# Neural Exploration

Patrick Saux

Reinforcement Learning

February 13th 2020

# Outline



Code available at : `https://github.com/sauxpa/neural_exploration`

# Contextual MDP

## Fixed horizon

Contextual MDP = regular MDP with observable state-action features.

# Contextual MDP

## Fixed horizon

Contextual MDP = regular MDP with observable state-action features.

### Definition (Contextual MDP)

$\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ is a *contextual fixed horizon MDP* when there exists a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and for all $h = 1, \ldots, H$ two mappings $\mathcal{R}_h : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{P}_h : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}_+$ and $(\xi_{h,s,a})_{h=1,\ldots,H, s\in\mathcal{S}, a\in\mathcal{A}}$ a collection of i.i.d sub-Gaussian random variables such that:

i. $\forall h, \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad r_h(s,a) = \mathcal{R}_h(\phi(s,a)) + \xi_{h,s,a},$

ii. $\forall h, \forall (s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \quad \mathbb{P}_h(s,a,s') = \mathcal{P}_h(\phi(s,a), s').$

# Contextual MDP

## Fixed horizon

Contextual MDP = regular MDP with observable state-action features.

### Definition (Contextual MDP)

$\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ is a *contextual fixed horizon MDP* when there exists a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and for all $h = 1, \dots, H$ two mappings $\mathcal{R}_h : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{P}_h : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}_+$ and $(\xi_{h,s,a})_{h=1,\dots,H, s\in\mathcal{S}, a\in\mathcal{A}}$ a collection of i.i.d sub-Gaussian random variables such that:

i. $\forall h, \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad r_h(s,a) = \mathcal{R}_h(\phi(s,a)) + \xi_{h,s,a},$

ii. $\forall h, \forall (s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \quad \mathbb{P}_h(s,a,s') = \mathcal{P}_h(\phi(s,a), s').$

- $\mathcal{M}$ is said to be linear if all $\mathcal{R}_h$ and $\mathcal{P}_h$ are linear maps;

# Contextual MDP

## Fixed horizon

Contextual MDP = regular MDP with observable state-action features.

**Definition (Contextual MDP)**

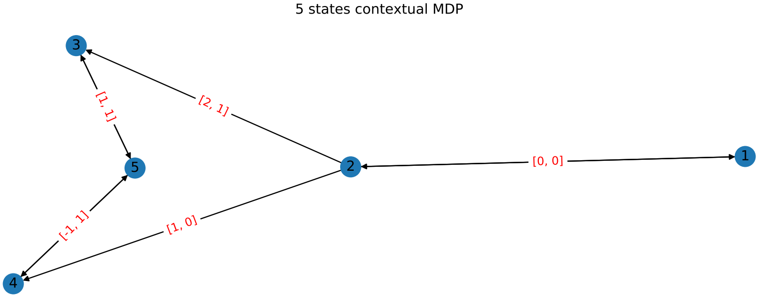
$\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ is a *contextual fixed horizon MDP* when there exists a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and for all $h = 1, \ldots, H$ two mappings $\mathcal{R}_h : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{P}_h : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}_+$ and $(\xi_{h,s,a})_{h=1,\ldots,H, s \in \mathcal{S}, a \in \mathcal{A}}$ a collection of i.i.d sub-Gaussian random variables such that:

i. $\forall h, \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad r_h(s,a) = \mathcal{R}_h(\phi(s,a)) + \xi_{h,s,a},$
ii. $\forall h, \forall (s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \quad \mathbb{P}_h(s,a,s') = \mathcal{P}_h(\phi(s,a), s').$

- $\mathcal{M}$ is said to be linear if all $\mathcal{R}_h$ and $\mathcal{P}_h$ are linear maps;
- Sub-Gaussian noise : for concentration bounds.

# Contextual MDP

## Example



# Contextual MDP

## Bandit

Contextual bandit = single state, single frame contextual MDP.

# Contextual MDP

## Bandit

Contextual bandit = single state, single frame contextual MDP.

**Definition (Contextual bandit)**

$\mathcal{M} = (\mathcal{A}, r)$ is a *contextual bandit* when there exists a feature map $\phi : \mathcal{A} \to \mathbb{R}^d$, a mapping $\mathcal{R} : \mathbb{R}^d \to \mathbb{R}$ and $(\xi_a)_{a \in \mathcal{A}}$ a collection of i.i.d sub-Gaussian random variables such that:

- $\forall a \in \mathcal{A}, \quad r(a) = \mathcal{R}(\phi(a)) + \xi_a.$

# Contextual MDP

## Bandit

Contextual bandit = single state, single frame contextual MDP.

**Definition (Contextual bandit)**

$\mathcal{M} = (\mathcal{A}, r)$ is a *contextual bandit* when there exists a feature map $\phi : \mathcal{A} \to \mathbb{R}^d$, a mapping $\mathcal{R} : \mathbb{R}^d \to \mathbb{R}$ and $(\xi_a)_{a \in \mathcal{A}}$ a collection of i.i.d sub-Gaussian random variables such that:

- $\forall a \in \mathcal{A}, \quad r(a) = \mathcal{R}(\phi(a)) + \xi_a.$

- Without loss of generality: $\forall a \in \mathcal{A}, \|\phi(a)\| \leq 1;$

# Contextual MDP

## Bandit

Contextual bandit = single state, single frame contextual MDP.

**Definition (Contextual bandit)**

$\mathcal{M} = (\mathcal{A}, r)$ is a *contextual bandit* when there exists a feature map $\phi : \mathcal{A} \to \mathbb{R}^d$, a mapping $\mathcal{R} : \mathbb{R}^d \to \mathbb{R}$ and $(\xi_a)_{a \in \mathcal{A}}$ a collection of i.i.d sub-Gaussian random variables such that:

- $\forall a \in \mathcal{A}, \quad r(a) = \mathcal{R}(\phi(a)) + \xi_a.$

- Without loss of generality: $\forall a \in \mathcal{A}, \|\phi(a)\| \leq 1$;
- $\phi(a)$ could change between episodes : $\phi_t(a)$.

# Regret

- Value function: $V^\pi(s) = \mathbb{E}\big[\sum_{h=0}^{H-1} r(s_h, \pi(s_h)|s_0 = s, \pi\big]$;
- Q-function: $Q^\pi(s,a) = \mathbb{E}\big[\sum_{h=0}^{H-1} r(s_h, a_h|s_0 = s, a_0 = a, \pi\big]$;
- Optimal policy:
  $\pi^*(s) \in \arg\max_a \max_\pi Q^\pi(s,a) \quad (or \in \arg\max_\pi V^\pi(s))$.

# Regret

- Value function: $V^{\pi}(s) = \mathbb{E}\left[\sum_{h=0}^{H-1} r(s_h, \pi(s_h)|s_0 = s, \pi\right]$;
- Q-function: $Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{h=0}^{H-1} r(s_h, a_h|s_0 = s, a_0 = a, \pi\right]$;
- Optimal policy:
  $\pi^*(s) \in \arg\max_a \max_\pi Q^\pi(s,a) \quad (or \in \arg\max_\pi V^\pi(s))$.

**Definition (Regret after $T$ episodes - MDP)**

$R_T = \sum_{t=1}^{T} V_0^*(s_0^t) - V_0^{\pi_t}(s_0^t)$.

# Regret

- Value function: $V^\pi(s) = \mathbb{E}\big[\sum_{h=0}^{H-1} r(s_h, \pi(s_h)|s_0 = s, \pi\big]$;
- Q-function: $Q^\pi(s,a) = \mathbb{E}\big[\sum_{h=0}^{H-1} r(s_h, a_h|s_0 = s, a_0 = a, \pi\big]$;
- Optimal policy:
  $\pi^*(s) \in \arg\max_a \max_\pi Q^\pi(s,a) \quad (or \in \arg\max_\pi V^\pi(s))$.

**Definition (Regret after $T$ episodes - MDP)**

$R_T = \sum_{t=1}^T V_0^*(s_0^t) - V_0^{\pi_t}(s_0^t)$.

**Definition (Regret after $T$ episodes - bandit)**

$R_T = \sum_{t=1}^T r(a_t^*) - r(a_t)$.

# Regret

- Value function: $V^\pi(s) = \mathbb{E}\big[\sum_{h=0}^{H-1} r(s_h, \pi(s_h)|s_0 = s, \pi\big]$;
- Q-function: $Q^\pi(s, a) = \mathbb{E}\big[\sum_{h=0}^{H-1} r(s_h, a_h|s_0 = s, a_0 = a, \pi\big]$;
- Optimal policy:
  $\pi^*(s) \in \arg\max_a \max_\pi Q^\pi(s, a) \quad (or \in \arg\max_\pi V^\pi(s))$.

**Definition (Regret after $T$ episodes - MDP)**

$R_T = \sum_{t=1}^{T} V_0^*(s_0^t) - V_0^{\pi_t}(s_0^t)$.

**Definition (Regret after $T$ episodes - bandit)**

$R_T = \sum_{t=1}^{T} r(a_t^*) - r(a_t)$.

Goal of episodic RL : minimum (sublinear growth) regret.

# Optimistic exploration

## UCB (general form)

**Algorithm:** UCB

**Initialize** $approximator_0$
**for** $t = 1, \dots, T$ **do**
  **for** $a \in \mathcal{A}$ **do**
    $\hat{\mu}_t(a) = approximator_{t-1}(\phi_t(a))$
    $B_t(a) = \hat{\mu}_t(a) + ExplorationBonus_t(a)$
  $a_t = \arg\max_{a \in \mathcal{A}} B_t(a)$
  Play $a_t$, receive reward $r(a_t)$
  Train $approximator_t$ on $a_t, r(a_t)$

# LinUCB

## Algorithm

**Algorithm:** LinUCB

**Exploration**:

$A_0 = \lambda I$

$A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\phi_t(a_t)\phi_t(a_t)^\top A_{t-1}^{-1}}{1+\phi_t(a_t)^\top A_{t-1}^{-1}\phi_t(a_t)}$ (Sherman-Morrison)

$ExplorationBonus_t(a) = \gamma\sqrt{\phi_t(a)^\top A_{t-1}^{-1}\phi_t(a)}$

**Training**:

$b_t = b_{t-1} + \phi_t(a_t)r(a_t)$

$\theta_t = A_t^{-1}b_t$

$\hat{\mu}_t(a) = \phi_t(a)^\top\theta_t$

# LinUCB

## Regret analysis

**Theorem (LinUCB regret)**

*With probability at least* $1 - \delta$*:*

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

# LinUCB

## Regret analysis

**Theorem (LinUCB regret)**

*With probability at least* $1 - \delta$*:*

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

Actually, a few assumptions:

- Sub-Gaussian concentration parameter,
- Strong enough regularization $\lambda$,
- Explicit (time-dependent) scaling factor $\gamma_t$.

# LinUCB

## Regret analysis

**Theorem (LinUCB regret)**

*With probability at least* $1-\delta$*:*

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

Actually, a few assumptions:

- Sub-Gaussian concentration parameter,
- Strong enough regularization $\lambda$,
- Explicit (time-dependent) scaling factor $\gamma_t$.

Can be improved to $\mathcal{O}\left(\sqrt{dT \log T}\right)$ using SupLinUCB and mutually exclusive samples.

# LinUCB

## Linear rewards


Figure: 4 arms, 20 features per arm, 10% Gaussian noise.
$r_a = 10\theta^\top \phi(a)$.
Frequency of optimal arm selection : $\geq 96\%$.

# LinUCB

## Quadratic rewards


Figure: 4 arms, 20 features per arm, 10% Gaussian noise.
$r_a = \theta^\top \phi(a) + 0.05(\theta^\top \phi(a))^2$.
Frequency of optimal arm selection : 75%.

# LinUCB

## Cosine rewards


Figure: 4 arms, 20 features per arm, 10% Gaussian noise.
$r_a = \cos(10\theta^\top \phi(a))$.
Frequency of optimal arm selection : 25%.

# Kernel UCB

## Algorithm

Feature map $\phi_t(a) \mapsto g \cdot \phi_t(a)$, associated with kernel matrix $\mathcal{K}$.

# Kernel UCB
## Algorithm

Feature map $\phi_t(a) \mapsto g \cdot \phi_t(a)$, associated with kernel matrix $\mathcal{K}$.

**Algorithm:** Kernel UCB

**Exploration**:

$A_0 = \lambda I$

$A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} g \cdot \phi_t(a_t) g \cdot \phi_t(a_t)^\top A_{t-1}^{-1}}{1 + g \cdot \phi_t(a_t)^\top A_{t-1}^{-1} g \cdot \phi_t(a_t)}$ (Sherman-Morrison)

$ExplorationBonus_t(a) = \gamma \sqrt{g \cdot \phi_t(a)^\top A_{t-1}^{-1} g \cdot \phi_t(a)}$

**Training**:

$\hat{\mu}_t(a) = KernelRidgeRegression(\mathcal{K}, \phi_t(a), \lambda)$

# Kernel UCB

## Regret Analysis

**Theorem (Kernel UCB regret)**

*With probability at least $1-\delta$:*

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

*where*

$$\tilde{d} = \frac{\log det(I + \mathcal{K}/\lambda)}{\log(1 + TK/\lambda)}$$

*is the effective dimension of the kernel $\mathcal{K}$ on the contexts $(\phi(a))_{a\in\mathcal{A}}$.*

# Kernel UCB

## Regret Analysis

**Theorem (Kernel UCB regret)**

*With probability at least* $1-\delta$*:*

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

*where*

$$\tilde{d} = \frac{\log det(I + \mathcal{K}/\lambda)}{\log(1 + TK/\lambda)}$$

*is the effective dimension of the kernel* $\mathcal{K}$ *on the contexts* $(\phi(a))_{a\in\mathcal{A}}$.

Again, can be improved to $\mathcal{O}\left(\sqrt{\tilde{d}T\log T}\right)$ using SupKernelUCB.

# Kernel UCB

## Effective dimension

Assume $g \cdot \phi_t(a) \in [-1, 1]^d$ and consider two extreme cases:

- Independant RKHS features: $\mathcal{K} = diag(\alpha_1, \ldots, \alpha_{TK})$:

$$\tilde{d} = \frac{\sum_{k=1}^{TK} \log(1 + \alpha_k/\lambda)}{\log(1 + TK/\lambda)} \approx \frac{\sum_{k=1}^{TK} \sum_{j=1}^{d_{\mathcal{K}}} |g \cdot \phi_t(a_k)_j|^2/\lambda}{TK/\lambda} \leq d_{\mathcal{K}}$$

- Fully correlated RKHS features: $\mathcal{K} = \mathbb{1}$:

$$\tilde{d} = \frac{\log(1 + TK/\lambda)}{\log(1 + TK/\lambda)} = 1.$$

# Optimistic exploration

## UCB-VI (general form)

**Algorithm:** UCB-VI

**Initialize** $approximator_h^0, h = 1, \dots, H$

**for** $t = 1, \dots, T$ **do**

  Receive initial state $s_1^t$

  **for** $h$=$H$, ..., $1$ **do**

    Train $approximator_h^t$ on $s_h^t, a_h^t, r(s_h^t, a_h^t)$

    **for** $s, a \in \mathcal{S} \times \mathcal{A}$ **do**

      $\hat{Q}_h^t(s,a) = approximator_h^t(\phi(s,a))$

      $\tilde{Q}_h^t(s,a) = \hat{Q}_h^t(s,a) + ExplorationBonus_h^t(s,a)$

  **for** $h = 1, \dots, H$ **do**

    $a_h^t = \arg\max_{a \in \mathcal{A}} \tilde{Q}_h^t(s_h^t, a_h^t)$

    Play $a_h^t$, receive reward $r(s_h^t, a_h^t)$, go to state $s_{h+1}^t$.

# LinUCB-VI

## Algorithm

**Algorithm:** LinUCB-VI

**Exploration**:

$A_{0,h} = \lambda I$

$A_{t,h}^{-1} = A_{t-1,h}^{-1} - \frac{A_{t-1,h}^{-1}\phi(s_h^t,a_h^t)\phi(s_h^t,a_h^t)^\top A_{t-1,h}^{-1}}{1+\phi(s_h^t,a_h^t)^\top A_{t-1,h}^{-1}\phi(s_h^t,a_h^t)}$

$ExplorationBonus_h^t(s,a) = \gamma\sqrt{\phi(s,a)^\top A_{t,h}^{-1}\phi(s,a)}$

**Training**:

$b_{t,h} = b_{t-1,h} + \phi(s_h^t, a_h^t)\big(r(s_h^t, a_h^t) + \max_a \tilde{Q}_{h+1}^t(s_{h+1}^t, a)\big)$

$\theta_{t,h} = A_{t,h}^{-1} b_{t,h}$

$\hat{Q}_h^t(s,a) = \phi(s,a)^\top \theta_{t,h}.$

# LinUCB-VI

## Regret analysis

**Theorem (LinUCB-VI regret [3])**

*With probability at least* $1-\delta$*:*

$$R_T \leq \mathcal{O}\left(\sqrt{d^3 H^3 T \log(2dT/\delta)^2}\right).$$

# LinUCB-VI

## Regret analysis

**Theorem (LinUCB-VI regret [3])**

*With probability at least* $1-\delta$*:*

$$R_T \leq \mathcal{O}\left(\sqrt{d^3 H^3 T \log(2dT/\delta)^2}\right).$$

Other results:

- TS version: $\mathcal{O}\left(d^2 H^2 \sqrt{T} + H^5 d^4\right)$ [5].
- Low rank transition: $\mathcal{O}\left(\sqrt{d^3 T} H^2 \log T\right)$ or even $\mathcal{O}\left(d\sqrt{T} H^2 \log T\right)$ if strong feature regularity [4].

# LinUCB-VI

## Linear rewards

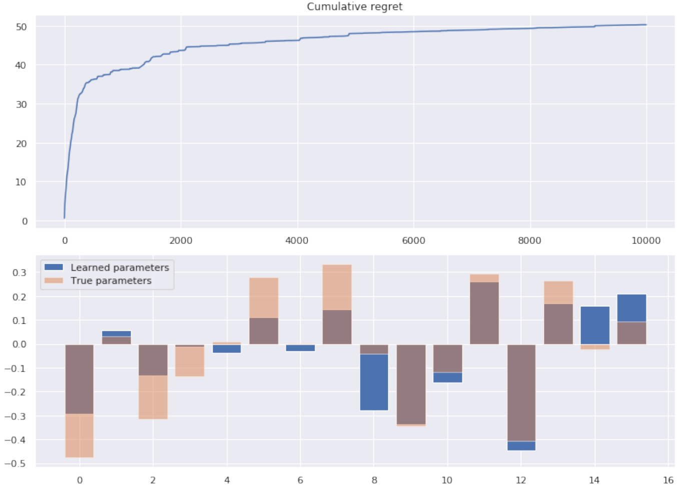



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$.

# LinUCB-VI

## Misspecified setting

### Theorem (LinUCB-VI for quasilinear MDP)

*Let $\varepsilon > 0$ a bound on the nonlinear expansion terms of $\mathcal{R}_h$ and $\mathcal{P}_h$. The regret analysis becomes:*

- *[3] :*
  $$R_T \leq \mathcal{O}\left(\sqrt{d^3 H^3 T \log(2dT/\delta)^2} + \boldsymbol{\varepsilon d H T \sqrt{\log(2dT/\delta)}}\right),$$
- *[5] :*
  $$R_T \leq \mathcal{O}\left(d^2 H^2 \sqrt{T} + H^5 d^4 + \boldsymbol{\varepsilon d H T (1 + \varepsilon d H^2)}\right).$$

# LinUCB-VI

## Quadratic rewards

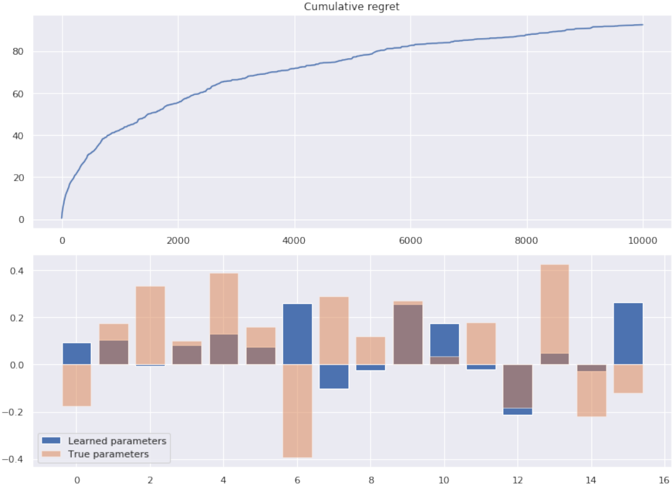



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$.

# LinUCB-VI

## Cosine rewards

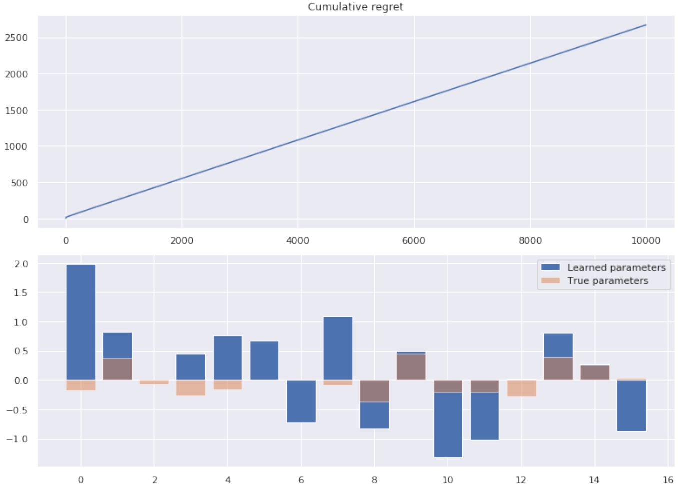



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$.

# Neural Tangent Kernel

## Intuition

Let :

- $f(\cdot;\theta)$ feed-forward neural network output,
- $\mathcal{L} = \sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$ supervised batch loss.

# Neural Tangent Kernel

## Intuition

Let :

- $f(\cdot;\theta)$ feed-forward neural network output,
- $\mathcal{L} = \sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$ supervised batch loss.

In the limit of small learning rate, $\partial_t \theta(t) = -\nabla \mathcal{L}(f(\cdot;\theta(t)))$.

# Neural Tangent Kernel

## Intuition

Let :

- $f(\cdot;\theta)$ feed-forward neural network output,
- $\mathcal{L} = \sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$ supervised batch loss.

In the limit of small learning rate, $\partial_t \theta(t) = -\nabla \mathcal{L}(f(\cdot;\theta(t)))$.
For a generic test data $x$:

$$\begin{aligned}
\partial_t f(x;\theta(t)) &= \nabla f(x;\theta(t))^\top \partial_t \theta(t) \\
&= -\nabla f(x;\theta(t))^\top \nabla \mathcal{L}(f(\cdot;\theta(t))) \\
&= -\sum_{i=1}^{n} \sum_{p=1}^{P} \partial_{\theta_p} f(x_i;\theta(t)) \partial_{\theta_p} f(x;\theta(t)) \partial_z \ell(z, y_i) \Big|_{z=f(x_i;\theta(t))}
\end{aligned}$$

# Neural Tangent Kernel

## Definition

**Definition (Neural Tangent Kernel)**

We call Neural Tangent Kernel the kernel form

$$\Theta(x_i, x_j) = \sum_{p=1}^{P} \partial_{\theta_p} f(x_i; \theta(t)) \partial_{\theta_p} f(x_j; \theta(t)).$$

It is the kernel matrix associated with the feature map

$$\left( \partial_{\theta_p} f(x_i; \theta(t)) \right)_{i=1}^{n}.$$

# Neural Tangent Kernel

## Limit kernel

### Definition (Limit NTK I)

Define recursively for $L$ layers with $m$ neurons and activation $\sigma$:

$$\Sigma^{(0)}(x, x') = x^\top x', c_\sigma = \left(\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\sigma(z)^2]\right)^{-1}$$

$$\Lambda^{(h)}(x, x') = \begin{pmatrix} \Sigma^{(h-1)}(x, x) & \Sigma^{(h-1)}(x, x') \\ \Sigma^{(h-1)}(x', x) & \Sigma^{(h-1)}(x', x') \end{pmatrix},$$

$$\Sigma^{(h)}(x, x') = c_\sigma \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)],$$

$$\dot{\Sigma}^{(h)}(x, x') = c_\sigma \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Lambda^{(h)})}[\dot{\sigma}(u)\dot{\sigma}(v)],$$

$$\dot{\Sigma}^{(L+1)}(x, x') = 1.$$

# Neural Tangent Kernel

## Limit kernel

**Definition (Limit NTK II)**

The limit NTK is:

$$\Theta^{(L)}(x,x') = \sum_{h=1}^{L+1} \left( \Sigma^{(h-1)}(x,x') \prod_{h'=h}^{L+1} \dot{\Sigma}^{(h')}(x,x') \right).$$

# Neural Tangent Kernel

## Convergence to the limit kernel

$$f^{(h)}(x) = W^{(h)} g^{(h-1)}(x) + b^{(h)}, \quad g^{(h)}(x) = \frac{c_\sigma}{\sqrt{m}} \sigma\left(f^{(h)}(x)\right).$$

Initialization: $W^{(h)} \sim \mathcal{N}\left(0, I\right), \quad b^{(h)} \sim \mathcal{N}\left(0, 1\right)$ i.i.d.

### Theorem (Convergence of NTK)

*Let $\varepsilon > 0$ and $\delta \in (0,1)$, $\sigma = ReLU$, $m \geq \Omega(\frac{L^6}{\varepsilon^4} \log(\frac{L}{\delta}))$, then for all $x, x'$ in the unit ball, with probability at least $1 - \delta$:*

$$\left| \nabla_\theta f(x;\theta)^\top \nabla_\theta f(x';\theta) - \Theta^{(L)}(x,x') \right| \leq (1+L)\varepsilon.$$

# Neural UCB

## Algorithm

**Algorithm:** Neural UCB

**Exploration**:

$A_0 = \lambda I$

$g_t(a) = \frac{1}{\sqrt{m}} \nabla_\theta f(\phi(a); \theta_{t-1})$

$A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} g_t(a_t) g_t(a_t)^\top A_{t-1}^{-1}}{1 + g_t(a_t)^\top A_{t-1}^{-1} g_t(a_t)}$

$ExplorationBonus_t(a) = \gamma \sqrt{g_t(a)^\top A_{t-1}^{-1} g_t(a)}$

**Training**:

$\theta_t = \mathrm{SGD}(\{\phi(a_i)\}_{i=1}^t, \{r(a_i)\}_{i=1}^t; \theta_{t-1})$

$\hat{\mu}_t(a) = f(\phi(a); \theta_t)$

# Neural UCB

## Regret analysis

### Theorem (Neural UCB regret)

*With probability at least* $1 - \delta$, *if* $m \geq PolyLog(T, L, K, \lambda, log(1/\delta))$:

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

*where*

$$\tilde{d} = \frac{\log det(I + \Theta^{(L)}/\lambda)}{\log(1 + TK/\lambda)}$$

*is the effective dimension of the neural tangent kernel.*

# Neural UCB

## Regret analysis

### Theorem (Neural UCB regret)

*With probability at least* $1-\delta$, *if* $m \geq PolyLog(T, L, K, \lambda, log(1/\delta))$*:*

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

*where*

$$\tilde{d} = \frac{\log det(I + \Theta^{(L)}/\lambda)}{\log(1 + TK/\lambda)}$$

*is the effective dimension of the neural tangent kernel.*

Problems:

- Bound on $m$ can be huge $(K^4, L^6, T^4, \dots)$,
- Scaling factor $\gamma$ not as explicit as LinUCB.

# Neural UCB

## Regret analysis

### Theorem (Neural UCB regret)

*With probability at least* $1-\delta$*, if* $m \geq PolyLog(T, L, K, \lambda, log(1/\delta))$*:*

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

*where*

$$\tilde{d} = \frac{\log det(I + \Theta^{(L)}/\lambda)}{\log(1 + TK/\lambda)}$$

*is the effective dimension of the neural tangent kernel.*

Problems:

- Bound on $m$ can be huge $(K^4, L^6, T^4, \dots)$,
- Scaling factor $\gamma$ not as explicit as LinUCB.

In practice : use small $m$ and tune $\gamma_t \equiv \gamma$ as an hyperparameter.

# Neural UCB

## Linear rewards

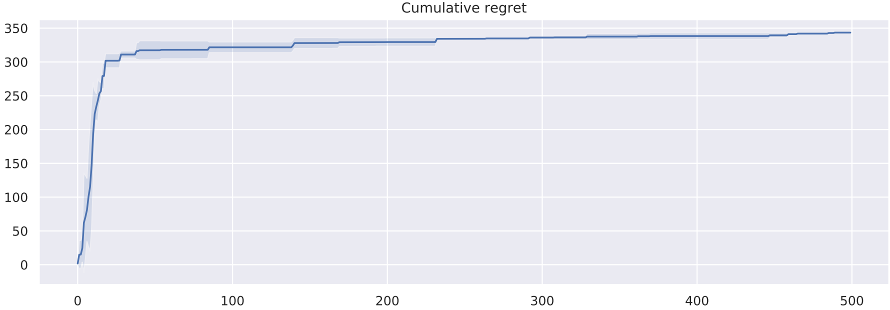



Figure: 4 arms, 16 features per arm, 10% Gaussian noise, 1 hidden layer of 64 neurons.

$r_a = 10\theta^\top \phi(a)$.

Frequency of optimal arm selection : $\geq 93\%$.

# Neural UCB
## Quadratic rewards

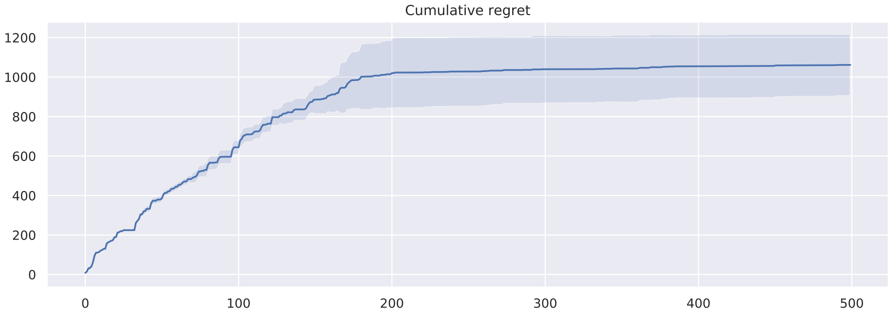



Figure: 4 arms, 16 features per arm, 10% Gaussian noise, 1 hidden layer of 64 neurons.
$r_a = 100(\theta^\top \phi(a))^2$.
Frequency of optimal arm selection : $\geq 73\%$.

# Neural UCB

## Cosine rewards


Figure: 4 arms, 20 features per arm, 10% Gaussian noise.
$r_a = \cos(10\theta^\top \phi(a))$.
Frequency of optimal arm selection : $\geq 80\%$.

# Neural UCB-VI

## Algorithm

**Algorithm:** Neural UCB-VI

**Exploration**:

$A_{0,h} = \lambda I$

$g_{t,h}(s,a) = \frac{1}{\sqrt{m}} \nabla_\theta f(\phi(s,a);\theta_{t-1,h})$

$A_{t,h}^{-1} = A_{t-1,h}^{-1} - \frac{A_{t-1,h}^{-1} g_{t,h}(s_h^t,a_h^t) g_{t,h}(s_h^t,a_h^t)^\top A_{t-1,h}^{-1}}{1+g_{t,h}(s_h^t,a_h^t)^\top A_{t-1,h}^{-1} g_{t,h}(s_h^t,a_h^t)}$

$ExplorationBonus_h^t(s,a) = \gamma \sqrt{g_{t,h}(s,a)^\top A_{t,h}^{-1} g_{t,h}(s,a)}$

**Training**:

$\theta_{t,h} =$
$\mathrm{SGD}(\{\phi(s_h^i,a_h^i)\}_{i=1}^t, \{r(s_h^i,a_h^i) + \max_a \tilde{Q}_{h+1}^i(s_{h+1}^i,a)\}_{i=1}^t; \theta_{t-1,h})$

$\hat{Q}_h^t(a) = f(\phi(s_h^t,a_h^t);\theta_{t,h})$

# Neural UCB-VI

## Algorithm

**Algorithm:** Neural UCB-VI

**Exploration**:

$A_{0,h} = \lambda I$

$g_{t,h}(s,a) = \frac{1}{\sqrt{m}} \nabla_\theta f(\phi(s,a); \theta_{t-1,h})$

$A_{t,h}^{-1} = A_{t-1,h}^{-1} - \frac{A_{t-1,h}^{-1} g_{t,h}(s_h^t, a_h^t) g_{t,h}(s_h^t, a_h^t)^\top A_{t-1,h}^{-1}}{1 + g_{t,h}(s_h^t, a_h^t)^\top A_{t-1,h}^{-1} g_{t,h}(s_h^t, a_h^t)}$

$ExplorationBonus_h^t(s,a) = \gamma \sqrt{g_{t,h}(s,a)^\top A_{t,h}^{-1} g_{t,h}(s,a)}$

**Training**:

$\theta_{t,h} =$
$\mathrm{SGD}(\{\phi(s_h^i, a_h^i)\}_{i=1}^t, \{r(s_h^i, a_h^i) + \max_a \tilde{Q}_{h+1}^i(s_{h+1}^i, a)\}_{i=1}^t; \theta_{t-1,h})$

$\hat{Q}_h^t(a) = f(\phi(s_h^t, a_h^t); \theta_{t,h})$

In practice, good and fast results by sharing $\theta_{t,h} \equiv \theta_t$ across frames.

# Neural UCB-VI

## Linear rewards

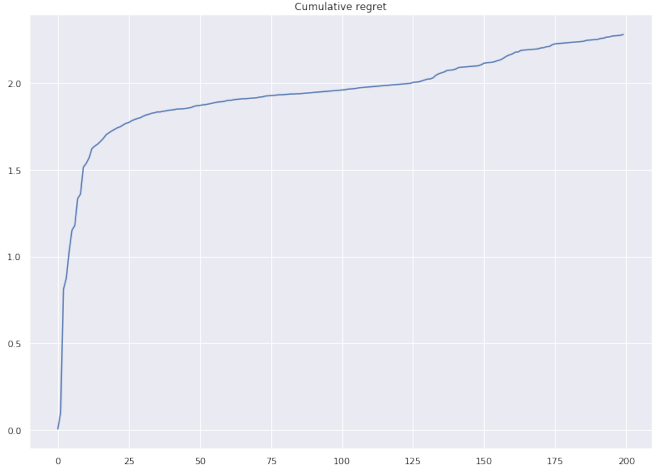



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$, 10% Gaussian noise, $L = 1$, $m = 64$.

# Neural UCB-VI

## Quadratic rewards

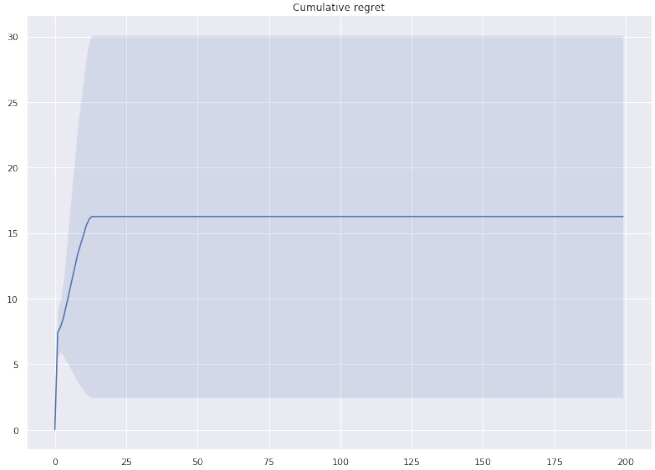



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$, 10% Gaussian noise, $L = 1$, $m = 64$.

# Neural UCB-VI

## Cosine rewards

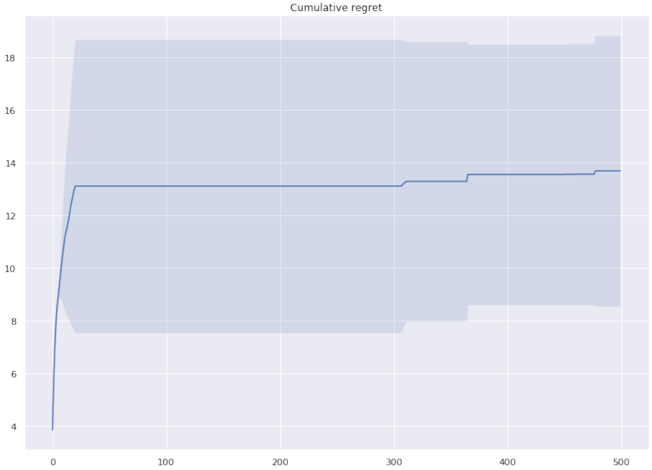



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$, 10% Gaussian noise, $L = 1$, $m = 64$.

# Neural UCB-VI

## Cosine rewards, larger features

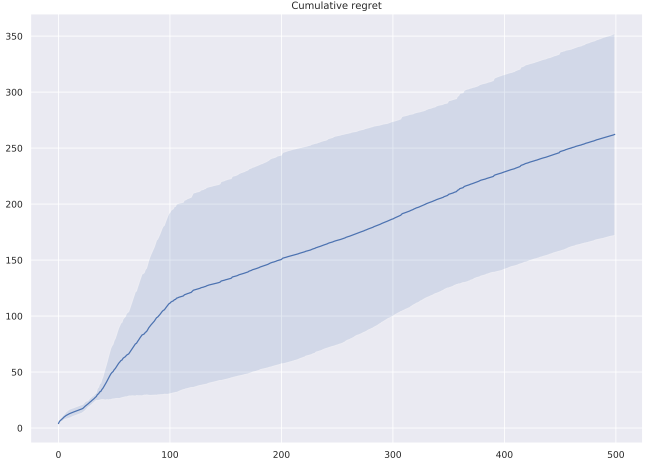



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 128$, 10% Gaussian noise, $L = 1$, $m = 64$.

# Neural UCB-VI

## Cosine rewards, larger MDP

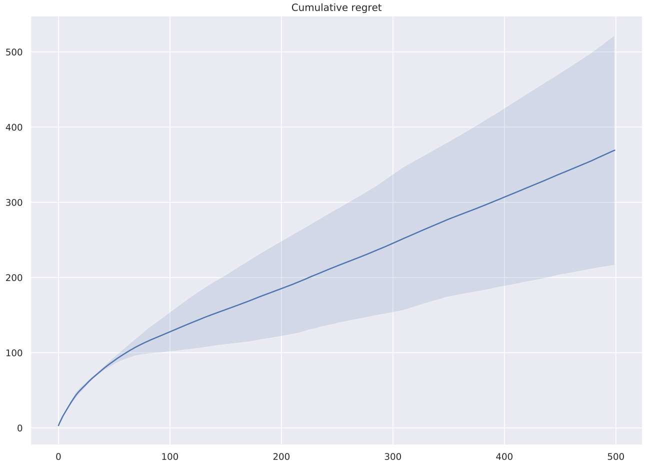



Figure: $H = 6$, $|\mathcal{S}| = 32$, $|\mathcal{A}| = 3$, $d = 16$, 10% Gaussian noise, $L = 1$, $m = 64$.

# Neural UCB-VI

## Linear rewards, unstable regret

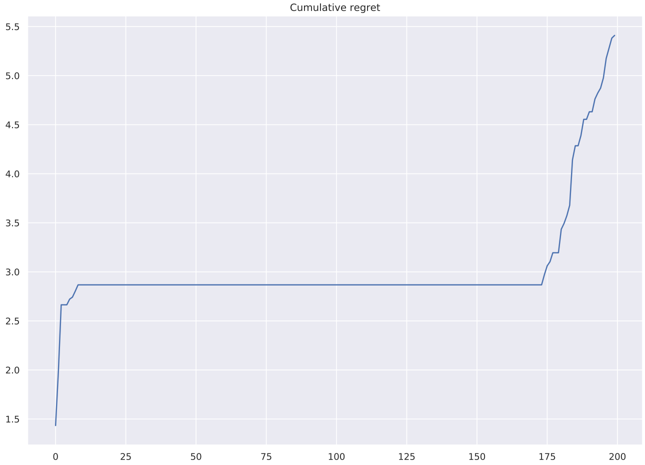



Figure: $H = 6$, $|\mathcal{S}| = 5$, $|\mathcal{A}| = 2$, $d = 16$, 10% Gaussian noise, $L = 1$, $m = 64$.

# Neural UCB-VI

## Regret analysis

**Conjecture (Neural UCB-VI regret)**

*With probability at least* $1-\delta$, *if* $m \geq PolyLog(T, L, |\mathcal{S}||\mathcal{A}|, \ \lambda, log(1/\delta))$*:*

$$R_T \leq \mathcal{O}\left(\tilde{d}^2 H^2 \sqrt{T \log T}\right).$$

*where*

$$\tilde{d} = \frac{\log det(I + \Theta^{(L)}/\lambda)}{\log(1 + T|\mathcal{S}||\mathcal{A}|/\lambda)}$$

*is the effective dimension of the neural tangent kernel.*

# Conclusion

Neural bandit exploration:

- Provably efficient,
- Equivalent to kernelized exploration with dynamic feature map,
- Unrealistic bounds on number of neurons because of NTK.

Neural MDP exploration:

- Conjectured efficient, proof ingredients are similar to those of neural bandit...
- ... therefore also impractical,
- Works decently well in practice on small MDP.

# References I


[1] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net, 2019.

[2] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks, 2018.

[3] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation, 2019.

[4] L. F. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound, 2019.

[5] A. Zanette, D. Brandfonbrener, E. Brunskill, M. Pirotta, and A. Lazaric. Frequentist regret bounds for randomized least-squares value iteration, 2019.

[6] D. Zhou, L. Li, and Q. Gu. Neural contextual bandits with upper confidence bound-based exploration, 2019.

[7] L. Zhou. A survey on contextual multi-armed bandits, 2015.