# On Convergence of Neural asynchronous Q-iteration

**Elena Smirnova** esmirnovae@gmail.com

## Abstract

Deep Q-network algorithm is a successful and scalable algorithm on a variety of application domains. The algorithm incrementally trains a Q-network using one-step gradient descent over a mini-batch of randomly sampled transitions in a replay buffer. In this work, we formalize the Deep Q-network algorithm using an abstract algorithm, Neural asynchronous Q-iteration. We show convergence at a rate $O(\tilde{\gamma}^N)$, where $N$ is the number of iterations, $\tilde{\gamma}$ is a contraction coefficient that depends on a neural network and a rate of a mini-batch update. In particular, we show that Neural Tangent Kernel of a multi-layer ReLU neural network is a non-expansive operator and asynchronous Bellman operator is a smooth Bellman operator on average. Compared to previous work, our analysis is carried in the infinite-dimensional value function space w.r.t. kernel-based value functions. In this space, the algorithm takes a simple form, called Kernel Value Iteration.

**Keywords:** deep reinforcement learning, approximate dynamic programming, neural tangent kernel

## 1. Introduction

Starting with the Deep Q-network (DQN) algorithm Mnih et al. (2015), the combination of deep neural networks and RL algorithms has been successful in solving large-scale problems. The DQN algorithm performs an iterative update of the Q-network parameters, similarly to a gradient descent on a squared Bellman residual, using a mini-batch of randomly sampled transitions from a replay buffer. Motivated by the success of DQN and derived algorithms, we propose a DQN model based on Q-iteration, analyse its convergence and error propagation.

Previous work analysed the DQN algorithm as an instance of Q-learning algorithm Xu and Gu (2020) and as an instance of fitted Q-iteration Fan et al. (2020). In these analysis, the optimal Q-network is assumed to be close to the optimal Q-function. Q-learning analysis Xu and Gu (2020) provides convergence in terms of expected performance averaged across iterations at a rate $O(1/\sqrt{N})$, where $N$ is the number of iterations. Fitted Q-iteration analysis Fan et al. (2020) provides convergence in terms of weighted $L_1$-distance to the optimal Q-function with a term decreasing at a rate $O(\gamma^{N+1})$, where $\gamma$ is a discount factor, and an error term due to the finite sample of transitions to approximate Q-function.

In this work, we model DQN as an instance of abstract algorithm, *Neural asynchronous Q-iteration*. This algorithm is a Q-iteration with asynchronous update of state-action pairs Barto et al. (1995) and neural network function approximator Riedmiller (2005). We provide convergence in terms of $L_\infty$-distance to the fixed point at a rate $O(\tilde{\gamma}^N)$, where $\tilde{\gamma} := \|K_{NTK}\|_{op}(\beta\gamma + 1 - \beta)$. Specifically, neural network function approximation results in the operator norm of the (empirical) *Neural Tangent Kernel* Jacot et al. (2018) $K_{NTK}$, asynchronous update results in a smoothing coefficient of *smooth Bellman operator* Smirnova and Dohmatob (2020) $\beta \in (0, 1]$, given by the ratio of mini-batch size to replay buffer size. As we show $\tilde{\gamma} < 1$, the Neural asynchronous Q-iteration converges at $N \to \infty$. Further, the sampling noise of approximate Bellman update is down-weighted by a smoothing coefficient.

In more detail, we view Neural asynchronous Q-iteration in parameter space as Neural Tangent Kernel (NTK) value iteration in value function space. In this view, value function is in a Hilbert space, induced by NTK kernel. We show that kernel-based value iteration converges with normalized positive definite kernel function, such as (empirical) NTK kernel. We use the smooth Bellman operator to model random mini-batch updates. We provide a combined performance bound of empirical NTK value iteration with smooth Bellman operator, sampled at one transition.

**Related work** The related work includes recent analysis of wide neural networks Jacot et al. (2018); Chizat et al. (2018), that propose to model neural network as a linear model w.r.t. the NTK kernel. Value iteration in Hilbert space has been considered as kernel smoothing Ormoneit and Sen (2002) and in regularized approximate setting massoud Farahmand et al. (2009). Q-iteration in parameter space of a neural network has been studied in the finite-sample case Riedmiller (2005). Fitted Q-iteration has been analysed w.r.t. a function approximation operator of Q-function Gordon (1995).

**Contributions** To summarize, our contributions are as follows. (1) We propose a model of random mini-batch update as a smooth Bellman operator (Section 3.1). (2) We formalize Deep Q-network using linearization of over-parametrized Q-function (Section 3.3). (3) We propose a model of one-step gradient descent update of Deep Q-network as kernel-based value iteration with Neural Tangent Kernel (Section 4). (4) We show convergence of the resulting model w.r.t. smoothing coefficient, given by the mini-batch sampling ratio, and the operator norm of empirical Neural Tangent Kernel (Section 4.2).

**Outline** We present our analysis as follows. First, we present the background on approximate dynamic programming framework (Section 2). Next, we formalize the DQN algorithm as Neural asynchronous Q-iteration (Section 3). We analyse this algorithm as an instance of kernel-based value iteration, with the NTK kernel (Section 4).

## 2. Approximate dynamic programming

$\Delta_X$ will denote the set of probability distributions over a finite set (or general measurable space) $X$ and $Y^X$ is a set of functions from set $X$ to set $Y$.

### 2.1 Markov Decision Process

We consider the framework of Markov Decision Process $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is a finite set of actions, $P(\cdot|s,a) : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is a transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a bounded reward function, $\gamma \in (0,1)$ is a discount factor. Taking action $a \in \mathcal{S}$ in a state $s \in \mathcal{S}$ results in a transition according to a probability distribution $P(\cdot|s,a)$ and a reward $R(s,a)$.

Define a value function $V : \mathcal{S} \to \mathbb{R}$ of a policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ by the expected sum of discounted rewards collected by this policy starting at a given state $V^\pi(s) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s]$. It is convenient to define a value function of a state-action pair, called Q-function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of a policy $\pi$, given by $Q^\pi(s,a) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a]$, such that $V^\pi(s) = Q^\pi(s, \pi(s))$. The goal is to find the optimal value function, defined as a maximum value function over a set of policies state-wise $V^* := \max_\pi V^\pi$.

Define the Bellman operator $\mathcal{T}^\pi : V \to V$ of a policy $\pi$ as follows $[\mathcal{T}^\pi V](s) := \mathbb{E}_{a \sim \pi(\cdot|s)}[r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V(s')]]$. Value function of a policy $\pi$ is a unique fixed point of Bellman operator of policy $\pi$, i.e. $\mathcal{T}^\pi V^\pi = V^\pi$, where the equality holds state-wise. Define the optimal Bellman operator $\mathcal{T} : V \to V$ as a maximizing operator across all policies $\mathcal{T}V := \max_\pi \mathcal{T}^\pi V$. The maximizing policy is given by the greedy policy $\pi(\cdot|s) = \mathcal{G}(V) = \arg\max_a [r(s,a) + \gamma \mathbb{E}_{s' \sim P(s'|s,a)}[V(s')]]$. Optimal value function is a unique fixed point of the optimal Bellman operator, i.e. $\mathcal{T}V^* = V^*$ and the optimal policy is given by $\pi^* = \mathcal{G}(V^*)$, where the equality holds state-wise. Similarly, Bellman operator and optimal Bellman operator can be defined for Q-function.

### 2.2 Q-iteration

Q-iteration is an abstract dynamic programming algorithm that creates a sequence of Q-functions by applying the optimal Bellman operator (QI) $Q_{k+1} \leftarrow \mathcal{T}Q_k$. It is known that this sequence converges to the optimal Q-function at a rate $O(\gamma^N)$, where $N$ is the number of iterations. In the case of large or infinite state space, a parametrized function class is used to represent Q-function. In addition, if transition function is unknown, the optimal Bellman operator is computed approximately. This motivates the approximate Q-iteration scheme

$$\text{(Approx QI)} \qquad Q_{k+1} \leftarrow \mathcal{T}Q_k + \epsilon_{k+1}, \tag{1}$$

where $\epsilon_k \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ is a per-state error vector.

## 3. Neural asynchronous Q-iteration

Motivated by the DQN algorithm, we analyse an abstract algorithm, Neural asynchronous Q-iteration. It builds on the top of asynchronous Q-iteration Barto et al. (1995), defined as a standard Q-iteration where a random subset of state-action pairs is updated at each iteration (Section 3.1). This scheme formalizes random mini-batch update that, in expectation, is given by a smooth Bellman operator. Thus, the smooth Bellman target represents the average mini-batch update. Regression problem w.r.t. neural network parameters to the smooth Bellman target using squared objective defines a step of Neural asynchronous Q-iteration. A gradient descent step on this regression objective formalizes the DQN incremental update (Section 3.2).

Recent work on analysis of wide neural networks Jacot et al. (2018); Chizat et al. (2018) suggests a regression problem w.r.t. a value function obtained by linearizing neural network around initialization. This results in Linearized Neural Value Iteration, defined in terms of value function (Section 3.3). Further, one-step gradient descent on a squared objective w.r.t. linearized value function results in a kernel-based value function, w.r.t. the Neural Tangent Kernel (Section 4).

### 3.1 Asynchronous Q-iteration

In the DQN algorithm a random subset of state-action pairs is updated at each time step, called mini-batch. In the following, we show that it results in a standard Q-iteration with a smooth Bellman operator $\mathcal{T}_\beta$ in expectation, where the coefficient $\beta$ is given by a probability of sampling a state-action pair from a replay buffer at each time-step. To show this, define an asynchronous Bellman operator for a Bernoulli random variable $y \sim \text{Bern}(\beta)$

$$[\hat{\mathcal{T}}_\beta]Q(s,a) := \begin{cases} [\mathcal{T}Q](s,a), & \text{if } y_{s,a} = 1 \\ Q(s,a), & \text{else.} \end{cases} \tag{2}$$

Taking $\beta = m/M$, where $m$ is a size of a mini-batch and $M$ is a size of a replay buffer, this operator formalizes the asynchronous process of the DQN. In expectation, we obtain the smooth Bellman operator

$$\mathcal{T}_\beta = \mathbb{E}_y[\hat{\mathcal{T}}_\beta] = \mathbb{E}_y[(1-y)I + y\mathcal{T}] = (1-\beta)I + \beta\mathcal{T}. \tag{3}$$

Therefore, the DQN algorithm implements on average the Q-iteration with a smooth Bellman operator $Q_{k+1} \leftarrow \mathcal{T}_\beta Q_k$. Alternatively, it can be seen as a standard Q-iteration at scale $T = 1/\beta$, i.e. on average the update every $T$ time-steps corresponds to the standard Bellman update.

### 3.2 Neural asynchronous Q-iteration

In the DQN algorithm, the Q-function is represented by a neural network $Q_\theta$, $\theta \in \mathbb{R}^d$. This motivates the following scheme

$$\text{(Neural async QI)} \quad \theta_{k+1} \leftarrow \arg\min_{\theta \in \mathbb{R}^d} \|Q_\theta - \mathcal{T}_\beta Q_k\|_2^2, \; Q_{k+1} \leftarrow Q_{\theta_{k+1}}, \tag{4}$$

that iteratively fits a Q-function, represented by a neural network with parameters $\theta$, into a smooth Bellman target that corresponds to an average mini-batch update.

### 3.3 Linearized Neural Value Iteration

In the DQN algorithm, the parameters of Q-function are continuously updated at each time step using a stochastic mini-batch gradient descent, e.g. $\theta_{k+1} \leftarrow \theta_k - \eta \nabla_\theta Q_\theta^T (Q_\theta - \mathcal{T}Q_k)\big|_{\theta=\theta_k}$, where $\eta > 0$ is a (small) learning rate. This setup is referred in the literature as "re-using" the Q-function.

In the following, we will use value function notation. Consider parametrized value function $V_\theta : \theta \in \mathbb{R}^d \to V \in \mathbb{R}^{\mathcal{S}}$. Consider linearization of value function around initialization $V_\theta = V_0 + DV_0(\theta - \theta_0)$, where $\theta_0 \in \mathbb{R}^d$ is an initial vector of parameters, $V_0 = V_{\theta_0}$ is an initial value function, $DV_0 : \mathcal{S} \to \mathbb{R}^d$ denotes the gradient of $V_\theta$ at a given state evaluated at $\theta = \theta_0$. For example, over-parametrized neural networks can be viewed as linear model around random initialization sampled from standard Gaussian distribution Jacot et al. (2018), i.e. $\theta_0 \sim \mathcal{N}(0, \sigma^2 I)$. More generally, parametrized functions in lazy regime behave as linear models around initial parameter values under gradient descent training Chizat et al. (2018).

Define a space of linearized value functions around initialization $\mathcal{F}_{\text{lin}} = \{V = V_0 + DV_0(\theta - \theta_0), \theta \in \mathbb{R}^d\}$, where $DV_0 : \mathcal{S} \to \mathbb{R}^d$ is a gradient of the value function w.r.t. parameters, evaluated at $\theta = \theta_0$. This suggests the following value iteration algorithm

$$\text{(Linearized VI)} \quad V_{k+1} \leftarrow \arg\min_{V \in \mathcal{F}_{\text{lin}}} \|V - \mathcal{T}V_k\|_2^2. \tag{5}$$

Each iteration is a functional regression of Bellman update to a set of linearized functions. In parameter space, each iteration is a quadratic optimization problem w.r.t. $\theta$

$$\text{(Linearized VI)} \quad \theta_{k+1} \leftarrow \arg\min_{\theta \in \mathbb{R}^d} \|V_0 + DV_0(\theta - \theta_0) - \mathcal{T}V_k\|_2^2, V_{k+1} \leftarrow V_{\theta_{k+1}}. \tag{6}$$

One-step gradient descent on the above regression objective results in the following update

$$\theta_{k+1} = \theta_0 - \eta(\nabla_\theta V_\theta^T (V_0 + DV_0(\theta - \theta_0) - \mathcal{T}V_k))\big|_{\theta = \theta_0} = \theta_0 + \eta \nabla V_{\theta_0}^T (\mathcal{T}V_k - V_0). \tag{7}$$

In value function space, we have

$$V_{\theta_{k+1}} = V_0 + \nabla V_{\theta_0}(\theta_{k+1} - \theta_0) = V_0 + \eta \nabla V_{\theta_0} \nabla V_{\theta_0}^T (\mathcal{T}V_k - V_0), \tag{8}$$

becomes

$$V_{k+1} = V_0 + \eta K_0 (\mathcal{T}V_k - V_0), \tag{9}$$

where $K_0 := \nabla V_{\theta_0} \nabla V_{\theta_0}^T$ is a tangent kernel at initialization, where $K_0(s, s') = \langle \nabla V_{\theta_0}(s), \nabla V_{\theta_0}(s') \rangle$.

# 4. Kernel Value Iteration

Let $V_0 = 0$. From (9), Linearized VI with one gradient step training defines (Tangent) Kernel Value Iteration

$$\text{(Kernel VI)} \quad V_{k+1} = K_0 \mathcal{T} V_k. \tag{10}$$

Kernel VI is defined w.r.t the kernel operator $K_0 : V \to V$, induced by kernel function $k_0 : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, e.g. tangent kernel of over-parametrized neural network at random initialization is given by $k_0(s, s') = \langle \nabla V_{\theta_0}(s), \nabla V_{\theta_0}(s') \rangle$ and $K_0 = \nabla V_{\theta_0} \nabla V_{\theta_0}^T$.

Note that Kernel VI with $K_0 = I$ corresponds to standard VI.

In the finite-sample case, $K_0 \in \mathbb{R}^{n \times n}$ is given by kernel Gram matrix $(K_0)_{ij} = k_0(s_i, s_j), i, j = [n]$ of a set of samples $\{s_i\}_{i=1}^n$.

In the following, we show convergence of Kernel VI with normalized positive definite kernel function (Proposition 1). In particular, convergence holds for Neural Tangent Kernel of a multi-layer ReLU neural network.

## 4.1 Convergence

Let state space $\mathcal{S} \subset \mathbb{R}^d$ be a compact set with normalized Euclidean measure, $\mu(\mathcal{S}) = 1$.

Consider a positive definite kernel function $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ with bounded norm, i.e. symmetric function that defines a valid dot-product in $\mathbb{R}^n$, $\langle a, a \rangle_K := \langle a, Ka \rangle > 0$, where $a \in \mathbb{R}^n$, $a \neq 0$, and $K_{ij} = k(s_i, s_j), i, j = [n]$ is a kernel Gram matrix of a set $\{s_i\}_{i=1}^n$.

By Mercer theorem, positive definite kernel function induces an integral operator $K : V \to V$, defined as

$$[KV](s) := \langle k(s, \cdot), V \rangle = \int_{\mathcal{S}} k(s, \cdot) V d\mu \quad \forall s \in \mathcal{S}. \tag{11}$$

Using standard construction, Kernel VI defines value iteration in the Reproducing Kernel Hilbert Space (RKHS), $\mathcal{H}_k = \overline{\text{span}(k(s, \cdot), s \in \mathcal{S})}$, induced by positive definite bounded kernel function.

Define the operator norm of integral operator $\|K\|_{op} := \inf\{c \geq 0 : \|KV\| \leq c\|V\|, V \in \mathbb{R}^{\mathcal{S}}\}$. Then, convergence of Kernel VI is given by the operator norm of $K$, namely, for any $V_1, V_2 \in \mathbb{R}^{\mathcal{S}}$

$$\|K\mathcal{T}V_1 - K\mathcal{T}V_2\| \leq \|K\|_{op}\|\mathcal{T}V_1 - \mathcal{T}V_2\|. \tag{12}$$

If $\|K\|_{op} \leq 1$, i.e. $K$ is a non-expanding operator, Kernel VI converges at a rate $\tilde{\gamma} := \|K\|_{op}\gamma < 1$, by Banach fixed-point theorem for contracting operator. The convergence is monotone since $K$ is a positive operator.

**Proposition 1** (Kernel Value Iteration convergence)**.** *Consider Kernel VI $V_{k+1} \leftarrow K\mathcal{T}V_k$, where $K : V \to V$ is integral operator* (11)*, induced by a positive definite kernel function $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$. Then, if $\|K\|_{op} = \sup_{s \in \mathcal{S}} \|k(s, \cdot)\|_1 \leq 1$, Kernel VI converges at a rate $\tilde{\gamma} := \|K\|_{op}\gamma$. In particular, convergence holds with normalized kernel function $k(s, s) = 1$.*

**Proof** See Appendix A.1.

Previous work has shown convergence of fitted value iteration with a set of function approximations, called averagers, given by an expectation over the Bellman targets Gordon (1995). In this work, Kernel VI with $K = \mathbb{E}$ converges, since the norm of expectation operator equals to 1. Proposition 1 states convergence of a larger class of linear operators, namely, integral operators induced by a normalized positive definite kernel function.

Proposition 1 extends to the finite-sample case with the empirical integral operator. Given $n$ independent samples $\{s_i\}_{i=1}^n$ from distribution $P_\mu$ induced by $\mu$, define empirical integral operator $K_n$ as kernel Gram matrix

$$(K_n)_{ij} = \frac{1}{n}k(s_i, s_j), \quad i, j = [n].$$

Then, $K_n$ is a finite-sample approximation of $K$ and Kernel VI with $K_n$ converges under conditions of Proposition 1 (see Appendix A.2).

**Neural Tangent Kernel** Neural Tangent Kernel (NTK) of a multi-layer ReLU neural network satisfies conditions of Proposition 1. NTK of a multi-layer ReLU neural network is given by a dot-product kernel function on a sphere, i.e. $k_{NTK}(s, s') = h_{NTK}(\langle s, s' \rangle)$, where $h_{NTK} : \mathbb{R} \to \mathbb{R}$ is related to arc-cosine kernel functions Bietti and Mairal (2019). In particular, NTK of a two-layer ReLU neural network is given by

$$h_{NTK}(t) := (1 + t)\kappa_0(t) + \kappa_1(t),$$

where $\kappa_0$ and $\kappa_1$ are arc-cosine kernel functions of degree 0 and 1. If $\mathcal{S} = \mathbb{S}^{d-1}$ is a $d$-dimensional unit sphere, then normalized NTK kernel function $h_{NTK}/h(1)$ is a normalized positive definite kernel function (Jacot et al., 2018, Proposition 2).

**Fixed point** By Banach fixed-point theorem, there exists and unique a fixed point $\tilde{V}^* \in \mathbb{R}^{\mathcal{S}}$ of $K\mathcal{T}$, i.e. $K\mathcal{T}\tilde{V}^* = \tilde{V}^*$. The difference between this fixed point and the optimal value function is upper-bounded $\|V^* - \tilde{V}^*\|_\infty \leq \epsilon$.

**Proof** It holds state-wise $|V^* - \tilde{V}^*| = |\mathcal{T}V^* - \mathcal{T}\tilde{V}^* + \mathcal{T}\tilde{V}^* - K\mathcal{T}\tilde{V}^*| \leq |\mathcal{T}V^* - \mathcal{T}\tilde{V}^*| + |\mathcal{T}\tilde{V}^* - K\mathcal{T}\tilde{V}^*| \leq \gamma\|V^* - \tilde{V}^*\|_\infty + |(I - K)\mathcal{T}\tilde{V}^*|$. Then, $\|V^* - \tilde{V}^*\|_\infty \leq \frac{1}{1-\gamma}\|(I - K)\mathcal{T}\tilde{V}^*\|_\infty$.

## 4.2 Error propagation

**Kernel Value Iteration with sampling** We analyse propagation of sampling noise due to the finite number of transitions used to compute Bellman operator. Define sampled Bellman operator $\hat{\mathcal{T}}V(s) := r(s, a) + \gamma V(s')$, where $a \sim \pi(\cdot|s) = \mathcal{G}(V)$[1], $s' \sim P(\cdot|s, a)$, such that $\mathbb{E}[\hat{\mathcal{T}}] = \mathcal{T}$. Kernel Value Iteration with sampled Bellman operator is given by

$$\text{(Kernel VI with sampling)} \quad V_{k+1} = K\hat{\mathcal{T}}V_k = K(\mathcal{T}V_k + \epsilon_{k+1}), \tag{13}$$

[1] Define Dirac measure $\delta_{a^*}$, where $a^* = \arg\max_a Q_V(\cdot, a)$. Then, $[\mathcal{T}V](s) = \mathbb{E}_{a \sim \pi(\cdot|s) = \mathcal{G}(V)}[Q_V(s, a)] := \int_{\mathcal{A}} Q_V(s, a)\delta_{a^*}(da) = Q_V(s, a^*)$, where $Q_V(s, a) := r(s, a) + \gamma\mathbb{E}_{s' \sim P(\cdot|s,a)}V(s')$.

where $\epsilon_k \in \mathbb{R}^{\mathcal{S}}$ is a noise vector, $\mathbb{E}[\epsilon_k] = 0$. The resulting noise term $\tilde{\epsilon}_k = K\epsilon_k$ is known to satisfy $\mathbb{E}[\tilde{\epsilon_k}] = 0$, as a linear combination of independent Gaussian random variables. Kernel VI does not increase the noise $\|K\epsilon_k\| \leq \|\epsilon_k\|$, since $\|K\|_{op} \leq 1$.

The error propagation of Kernel VI with sampling (13) is given by the standard approximate VI error bound Scherrer et al. (2015) with $\tilde{\gamma} := \|K\|_{op}\gamma$ and $\tilde{\epsilon}_k := K\epsilon_k$.

**Proposition 2** (Kernel Value Iteration with sampling error propagation)**.** *Let $V_0$ be initial value function. Consider Kernel VI with sampling* (13)*. Then, the $L_\infty$-distance at iteration $N$ to the fixed point $\tilde{V}^*$ of Kernel VI is given by*

$$\|V_N - \tilde{V}^*\|_\infty \leq \sum_{k=1}^{N} \tilde{\gamma}^{N-k} \|\tilde{\epsilon}_k\|_\infty + \tilde{\gamma}^N \|V_0 - \tilde{V}^*\|_\infty, \tag{14}$$

*where $\tilde{\gamma} := \|K\|_{op}\gamma$ and $\tilde{\epsilon}_k := K\epsilon_k$.*

**Proof** See Appendix A.3.

**Smooth Kernel Q-iteration** Motivated by the Neural asynchronous Q-iteration, we provide error bound of Smooth Kernel Q-iteration that combines (empirical) Kernel Q-iteration with smooth sampled Bellman operator.

Define empirical integral operator $K_n$ w.r.t. $n$ independent samples of state-action pairs $(s_i, a_i)$ from distribution $P_\mu$, as $[K_n Q](s,a) := \frac{1}{n}\sum_{i=1}^{n} k((s,a),(s_i,a_i))Q(s_i,a_i)$, and sampled Bellman operator $\hat{\mathcal{T}}Q(s,a) := r(s,a) + \gamma \max_{a'} Q(s',a')$, where $s' \sim P(\cdot|s,a)$. Then, Smooth Kernel Q-iteration is given by

$$\text{(Smooth Kernel QI)} \quad Q_{k+1} = K_n \hat{\mathcal{T}}_\beta Q_k, \tag{15}$$

where $\hat{\mathcal{T}}_\beta := (1-\beta)I + \beta\hat{\mathcal{T}}$ is a smooth sampled Bellman operator.

Define sampling noise vector $\epsilon_{k+1} := \hat{\mathcal{T}}Q_k - \mathcal{T}Q_k$ due to one-transition used to compute Bellman operator. Denote $\tilde{\gamma} := \|K_n\|_{op}\bar{\gamma}$, where $\bar{\gamma} := \beta\gamma + (1-\beta)$ is a contraction coefficient of smooth Bellman operator $\mathcal{T}_\beta$ Smirnova and Dohmatob (2020).

**Proposition 3** (Smooth Kernel Q-iteration error propagation)**.** *Let $Q_0$ be an initial Q-function. Consider Smooth Kernel QI* (15)*. Then, $L_\infty$-distance at iteration $N$ to the fixed point $\tilde{Q}^*$ of Kernel Q-iteration is given by*

$$\|Q_N - \tilde{Q}^*\|_\infty \leq \sum_{k=1}^{N} \tilde{\gamma}^{N-k} \|\beta\tilde{\epsilon}_k\|_\infty + \tilde{\gamma}^N \|Q_0 - \tilde{Q}^*\|_\infty, \tag{16}$$

*where $\tilde{\gamma} := \|K_n\|_{op}\bar{\gamma}$, $\bar{\gamma} := \beta\gamma + (1-\beta)$ and $\tilde{\epsilon}_k := K_n\epsilon_k$.*

**Proof** See Appendix A.3.

Smooth Kernel QI with empirical NTK kernel $K_n = K_{n,NTK}$ and smoothing coefficient $\beta = m/M$, given by the ratio of the size of a mini-batch $m$ to the size of a replay buffer $M$, corresponds to Neural asynchronous Q-iteration (4) with sampled Bellman operator. Since NTK of a multi-layer ReLU neural network satisfies conditions of Proposition 1, the induced NTK Kernel Q-iteration converges at a rate $\tilde{\gamma} := \|K_{n,NTK}\|_{op}\bar{\gamma} \leq \bar{\gamma} < 1$.

**Numerical illustration** We present numerical illustration of the error bound (16) of Smooth Kernel Value Iteration (15). We experiment on a stochastic gridworld problem, known as cliff walking problem Sutton and Barto (2018). We set $\gamma = 0.9$. We use $\beta = 0.1$ and normalized NTK kernel of a two-layer ReLU neural network $h_{NTK}(t)/2 = (t\kappa_0(t) + \kappa_1(t))/2$, where $\kappa_0$ and $\kappa_1$ are arc-cosine kernel functions of degree 0 and 1 Bietti and Mairal (2019).

For a finite state space $\mathcal{S} = \{s_i\}_{i=1}^n$, define a normalized discrete measure $\mu = \frac{1}{n}\sum_{i=1}^{n}\delta_{s_i}$. Then, integral operator is given by $[K_n V](s) = \frac{1}{n}\sum_{i=1}^{n} k(s,s_i)V(s_i)$. Equivalently, define kernel Gram matrix $(K_n)_{ij} = \frac{1}{n}k(s_i,s_j), i,j = [n]$, then Kernel VI in the finite state space writes as multiplication of kernel Gram matrix and per-state value vector $V_{k+1} \leftarrow K_n\mathcal{T}V_k$. Note that the operator norm of integral operator, induced by normalized kernel function, is upper-bounded by the operator norm of identity operator, i.e. $\|K_n\|_{op} \leq \|I\|_{op}$, that implies the same or faster convergence than standard Value Iteration.

Define a valid dot-product in $\mathcal{S}$ as $\langle s, s' \rangle := \mathbf{1}_{s=s'}$. Then, for normalized NTK kernel function $k_{NTK}(s_i, s_j) = h_{NTK}(\langle s_i, s_j \rangle)/h(1)$, the NTK Gram matrix is given by $(K_n)_{ij} = \frac{1}{n} \begin{cases} 1, & \text{if } i = j, \\ \frac{h(0)}{h(1)}, & \text{else} \end{cases}, i, j = [n]$. It can be shown that this matrix is positive definite. By spectral theorem, its eigendecomposition has positive real eigenvalues $K_n = U^{-1}\Lambda_n U$, where $\Lambda_n = \operatorname{diag}((\lambda_i)_{i=1}^n)$. In the eigenbasis of $K_n$, the convergence of Kernel VI $V_{k+1} \leftarrow K_n \mathcal{T} V_k = U^{-1}\Lambda_n U \mathcal{T} V_k \iff UV_{k+1} \leftarrow U\Lambda_n \mathcal{T} V_k$ is given by the largest eigenvalue of $K_n$, i.e. $\|\Lambda_n\|_{op} = \lambda_{\max}(K_n)$.

Figure 1 shows convergence to the fixed point in terms of $\|V_N - \tilde{V}^*\|_\infty$ across iterations averaged over 30 runs[2]. We plot theoretical convergence $C\tilde{\gamma}^N$, where the rate is given by $\tilde{\gamma} = \|K_n\|_{op}\bar{\gamma}$, $\bar{\gamma} := \beta\gamma + 1 - \beta$. Empirically, $\tilde{\gamma} \approx 0.19$, where $\bar{\gamma} = 0.99$ and the operator norm of the NTK Gram matrix $\|K_n\|_{op} \approx 0.19$. As can be seen from the Figure, the Smooth Kernel Value Iteration follows the theoretical rate. The rate $\tilde{\gamma} < \gamma$ results in a faster convergence than standard Value Iteration, to a related fixed point.

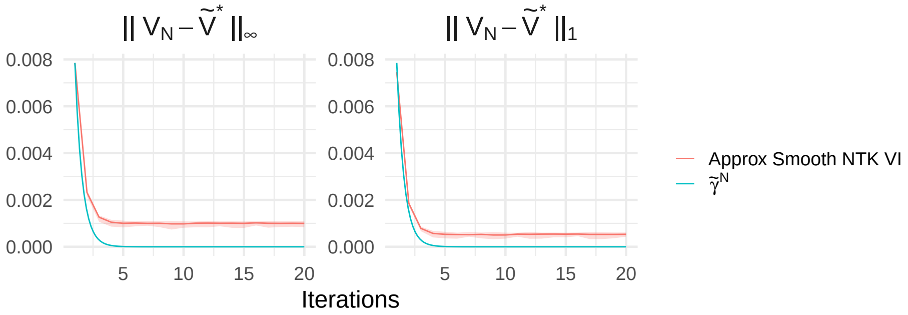

Figure 1: Performance of Smooth Kernel Value Iteration (15) with normalized NTK kernel on a stochastic gridworld problem across iterations averaged over 30 runs.

## 5. Conclusion

In this work, we showed implications of neural network function approximator in linear regime on Q-iteration algorithm, namely, contraction coefficient, fixed point and error propagation. We showed that, with appropriate normalization, the contraction coefficient of Q-iteration does not increase, the iteration converges to a related fixed point and the sampling noise is not increasing across iterations.

## References

Andrew G Barto, Steven J Bradtke, and Satinder P Singh. Learning to act using real-time dynamic programming. *Artificial intelligence*, 72(1-2):81–138, 1995.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pages 486–489. PMLR, 2020.

[2] We use random seeds in range $[0, 1000]$.

Geoffrey J Gordon. Stable function approximation in dynamic programming. In *Machine learning proceedings 1995*, pages 261–268. Elsevier, 1995.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Amir massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized fitted q-iteration for planning in continuous-space markovian decision problems. In *2009 American Control Conference*, pages 725–730. IEEE, 2009.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2):161–178, 2002.

Martin L Puterman. *Markov decision processes*. Wiley, New York, 1994.

Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer, 2005.

Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 16:1629–1676, 2015.

Elena Smirnova and Elvis Dohmatob. On the convergence of smooth regularized approximate value iteration schemes. *Advances in Neural Information Processing Systems*, 33:6540–6550, 2020.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2018.

Pan Xu and Quanquan Gu. A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pages 10555–10565. PMLR, 2020.

# Appendix A. Proofs

## A.1 Kernel Value Iteration

Define Kernel Value Iteration $V_{k+1} \leftarrow K\mathcal{T}V_k$, where $K : V \rightarrow V$ is an integral operator, induced by kernel function $k : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$.

**Proposition 4** (Kernel Value Iteration convergence)**.** *Consider Kernel VI $V_{k+1} \leftarrow K\mathcal{T}V_k$, where $K : V \rightarrow V$ is integral operator* (11)*, induced by a positive definite kernel function $k : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$. Then, if $\|K\|_{op} = \sup_{s\in\mathcal{S}} \|k(s,\cdot)\|_1 \leq 1$, Kernel VI converges at a rate $\tilde{\gamma} := \|K\|_{op}\gamma$. In particular, convergence holds with normalized kernel function $k(s,s) = 1$.*

**Proof** To show convergence of Kernel VI, we show that composition of operators $K\mathcal{T}$ defines a valid Bellman operator, i.e. satisfies contraction, monotonicity and distributivity properties (Puterman, 1994).

(Contraction) By Holder inequality, integral product is upper-bounded $|[KV](s)| \leq \|k(s,\cdot)\|_1\|V\|_\infty$. If $\|k(s,\cdot)\|_1 \leq 1, \forall s \in \mathcal{S}$, $K$ is a non-expansion operator in sup-norm with coefficient $\|K\|_{op} = \sup_{s\in\mathcal{S}} \|k(s,\cdot)\|_1$. Then, the composed operator $K\mathcal{T}$ remains a sup-norm contraction with coefficient $\|K\|_{op}\gamma$.

(Monotonicity) By property of a positive operator, $K$ is a monotone operator, i.e. if $V_1 \leq V_2$, $KV_1 \leq KV_2$. Then, the composed operator $K\mathcal{T}$ is also a monotone operator.

(Distributivity) The distributivity property holds for linear operators $K(V + c\mathbf{1}) = KV + cK\mathbf{1}$, $c \in \mathbb{R}$. Then, the composed operator $K\mathcal{T}$ also satisfies distributivity property.

Thus, $K\mathcal{T}$ is a valid Bellman operator. The induced value iteration, given by Kernel Value Iteration, converges to a fixed point at a rate $\|K\|_{op}\gamma$.

We show that normalized kernel function, i.e. kernel function that satisfies $k(s,s) = 1$, defines a kernel function with unit-bounded norm $\|k(s,\cdot)\|_1 \leq 1$. By property of a positive-definite kernel function (Cauchy-Schwartz inequality), $|k(s,s')|^2 \leq k(s,s)k(s',s') = 1$. Then, $\|k(s,\cdot)\|_1 \leq \int_{\mathcal{S}} |k(s,\cdot)|d\mu \leq 1$, since $\mu(\mathcal{S}) = \int_{\mathcal{S}} d\mu = 1$.

## A.2 Finite-sample case

Convergence of Kernel VI implies convergence of empirical Kernel VI in the finite-sample case.

**Proposition 5** (Empirical Kernel)**.** *Given $\{s_i\}_{i=1}^n$ independent samples from distribution $P_\mu$, induced by $\mu$, define empirical integral operator $K_n$ as a kernel Gram matrix $(K_n)_ij = \frac{1}{n}k(s_i, s_j)$, $i,j = [n]$. Then, $K_n$ is a finite-sample approximation of $K$ and Kernel VI with $K_n$ converges.*

**Proof** Define empirical measure $\mu_n$, centered at independent samples $\{s_i\}_{i=1}^n$ from distribution $P_\mu$, induced by $\mu$. Specifically, define a normalized discrete measure $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{s_i}$, where $\delta_{s_i}$ is a Dirac measure at a point $s_i$. Then, $[K_nV](s) = \int_{\mathcal{S}} k(s,\cdot)V d\mu_n = \frac{1}{n}\sum_{i=1}^n k(s,s_i)V(s_i)$. Equivalently, the empirical integral operator $K_n$ w.r.t. $\mu_n$ is given by the kernel Gram matrix $(K_n)_{ij} = \frac{1}{n}k(s_i,s_j)$, $i,j = [n]$.

As empirical measure converges to $\mu$ at increased number of samples, i.e. $\mu_n \xrightarrow[n\rightarrow\infty]{} \mu$, the empirical integral operator approximates the integral operator $K_n \xrightarrow[n\rightarrow\infty]{} K$ uniformly state-wise.

Convergence follows from Proposition 1 with empirical integral operator induced by normalized positive definite kernel.

## A.3 Error propagation

Consider propagation of sampling noise due to the sampling of Bellman operator in Kernel VI (Proposition 2) and Smooth Kernel VI (Proposition 3).

**Proposition 6** (Kernel Value Iteration with sampling error propagation)**.** *Let $V_0$ be initial value function. Consider Kernel VI with sampling* (13)*. Then, the $L_\infty$-distance at iteration $N$ to the fixed point $\tilde{V}^*$ of Kernel VI is given by*

$$\|V_N - \tilde{V}^*\|_\infty \leq \sum_{k=1}^{N} \tilde{\gamma}^{N-k} \|\tilde{\epsilon}_k\|_\infty + \tilde{\gamma}^N \|V_0 - \tilde{V}^*\|_\infty, \tag{17}$$

*where $\tilde{\gamma} := \|K\|_{op}\gamma$ and $\tilde{\epsilon}_k := K\epsilon_k$.*

**Proof** $|V_N(s) - \tilde{V}^*(s)| = |K\mathcal{T}V_{N-1}(s) + \tilde{\epsilon_N}(s) - K\mathcal{T}\tilde{V}^*(s)| \leq \tilde{\gamma}\|V_{N-1} - \tilde{V}^*\|_\infty + \|\tilde{\epsilon_N}\|_\infty$. Recursively, $|V_N(s) - \tilde{V}^*(s)| \leq \tilde{\gamma}^N \|V_0 - \tilde{V}^*\|_\infty + \sum_{k=1}^N \tilde{\gamma}^{N-k}\|\tilde{\epsilon}_k\|_\infty$.

**Proposition 7** (Smooth Kernel Q-iteration error propagation)**.** *Let $Q_0$ be an initial Q-function. Consider Smooth Kernel QI* (15)*. Then, $L_\infty$-distance at iteration $N$ to the fixed point $\tilde{Q}^*$ of Kernel Q-iteration is given by*

$$\|Q_N - \tilde{Q}^*\|_\infty \leq \sum_{k=1}^{N} \tilde{\gamma}^{N-k} \|\beta\tilde{\epsilon}_k\|_\infty + \tilde{\gamma}^N \|Q_0 - \tilde{Q}^*\|_\infty, \tag{18}$$

*where $\tilde{\gamma} := \|K_n\|_{op}\bar{\gamma}$, $\bar{\gamma} := \beta\gamma + (1-\beta)$ and $\tilde{\epsilon}_k := K_n\epsilon_k$.*

**Proof** By definition of smooth Bellman operator, $Q_{k+1} = K_n\hat{\mathcal{T}}_\beta Q_k = K_n((1-\beta)Q_k + \beta\hat{\mathcal{T}}Q_k) = K_n(\mathcal{T}_\beta Q_k + \beta\epsilon_{k+1})$. The bound follows from (14) with coefficient $\tilde{\gamma} = \|K_n\|_{op}\bar{\gamma}$ and sampling noise $\beta\tilde{\epsilon}_k$, taken for a Q-function.