\; w' \in \mathbb{R}^M,$$

$$\begin{aligned} Q(x, u, \mathbf{w}) &= ([1 \;\; x^\intercal \; u^\intercal]w)_+ \, w' \\ &= \sum_{i=1}^{M} ([1 \;\; x^\intercal \; u^\intercal]w_i)_+ \, w'_i. \end{aligned}$$

Let $\mathbf{w}^\star$ be the optimal network parameters. Then, they satisfy the Bellman equation

$$Q(x, u, \mathbf{w}^\star) = c(x, u) + \mathbf{P}_\mathbf{Q}\left(\gamma \min_{u_+} Q(x_+, u_+, \mathbf{w}^\star)\right) \tag{4}$$

where

$$\mathbf{Q} = \left\{Q : Q = (Xw)_+ w', w \in \mathbb{R}^{d\times M}, w' \in \mathbb{R}^M, M \in \mathbb{N}\right\}$$

The training of the Q function parameters is to minimize the squared error function with respect to the weights $\mathbf{w}$ of a neural network, that is minimizing the loss

$$\min_{\mathbf{w}} \sum_{t=1}^{T} l(\mathbf{w})$$

where

$$\begin{aligned} l(\mathbf{w}) = &\left(c(x, u) + \gamma \min_{u_+} Q(x_+, u_+, \mathbf{w}_-) - Q(x, u, \mathbf{w})\right)^2 \\ &+ \rho R(\mathbf{w}) \end{aligned}$$

and $R(\mathbf{w})$ is some regularization term that can be chosen appropriately.

The minimization with respect to the loss function $l(\mathbf{w})$ is not necessarily a convex optimization problem, and it could be hard to find the right neural network approximation. To get around this problem, we will consider a so called episodic setting, which we will describe here. Consider learning to control the dynamical system over K episodes, where each episode has a time horizon of T time steps. The input data will be given by

$$X = \begin{pmatrix} 1 & x_1^\intercal & u_1^\intercal \\ 1 & x_2^\intercal & u_2^\intercal \\ \vdots & \vdots & \vdots \\ 1 & x_{T-1}^\intercal & u_{T-1}^\intercal \\ 1 & x_T^\intercal & u_T^\intercal \end{pmatrix}$$

and output data will be given by

$$y_t = c(x_t, u_t) + \gamma Q(x_{t+1}, u_{t+1}, \mathbf{w}_k),$$

for $t = 1, ..., T$.

Now, let

$$c = [c(x_1, u_1) \;\; c(x_2, u_2) \;\; \cdots \;\; (x_T, u_T)]^\intercal$$

and

$$Z = \begin{pmatrix} 1 & x_2^\intercal & u_2^\intercal \\ 1 & x_3^\intercal & u_3^\intercal \\ \vdots & \vdots & \vdots \\ 1 & x_T^\intercal & u_T^\intercal \\ 1 & x_{T+1}^\intercal & u_{T+1}^\intercal \end{pmatrix}$$

Suppose that the regularization term R is given by

$$R(\mathbf{w}) = \|w\|_F^2 + |w'|^2$$

The optimization problem of training a two layer neural network becomes

$$\begin{aligned} &\min_{w,w'} |(Xw)_+ w' - y|^2 + \rho\left(\|w\|_F^2 + |w'|^2\right) = \\ &\min_{w,w'} \left|\sum_{i=1}^{M} (Xw_i)_+ w'_i - y\right|^2 + \rho \sum_{i=1}^{M}\left(|w_i|^2 + |w'_i|^2\right) \end{aligned}$$

where

$$\begin{aligned} w &= [w_1 \; w_2 \;\; \cdots \;\; w_M], \\ w' &= [w'_1 \; w'_2 \;\; \cdots \;\; w'_M]^\intercal, \end{aligned}$$

$w_i \in \mathbb{R}^d$, $w'_i \in \mathbb{R}$, for $i = 1, ..., M$.

Using the framework by Pilanci *et. al.* (Pilanci & Ergen, 2020), the optimization problem of minimizing the loss $l(\mathbf{w})$ becomes convex if we approximate the Q-function with a two-layer neural network. We can use the result in (Pilanci & Ergen, 2020) by generating a set of D matrices for each episode, and then optimize for the weights $\mathbf{w}$.

The equivalent convex optimization problem is given by

$$\begin{aligned} \min_{w \in \mathcal{W}} &\left|\sum_{p=1}^{P} D_p X(w_{1,p} - w_{2,p}) - y\right|^2 \\ &+ \rho \sum_{p=1}^{P} (|w_{1,p}| + |w_{2,p}|) \end{aligned} \tag{5}$$

where $w = (w_1, ..., w_P) \in \mathbb{R}^{2(m+n+1)\times P}$. Note that the optimization problem in (5) is equivalent to

$$\begin{aligned} \min_{\substack{w \\ v>0}} &\left|\sum_{p=1}^{P} D_p X(v_{1,p} w_{1,p} - v_{2,p} w_{2,p}) - y\right|^2 \\ &+ \rho \sum_{p=1}^{P} (|v_{1,p} w_{1,p}| + |v_{2,p} w_{2,p}|) \end{aligned} \tag{6}$$

$$\begin{aligned} \text{subject to} \;\; & 0 \le (2D_p - 1)Xw_{1,p}, \;\; p = 1, ..., P \\ & 0 \le (2D_p - 1)Xw_{2,p}, \;\; p = 1, ..., P. \end{aligned}$$

The learning algorithm we propose is given by Algorithm 1. The main theoretical result of the paper provides conditions for which Algorithm 1 converges. Before stating our theorem, let $v_{i,p}$ be the optimal solution in (6), , for $p = 1, .., P$, $i = 1, 2$, and introduce

$$F_1 = \begin{bmatrix} \frac{\rho}{2v_{1,1}^2} & 0 & \cdots & 0 \\ 0 & \frac{\rho}{2v_{1,2}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\rho}{2v_{1,p}^2} \end{bmatrix},$$

$$F_2 = \begin{bmatrix} \frac{\rho}{2v_{2,1}^2} & 0 & \cdots & 0 \\ 0 & \frac{\rho}{2v_{2,2}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\rho}{2v_{2,p}^2} \end{bmatrix},$$

and

$$F = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix}.$$

Now we are ready to state our main result.

Theorem 1. *Consider Algorithm 1 and let λ and β be positive real numbers with $\beta \geq 1$, such that*

$$\lambda I \preceq \frac{1}{T}\left(\sum_{t=1}^{T}\begin{bmatrix} X_t^\intercal \\ -X_t^\intercal \end{bmatrix}\begin{bmatrix} X_t & -X_t \end{bmatrix} + F\right)$$

and

$$|u_t|^2 \leq \beta, \quad |x_t|^2 \leq \beta, \quad \text{for } t = 1, ..., T.$$

Further, let α_k be either a positive real constant, $\alpha_k = \alpha < 1$ for all $k \in \mathbb{N}$, or such that $\alpha_k = 1/k$. If

$$T > \frac{3}{2}\left(\frac{\gamma\beta}{\lambda}\right)^2,$$

then $w_k \to w$ where

$$\|w - w^\star\| \leq C \cdot \frac{\rho}{T}$$

for some positive real constant C.

Proof. The proof of this theorem is provided in a later section.

Remark 1. *Note that Algorithm 1 above is useful for both cases when the system parameters are known as well as when they are unknown. In the case of known parameters, it's hard in general to have a closed form solution for the controller. Training a convex Q-function is a great way to find a controller that is near optimal. In the case of unknown system parameters and cost function, we can just use the measured values of $c(x, u)$ and the state x. It would be interesting to compare both results when assuming known and unknown system parameters.*

Initial State	Lower Bound	Convex Optimization
0.25	0.346	0.364
0.50	1.493	1.548
0.75	0.315	0.364
1.00	8.140	9.981

Table 1. The table shows the performance of the trained neural network, compared to the lower bound given by the optimal finite horizon controller found by numerically solving the Bellman equation. We considered a time horizon $T = 5$, and tested different initial states x_0 in the interval $[0, 1]$. We see clearly the the convex optimization approach presented in this paper gets very close to the optimal solution after 1000 episodes.

3. Experiments

In this section we will consider reinforcement learning for nonlinear dynamical system given by

$$x_{t+1} = 0.9x_t^2 + 0.1u_t$$

with a cost function given by

$$c(x, u) = x^2 + (0.1u - 2x)^2$$

and constraint $|u_t| \leq 5$. for simplicity, We restrict the training to initial states $x_0 \in [0, 1]$. Since it's hard to numerically find the optimal stationary (time-invariant) controller with full knowledge of the system model and cost function, we solve the finite horizon problem using dynamic programming and gridding of the state space applied to the Bellman equation, to find the optimal time-varying controller. This would provide a lower bound to the cost using a stationary controller. In particular we verify the cost over 5 time steps, that is $T = 5$. The number of episodes was set $K = 1000$ to train a two layer neural network that approximates the optimal Q-function, and the regularization parameter is set to $\rho = 10^{-4}$. Table 3 summarizes the results for different initial states. We can see that the neural networks found by the convex optimization based episodic learning algorithm presented in this paper gets very close to the lower bound of the optimal controller.

4. Proof of Theorem 1

Let

$$c = \begin{bmatrix} c(x_1, u_1) \\ c(x_2, u_2) \\ \vdots \\ c(x_T, u_T) \end{bmatrix}$$

and

$$Z = \begin{pmatrix} 1 & x_2^\intercal & u_2^\intercal \\ 1 & x_3^\intercal & u_3^\intercal \\ \vdots & \vdots & \vdots \\ 1 & x_{T+1}^\intercal & u_{T+1}^\intercal \end{pmatrix}$$

We have that

$$y = c + \sum_{p=1}^{P} D'_p Z(w_{1,p,k} - w_{2,p,k})$$

where w_k are the parameters used in episode k and

$$[D'_p]_{tt} = \begin{cases} [D_p]_{(t+1)(t+1)} & \text{for } 1 \le t \le T-1 \\ 0 & \text{for } t = T \end{cases}$$

Let $\mathcal{W}$ be the linear space given by

$$\mathcal{W} = \{w:\ 0 \le (2D_p - 1)Xw_p,\ p = 1, ..., P, \}$$

and let $\mathcal{Q}$ be the space of functions given by

$$\mathcal{Q} = \Big\{Q:\ Q = \sum_{p=1}^{P} D_p X(w_{1,p} - w_{2,p}), \\ Q \ge 0,\ 0 \le (2D_p - 1)Xw_{i,p}, i = 1, 2\Big\}.$$

The optimal weights $w^\star$ are given by the Bellman equation

$$\sum_{p=1}^{P} D_p X(w^\star_{1,p} - w^\star_{2,p}) = \\ c + \mathbf{P}_{\mathcal{Q}}\left(\gamma \min_u \sum_{p=1}^{P} D'_p Z(w^\star_{1,p} - w^\star_{2,p})\right) \tag{7}$$

Since

$$|w_{i,p} v_{i,p}| \le \frac{1}{2}\left(|w_{i,p}|^2 + v_{i,p}^2\right),$$

with equality if and only if $|w_p| = v_p$, we see that (6) is equivalent to

$$\min_{\substack{w \\ v>0}} \left|\sum_{p=1}^{P} D_p X(v_{1,p}w_{1,p} - v_{2,p}w_{2,p}) - y\right|^2 \\ + \frac{\rho}{2}\sum_{p=1}^{P} \frac{1}{2}\left(|w_{i,p}|^2 + v_{i,p}^2\right) \\ \text{subject to } 0 \le (2D_p - 1)Xw_{i,p},\quad p = 1, ..., P,\quad i = 1, 2 \tag{8}$$

Now the transformation $w_{i,p}v_{i,p} \to w_{i,p}$ implies the equivalent optimization problem

$$\min_{v>0}\min_{w} \left|\sum_{p=1}^{P} D_p X(w_{1,p} - w_{2,p}) - y\right|^2 \\ + \frac{\rho}{2}\sum_{p=1}^{P}\left(|w_{i,p}/v_{i,p}|^2 + v_{i,p}^2\right) \tag{9} \\ \text{subject to } \begin{array}{l} 0 \le (2D_p - 1)Xw_{1,p},\quad p = 1, ..., P \\ 0 \le (2D_p - 1)Xw_{2,p},\quad p = 1, ..., P. \end{array}$$

Introduce

$$D_X = [D_1X\ D_2X\ \cdots\ D_pX] \\ D_Z = \left[D'_1Z\ D'_2Z\ \cdots\ D'_pZ\right],$$

The objective function in (9) is given by

$$\left|\sum_{p=1}^{P} D_p X(w_{1,p} - w_{2,p}) - y\right|^2 \\ + \frac{\rho}{2}\sum_{p=1}^{P}\left(|w_{1,p}/v_{1,p}|^2 + |w_{2,p}/v_{2,p}|^2 + v_{1,p}^2 + v_{2,p}^2\right) \\ = \left|\sum_{p=1}^{P} D_p X(w_{1,p} - w_{2,p}) - c \right. \\ \left. -\gamma\sum_{p=1}^{P} D'_p Z(w^k_{1,p} - w^k_{2,p})\right|^2 \\ + \frac{\rho}{2}\sum_{p=1}^{P}\left(|w_{1,p}/v_{1,p}|^2 + |w_{2,p}/v_{2,p}|^2 + v_{1,p}^2 + v_{2,p}^2\right) \\ = \left|\sum_{p=1}^{P} D_p X(w_{1,p} - w_{2,p}) \right. \\ \left. -c - \mathbf{P}_{\mathcal{Q}}\left(\gamma\sum_{p=1}^{P} D'_p Z(w^k_{1,p} - w^k_{2,p})\right)\right|^2 \\ + \left|(I - \mathbf{P}_{\mathcal{Q}})\left(\gamma\sum_{p=1}^{P} D'_p Z(w^k_{1,p} - w^k_{2,p})\right)\right|^2 \\ + w_1^\intercal F_1 w_1 + w_2^\intercal F_2 w_2 + \frac{\rho}{2}\sum_{p=1}^{P}|v_p|^2 \\ = |D_X(w_1 - w_2) - \bar{y}|^2 \\ + \left|(I - \mathbf{P}_{\mathcal{Q}})\left(\gamma\sum_{p=1}^{P} D'_p Z(w^k_{1,p} - w^k_{2,p})\right)\right|^2 \\ + w_1^\intercal F_1 w_1 + w_2^\intercal F_2 w_2 + \frac{\rho}{2}\sum_{p=1}^{P}|v_p|^2 \\ = \begin{bmatrix}\bar{y} \\ w_1 \\ w_2\end{bmatrix}^\intercal H \begin{bmatrix}\bar{y} \\ w_1 \\ w_2\end{bmatrix} + \frac{\rho}{2}\sum_{p=1}^{P} v_p^2$$

where

$$H = \begin{bmatrix} I & -D_X & D_X \\ -D_X^\intercal & D_X^\intercal D_X + F_1 & -D_X^\intercal D_X \\ D_X^\intercal & -D_X^\intercal D_X & D_X^\intercal D_X + F_2 \end{bmatrix},$$

$$\Lambda = \begin{bmatrix} D_X^\intercal D_X + F_1 & -D_X^\intercal D_X \\ -D_X^\intercal D_X & D_X^\intercal D_X + F_2 \end{bmatrix}$$
$$L = \Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix},$$
$$M = I - L^\intercal \Lambda^{-1} L.$$

Using the Bellman equation (7), we get

$$\begin{aligned}\bar{y} &= c + \mathbf{P}_{\mathcal{Q}}(\gamma D_Z(w_{1,k} - w_{2,k})) \\ &= D_X(w_1^\star - w_2^\star) - \mathbf{P}_{\mathcal{Q}}(\gamma D_Z(w_1^\star - w_2^\star)) \\ &\quad + \mathbf{P}_{\mathcal{Q}}(\gamma D_Z(w_{1,k} - w_{2,k})) \\ &= [D_X \quad -D_X]w^\star + \gamma \mathbf{P}_{\mathcal{Q}}([D_Z \quad -D_Z]w_k) \\ &\quad - \mathbf{P}_{\mathcal{Q}}(\gamma [D_Z \quad -D_Z]w^\star)\end{aligned}$$

Then, completion of squares gives the relation

$$\begin{bmatrix} \bar{y} \\ w \end{bmatrix}^\intercal H \begin{bmatrix} \bar{y} \\ w \end{bmatrix} = (w - L\bar{y})^\intercal \Lambda (w - L\bar{y}) + \bar{y}^\intercal M \bar{y}$$

Since $\Lambda \prec 0$, we may define the norm

$$\|w\|_\Lambda \triangleq w^\intercal \Lambda w.$$

Let $\mathcal{W}$ be the linear space given by

$$\mathcal{W} = \{w : \ 0 \leq (2D_p - 1)Xw_p, \ p = 1, ..., P\}.$$

Standard Hilbert Space Theory implies that the optimal solution $w^\star$ to

$$\min_{w \in \mathcal{W}} \|w - L\bar{y}\|_\Lambda$$

is the projection of Lv on $\mathcal{W}$ under the norm $\|\cdot\|_\Lambda$, that is

$$w_k^\star = \mathbf{P}_{\mathcal{W}}(L\bar{y}).$$

Thus,

$$w_k^\star = \mathbf{P}_{\mathcal{W}}\Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \left(c + \mathbf{P}_{\mathcal{Q}}\left(\gamma[D_Z \quad -D_Z]w_k\right)\right)\Big)$$

and

$$\begin{aligned}&w_k^\star - w^\star \\ &= \mathbf{P}_{\mathcal{W}}\Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \left(c + \mathbf{P}_{\mathcal{Q}}\left(\gamma[D_Z \quad -D_Z]w_k\right)\right) - w^\star\Big) \\ &= \mathbf{P}_{\mathcal{W}}\Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \left(c + \mathbf{P}_{\mathcal{Q}}\left(\gamma[D_Z \quad -D_Z]w_k\right)\right) \\ &\quad - \Lambda^{-1}\Big(\begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix}[D_X \quad -D_X] + F\Big)w^\star\Big) \\ &= \mathbf{P}_{\mathcal{W}}\Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \left(c + \mathbf{P}_{\mathcal{Q}}\left(\gamma[D_Z \quad -D_Z]w_k\right)\right. \\ &\quad \left. - [D_X \quad -D_X]w^\star\right) - \Lambda^{-1}Fw^\star\Big) \\ &= \mathbf{P}_{\mathcal{W}}\Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \left(\mathbf{P}_{\mathcal{Q}}\left(\gamma[D_Z \quad -D_Z](w_k - w^\star)\right)\right. \\ &\quad - \Lambda^{-1}Fw^\star\Big)\end{aligned}$$

Now the update rule for w_k implies that

$$w_{k+1} - w^\star = (1 - \alpha_k)(w_k - w^\star) + \alpha_k(w_k^\star - w^\star)$$

and

$$\begin{aligned}&w_{k+1} - w^\star \\ &= (1 - \alpha_k)(w_k - w^\star) + \alpha_k(w_k^\star - w^\star) \\ &= (1 - \alpha_k)(w_k - w^\star) \\ &+ \alpha_k \mathbf{P}_{\mathcal{W}}\Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \left(\mathbf{P}_{\mathcal{Q}}\left(\gamma[D_Z \quad -D_Z](w_k - w^\star)\right)\right) \\ &\qquad - \Lambda^{-1}Fw^\star\Big)\end{aligned}$$

Introduce

$$A = \begin{bmatrix} D_X & -D_X \end{bmatrix},$$

Then,

$$\begin{aligned}A^\intercal A &= \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} [D_X \quad -D_X] \\ &= \begin{bmatrix} D_X^\intercal D_X & -D_X^\intercal D_X \\ -D_X^\intercal D_X & D_X^\intercal D_X \end{bmatrix},\end{aligned}$$

$$\begin{aligned}AA^\intercal &= \begin{bmatrix} D_X & -D_X \end{bmatrix} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix}, \\ &= 2D_X D_X^\intercal\end{aligned}$$

Also, note that

$$\begin{aligned}A^\intercal A + F &= \sum_{t=1}^{T} \sum_{p=1}^{P} [D_p]_{tt} \begin{bmatrix} X_t^\intercal \\ -X_t^\intercal \end{bmatrix} [X_t \quad -X_t] + F \\ &\succeq \sum_{t=1}^{T} \sum_{p=1}^{P} \begin{bmatrix} X_t^\intercal \\ -X_t^\intercal \end{bmatrix} [X_t \quad -X_t] + F \\ &\succeq T \cdot \lambda I\end{aligned}$$

and

$$\begin{aligned}\|\Lambda^{-1}F\| &= \|(A^\intercal A + F)^{-1}F\| \\ &\leq \|(T \cdot \lambda I)^{-1}F\| \\ &\leq \|(T \cdot \lambda I)^{-1}\rho f_{\max}\| \\ &\leq \frac{\rho f_{\max}}{\lambda T}\end{aligned} \tag{10}$$

Now we have that

$$(A^\intercal A + F)^{-1}A^\intercal A(A^\intercal A + F)^{-1} \preceq (A^\intercal A + F)^{-1}$$

Introduce

$$f_{\min} = \min_{i,p} \left\{\frac{1}{v_{i,p}}\right\}$$
$$f_{\max} = \max_{i,p} \left\{\frac{1}{v_{i,p}}\right\},$$

and note that for any $f \in (0, \infty)$, we have the relation

$$(A^\intercal A + fI)^{-1} A^\intercal = A^\intercal (AA^\intercal + fI)^{-1}$$

Thus,

$$\begin{aligned}
\left\| \Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \right\| &= \left\| (A^\intercal A + F)^{-1} A^\intercal \right\| \\
&\le \left\| (A^\intercal A + \rho f_{\min} I)^{-1} A^\intercal \right\| \\
&= \left\| A^\intercal (AA^\intercal + \rho f_{\min} I)^{-1} \right\| \\
&= \left\| \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} (2 D_X D_X^\intercal + \rho f_{\min} I)^{-1} \right\| \\
&= \left\| D_X^\intercal (2 D_X D_X^\intercal + \rho f_{\min} I)^{-1} \right\| \\
&= \left\| (2 D_X^\intercal D_X + \rho f_{\min} I)^{-1} D_X^\intercal \right\| \\
&\le \left\| (2 D_X^\intercal D_X + \rho f_{\min} I)^{-1} \right\| \left\| D_X^\intercal \right\| \\
&\le \left\| (2T \cdot \lambda I + \rho f_{\min} I)^{-1} \right\| \cdot \sqrt{T\beta} \\
&= \frac{\sqrt{T\beta}}{2T\lambda + \rho f_{\min}} \\
&\le \sqrt{\frac{\beta}{4\lambda^2 T}}
\end{aligned} \tag{11}$$

Thus,

$$\begin{aligned}
&\|w_{k+1} - w^\star\| \\
&\le (1 - \alpha_k) \|w_k - w^\star\| \\
&+ \alpha_k \left\| \mathbf{P}_\mathcal{W} \Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \mathbf{P}_\mathcal{Q} \left(\gamma [D_Z \quad - D_Z] (w_k - w^\star) \right) \Big) \right\| \\
&+ \alpha_k \left\| \mathbf{P}_\mathcal{W} \left(\Lambda^{-1} F w^\star \right) \right\| \\
&\le (1 - \alpha_k) \|w_k - w^\star\| \\
&+ \alpha_k \left\| \mathbf{P}_\mathcal{W} \Big(\Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \Big) \right\| \left\| \mathbf{P}_\mathcal{Q} \left(\gamma [D_Z \quad - D_Z] \right) \right\| \|w_k - w^\star\| \\
&+ \alpha_k \left\| \mathbf{P}_\mathcal{W} \left(\Lambda^{-1} F w^\star \right) \right\| \\
&\le (1 - \alpha_k) \|w_k - w^\star\| \\
&+ \alpha_k \left\| \Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \right\| \left\| \gamma [D_Z \quad - D_Z] \right\| \|w_k - w^\star\| \\
&+ \alpha_k \left\| \Lambda^{-1} F w^\star \right\| \\
&\le (1 - \alpha_k) \|w_k - w^\star\| \\
&+ \alpha_k (\gamma \sqrt{6\beta}) \left\| \Lambda^{-1} \begin{bmatrix} D_X^\intercal \\ -D_X^\intercal \end{bmatrix} \right\| \|w_k - w^\star\| \\
&+ \alpha_k \left\| \Lambda^{-1} F w^\star \right\| \\
&\le (1 - \alpha_k) \|w_k - w^\star\| + \alpha_k (\gamma \sqrt{6\beta}) \sqrt{\frac{\beta}{4\lambda^2 T}} \|w_k - w^\star\| \\
&+ \alpha_k \left\| \Lambda^{-1} F w^\star \right\| \\
&= (1 - \alpha_k) \|w_k - w^\star\| + \alpha_k \sqrt{\frac{3\gamma^2\beta^2}{2\lambda^2 T}} \|w_k - w^\star\| \\
&+ \alpha_k \left\| \Lambda^{-1} F w^\star \right\| \\
&\le \left(1 - \alpha_k + \alpha_k \sqrt{\frac{3\gamma^2\beta^2}{2\lambda^2 T}} \right) \|w_k - w^\star\| + \alpha_k \frac{\rho f_{\max}}{\lambda T} \|w^\star\|
\end{aligned}$$

The inequality

$$T > \frac{3}{2} \left(\frac{\gamma\beta}{\lambda} \right)^2$$

implies that

$$\frac{3\gamma^2\beta^2}{2\lambda^2 T} < 1$$

Introduce

$$\mu = 1 - \sqrt{\frac{3\gamma^2\beta^2}{2\lambda^2 T}} < 1$$

Since $\alpha_k \le 1$, we have that

$$1 - \mu\alpha_k < 1.$$

For the case where $\alpha_k = \alpha$, for all $k \in \mathbb{N}$, we see that the inequality in (11) implies that $w_k - w^\star$ converges to some $w - w\star$ where

$$\|w - w^\star\| \le \frac{\mu\alpha}{1 - \mu\alpha} \frac{\rho f_{\max}}{\lambda T} \|w^\star\|$$

Similarly, for $\alpha_k = \frac{1}{k}$, we have that $\|w_k - w^\star\|$ is the state Δ_k of a stable dynamical system given by

$$\Delta_{k+1} \le (1 - \mu\alpha_k) \cdot \Delta_k + \alpha_k \frac{\rho f_{\max}}{\lambda T} \|w^\star\|$$

More explicitely, we have that

$$\Delta_K \le \prod_{k=1}^{K} (1 - \mu\alpha_k) \Delta_0 + \sum_{k=1}^{K} \prod_{i=k+1}^{K} (1 - \mu\alpha_i) \alpha_k \frac{\rho f_{\max}}{\lambda T} \|w^\star\| \tag{12}$$

Note that

$$\begin{aligned}
\prod_{k=1}^{K} (1 - \mu\alpha_k) &\le \prod_{k=1}^{K} \exp(-\mu\alpha_k) \\
&= \exp(-\mu \sum_{k=1}^{n} K\alpha_k) \\
&\le \exp(-\mu \ln K) \\
&= \frac{1}{K^\mu}
\end{aligned} \tag{13}$$

which goes to zero as $K \to \infty$. Similarly, for $k > 0$, we have

$$\begin{aligned}
\prod_{i=k+1}^{K} (1 - \mu\alpha_i) &\le \prod_{i=k+1}^{K} \exp(-\mu\alpha_i) \\
&= \exp\left(-\mu \sum_{i=k+1}^{K} \alpha_i \right) \\
&\le \exp\left(-\mu (\ln n - \ln k - 1) \right) \\
&= \left(\frac{ek}{K} \right)^\mu
\end{aligned} \tag{14}$$

Thus,

$$\begin{aligned}\sum_{k=1}^{K}\prod_{i=k+1}^{K}(1-\mu\alpha_i)\alpha_k &\leq \sum_{k=1}^{K}\left(\frac{ek}{K}\right)^{\mu}\frac{1}{k}\\ &= \left(\frac{e}{K}\right)^{\mu}\sum_{k=1}^{K}\left(\frac{1}{k}\right)^{1-\mu}\\ &\leq \left(\frac{e}{K}\right)^{\mu}\int_{k=1}^{K+1}x^{\mu-1}dx\\ &= \left(\frac{e}{K}\right)^{\mu}\frac{1}{\mu}\left((K+1)^{\mu}-1\right)\end{aligned} \tag{15}$$

Since

$$\lim_{K\to\infty}\frac{e^{\mu}}{\mu}\frac{1}{K^{\mu}}\left((K+1)^{\mu}-1\right)=\frac{e^{\mu}}{\mu},$$

we get the inequality

$$\sum_{k=1}^{\infty}\prod_{i=k+1}^{\infty}(1-\mu\alpha_i)\alpha_k \leq \frac{e^{\mu}}{\mu}. \tag{16}$$

We conclude that

$$\|w-w^{\star}\| = \Delta_{\infty} \leq C\cdot\frac{\rho}{T}$$

with

$$C=\frac{e^{\mu}}{\mu}\frac{f_{\max}}{\lambda}\|w^{\star}\|.$$

This completes the proof.

□

5. Conclusions

We have considered the problem of reinforcement learning for optimal control of stable nonlinear systems in an episodic setting, where we at each episode approximate the Q-function with a two-layer neural network for which the optimal parameters per episode are found by using convex optimization. We show that as the number of episodes goes to infinity, the algorithm converges to neural network parameters given by w such that the distance to the optimal network parameters $w^{\star}$ is bounded according to the inequality

$$\|w-w^{\star}\| \leq C\cdot\frac{\rho}{T},$$

for some constant C. In particular, we can see that as regularization parameter ρ decreases and as the time horizon T increases, the converging neural network parameters get arbitrarily close to the optimal ones.

Future work includes applications to reinforcement learning for mixed continuous and discrete state and action spaces. It would also be interesting to apply the algorithm to fine-tuning Large Language Models, since the training would be fast and computationally very efficient, and convergence to the optimal neural network parameters is guaranteed.

References

Khalil, H. K. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002. URL https://cds.cern.ch/record/1173048. The book can be consulted by contacting: PH-AID: Wallet, Lionel.

Melo, F. S. and Ribeiro, M. I. Convergence of q-learning with linear function approximation. In *2007 European Control Conference (ECC)*, pp. 2671–2678, 2007. doi: 10.23919/ECC.2007.7068926.

Pilanci, M. and Ergen, T. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7695–7705. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/pilanci20a.htm

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Driessche, G. V. D., Graepel, T., and Hassabis, D. Mastering the game of go without human knowledge. *Nature*, 550:354 – 359, 10 2017. URL http://dx.doi.org/10.1038/nature24270.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html

van Hasselt, H. *Reinforcement Learning in Continuous State and Action Spaces*, pp. 207–251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_7. URL https://doi.org/10.1007/978-3-642-27645-3_7.