Recent Advances in Neural Bandits

Yikun Ban

11/22, 2021

Roadmap

- Background
- NeuralUCB
- NeuralTS
- EE-Net

Background

- Sequential decision-making problem is everywhere.
  - Personalized recommendation.
  - Online Advertising.
  - Clinical Trials.
- Exploitation-exploration dilemma exists in decision making.
  - Exploitation: Make greedy decisions by exploiting past data.
  - Exploration: Take risks to explore new knowledge.
- Powerful tool: Contextual multi-armed bandits.

Background: Contextual Bandit

n-armed contextual bandit problem:

- Learner observes n d-dimensional contextual vectors (arms) in a round t

$$\{\mathbf{x}_{t,i} \in \mathbb{R}^d | i \in [n]\}$$

- Learner selects an arm $\mathbf{x}_{t,i'}$ and receives a reward $r_{t,i'}$. For brevity, denote by $\mathbf{x}_t$ the selected arm in t and by r_t its reward.
- The goal is to minimize the following pesudo regret:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}(r_t^* - r_t)\right] \tag{1}$$

where $r_t^* = \max_{i\in[n]} \mathbb{E}[r_{t,i}]$.

Background: Linear Contextual Bandit

- Given an arm $\mathbf{x}_{t,i}, i \in [n]$, its reward $r_{t,i}$ is assumed to be a linear function:

$$r_{t,i} = \boldsymbol{\theta}^\top \mathbf{x}_{t,i} + \eta_{t,i}, \quad \eta_{t,i} \sim \nu - \text{sub-Gaussian} \tag{2}$$

  where $\boldsymbol{\theta}$ is unknown.

- To approximate $\boldsymbol{\theta}$, in rount t, based on the past data $\{\mathbf{x}_i, r_i\}_{i=1}^t$, Ridge regression is applied

$$\hat{\boldsymbol{\theta}}_t = \mathbf{A}_{i_t,t}{}^{-\mathbf{1}}\mathbf{b}_{i_t,t}, \quad \mathbf{A}_{i_t,t} = \mathbf{I} + \sum_{i=1}^{t} \mathbf{x}_i \mathbf{x}_i^\intercal, \quad \mathbf{b}_{i_t,t} = \sum_{i=1}^{t} \mathbf{x}_i r_i, \tag{3}$$

  where $\mathbf{I}$ is a $d \times d$ identity matrix.

Background: Linear Contextual Bandit

Upper Confidence Bound: With probability $1-\delta$,

$$\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\| \leq \text{UCB}. \tag{4}$$

Exploration strategies:

- ϵ-greedy: With probability $1-\epsilon$, $\mathbf{x}_t = \arg_{i\in[n]} \max \hat{\boldsymbol{\theta}}^\top \mathbf{x}_{t,i}$; Otherwise, randomly choose $\mathbf{x}_t$.
- UCB:
$$\mathbf{x}_t = \arg_{i\in[n]} \max \left(\hat{\boldsymbol{\theta}}^\top \mathbf{x}_{t,i} + \text{UCB}_{t,i} \right) \tag{5}$$
- Thompson Sampling:
$$\mathbf{x}_t = \arg_{i\in[n]} \max \hat{\boldsymbol{\theta}}^\top \mathbf{x}_{t,i}, \;\; \hat{\boldsymbol{\theta}} \sim \mathcal{N}(\mathbf{A}_{i_t,t}{}^{-\mathbf{1}} \mathbf{b}_{i_t,t}, \sigma_{t,i}^2) \tag{6}$$
where $\sigma_{t,i}$ can be thought of as an UCB.

Background: Neural Contextual Bandit

- Given an arm $\mathbf{x}_{t,i}, i \in [n]$, its reward $r_{t,i}$ is assumed to be a linear/non-linear function:

$$r_{t,i} = h(\mathbf{x}_{t,i}) + \eta_{t,i}, \quad \eta_{t,i} \sim \nu - \text{sub-Gaussian}$$

  where h is unknown and $0 \leq h(\mathbf{x}) \leq 1$.

- The goal is to minimize the following pesudo regret:

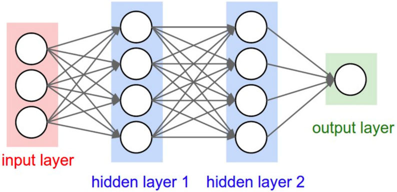
$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}(r_t^* - r_t)\right] = \sum_{t=1}^{T}(h(\mathbf{x}_t^*) - h(\mathbf{x}_t))$$

  where $\mathbf{x}_t^* = \arg_{i \in [n]} \max h(\mathbf{x}_{t,i})$.

NeuralUCB: Network Function

- To learn some universal reward function h, use the universal function approximator, such as neural networks.
- Here, use fully-connected neural network:

$$f(\mathbf{x}_{t,i}; \boldsymbol{\theta}) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\ldots \sigma(\mathbf{W}_1 \mathbf{x}_{t,i}))).$$



where σ is the ReLU activation function and $\boldsymbol{\theta} = \big(\text{vec}(\mathbf{W}_L)^\intercal, \ldots, \text{vec}(\mathbf{W}_1)^\intercal\big)^\intercal \in \mathbb{R}^p$.

NeuralUCB: Selection Criterion

- Let $g(\mathbf{x}_{t,i};\boldsymbol{\theta})$ be the gradient $\nabla_{\boldsymbol{\theta}} f(\mathbf{x}_{t,i};\boldsymbol{\theta})$.
- In round t, given n arms $\{\mathbf{x}_{t,1},\dots,\mathbf{x}_{t,n}\}$, we select arm by

$$\mathbf{x}_t = \arg_{i\in[n]}\max\left(\underbrace{f(\mathbf{x}_{t,i};\boldsymbol{\theta}_{t-1})}_{\text{Exploitation: Estimated reward}} + \underbrace{\gamma_{t-1}\sqrt{g(\mathbf{x}_{t,i};\boldsymbol{\theta}_{t-1})^\top \mathbf{Z}_{t-1}^{-1} g(\mathbf{x}_{t,i};\boldsymbol{\theta}_{t-1})/m}}_{\text{Exploration: UCB}}\right) \tag{7}$$

where γ_{t-1} is a tuning parameter and $\mathbf{Z}_{t-1} = \mathbf{I} + \sum_{t'=1}^{t} g(\mathbf{x}_{t'};\boldsymbol{\theta})g(\mathbf{x}_{t'};\boldsymbol{\theta})^\top$ is the gradient outer product matrix.

NeuralUCB: Update θ

- In round t, after selecting $\mathbf{x}_t$, receive r_t.
- Based on past data $\{\mathbf{x}_i, r_i\}_{i=1}^t$, define loss function:

$$\mathcal{L} = \sum_{i=1}^{t} (f(\mathbf{x}_i; \boldsymbol{\theta}) - r_i)^2 + m\lambda \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2/2. \quad (8)$$

  where $\boldsymbol{\theta}_0$ are the parameters at initialization.
- Conduct gradient descent on $\boldsymbol{\theta}$

NeuralUCB: Workflow

for $t = 1, \ldots, T$ **do**
    Observe $\{\mathbf{x}_{t,a}\}_{a=1}^{K}$
    for $a = 1, \ldots, K$ **do**
        Compute $U_{t,a} = f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1}) + \gamma_{t-1}\sqrt{\mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})^\top \mathbf{Z}_{t-1}^{-1}\mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})/m}$
        Let $a_t = \mathrm{argmax}_{a \in [K]} U_{t,a}$
    end for
    Play a_t and observe reward r_{t,a_t}
    Compute $\mathbf{Z}_t = \mathbf{Z}_{t-1} + \mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_{t-1})\mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_{t-1})^\top/m$
    Let $\boldsymbol{\theta}_t = \mathrm{TrainNN}(\lambda, \eta, J, m, \{\mathbf{x}_{i,a_i}\}_{i=1}^{t}, \{r_{i,a_i}\}_{i=1}^{t}, \boldsymbol{\theta}_0)$

NeuralUCB: Regret Upper Bound

Regret upper bound complexity:

$$R_T \leq \mathcal{O}(\sqrt{T\tilde{d}} \log T)$$

Theorem 4.5. Let $\widetilde{d}$ be the effective dimension, and $\mathbf{h} = [h(\mathbf{x}^i)]_{i=1}^{TK} \in \mathbb{R}^{TK}$. There exist constant $C_1, C_2 > 0$, such that for any $\delta \in (0, 1)$, if

$$m \geq \text{poly}(T, L, K, \lambda^{-1}, \lambda_0^{-1}, S^{-1}, \log(1/\delta)), \quad (4.2)$$
$$\eta = C_1(mTL + m\lambda)^{-1},$$

$\lambda \geq \max\{1, S^{-2}\}$, and $S \geq \sqrt{2\mathbf{h}^\top \mathbf{H}^{-1}\mathbf{h}}$, then with probability at least $1 - \delta$, the regret of Algorithm 1 satisfies

$$\begin{aligned} R_T \leq 3\sqrt{T}\sqrt{\widetilde{d}\log(1 + TK/\lambda) + 2} \\ \cdot \Big[\nu\sqrt{\widetilde{d}\log(1 + TK/\lambda) + 2 - 2\log\delta} \\ + (\lambda + C_2TL)(1 - \lambda/(TL))^{J/2}\sqrt{T/\lambda} \\ + +2\sqrt{\lambda}S\Big] + 1. \end{aligned} \quad (4.3)$$

NeuralUCB: Regret Upper Bound

- $\tilde{d}$ is defined as the effective dimension, which can be thought of as the eigenvalues of context NTK.

Definition 4.1 (Jacot et al. (2018); Cao & Gu (2019)). Let $\{\mathbf{x}^i\}_{i=1}^{TK}$ be a set of contexts. Define

$$\widetilde{\mathbf{H}}_{i,j}^{(1)} = \mathbf{\Sigma}_{i,j}^{(1)} = \langle \mathbf{x}^i, \mathbf{x}^j \rangle, \qquad \mathbf{A}_{i,j}^{(l)} = \begin{pmatrix} \mathbf{\Sigma}_{i,i}^{(l)} & \mathbf{\Sigma}_{i,j}^{(l)} \\ \mathbf{\Sigma}_{i,j}^{(l)} & \mathbf{\Sigma}_{j,j}^{(l)} \end{pmatrix},$$

$$\mathbf{\Sigma}_{i,j}^{(l+1)} = 2\mathbb{E}_{(u,v)\sim N(\mathbf{0},\mathbf{A}_{i,j}^{(l)})}\left[\sigma(u)\sigma(v)\right],$$

$$\widetilde{\mathbf{H}}_{i,j}^{(l+1)} = 2\widetilde{\mathbf{H}}_{i,j}^{(l)}\mathbb{E}_{(u,v)\sim N(\mathbf{0},\mathbf{A}_{i,j}^{(l)})}\left[\sigma'(u)\sigma'(v)\right] + \mathbf{\Sigma}_{i,j}^{(l+1)}.$$

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

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

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

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

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

NeuralUCB: Regret Upper Bound

To derive an Upper Confidence Bound:

$$|f(\mathbf{x}_t;\boldsymbol{\theta}) - h(\mathbf{x}_t)| \leq \text{UCB}$$

- $h(\mathbf{x}_t)$ is linear with respect to gradient.

Lemma 5.1. There exists a positive constant $\bar{C}$ such that for any $\delta \in (0,1)$, if $m \geq \bar{C}T^4K^4L^6\log(T^2K^2L/\delta)/\lambda_0^4$, then with probability at least $1-\delta$, there exists a $\boldsymbol{\theta}^* \in \mathbb{R}^p$ such that

$$h(\mathbf{x}^i) = \langle \mathbf{g}(\mathbf{x}^i;\boldsymbol{\theta}_0), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0\rangle,$$

- (1) Apply Ridge regression on $g(\mathbf{x};\theta_0)$. Calculated the distance between $h(\mathbf{x}_t)$ and Ridge regression.

$$\|\sqrt{m}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) - \bar{\mathbf{Z}}_t^{-1}\bar{\mathbf{b}}_t\|_{\bar{\mathbf{Z}}_t} \leq \bar{\gamma}_t.$$

NeuralUCB: Regret Upper Bound

- (2) Apply NTK objective $< g(\mathbf{x};\theta_0), \boldsymbol{\theta}_t - \boldsymbol{\theta}_0 >$. Calculated the distance between Ridge regression and NTK objective.

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0 - \bar{\mathbf{Z}}_t^{-1}\bar{\mathbf{b}}_t/\sqrt{m}\|_2 \le (1-\eta m\lambda)^{J/2}\sqrt{t/(m\lambda)} + \bar{C}_5 m^{-2/3}\sqrt{\log m}L^{7/2}t^{5/3}\lambda^{-5/3}(1+\sqrt{t/\lambda}).$$

- (3) Calculated the distance between NTK objective and Network function.

Lemma B.4 (Lemma 4.1, Cao & Gu (2019)). There exist constants $\{\bar{C}_i\}_{i=1}^3 > 0$ such that for any $\delta > 0$, if τ satisfies that

$$\bar{C}_1 m^{-3/2} L^{-3/2}[\log(TKL^2/\delta)]^{3/2} \le \tau \le \bar{C}_2 L^{-6}[\log m]^{-3/2},$$

then with probability at least $1-\delta$, for all $\widetilde{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}}$ satisfying $\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|_2 \le \tau$, $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\|_2 \le \tau$ and $j \in [TK]$ we have

$$\Big| f(\mathbf{x}^j;\widetilde{\boldsymbol{\theta}}) - f(\mathbf{x}^j;\widehat{\boldsymbol{\theta}}) - \langle \mathbf{g}(\mathbf{x}^j;\widehat{\boldsymbol{\theta}}), \widetilde{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}\rangle \Big| \le \bar{C}_3 \tau^{4/3} L^3 \sqrt{m\log m}.$$

- Putting them together, we can calculate the upper bound for $|f(\mathbf{x}_t;\boldsymbol{\theta}) - h(\mathbf{x}_t)|$!.

Neural Thompson Sampling

- Given an arm $\mathbf{x}_{t,i}$, to learn the expected reward $h(\mathbf{x}_{t,i})$, use the neural network

$$f(\mathbf{x}_{t,i}; \boldsymbol{\theta}) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\dots \sigma(\mathbf{W}_1 \mathbf{x}_{t,i}))).$$

- In round t, given n arms $\{\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,n}\}$, select an arm by

$$\forall i \in [n], \text{draw } \hat{r}_{t,i} \sim \mathcal{N}(\underbrace{f(\mathbf{x}_{t,i}; \boldsymbol{\theta})}_{\text{Mean: Exploitation}}, \underbrace{\sigma^2}_{\text{Variance: Exploration}}) \tag{9}$$

$$\text{Select } \mathbf{x}_t = \arg_{i \in [n]} \max \hat{r}_{t,i}.$$

where $\sigma = \nu g(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1})^\top \mathbf{Z}_{t-1}^{-1} g(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1})$.

- Receive reward and update parameters.

Neural Thompson Sampling: Regret Upper Bound

- Regret bound complexity:

$$R_T \leq \mathcal{O}(\sqrt{T\tilde{d}}\log T).$$

Theorem 3.5. Under Assumption 3.4, set the parameters in Algorithm 1 as $\lambda = 1 + 1/T$, $\nu = B + R\sqrt{\tilde{d}\log(1+TK/\lambda) + 2 + 2\log(1/\delta)}$ where $B = \max\left\{1/(22e\sqrt{\pi}), \sqrt{2\mathbf{h}^\top\mathbf{H}^{-1}\mathbf{h}}\right\}$ with $\mathbf{h} = (h(\mathbf{x}^1), \ldots, h(\mathbf{x}^{TK}))^\top$, and R is the sub-Gaussian parameter. In line 9 of Algorithm 1, set $\eta = C_1(m\lambda + mLT)^{-1}$ and $J = (1 + LT/\lambda)(C_2 + \log(T^3L\lambda^{-1}\log(1/\delta)))/C_1$ for some positive constant C_1, C_2. If the network width m satisfies:

$$m \geq \text{poly}\Big(\lambda, T, K, L, \log(1/\delta), \lambda_0^{-1}\Big),$$

then, with probability at least $1-\delta$, the regret of Algorithm 1 is bounded as

$$R_T \leq C_2(1+c_T)\nu\sqrt{2\lambda L(\tilde{d}\log(1+TK)+1)T} + (4 + C_3(1+c_T)\nu L)\sqrt{2\log(3/\delta)T} + 5,$$

where C_2, C_3 are absolute constants, and $c_T = \sqrt{4\log T + 2\log K}$.

Neural Thompson Sampling: Regret Upper Bound

- (1) Calculate variance σ^2, which can be thought of as the UCB of $|f(\mathbf{x}_{t,i};\boldsymbol{\theta}) - h(\mathbf{x}_{t,i})|$.
  1. Calculate the distance between $h(\mathbf{x}_{t,i})$ and Ridge regression.
  2. Calculate the distance between Ridge regression and NTK.
  3. Calculate the distance between NTK and $f(\mathbf{x}_{t,i};\boldsymbol{\theta})$.
- (2) Use concentration inequalities to upper bound $|f(\mathbf{x}_{t,i};\boldsymbol{\theta}) - r_{t,i}|$.

EE-Net: Exploitation-Exploration Neural Networks

- Same, given an arm $\mathbf{x}_{t,i}$, to learn the expected reward $h(\mathbf{x}_{t,i})$, use the neural network
$$f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1}\sigma(\ldots\sigma(\mathbf{W}_1 \mathbf{x}_{t,i}))).$$
- Why explore? To fill the gap between expected reward and estimated reward.

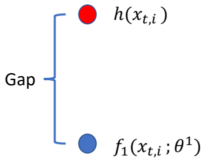


Figure 1: Case 1: When expected reward is larger than estiamted reward.

EE-Net: Exploitation-Exploration Neural Networks

- Instead of calculating a statistic upper bound for $|h(\mathbf{x}_{t,i}) - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^2)|$, EE-Net uses a neural network f_2 to learn $h(\mathbf{x}_{t,i}) - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^2)$.

$$f_2(\mathbf{x}_{t,i}; \boldsymbol{\theta}^2) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\dots \sigma(\mathbf{W}_1 \mathbf{x}_{t,i}))).$$

- Ground truth: $h(\mathbf{x}_{t,i}) - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$, i.e., $r_{t,i} - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$.
- $h(\mathbf{x}_{t,i}) - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$ indicates exploration direction: "Upward" or "Downward" exploration.

Case 1: Upward Exploration

Case 2: Downward Exploration

EE-Net: Exploitation-Exploration Neural Networks

- Input: Gradient $\nabla_{\boldsymbol{\theta}_1} f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$. Why?
- $\nabla_{\boldsymbol{\theta}_1} f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$ contains two sides of information.
  1. Arm feature $\mathbf{x}_{t,i}$.
  2. Discriminative ability of f_1(Exploration depending on the exploitation).
- Build loss function $\mathcal{L}_2$

$$\mathcal{L}_2 = \frac{1}{2} \sum_{i=1}^{t} \left(f_2\left(\nabla_{\boldsymbol{\theta}^1} f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1); \boldsymbol{\theta}^2\right) - \underbrace{(r_i - f_1(\mathbf{x}_i; \boldsymbol{\theta}^1))}_{\text{Ground truth}} \right)^2$$

- After receiving r_t in round t, based on $\left\{\nabla_{\boldsymbol{\theta}_1} f_1(\mathbf{x}_i; \boldsymbol{\theta}_i^1), r_i - f_1(\mathbf{x}_i; \boldsymbol{\theta}_i^1)\right\}_{i=1}^{t}$, use gradient descent to update $\boldsymbol{\theta}^2$.

EE-Net: Selection Criterion 1

- In round t, given n arms $\{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,n}\}$, we select arm by

$$\mathbf{x}_t = \arg_{i \in [n]} \max \left(\underbrace{f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1_{t-1})}_{\text{Exploitation}} + \underbrace{f_2\left(\nabla_{\boldsymbol{\theta}^1_{t-1}} f_1(\mathbf{x}_i; \boldsymbol{\theta}^1_{t-1}); \boldsymbol{\theta}^2_{t-1}\right)}_{\text{Exploration}} \right) \tag{10}$$

- Receive reward r_t and update $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2$.

EE-Net: Selection Criterion 2

Build Decision Maker $f_3(\cdot;\boldsymbol{\theta}^3)$.

- In roung t, given an arm $\mathbf{x}_{t,i}$, calculate its f_1, f_2 scores.
- Build a neural network $f_3(\cdot;\boldsymbol{\theta}^3)$.
- Input: $f_1(\mathbf{x}_{t,i};\boldsymbol{\theta}^1_{t-1}), f_2(\triangledown_{\boldsymbol{\theta}^1_{t-1}} f_1;\boldsymbol{\theta}^2_{t-1})$.
- Ground truth: $p_{t,i}$, i.e., the probability of $\mathbf{x}_{t,i}$ being the optimal arm in round t.
  1. Binary reward $(0,1)$: $p_{t,i} = 1.0$ if $r_{t,i} = 1$; Otherwise, $p_{t,i} = 0.0$ if $r_{t,i} = 0$.
  2. Continuous reward $[0,1]$: (1) $p_{t,i} = \frac{r_{t,i} - 0}{1 - 0} = r_{t,i}$; (2) Set a threshold γ. $p_{t,i} = 1.0$ if $r_{t,i} > \gamma$; Otherwise $p_{t,i} = 0.0$.
- Build loss function:

$$\mathcal{L}_3 = -\frac{1}{t}\sum_{i=1}^{t}\left[p_t \log f_3((f_1, f_2);\boldsymbol{\theta}^3) + (1 - p_t)\log(1 - f_3((f_1, f_2);\boldsymbol{\theta}^3))\right]. \quad (11)$$

- Update $\boldsymbol{\theta}^3$ in each round.

EE-Net: Selection Criterion 2

- In round t, given n arms $\{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,n}\}$, we select arm by

$$1.\ \text{Calculated}\ f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1_{t-1}), f_2(\nabla_{\boldsymbol{\theta}^1_{t-1}} f_1; \boldsymbol{\theta}^2_{t-1}) \tag{12}$$

$$2.\ \mathbf{x}_t = \arg_{i \in [n]} \max f_3 \left((f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1_{t-1}), f_2(\nabla_{\boldsymbol{\theta}^1_{t-1}} f_1; \boldsymbol{\theta}^2_{t-1})); \boldsymbol{\theta}^3_{t-1} \right) \tag{13}$$

- Receive reward r_t and update $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \boldsymbol{\theta}^3$.

EE-Net: Regret Upper Bound

- Regret bound complexity:

$$R_T \leq \mathcal{O}(\sqrt{T \log T}).$$

Theorem 1. *Let f_1, f_2 follow the setting of f (Eq. (5.1)) with width m, m' respectively and same depth L. Let $\mathcal{L}_1, \mathcal{L}_2$ be loss function defined in Algorithm 1. Set f_3 as $f_3 = f_1 + f_2$. Given two constants ϵ_1, ϵ_2, $0 < \epsilon_1, \epsilon_2 < 1$, assume*

$$\begin{aligned}
& m \geq poly(T, n, L, \log(1/\delta) \cdot d \cdot e^{\sqrt{\log 1/\delta}}), \ m' \geq \Omega(m^2 L) \\
& \eta_1 = \Theta\left(\frac{d\delta}{poly(T,n,L) \cdot m}\right), \ \eta_2 = \Theta\left(\frac{\mathcal{O}(m^2 L)\delta}{poly(T,n,L) \cdot m'}\right) \\
& K_1 = \Theta\left(\frac{poly(T,n,L)}{\delta^2} \cdot \log\left((\epsilon_1/2)^{-1}\right)\right), \ K_2 = \Theta\left(\frac{poly(T,n,L)}{\delta^2} \cdot \log\left(\epsilon_2^{-1}\right)\right),
\end{aligned} \tag{5.3}$$

then with probability at least $1-\delta$, the expected cumulative regret of EE-Net in T rounds satisfies

$$\mathbf{R}_T \leq \mathcal{O}\left((2\sqrt{T}-1)\sqrt{2\epsilon_2}\right) + \mathcal{O}\left((\xi_2 + \epsilon_1)(2\sqrt{T}-1)\sqrt{2\log(\mathcal{O}(Tn)/\delta)}\right). \tag{5.4}$$

EE-Net: Regret Upper Bound

Proof Workflow:

- $\forall t \in [T], i \in [n]$, assume $(\mathbf{x}_{t,i}, r_{t,i})$ are i.i.d random variables, generated from unknown $\mathcal{D}$ and $f_3 = f_1 + f_2$.
- Given $\{\mathbf{x}_i, r_i\}_{i=1}^{t-1}$, calculate convergency error of f_3.

Lemma B.3. *Suppose $m \geq \max\left(poly(n, L, \delta^{-1} \cdot d), \Omega(e^{\sqrt{\log 1/\delta}})\right)$, the learning rate $\eta = \Omega(\frac{\delta d}{poly(T,n,L)m})$, the number of iterations K satisfies the conditions in Eq. (C.1), then with probability at least $1-\delta$, given a constant $0 < \epsilon < 1$, starting from random initialization,*

(1) (Theorem 1 in (Allen-Zhu et al., 2019)) The loss satisfies $\mathcal{L} \leq \epsilon$ (Eq. (5.2)) in $K = \Omega(\frac{poly(T,n,L)}{\delta^2} \cdot \log \epsilon^{-1})$ iterations,

- Calculate the generalization bound of f_3 with respect to h, such that we can upper bound $|f_3(\cdot; \boldsymbol{\theta}^3) - h(\cdot)|$.

Lemma B.1. *Given $0 < \epsilon_1, \epsilon_2 < 1$, suppose m, η, K_1, K_2 satisfy the conditions in Eq. (C.1). Then, with probability at least $1-\delta$, for any $t \in [T], i \in [n]$, it holds uniformly that*

$$\mathbb{E}_{(\mathbf{x}_{t,i}, r_{t,i}) \sim \mathcal{D}}[|f_2(\nabla_{\boldsymbol{\theta}_1} f_1 / c_1\sqrt{mL}; \boldsymbol{\theta}_t^2) - (r_{t,i} - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}_t^1))\,|] \leq \sqrt{\frac{2\epsilon_2}{t}} + (\xi_2 + \epsilon_1)\sqrt{\frac{2\log(\mathcal{O}(Tn)/\delta)}{t}}.$$

Comparison 1: Selection Criterion

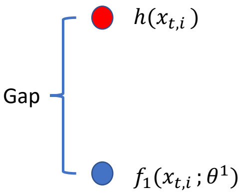
Table 1: Selection Criterion Comparison ($\mathbf{x}_t$: selected arm in round t).

Methods	Selection Criterion
Neural Epsilon-greedy	With probability $1-\delta$, $\mathbf{x}_t = \arg\max_{i \in [n]} f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$; Otherwise, select $\mathbf{x}_t$ randomly.
NeuralTS (Zhang et al., 2020)	For $\mathbf{x}_{t,i}, \forall i \in [n]$, draw $\hat{r}_{t,i}$ from $\mathcal{N}(f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1), {\sigma_{t,i}}^2)$. Then, $\mathbf{x}_t = \arg\max_{i \in [n]} \hat{r}_{t,i}$.
NeuralUCB (Zhou et al., 2020)	$\mathbf{x}_t = \arg\max_{i \in [n]} \left(f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1) + \mathrm{UCB}_{t,i}\right)$.
EE-Net (Our approach)	$\forall i \in [n]$, compute $f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1)$, $f_2\left(\nabla_{\boldsymbol{\theta}^1} f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}^1); \boldsymbol{\theta}^2\right)$ (Exploration Net). Then $\mathbf{x}_t = \arg\max_{i \in [n]} f_3(f_1, f_2; \boldsymbol{\theta}^3)$.

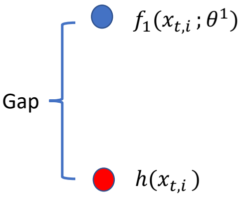
Comparison 2: Exploration Direction

Table 3: Exploration Direction Comparison.

Methods	"Upward" Exploration	"Downward" Exploration
NeuralUCB	√	×
NeuralTS	Randomly	Randomly
EE-Net	√	√



Case 1: Upward Exploration



Case 2: Downward Exploration

Comparison 3: Running Complexity

Table 5: Running Time/Space Complexity Comparison (p is number of parameters of f_1).

Methods	Time	Space	Training Time (# Neural Networks)
NeuralUCB	$\mathcal{O}(p^2)$	$\mathcal{O}(p^2)$	1
NeuralTS	$\mathcal{O}(p^2)$	$\mathcal{O}(p^2)$	1
EE-Net	$\mathcal{O}(p)$	$\mathcal{O}(p)$	2-3

Comparison 4: Regret Bound

Table 4: Regret Bound Comparison.

Methods	Regret Upper Bound	Effective Dimension $\tilde{d}$
NeuralUCB	$\mathcal{O}(\sqrt{\tilde{d}T} \log T)$	Yes
NeuralTS	$\mathcal{O}(\sqrt{\tilde{d}T} \log T)$	Yes
EE-Net	$\mathcal{O}(\sqrt{T}\sqrt{\log T})$	No

Comparison 5: Empirical Performance

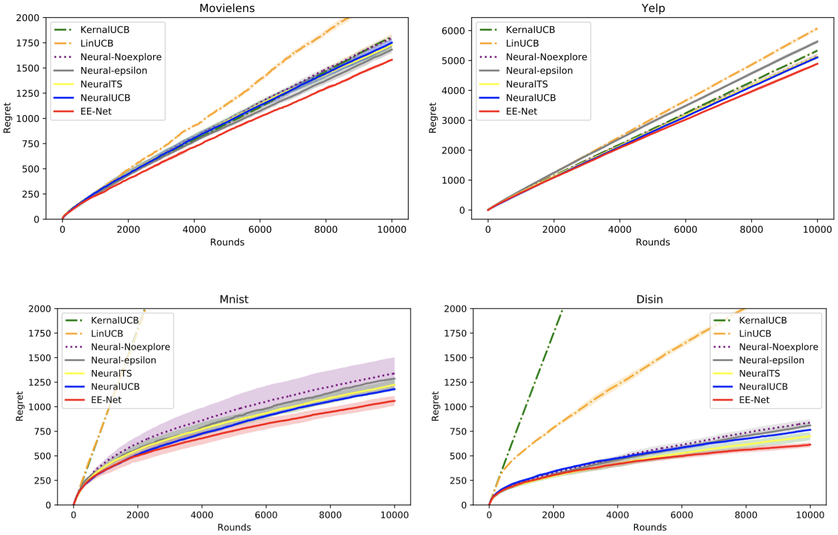


Figure 2: Regret comparison on Mnist and Disin (mean of 10 runs with standard deviation (shadow)). With the same exploitation network f_1, EE-Net outperforms all baselines.

Summary

- Background
- Rule-based Exploration
  1. NeuralUCB
  2. NeuralTS
- Neural-based Exploration
  1. EE-Net

Thanks