NEURALUCB: CONTEXTUAL BANDITS WITH NEURAL NETWORK-BASED EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the stochastic contextual bandit problem, where the reward is generated from an unknown bounded function with additive noise. We propose the NeuralUCB algorithm, which leverages the representation power of deep neural networks and uses the neural network-based random feature mapping to construct an upper confidence bound (UCB) of reward for efficient exploration. We prove that, under mild assumptions, NeuralUCB achieves $\widetilde{O}(\sqrt{T})$ regret bound, where T is the number of rounds. To the best of our knowledge, our algorithm is the first neural network-based contextual bandit algorithm with near-optimal regret guarantee. Preliminary experiment results on synthetic data corroborate our theory, and shed light on potential applications of our algorithm to real-world problems.

1 INTRODUCTION

The stochastic contextual bandit problem has been extensively studied in machine learning: at round $t \in \{1, 2, \ldots\}$, an agent is presented with a set of K actions, each of which is associated with a d-dimensional feature vector. After choosing an action, the agent will receive a stochastic reward generated from some unknown distribution conditioned on the chosen action's feature vector. The goal of the agent is to maximize the expected cumulative rewards over total T rounds. Contextual bandit algorithms have been applied to many real-world applications, such as personalized recommendation, advertising and Web search (e.g., Agarwal et al., 2009; Li et al., 2010).

The most studied model in the literature is linear contextual bandits (Auer, 2002; Abe et al., 2003; Dani et al., 2008; Rusmevichientong and Tsitsiklis, 2010; Chu et al., 2011; Li et al., 2010; Abbasi-Yadkori et al., 2011), which assumes that the expected reward at each round is a linear function of the feature vector. Linear bandit algorithms have achieved great success in both theory and practice, such as news article recommendation (Li et al., 2010). However, the linear-reward assumption often fails to hold exactly in practice, which motivates the study of nonlinear contextual bandits (e.g., Filippi et al., 2010; Srinivas et al., 2010; Bubeck et al., 2011; Valko et al., 2013). However, they still require fairly strong assumptions on the reward function. For instance, Valko et al. (2013) assume the reward function is in some predefined Reproducing Kernel Hilbert Space (RKHS), and Bubeck et al. (2011) require it to have a Lipschitz continuous property. Therefore, these algorithms are not very practical since the RKHS or the metric space is often unknown.

In order to overcome the above shortcomings, deep neural networks (DNNs) (LeCun et al., 2015) have been introduced to learn the underlying reward function in contextual bandit problem, thanks to their strong representation power. Given the fact that DNNs enable the agent to make use of nonlinear models with less domain knowledge, existing work (Riquelme et al., 2018; Zahavy and Mannor, 2019) focuses on the idea called *neural-linear bandit*. More precisely, they use the first $L - 1$ layers of a DNN as a feature map, which transforms contexts from the raw input space to a low-dimensional space, usually with better representation and less frequent update. Then they learn a linear exploration policy on top of the last hidden layer of the DNN with a more frequent update. These attempts have achieved great empirical success. However, none of these work has provided a theoretical guarantee on the regret of the algorithms.

In this paper, we take the first step towards provable efficient contextual bandit algorithms based on deep neural networks. Specifically, we propose a new algorithm, NeuralUCB, which uses a deep neural network to learn the underlying reward function. At the core of the algorithm is an upper confidence bound constructed by deep neural network-based random feature mappings. Our regret

analysis of NeuralUCB is built on recent results on optimization and generalization of deep neural networks (Jacot et al., 2018; Arora et al., 2019; Cao and Gu, 2019a). While the main focus of our paper is mostly theoretical, we also carry out proof-of-concept experiments on synthetic data to validate the effectiveness of our proposed algorithm.

Our contributions are summarized as follows:

- We propose a neural contextual bandit algorithm using neural network-based exploration. It can be regarded as an extension of LinUCB (Li et al., 2010) and OFUL (Abbasi-Yadkori et al., 2011), from linear reward function to any bounded reward function.
- We prove that, under mild assumptions, our algorithm is able to achieve a $\widetilde{O}(\widetilde{d}\sqrt{T})$ regret, where $\widetilde{d}$ is the effective dimension of a neural tangent kernel matrix and T is the number of rounds. Our regret bound recovers the $\widetilde{O}(d\sqrt{T})$ regret for linear contextual bandit as a special case (Abbasi-Yadkori et al., 2011), where d is the dimension of context.
- We provide empirical evidence in several proof-of-concept experiments to demonstrate potential applications of our algorithm to real-world problems.

Notation: Scalars are denoted by lower case letters, vectors by lower case bold face letters, and matrices by upper case bold face letters. For a positive integer k, $[k]$ denotes $\{1, \ldots, k\}$. For a vector $\boldsymbol{\theta} \in \mathbb{R}^d$, we denote its ℓ_2 norm by $\|\boldsymbol{\theta}\|_2 = \sqrt{\sum_{i=1}^d \theta_i^2}$. For a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, we denote its spectral norm, Frobenius norm, and (i,j)-th entry by $\|\mathbf{A}\|_2$, $\|\mathbf{A}\|_F$, and $[\mathbf{A}]_{i,j}$, respectively. We denote a sequence of vectors by $\{\boldsymbol{\theta}_j\}_{j=1}^t$, and similarly for matrices. For two sequences $\{a_n\}$ and $\{b_n\}$, we use $a_n = O(b_n)$ to denote that there exists some constant $C > 0$ such that $a_n \leq Cb_n$, $a_n = \Omega(b_n)$ to denote that there exists some constant $C' > 0$ such that $a_n \geq C'b_n$. In addition, we use $\widetilde{O}(\cdot)$ to hide logarithmic factors. We say a random variable X is ν-sub-Gaussian if $\mathbb{E}\exp(\lambda(X - \mathbb{E}X)) \leq \exp(\lambda^2\nu^2/2)$ for any $\lambda > 0$.

2 Related Work

2.1 Contextual Bandits

There is a line of extensive work on linear bandits (e.g., Auer, 2002; Abe et al., 2003; Dani et al., 2008; Rusmevichientong and Tsitsiklis, 2010; Li et al., 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011). For the setting with finitely many arms, Abe et al. (2003) formalized the linear bandit setting and analyzed some of the earliest algorithms. Auer (2002) proposed SupLinRel algorithm that achieves $\widetilde{O}\sqrt{dT}$ regret. Chu et al. (2011) obtained the same regret with SupLinUCB that is based on LinUCB (Li et al., 2010); the authors also provided an lower bound of $\Omega(\sqrt{dT})$. For the other setting with infinitely many arms, a few authors (Dani et al., 2008; Rusmevichientong and Tsitsiklis, 2010; Abbasi-Yadkori et al., 2011) proposed algorithms that achieve $\widetilde{O}(d\sqrt{T})$ regret. Dani et al. (2008) also showed an $\Omega(d\sqrt{T})$ lower bound on the regret in this setting.

While most algorithms above are based on the idea of upper confidence bounding, it is also possible to use proper randomization to achieve strong regret guarantees, such as Thompson sampling and reward perturbation (Thompson, 1933; Chapelle and Li, 2011; Agrawal and Goyal, 2013; Russo and Van Roy, 2014; 2016; Kveton et al., 2019).

To deal with nonlinearity, generalized linear bandit has been considered, which assumes that the reward function can be written as a composition of a linear function and a link function. In particular, Filippi et al. (2010) proposed a GLM-UCB algorithm, which attains $\widetilde{O}(d\sqrt{T})$ regret. Li et al. (2017) proposed SupCB-GLM for generalized contextual bandit problems and showed a $\widetilde{O}(\sqrt{dT})$ regret that matches the lower bound. Jun et al. (2017) studied how to scale up algorithms for GLM bandits.

A few authors have also explored more general nonlinear bandits without making strong modeling assumptions. One line of work is variants of expert learning algorithms (Auer et al., 2002), which typically has time complexity linear in the number of experts (roughly exponential in number of parameters). Another approach is to reduce a bandit problem into supervised learning, starting from the epoch-greedy algorithm (Langford and Zhang, 2008) that has an $O(T^{2/3})$ regret. Later,

Agarwal et al. (2014) develop an algorithm that yields a near-optimal regret bound, but relies on an optimization oracle that can be expensive. A third approach uses nonparametric modeling, such as Gaussian process and kernels (Srinivas et al., 2010; Krause and Ong, 2011; Valko et al., 2013). More specifically, Srinivas et al. (2010) assumed that the reward function is generated from a Gaussian process with known mean and covariance functions. They proposed a GP-UCB algorithm which achieves $\widetilde{O}(\sqrt{T\gamma_T})$ regret, where γ_T is the maximum information gain. Krause and Ong (2011) assumed the reward function is defined over the join space of contexts and arms and proposed a Contextual GP-UCB in this setting. Valko et al. (2013) assumed that the reward function lies in a RKHS defined by some known kernel function with bounded RKHS norm. They proposed a SupKernelUCB algorithm and showed a $\widetilde{O}(\sqrt{\widetilde{d}T})$ regret, where $\widetilde{d}$ is effective dimension of the kernel that can be seen as an generalized notion of the dimension of contexts. There is also work focusing on bandit problems in general metric space with Lipschitz continuous property on the context (Kleinberg et al., 2008; Bubeck et al., 2011).

2.2 Neural Networks

Different lines of research have been done to provide theoretical understandings of DNNs from different aspects. For example, to understand how the expressive power of DNNs are related to their architecture, Telgarsky (2015; 2016); Liang and Srikant (2016); Yarotsky (2017; 2018); Hanin (2017) showed that deep neural networks can express more function classes than shallow networks. Lu et al. (2017); Hanin and Sellke (2017) suggested that the width of neural networks is crucial to improve the expressive power of neural networks. For the optimization of DNNs, a series of work have been proposed to show that (stochastic) gradient descent can find the global minima of training loss (Li and Liang, 2018; Du et al., 2019b; Allen-Zhu et al., 2019; Du et al., 2019a; Zou et al., 2019; Zou and Gu, 2019). For the generalization of DNNs, a series of work (Daniely, 2017; Cao and Gu, 2019b;a; Arora et al., 2019) have been proposed to show that by using (stochastic) gradient descent, the parameters of a DNN are located in a particular regime and the generalization bound of DNNs can be characterized by the best function in the corresponding neural tangent kernel space (Jacot et al., 2018).

3 Problem Setting

We consider the stochastic K-armed contextual bandit problem, where the total number of rounds T is known. At round $t \in [T]$, the agent observes the tth context consisting of K feature vectors: $\{\mathbf{x}_{t,a} \in \mathbb{R}^d \mid a \in [K]\}$. The agent selects an action a_t and receive a reward r_{t,a_t}. For simplicity, we denote $\{\mathbf{x}^i\}_{i=1}^{TK}$ as the collection of $\{\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{T,K}\}$. Our goal is to maximize the following pseudo regret (Audibert et al., 2009):

$$R_T = \mathbb{E}\bigg[\sum_{t=1}^{T}(r_{t,a_t^*} - r_{t,a_t})\bigg], \tag{3.1}$$

where a_t^* is the action which maximizes the expected reward at round t, i.e., $a_t^* = \mathrm{argmax}_{a\in[K]}\, \mathbb{E}[r_{t,a}]$. In this paper, we simply use *regret* to refer to the pseudo regret in (3.1).

This work makes the following assumption on reward generation: for any round t,

$$r_{t,a_t} = h(\mathbf{x}_{t,a_t}) + \xi_t, \tag{3.2}$$

where h is some unknown function satisfying $0 \le h(\mathbf{x}) \le 1$ for any $\mathbf{x}$, and ξ_t is ν-sub-Gaussian noise conditioned on $\mathbf{x}_{1,a_1}, \ldots, \mathbf{x}_{t-1,a_{t-1}}$. Note that the ν-sub-Gaussian noise assumption for ξ_t is a standard assumption in stochastic bandit literature (Abbasi-Yadkori et al., 2011); in particular, any bounded noise satisfies such an assumption. It is worth noting that we do not impose any structural assumption on reward function $h(\mathbf{x})$, unlike those made in the literature for linear bandits, generalized linear bandits, RKHS realizability, etc. In other words, our reward function class contains the function classes of linear, generalized linear, Gaussian process and bounded RKHS norm.

In order to learn the reward function h in (3.2), we propose to use a fully connected deep neural networks with depth $L \ge 2$:

$$f(\mathbf{x};\boldsymbol{\theta}) = \sqrt{m}\cdot \mathbf{W}_L\sigma\Big(\mathbf{W}_{L-1}\sigma\big(\cdots\sigma(\mathbf{W}_1\mathbf{x})\big)\Big), \tag{3.3}$$

where $\sigma(x) = \max\{x, 0\}$ is the rectified linear unit (ReLU) activation function, $\mathbf{W}_1 \in \mathbb{R}^{m\times d}, \mathbf{W}_i \in \mathbb{R}^{m\times m}, 2 \leq i \leq L-1, \mathbf{W}_L \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{\theta} = [\text{vec}(\mathbf{W}_1)^\top, \ldots, \text{vec}(\mathbf{W}_L)^\top]^\top \in \mathbb{R}^p$ with $p = m + md + m^2(L-1)$. Without loss of generality, we assume that the width of each hidden layer is the same (i.e., m) for convenience in analysis. We denote the gradient of the neural network function by $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}^p$.

4 THE NEURALUCB ALGORITHM

We present in Algorithm 1 our algorithm, NeuralUCB. The key idea is to use the gradient of the initial neural network $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta}_0)$ as a random feature mapping, together with upper confidence bound-based exploration used in Li et al. (2010); Chu et al. (2011); Abbasi-Yadkori et al. (2011).

In particular, Algorithm 1 first initializes $\boldsymbol{\theta}_0 \in \mathbb{R}^p$ by randomly generates each entry of $\boldsymbol{\theta}_0$ from appropriate Gaussian distributions. With $\boldsymbol{\theta}_0$, we define a feature mapping through the network gradient: $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}_0)/\sqrt{m}$. At round t, Algorithm 1 observes the context set for each action $\{\mathbf{x}_{t,a}\}_{a=1}^K$. Then, it chooses action a_t by using upper confidence bound-based exploration in (4.1), and receives the corresponding reward r_{t,a_t}. At the end of round t, it constructs a new confidence set $\mathcal{C}_t$ as in (4.2).

Algorithm 1 NeuralUCB

1: **Input:** number of rounds T, regularization parameter λ, exploration parameter ν, confidence parameter δ, norm parameter S, network width m, network depth L
2: **Initialization:** Generate each entry of $\mathbf{W}_l$ independently from $N(0, 2/m)$ for $1 \leq l \leq L-1$, and each entry of $\mathbf{W}_L$ independently from $N(0, 1/m)$. Define $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}_0)/\sqrt{m}$, where $\boldsymbol{\theta}_0 = [\text{vec}(\mathbf{W}_1)^\top, \ldots, \text{vec}(\mathbf{W}_L)^\top]^\top \in \mathbb{R}^p$
3: $\mathbf{Z}_0 = \lambda\mathbf{I},\ \mathbf{b}_0 = \mathbf{0}$
4: **for** $t = 1, \ldots, T$ **do**
5: Observe $\{\mathbf{x}_{t,a}\}_{a=1}^K$ and compute

$$(a_t, \widetilde{\boldsymbol{\theta}}_{t,a_t}) = \underset{a\in[K], \boldsymbol{\theta}\in\mathcal{C}_{t-1}}{\text{argmax}} \langle \boldsymbol{\phi}(\mathbf{x}_{t,a}), \boldsymbol{\theta} - \boldsymbol{\theta}_0 \rangle \tag{4.1}$$

6: Play a_t and receive reward r_{t,a_t}
7: Compute

$$\mathbf{Z}_t = \mathbf{Z}_{t-1} + \boldsymbol{\phi}(\mathbf{x}_{t,a_t})\boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top \in \mathbb{R}^{p\times p}, \quad \mathbf{b}_t = \mathbf{b}_{t-1} + r_{t,a_t}\boldsymbol{\phi}(\mathbf{x}_{t,a_t}) \in \mathbb{R}^p$$

8: Compute $\boldsymbol{\theta}_t = \mathbf{Z}_t^{-1}\mathbf{b}_t + \boldsymbol{\theta}_0 \in \mathbb{R}^p$
9: Construct $\mathcal{C}_t$ as

$$\mathcal{C}_t = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}_t - \boldsymbol{\theta}\|_{\mathbf{Z}_t} \leq \gamma_t\}, \quad \text{where} \quad \gamma_t = \nu\sqrt{\log\frac{\det \mathbf{Z}_t}{\det \lambda\mathbf{I}} - 2\log\delta} + \sqrt{\lambda}S \tag{4.2}$$

10: **end for**

Comparison to Existing Algorithms Here we compare NeuralUCB with other neural network based contextual bandit algorithms. Allesiardo et al. (2014) proposed NeuralBandit which consists of K neural networks. It uses a committee of networks to compute the score of each action and choose the action by ϵ-greedy policy. In contrast, our NeuralUCB uses upper confidence bound based exploration, which is more effective than ϵ-greedy. In addition, our algorithm only used one neural network instead of K neural networks, thus can be computationally more efficient.

Riquelme et al. (2018) proposed NeuralLinear, which uses the first $L-1$ layer of a L-layer DNN to learn a representation, then applies Thompson sampling on the last layer to choose action. Zahavy and Mannor (2019) proposed a NeuralLinear with limited memory (NeuralLinearLM), which also uses the first $L-1$ layer of a L-layer DNN to learn a representation and applies Thompson sampling on the last layer. Instead of computing the exact mean and variance in Thompson sampling, NeuralLinearLM only computes their approximation. Unlike NeuralLinear and NeuralLinearLM, NeuralUCB uses the entire DNN to learn the representation and constructs the upper confidence bound based on the random feature mapping defined by the neural network gradient.

Efficient Implementation Algorithm 1 can be implemented efficiently. At round t, we define

$$\mathbf{d}_{t,a} = \mathbf{Z}_{t-1}^{-1}\boldsymbol{\phi}(\mathbf{x}_{t,a}),$$

and we will discuss about how to compute $\mathbf{d}_{t,a}$ efficiently later in this section. Based on $\mathbf{d}_{t,a}$, we can update $\boldsymbol{\theta}_t$ by the Sherman-Morrison formula (Golub and Van Loan, 1996):

$$\begin{aligned}
\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 &= \mathbf{Z}_t^{-1}\mathbf{b}_t \\
&= \left(\mathbf{Z}_{t-1}^{-1} - \frac{\mathbf{Z}_{t-1}^{-1}\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top\mathbf{Z}_{t-1}^{-1}}{1+\phi(\mathbf{x}_{t,a_t})^\top\mathbf{Z}_{t-1}^{-1}\phi(\mathbf{x}_{t,a_t})}\right)(\mathbf{b}_{t-1} + r_{t,a_t}\boldsymbol{\phi}(\mathbf{x}_{t,a_t})) \\
&= \boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0 - \frac{\mathbf{d}_t(\boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top(\boldsymbol{\theta}_{t-1}-\boldsymbol{\theta}_0))}{1+\phi(\mathbf{x}_{t,a_t})^\top\mathbf{d}_t} + r_{t,a_t}\mathbf{d}_t - r_{t,a_t}\frac{\mathbf{d}_t(\boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top\mathbf{d}_t)}{1+\phi(\mathbf{x}_{t,a_t})^\top\mathbf{d}_t}. \quad (4.3)
\end{aligned}$$

We also update $\det(\mathbf{Z}_t)$ by matrix determinant lemma (Golub and Van Loan, 1996):

$$\det(\mathbf{Z}_t) = \det\left[\mathbf{Z}_{t-1} + \boldsymbol{\phi}(\mathbf{x}_{t,a_t})\boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top\right] = \left[1+\boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top\mathbf{Z}_{t-1}^{-1}\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\right]\det(\mathbf{Z}_{t-1}). \quad (4.4)$$

Note that both (4.3) and (4.4) only require to compute vector inner products, which only requires $O(p)$ time. Now we show how to compute $\mathbf{d}_{t,a}$ efficiently. By the definition of $\mathbf{d}_{t,a}$, we have

$$\boldsymbol{\phi}(\mathbf{x}_{t,a}) = \mathbf{Z}_{t-1}\mathbf{d}_{t,a} = \left(\lambda\mathbf{I} + \sum_{i=1}^{t-1}\boldsymbol{\phi}(\mathbf{x}_{i,a_i})\boldsymbol{\phi}(\mathbf{x}_{i,a_i})^\top\right)\mathbf{d}_{t,a}.$$

Thus, $\mathbf{d}_{t,a}$ is the global minimizer of the following convex optimization problem:

$$\min_{\mathbf{d}\in\mathbb{R}^p}\left\|\left(\lambda\mathbf{I} + \sum_{i=1}^{t-1}\boldsymbol{\phi}(\mathbf{x}_{i,a_i})\boldsymbol{\phi}(\mathbf{x}_{i,a_i})^\top\right)\mathbf{d} - \boldsymbol{\phi}(\mathbf{x}_{t,a})\right\|_2^2,$$

and we can use (stochastic) gradient descent to find $\mathbf{d}_{t,a}$ efficiently.

5 Regret Analysis

In this section, we present a regret analysis for Algorithm 1. Recall that $\{\mathbf{x}^i\}_{i=1}^{TK}$ is the collection of all $\{\mathbf{x}_{t,a}\}$. Since our regret analysis is built upon the recently proposed neural tangent kernel matrix (Jacot et al., 2018), here we provide its formal definition for completeness.

Definition 5.1 (Jacot et al. (2018); Cao and Gu (2019a))**.** For a set of contexts $\{\mathbf{x}^i\}_{i=1}^{TK}$, define

$$\begin{aligned}
\widetilde{\mathbf{H}}_{i,j}^{(1)} &= \boldsymbol{\Sigma}_{i,j}^{(1)} = \langle\mathbf{x}^i, \mathbf{x}^j\rangle, \\
\mathbf{A}_{i,j}^{(l)} &= \begin{pmatrix}\boldsymbol{\Sigma}_{i,i}^{(l)} & \boldsymbol{\Sigma}_{i,j}^{(l)} \\ \boldsymbol{\Sigma}_{i,j}^{(l)} & \boldsymbol{\Sigma}_{j,j}^{(l)}\end{pmatrix}, \\
\boldsymbol{\Sigma}_{i,j}^{(l)} &= 2\mathbb{E}_{(u,v)\sim N(\mathbf{0},\mathbf{A}_{i,j}^{(l)})}\sigma(u)\sigma(v), \\
\widetilde{\mathbf{H}}_{i,j}^{(l+1)} &= 2\widetilde{\mathbf{H}}_{i,j}^{(l)}\mathbb{E}_{(u,v)\sim N(\mathbf{0},\mathbf{A}_{i,j}^{(l)})}\sigma'(u)\sigma'(v) + \boldsymbol{\Sigma}_{i,j}^{(l+1)}.
\end{aligned}$$

Then, $\mathbf{H} = (\widetilde{\mathbf{H}}^{(L)} + \boldsymbol{\Sigma}^{(L)})/2$ is called the neural tangent kernel (NTK) matrix on the context set.

Based on Definition 5.1, we first lay out the assumption on the contexts $\{\mathbf{x}^i\}_{i=1}^{TK}$.

Assumption 5.2. For any $1 \le i \le TK$, $\|\mathbf{x}^i\|_2 \le 1$. Meanwhile, $\mathbf{H} \succeq \lambda_0\mathbf{I}$.

Assumption 5.2 says that the neural tangent kernel matrix is non-singular, which is a very mild assumption made in the related literature (Du et al., 2019a; Arora et al., 2019; Cao and Gu, 2019a). It can be satisfied as long as any pair of contexts in $\{\mathbf{x}^i\}_{i=1}^{TK}$ are not parallel (or identical).

Next we define the effective dimension $\widetilde{d}$ of the neural tangent kernel matrix on contexts $\{\mathbf{x}^i\}_{i=1}^{TK}$.

Definition 5.3. The effective dimension $\widetilde{d}$ of the neural tangent kernel matrix on contexts $\{\mathbf{x}^i\}_{i=1}^{TK}$ is defined as

$$\widetilde{d} = \frac{\log\det(\mathbf{I}+\mathbf{H}/\lambda)}{\log(1+TK)}. \quad (5.1)$$

Remark 5.4. The notion of effective dimension was introduced by Valko et al. (2013) for analyzing kernel contextual bandits, which was defined by the eigenvalues of any kernel matrix restricted on the given contexts. We adopt a similar but different definition from Yang and Wang (2019), which was used for the analysis of kernel-based Q learning. Suppose the effective dimension of the reproducing kernel Hilbert space induced by the given kernel is $\widehat{d}$ and the feature mapping $\boldsymbol{\psi}$ induced by the given kernel satisfies $\|\boldsymbol{\psi}(\mathbf{x})\|_2 \leq 1$ for any $\mathbf{x} \in \mathbb{R}^d$. Then it is easy to verify that if $\lambda \geq 1$, we always have $\widetilde{d} \leq \widehat{d}$ (See Appendix A.2 for the verification).

Now we are ready to present the main result, which provides the regret bound R_T of Algorithm 1.

Theorem 5.5. Let $\widetilde{d}$ be the effective dimension defined in Definition 5.3. Let $\mathbf{h} = [h(\mathbf{x}^i)]_{i=1}^{TK} \in \mathbb{R}^{TK}$. There exists some universal constant $C > 0$, such that for any $\delta \in (0, 1)$, if $m \geq CT^4K^4L^6\log(T^2K^2L/\delta)/\lambda_0^4$, and $S \geq \sqrt{2\mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}}$, then with probability at least $1-\delta$ over the random initialization of $\boldsymbol{\theta}_0$, the regret of Algorithm 1 satisfies

$$R_T \leq 8\sqrt{T}\bigg(\nu\sqrt{2\widetilde{d}\log(1+TK) - 2\log(\delta/3)} + \sqrt{\lambda}S\bigg)\sqrt{\widetilde{d}\log(1+TK)}, \tag{5.2}$$

where ν is the variance of sub-Gaussian noise in the reward model in (3.1).

Remark 5.6. It is worth noting that, simply applying results for linear bandit to our algorithm would lead to a linear dependence of p or $\sqrt{p}$ in the regret. Such a bound is vacuous since in our setting p would be very large compared with the number of rounds T and the input context dimension d. In contrast, our regret bound only depends on $\widetilde{d}$, which is much smaller than p.

Remark 5.7. Treating ν and λ as constants and taking $S = \sqrt{2\mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}}$, then the regret bound in (5.2) becomes $R_T = \widetilde{O}\Big(\sqrt{\widetilde{d}T}\sqrt{\max\{\widetilde{d}, \mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}\}}\Big)$. Specifically, if h belongs to the RKHS $\mathcal{H}$ induced by the neural tangent kernel with bounded RKHS norm $\|h\|_{\mathcal{H}}$, we have $\|h\|_{\mathcal{H}} = \sqrt{\mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}}$, and our regert bound can be further written as $R_T = \widetilde{O}\Big(\sqrt{\widetilde{d}T}\sqrt{\max\{\widetilde{d}, \|h\|_{\mathcal{H}}\}}\Big)$.

The high-probability result in Theorem 5.5 can be used to obtain a bound on the expected regret.

Corollary 5.8. Let $\widetilde{d}$ be the effective dimension in Definition 5.3. Let $\mathbf{h} = [h(\mathbf{x}^i)]_{i=1}^{TK}$. There exists some constant $C > 0$ such that, if $m \geq CT^4K^4L^6\log(T^2K^2L)/\lambda_0^4$ and $S \geq \sqrt{2\mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}}$, then

$$\mathbb{E}R_T \leq 8\sqrt{T}\bigg(\nu\sqrt{2\widetilde{d}\log(1+TK) + 2\log(3T)} + \sqrt{\lambda}S\bigg)\sqrt{\widetilde{d}\log(1+TK)} + 1.$$

6 Proof of the Main Results

This section provides the proof of Theorem 5.5. We first point out two main technical challenges in this proof:

- Unlike previous work (Chu et al., 2011; Abbasi-Yadkori et al., 2011; Filippi et al., 2010; Valko et al., 2013; Srinivas et al., 2010), we do not make any strong assumption on the reward function h such as linear realizability or belonging to some RKHS, which makes the regret analysis of NeuralUCB more difficult.
- In practice, the neural network is often overparametrized, which implies m is very big. Thus, we need to make sure the regret bound is independent of m.

The two challenges above are addressed by the following technical lemmas. Their proofs are found in the appendix.

Lemma 6.1. There exists some constant $C > 0$ such that for any $\delta \in (0, 1)$, if $m \geq CT^4K^4L^6\log(T^2K^2L/\delta)/\lambda_0^4$, then with probability at least $1-\delta$ over the random initialization of $\boldsymbol{\theta}_0$, there exists a $\boldsymbol{\theta}^* \in \mathbb{R}^p$ such that

$$h(\mathbf{x}^i) = \langle\boldsymbol{\phi}(\mathbf{x}^i), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle, \quad \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|_2 \leq \sqrt{2\mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}}, \tag{6.1}$$

for all $1 \leq i \leq TK$.

Lemma 6.1 suggests that with high probability, the reward function restricted on $\{\mathbf{x}^i\}_{i=1}^{TK}$ can be regarded as a linear function of $\boldsymbol{\phi}(\mathbf{x}^i)$. Equipped with Lemma 6.1, we can utilize existing results on linear bandits (Abbasi-Yadkori et al., 2011) to show that with high probability, $\boldsymbol{\theta}^*$ lies in the sequence of confidence sets constructed in our algorithm.

Lemma 6.2. Under the same conditions of Lemma 6.1, with probability at least $1-2\delta$, $\boldsymbol{\theta}^* \in \mathcal{C}_t$ for all $1 \leq t \leq T$, where $\mathcal{C}_t$ is defined in (4.2).

It is worth noting that γ_t in (4.2) has a term $\log|\det \mathbf{Z}_t|^{1/2}$. A trivial upper bound of $\log|\det \mathbf{Z}_t|^{1/2}$ would result in an dependence on the neural network width m, since the dimension of $\mathbf{Z}_t$ is $p = md + m^2(L-2) + m$. The next lemma establishes an upper bound which is independent of m, and is only related to effective dimension $\widetilde{d}$.

Lemma 6.3. Under the same conditions of Lemma 6.1, let $\widetilde{d}$ be the effective dimension in Definition 5.3. Let $\boldsymbol{\phi}(\cdot)$ be as defined in Algorithm 1. Then with probability at least $1-\delta$, we have

$$\sum_{t=1}^{T} \gamma_{t-1}^2 \min\bigg\{\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}}^2, 1\bigg\} \leq 4\bigg(\nu\sqrt{2\widetilde{d}\log(1+TK) - 2\log\delta} + \sqrt{\lambda}S\bigg)^2 \widetilde{d}\log(1+TK).$$

With Lemmas 6.1, 6.2 and 6.3, we are ready to provide a proof for our main result.

Proof of Theorem 5.5. Denote $a_t^* = \mathrm{argmax}_{a\in[K]}\, h(\mathbf{x}_{t,a})$. First we bound R_T as follows:

$$R_T = \sum_{t=1}^{T}\big[h\big(\mathbf{x}_{t,a_t^*}\big) - h\big(\mathbf{x}_{t,a_t}\big)\big] \leq \sqrt{T\sum_{t=1}^{T}\big[h\big(\mathbf{x}_{t,a_t^*}\big) - h\big(\mathbf{x}_{t,a_t}\big)\big]^2}, \tag{6.2}$$

where the inequality holds due to Cauchy-Schwarz inequality. By Lemma 6.2, with probability at least $1-2\delta$, for all $1 \leq t \leq T$, we have $\boldsymbol{\theta}^* \in \mathcal{C}_t$. Thus, $h(\mathbf{x}_{t,a_t^*}) - h(\mathbf{x}_{t,a_t})$ can be bounded as follows:

$$\begin{aligned}
&h(\mathbf{x}_{t,a_t^*}) - h(\mathbf{x}_{t,a_t}) \\
&= \langle\boldsymbol{\phi}(\mathbf{x}_{t,a_t^*}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle - \langle\boldsymbol{\phi}(\mathbf{x}_{t,a_t}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle \\
&\leq \langle\boldsymbol{\phi}(\mathbf{x}_{t,a_t}), \widetilde{\boldsymbol{\theta}}_{t,a_t} - \boldsymbol{\theta}_0\rangle - \langle\boldsymbol{\phi}(\mathbf{x}_{t,a_t}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle \\
&= \langle\boldsymbol{\phi}(\mathbf{x}_{t,a_t}), \widetilde{\boldsymbol{\theta}}_{t,a_t} - \boldsymbol{\theta}_{t-1}\rangle - \langle\boldsymbol{\phi}(\mathbf{x}_{t,a_t}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_{t-1}\rangle \\
&\leq \|\widetilde{\boldsymbol{\theta}}_{t,a_t} - \boldsymbol{\theta}_{t-1}\|_{\mathbf{Z}^{t-1}}\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}} + \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{t-1}\|_{\mathbf{Z}^{t-1}}\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}} \\
&\leq 2\gamma_{t-1}\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}},
\end{aligned} \tag{6.3}$$

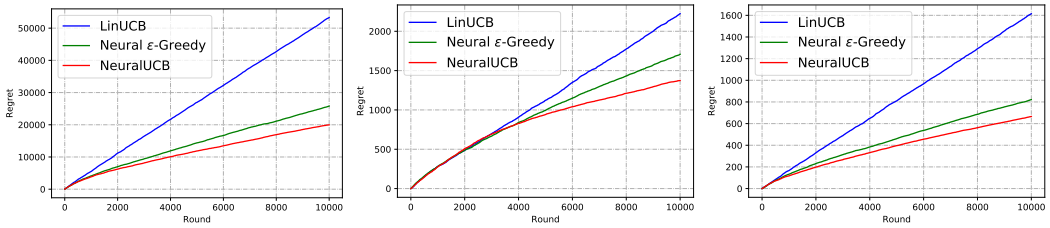
where the first inequality holds due to optimality condition in (4.1), the second inequality is by Cauchy-Schwarz inequality, and the third inequality holds by the definition of $\mathcal{C}_{t-1}$. Using the fact that $h(\mathbf{x}_{t,a_t^*}) - h(\mathbf{x}_{t,a_t}) \leq 1$ and (6.3), we have

$$h(\mathbf{x}_{t,a_t^*}) - h(\mathbf{x}_{t,a_t}) \leq \min\Big\{2\gamma_{t-1}\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}}, 1\Big\} \leq 2\gamma_{t-1}\min\Big\{\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}}, 1\Big\}. \tag{6.4}$$

Substituting (6.4) into (6.2), we have that, with probability at least $1-3\delta$,

$$\begin{aligned}
R_T &\leq 4\sqrt{T\sum_{t=1}^{T}\gamma_{t-1}^2\min\Big\{\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}}^2, 1\Big\}} \\
&\leq 8\sqrt{T}\bigg(\nu\sqrt{2\widetilde{d}\log(1+TK) - 2\log\delta} + \sqrt{\lambda}S\bigg)\sqrt{\widetilde{d}\log(1+TK)},
\end{aligned}$$

where the last inequality holds due to Lemma 6.3. Finally, by replacing δ with $\delta/3$, we complete the proof. □



(a) $h_1(\mathbf{x}) = 10(\mathbf{x}^\top \mathbf{a})^2$ (b) $h_2(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x}$ (c) $h_3(\mathbf{x}) = \cos(3\mathbf{x}^\top \mathbf{a})$

Figure 1: Comparison of LinUCB, Neural ϵ-Greedy and NeuralUCB.

7 Experiments

While our focus in this work is mostly on theoretical analysis of regret, we present results in proof-of-concept experiments in simulated problems. We compare it with two representative baselines: (1) LinUCB, and (2) Neural ϵ-Greedy, which replaces the UCB based exploration in Algorithm 1 by ϵ-greedy based exploration. We use the accumulated regret as the performance metric.

In our simulation, we use contextual bandit problems with context dimension $d = 20$, the number of actions $K = 4$ and the number of rounds $T = 10000$. The contextual vectors $\{\mathbf{x}_{1,1}, \ldots, \mathbf{x}_{T,K}\}$ are randomly chosen from $N(\mathbf{0}, \mathbf{I})$ and then normalized to have unit norm, i.e., $\|\mathbf{x}_{t,a}\|_2 = 1$. For the reward function h, we investigate the following nonlinear functions:

$$h_1(\mathbf{x}) = 10(\mathbf{x}^\top \mathbf{a})^2, \quad h_2(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x}, \quad h_3(\mathbf{x}) = \cos(3\mathbf{x}^\top \mathbf{a}),$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$ and each entry of $\mathbf{A}$ is randomly generated from $N(0, 1)$, $\mathbf{a}$ is randomly chosen from $N(\mathbf{0}, \mathbf{I})$ and normalized to have $\|\boldsymbol{\theta}^*\|_2 = 1$. For each $h_i(\cdot)$, the reward at round t for action a is generated by $r_{t,a} = h_i(\mathbf{x}_{t,a}) + \xi_t$, where ξ_t is independently drawn from $N(0, 1)$.

For LinUCB, we follow Li et al. (2010) to implement it with a constant radius α. We do a grid search for α over $\{0.01, 0.1, 1, 10\}$ and choose the best α for comparison. For NeuralUCB and Neural ϵ-Greedy, we choose a two-layer neural network $f(\mathbf{x}; \boldsymbol{\theta}) = \sqrt{m}\mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x})$ with network width $m = 20$, where $\boldsymbol{\theta} = [\text{vec}(\mathbf{W}_1)^\top, \text{vec}(\mathbf{W}_2)^\top] \in \mathbb{R}^p$ and $p = md + m$. For NeuralUCB, we choose $m = 20$, $L = 2$, $\nu = 1, \lambda = 1, \delta = 0.1$, so $p = 420$. For hyper-parameter S, we do a grid search over $\{0.01, 0.1, 1, 10\}$ and choose the best S for comparison. For Neural ϵ-Greedy, we do a grid search for ϵ over $\{0.001, 0.01, 0.1, 0.2\}$ and choose the best ϵ for comparison. For all the algorithms, we repeat the experiment for 10 runs and report the averaged results for comparison.

We plot the accumulative regret of LinUCB, Neural ϵ-Greedy and NeuralUCB in Figure 1, for reward function $h \in \{h_1, h_2, h_3\}$. We can see that due to the nonlinearity of reward function h, LinUCB fails to learn the true reward function and hence achieve an almost linear regret, as expected. In contrast, thanks to the neural network representation and efficient exploration, NeuralUCB achieves a sublinear regret which is much lower than that of LinUCB. The performance of Neural ϵ-Greedy is in-between. This suggests that while Neural ϵ-greedy can capture the nonlinearity of the underlying reward function, ϵ-Greedy based exploration is not as effective as UCB based exploration. This confirms the effectiveness of NeuralUCB for contextual bandit problems with any bounded (nonlinear) reward function. It is interesting to note that although the network width m in the experiment is not as large as our theory suggests, NeuralUCB still achieves a sublinear regret for nonlinear reward functions. We leave it as a future work to investigate the impact of m on regret.

8 Conclusions and Future Work

In this work, we proposed a new algorithm NeuralUCB for stochastic contextual bandit problems based on neural networks. We show that for arbitrary bounded reward function, our algorithm achieves $\widetilde{O}(\widetilde{d}\sqrt{T})$ regret bound. Our preliminary experiment results on synthetic data corroborate our theoretical findings. In the future, we are interested in a systematic empirical evaluation of NeuralUCB on real world datasets, and compare it with the state-of-the-art neural network based contextual bandit algorithms (without provable guarantee in regret) (Riquelme et al., 2018; Zahavy and Mannor, 2019). Another interesting direction is provably efficient exploration with neural network using other strategies like Thompson sampling.

REFERENCES

ABBASI-YADKORI, Y., PÁL, D. and SZEPESVÁRI, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*.

ABE, N., BIERMANN, A. W. and LONG, P. M. (2003). Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica* **37** 263–293.

AGARWAL, A., HSU, D., KALE, S., LANGFORD, J., LI, L. and SCHAPIRE, R. E. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*.

AGARWAL, D., CHEN, B.-C., ELANGO, P., MOTGI, N., PARK, S.-T., RAMAKRISHNAN, R., ROY, S. and ZACHARIAH, J. (2009). Online models for content optimization. In *Advances in Neural Information Processing Systems*.

AGRAWAL, S. and GOYAL, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*.

ALLEN-ZHU, Z., LI, Y. and SONG, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*.

ALLESIARDO, R., FÉRAUD, R. and BOUNEFFOUF, D. (2014). A neural networks committee for the contextual bandit problem. In *International Conference on Neural Information Processing*. Springer.

ARORA, S., DU, S. S., HU, W., LI, Z., SALAKHUTDINOV, R. and WANG, R. (2019). On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*.

AUDIBERT, J.-Y., MUNOS, R. and SZEPESVÁRI, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* **410** 1876–1902.

AUER, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* **3** 397–422.

AUER, P., CESA-BIANCHI, N., FREUND, Y. and SCHAPIRE, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* **32** 48–77.

BUBECK, S., MUNOS, R., STOLTZ, G. and SZEPESVÁRI, C. (2011). X-armed bandits. *Journal of Machine Learning Research* **12** 1655–1695.

CAO, Y. and GU, Q. (2019a). Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*.

CAO, Y. and GU, Q. (2019b). A generalization theory of gradient descent for learning over-parameterized deep relu networks. *arXiv preprint arXiv:1902.01384* .

CHAPELLE, O. and LI, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*.

CHU, W., LI, L., REYZIN, L. and SCHAPIRE, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*.

DANI, V., HAYES, T. P. and KAKADE, S. M. (2008). Stochastic linear optimization under bandit feedback .

DANIELY, A. (2017). Sgd learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*.

DU, S., LEE, J., LI, H., WANG, L. and ZHAI, X. (2019a). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*.

DU, S. S., ZHAI, X., POCZOS, B. and SINGH, A. (2019b). Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=S1eK3i09YQ

FILIPPI, S., CAPPE, O., GARIVIER, A. and SZEPESVÁRI, C. (2010). Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*.

GOLUB, G. H. and VAN LOAN, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

HANIN, B. (2017). Universal function approximation by deep neural nets with bounded width and ReLU activations. *arXiv preprint arXiv:1708.02691* .

HANIN, B. and SELLKE, M. (2017). Approximating continuous functions by ReLU nets of minimal width. *arXiv preprint arXiv:1710.11278* .

JACOT, A., GABRIEL, F. and HONGLER, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*.

JUN, K.-S., BHARGAVA, A., NOWAK, R. D. and WILLETT, R. (2017). Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems 30 (NIPS)*.

KLEINBERG, R., SLIVKINS, A. and UPFAL, E. (2008). Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM.

KRAUSE, A. and ONG, C. S. (2011). Contextual gaussian process bandit optimization. In *Advances in neural information processing systems*.

KVETON, B., SZEPESVÁRI, C., GHAVAMZADEH, M. and BOUTILIER, C. (2019). Perturbed-history exploration in stochastic linear bandits. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*.

LANGFORD, J. and ZHANG, T. (2008). The epoch-greedy algorithm for contextual multi-armed bandits. In *Advances in Neural Information Processing Systems 20 (NIPS)*.

LECUN, Y., BENGIO, Y. and HINTON, G. (2015). Deep learning. *nature* **521** 436.

LI, L., CHU, W., LANGFORD, J. and SCHAPIRE, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM.

LI, L., LU, Y. and ZHOU, D. (2017). Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

LI, Y. and LIANG, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*.

LIANG, S. and SRIKANT, R. (2016). Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161* .

LU, Z., PU, H., WANG, F., HU, Z. and WANG, L. (2017). The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*.

RIQUELME, C., TUCKER, G. and SNOEK, J. (2018). Deep bayesian bandits showdown. In *International Conference on Learning Representations*.

RUSMEVICHIENTONG, P. and TSITSIKLIS, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research* **35** 395–411.

RUSSO, D. and VAN ROY, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research* **39** 1221–1243.

RUSSO, D. and VAN ROY, B. (2016). An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research* **17** 2442–2471.

SRINIVAS, N., KRAUSE, A., KAKADE, S. and SEEGER, M. (2010). Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress.

TELGARSKY, M. (2015). Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101* .

TELGARSKY, M. (2016). Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485* .

THOMPSON, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **25** 285–294.

VALKO, M., KORDA, N., MUNOS, R., FLAOUNAS, I. and CRISTIANINI, N. (2013). Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869* .

YANG, L. F. and WANG, M. (2019). Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389* .

YAROTSKY, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Networks* **94** 103–114.

YAROTSKY, D. (2018). Optimal approximation of continuous functions by very deep ReLU networks. *arXiv preprint arXiv:1802.03620* .

ZAHAVY, T. and MANNOR, S. (2019). Deep neural linear bandits: Overcoming catastrophic forgetting through likelihood matching. *arXiv preprint arXiv:1901.08612* .

ZOU, D., CAO, Y., ZHOU, D. and GU, Q. (2019). Stochastic gradient descent optimizes over-parameterized deep relu networks. *Machine Learning* .

ZOU, D. and GU, Q. (2019). An improved analysis of training over-parameterized deep neural networks. In *Advances in Neural Information Processing Systems*.

A PROOF OF MAIN THEOREM

A.1 PROOF OF COROLLARY 5.8

Proof of Corollary 5.8. Notice that $R_T \leq T$ since $0 \leq h(\mathbf{x}) \leq 1$. Thus, with the fact that with probability at least $1-\delta$, (5.2) holds, we can bound $\mathbb{E}R_T$ as

$$\mathbb{E}R_T \leq (1-\delta)\left(8\sqrt{T}\left(\nu\sqrt{2\widetilde{d}\log(1+TK)-2\log\delta/2}+\sqrt{\lambda}S\right)\sqrt{\widetilde{d}\log(1+TK)}\right)+\delta T. \tag{A.1}$$

Taking $\delta = 1/T$, our statement holds. □

A.2 VERIFICATION OF THE CLAIM IN REMARK 5.4

Suppose there exists a mapping $\boldsymbol{\psi} : \mathbb{R}^d \to \mathbb{R}^{\widehat{d}}$ satisfying $\|\boldsymbol{\psi}(\mathbf{x})\|_2 \leq 1$ which maps any context $\mathbf{x} \in \mathbb{R}^d$ to the Hilbert space $\mathcal{H}$ associated with the Gram matrix $\mathbf{H} \in \mathbb{R}^{TK\times TK}$ over contexts $\{\mathbf{x}^i\}_{i=1}^{TK}$. Then $\mathbf{H} = \boldsymbol{\Psi}^\top\boldsymbol{\Psi}$, where $\boldsymbol{\Psi} = [\boldsymbol{\psi}(\mathbf{x}^1),\ldots,\boldsymbol{\psi}(\mathbf{x}^{TK})] \in \mathbb{R}^{\widehat{d}\times TK}$. Thus, we can bound the effective dimension $\widetilde{d}$ as follows

$$\widetilde{d} = \frac{\log\det[\mathbf{I}+\mathbf{H}/\lambda]}{\log(1+TK)} = \frac{\log\det\left[\mathbf{I}+\boldsymbol{\Psi}\boldsymbol{\Psi}^\top/\lambda\right]}{\log(1+TK)} \leq \widehat{d}\cdot\frac{\log\left\|\mathbf{I}+\boldsymbol{\Psi}\boldsymbol{\Psi}^\top/\lambda\right\|_2}{\log(1+TK)} \leq \widehat{d},$$

where the second equality holds due to the fact that $\det(\mathbf{I}+\mathbf{A}^\top\mathbf{A}/\lambda) = \det(\mathbf{I}+\mathbf{A}\mathbf{A}^\top/\lambda)$ holds for any matrix $\mathbf{A}$, the first inequality holds since $\det\mathbf{A} \leq \|\mathbf{A}\|_2^{\widehat{d}}$ for any $\mathbf{A} \in \mathbb{R}^{\widehat{d}\times\widehat{d}}$, the last inequality holds because

$$\left\|\mathbf{I}+\boldsymbol{\Psi}\boldsymbol{\Psi}^\top/\lambda\right\|_2 \leq 1+\left\|\boldsymbol{\Psi}\boldsymbol{\Psi}^\top\right\|_2 \leq 1+\sum_{i=1}^{TK}\left\|\boldsymbol{\psi}(\mathbf{x}^i)\boldsymbol{\psi}(\mathbf{x}^i)^\top\right\|_2 \leq 1+TK.$$

Here the first inequality is due to triangle inequality and the fact $\lambda \geq 1$, the second inequality holds due to the definition of $\boldsymbol{\Psi}$ and triangle inequality, and the last inequality is by $\|\boldsymbol{\psi}(\mathbf{x}^i)\|_2 \leq 1$ for any $1 \leq i \leq TK$.

B PROOF OF TECHNICAL LEMMAS

B.1 PROOF OF LEMMA 6.1

In order to prove Lemma 6.1, we need the following lemma:

Lemma B.1. Let $\mathbf{G} = [\boldsymbol{\phi}(\mathbf{x}^1),\ldots,\boldsymbol{\phi}(\mathbf{x}^{TK})] \in \mathbb{R}^{p\times(TK)}$. We denote the neural tangent kernel matrix $\mathbf{H}$ the same as Definition 5.1. For any $\delta \in (0,1)$, if

$$m = \Omega\left(\frac{T^4K^4L^6\log(T^2K^2L/\delta)}{\lambda_0^4}\right),$$

then with probability at least $1-\delta$ over the random initialization of $\boldsymbol{\theta}_0$, we have

$$\|\mathbf{G}^\top\mathbf{G}-\mathbf{H}\|_F \leq \lambda_0/3.$$

Proof of Lemma 6.1. By Assumption 5.2, we know that $\lambda_0 > 0$. By the choice of m and Lemma B.1, with probability at least $1-\delta$, $\mathbf{G}^\top\mathbf{G} \succeq 2\lambda_0\mathbf{I}/3 \succ 0$. Thus, suppose the singular value decomposition of $\mathbf{G}$ is $\mathbf{G} = \mathbf{P}\mathbf{A}\mathbf{Q}^\top$, $\mathbf{P} \in \mathbb{R}^{p\times TK}, \mathbf{A} \in \mathbb{R}^{TK\times TK}, \mathbf{Q} \in \mathbb{R}^{TK\times TK}$, we have $\mathbf{A} \succ 0$ and $\boldsymbol{\theta}^* = \boldsymbol{\theta}_0 + \mathbf{P}\mathbf{A}^{-1}\mathbf{Q}\mathbf{h}$ satisfies (6.1). To validate that $\boldsymbol{\theta}^*$ satisfies (6.1), first we have

$$\mathbf{G}^\top(\boldsymbol{\theta}^*-\boldsymbol{\theta}_0) = \mathbf{Q}\mathbf{A}\mathbf{P}^\top\mathbf{P}\mathbf{A}^{-1}\mathbf{Q}^\top\mathbf{h} = \mathbf{h},$$

which suggests that for any i, $\langle\boldsymbol{\phi}(\mathbf{x}^i),\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\rangle = h(\mathbf{x}^i)$. We also have

$$\|\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\|_2^2 = \mathbf{h}^\top\mathbf{Q}^\top\mathbf{A}^{-2}\mathbf{Q}\mathbf{h} = \mathbf{h}^\top\mathbf{G}^\top\mathbf{G}\mathbf{h} = \mathbf{h}^\top\mathbf{H}\mathbf{h}+\mathbf{h}^\top(\mathbf{G}^\top\mathbf{G}-\mathbf{H})\mathbf{h} \leq 2\mathbf{h}^\top\mathbf{H}\mathbf{h},$$

where the last inequality holds because $\mathbf{G}^\top\mathbf{G}-\mathbf{H} \preceq \mathbf{H}$, since

$$\mathbf{G}^\top\mathbf{G}-\mathbf{H} \preceq \|\mathbf{G}^\top\mathbf{G}-\mathbf{H}\|_F\mathbf{I} \preceq \lambda_0\mathbf{I} \preceq \mathbf{H},$$

where the second inequality holds due to Lemma B.1. Thus, our statement holds. □

B.2 PROOF OF LEMMA 6.2

Proof of Lemma 6.2. By Lemma 6.1, with probability at least $1-\delta$, there exists $\boldsymbol{\theta}^*$ such that for any $1 \le t \le T$,

$$h(\mathbf{x}_{t,a_t}) = \langle \boldsymbol{\phi}(\mathbf{x}_{t,a_t}), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0 \rangle,\ \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|_2 \le \sqrt{2\mathbf{h}^\top \mathbf{H}\mathbf{h}} \le S.$$

Thus, by Theorem 2 in Abbasi-Yadkori et al. (2011), with probability at least $1-2\delta$, $\boldsymbol{\theta}^* \in \mathcal{C}_t$ for any $1 \le t \le T$. □

B.3 PROOF OF LEMMA 6.3

To prove Lemma 6.3, we need the following lemma from Abbasi-Yadkori et al. (2011).

Lemma B.2 (Lemma 11, Abbasi-Yadkori et al. (2011))**.** We have the following inequality:

$$\sum_{t=1}^{T} \min\left\{ \|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|^2_{\mathbf{Z}_{t-1}^{-1}}, 1 \right\} \le 2 \log \frac{\det \mathbf{Z}_t}{\det \lambda \mathbf{I}}. \tag{B.1}$$

Proof of Lemma 6.3. First by the definition of γ_t, we know that γ_t is a monotonic function w.r.t. $\det \mathbf{Z}_t$. By the definition of $\mathbf{Z}_t$, we know that $\mathbf{Z}_T \succeq \mathbf{Z}_t$, which implies that $\det \mathbf{Z}_T \le \det \mathbf{Z}_t$. Thus, $\gamma_t \le \gamma_T$. Second, by Lemma B.2 we know that

$$\sum_{t=1}^{T} \min\left\{ \|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|^2_{\mathbf{Z}_{t-1}^{-1}}, 1 \right\} \le 2 \log \frac{\det \mathbf{Z}_T}{\det \lambda \mathbf{I}}. \tag{B.2}$$

Next we are going to bound $\log \det \mathbf{Z}_t$. Denote $\mathbf{G} = [\boldsymbol{\phi}(\mathbf{x}^1), \ldots, \boldsymbol{\phi}(\mathbf{x}^{TK})] \in \mathbb{R}^{p \times (TK)}$, then we have

$$\begin{aligned}
\log \frac{\det \mathbf{Z}_T}{\det \lambda \mathbf{I}} &= \log \frac{\det(\lambda \mathbf{I} + \sum_{t=1}^{T} \phi(\mathbf{x}_{t,a_t}))}{\lambda^p} \\
&= \log \det \left(\mathbf{I} + \sum_{t=1}^{T} \phi(\mathbf{x}_{t,a_t}) \phi(\mathbf{x}_{t,a_t})^\top / \lambda \right) \\
&\le \log \det \left(\mathbf{I} + \sum_{i=1}^{TK} \phi(\mathbf{x}^i) \phi(\mathbf{x}^i)^\top / \lambda \right) \\
&= \log \det \left(\mathbf{I} + \mathbf{G}\mathbf{G}^\top / \lambda \right) \\
&= \log \det \left(\mathbf{I} + \mathbf{G}^\top \mathbf{G} / \lambda \right),
\end{aligned} \tag{B.3}$$

where the fourth equality holds since for any matrix $\mathbf{A} \in \mathbb{R}^{p \times TK}$, we have $\det(\mathbf{I} + \mathbf{A}\mathbf{A}^\top) = \det(\mathbf{I} + \mathbf{A}^\top \mathbf{A})$. We can can be further bound (B.3) by the follows:

$$\begin{aligned}
\log \det \left(\mathbf{I} + \mathbf{G}^\top \mathbf{G} / \lambda \right) &= \log \det \left(\mathbf{I} + \mathbf{H}/\lambda + (\mathbf{G}^\top \mathbf{G} - \mathbf{H})/\lambda \right) \\
&\le \log \det \left(\mathbf{I} + \mathbf{H}/\lambda \right) + \langle (\mathbf{I} + \mathbf{H}/\lambda)^{-1}, (\mathbf{G}^\top \mathbf{G} - \mathbf{H})/\lambda \rangle \\
&\le \log \det \left(\mathbf{I} + \mathbf{H}/\lambda \right) + \|(\mathbf{I} + \mathbf{H}/\lambda)^{-1}\|_2 \|\mathbf{G}^\top \mathbf{G} - \mathbf{H}\|_F / \lambda \\
&\le \log \det \left(\mathbf{I} + \mathbf{H}/\lambda \right) + \lambda \cdot \lambda_0 \|\mathbf{G}^\top \mathbf{G} - \mathbf{H}\|_F / \lambda \\
&\le \widetilde{d} \log(1 + TK) + 1 \\
&\le 2\widetilde{d} \log(1 + TK),
\end{aligned} \tag{B.4}$$

where the first inequality holds due to the concavity of $\log\det(\cdot)$, the third inequality holds due to the fact that $\mathbf{I} + \mathbf{H}/\lambda \succeq \mathbf{H}/\lambda$. The fourth inequality holds by Lemma B.1 and the definition of effective dimension in Definition 5.3. Finally, substituting (B.4) into (B.3) and using (B.2), we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \gamma_{t-1}^2 \min\bigg\{\|\boldsymbol{\phi}(\mathbf{x}_{t,a_t})\|_{\mathbf{Z}_{t-1}^{-1}}^2, 1\bigg\} \\
&\leq 2\gamma_T^2 \log\det \mathbf{Z}_T \\
&= 2\bigg(\nu\sqrt{\log\frac{\det \mathbf{Z}_t}{\delta^2}} + \sqrt{\lambda}S\bigg)^2 \log\det \mathbf{Z}_T \\
&\leq 4\bigg(\nu\sqrt{2\widetilde{d}\log(1+TK) - 2\log\delta} + \sqrt{\lambda}S\bigg)^2 \widetilde{d}\log(1+TK).
\end{aligned}
$$

□

C Proofs of the Lemmas in Appendix B

C.1 Proof of Lemma B.1

To prove Lemma B.1, we need the following lemma from Arora et al. (2019):

Lemma C.1 (Theorem 3.1, Arora et al. (2019))**.** Fix $\epsilon > 0$ and $\delta \in (0, 1)$. Suppose that

$$
m = \Omega\bigg(\frac{L^6\log(L/\delta)}{\epsilon^4}\bigg),
$$

then for any $i, j \in [TK]$, with probability at least $1-\delta$ over random initialization of $\boldsymbol{\theta}_0$, we have

$$
|\langle \mathbf{g}(\mathbf{x}^i;\boldsymbol{\theta}_0), \mathbf{g}(\mathbf{x}^j;\boldsymbol{\theta}_0)\rangle/m - \mathbf{H}_{i,j}| \leq \epsilon. \tag{C.1}
$$

Proof of Lemma B.1. Taking union bound over $i, j \in [TK]$, we have that if

$$
m = \Omega\bigg(\frac{L^6\log(T^2K^2L/\delta)}{\epsilon^4}\bigg),
$$

then with probability at least $1-\delta$, (C.1) holds for all $(i, j) \in [TK] \times [TK]$. Therefore, we have

$$
\|\mathbf{G}^\top\mathbf{G} - \mathbf{H}\|_F = \sqrt{\sum_{i=1}^{TK}\sum_{j=1}^{TK} |\langle \mathbf{g}(\mathbf{x}^i;\boldsymbol{\theta}_0), \mathbf{g}(\mathbf{x}^j;\boldsymbol{\theta}_0)\rangle/m - \mathbf{H}_{i,j}|^2} \leq TK\epsilon.
$$

Taking $\epsilon = \lambda_0/(3TK)$, if

$$
m = \Omega\bigg(\frac{T^4K^4L^6\log(T^2K^2L/\delta)}{\lambda_0^4}\bigg),
$$

we have $\|\mathbf{G}^\top\mathbf{G}/m - \mathbf{H}\|_F \leq \lambda_0/3$.

□

D Equivalent Version of Algorithm 1

In this section, we present an equivalent version of Algorithm 1, which is written in the style of LinUCB (Li et al., 2010; Chu et al., 2011). Our regret analysis in Section 5 is directly applicable to Algorithm 2.

Algorithm 2 NeuralUCB

Input: Number of rounds T, regularization parameter λ, exploration parameter ν, confidence parameter δ, norm parameter S, network width m, network depth L.
Initialization: Generate each entry of $\mathbf{W}_l$ independently from $N(0, 2/m)$ for $1 \leq l \leq L-1$, and each entry of $\mathbf{W}_L$ independently from $N(0, 1/m)$, define $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}_0)/\sqrt{m}$, where $\boldsymbol{\theta}_0 = [\text{vec}(\mathbf{W}_1)^\top, \ldots, \text{vec}(\mathbf{W}_L)^\top]^\top \in \mathbb{R}^p$
Initialize $\mathbf{Z}_0 = \lambda \mathbf{I}, \mathbf{b}_0 = \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  Observe $\{\mathbf{x}_{t,a}\}_{a=1}^K$
  for $a = 1, \ldots, K$ **do**
    Compute

$$p_{t,a} = (\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_0)^\top \boldsymbol{\phi}(\mathbf{x}_{t,a}) + \gamma_{t-1} \sqrt{\boldsymbol{\phi}(\mathbf{x}_{t,a})^\top \mathbf{Z}_{t-1}^{-1} \boldsymbol{\phi}(\mathbf{x}_{t,a})}$$

    Let $a_t = \text{argmax}_{a \in [K]} p_{t,a}$
  end for
  Play a_t and observe reward r_{t,a_t}
  Compute

$$\mathbf{Z}_t = \mathbf{Z}_{t-1} + \boldsymbol{\phi}(\mathbf{x}_{t,a_t}) \boldsymbol{\phi}(\mathbf{x}_{t,a_t})^\top \in \mathbb{R}^{p \times p}, \ \mathbf{b}_t = \mathbf{b}_{t-1} + r_{t,a_t} \boldsymbol{\phi}(\mathbf{x}_{t,a_t}) \in \mathbb{R}^p$$

  Compute $\boldsymbol{\theta}_t = \mathbf{Z}_t^{-1} \mathbf{b}_t + \boldsymbol{\theta}_0 \in \mathbb{R}^p$
  Compute

$$\gamma_t = \nu \sqrt{\log \frac{\det \mathbf{Z}_t}{\det \lambda \mathbf{I}} - 2 \log \delta} + \sqrt{\lambda} S$$

end for