Image Contextual Bandits: A Visual Transformer Approach

Ryan Wong

Problem: Image Contextual Bandits

- Agent repeatedly chooses from a set of actions (termed "arms") with initially unknown reward distributions
- Exploration vs. Exploitation
- Agent goal: maximize expected cumulative reward (minimize regret)
- At each time step, each arm has an image context
- Recommender System
  - Arm is which show to recommend
  - Context is thumbnail
  - Reward is user interaction

Related work

- Lin UCB
  - Assume reward is linear to context vector
  - $\mathbb{E}[r_{t,a}] = \theta^{*\top} x_{t,a}$
  - $$U_{t,a} = \hat{\theta}_t^\top x_{t,a} + \alpha \sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}}$$
- Neural UCB and CNN UCB
  - Makes no assumption of relationship between context and expected reward
  - Only assumption is sub gaussian noise

ViT UCB

- Use a ViT (+MLP head) to help agent with decision making
- g is the gradient of the parameters
- $A_t = \lambda I + \sum_{s=1}^{t} g(x_s;\theta_{s-1}) g(x_s;\theta_{s-1})^\top$
- According to Neural Tangent Kernel Theory, an overparameterized network behaves linearly in its parameters
- In linucb, their exploration term is $\alpha\sqrt{x_{t,a}^\top A^{-1} x_{t,a}}$
- So by NTK theory, we can replace x with g

$$UCB_{t,a} = \underbrace{f_{ViT}(x_{t,a};\theta_t)}_{\text{Reward Estimate}} + \underbrace{\alpha || \frac{g_{ViT}(x_{t,a},\theta_t)}{\sqrt{d_{LoRA}}} ||_{A_t^{-1}}}_{\text{Exploration Bonus}}$$

Algorithm 1 ViT-UCB with LoRA Parameters

Require: Vision Transformer $f_{\mathrm{ViT}}(x;\theta_{\mathrm{LoRA}})$, regularization $\lambda > 0$, exploration coefficient $\alpha > 0$

Initialize LoRA parameters θ^0_{LoRA} of the ViT

Initialize $A_0 = \lambda I$

for each round $t = 1, 2, \ldots, T$ **do**

 Observe candidate arms $\mathcal{X}_t = \{x_{t,1}, \ldots, x_{t,K}\}$

 for each arm $x_{t,i} \in \mathcal{X}_t$ **do**

  Compute ViT prediction: $\hat{r}_{t,i} = f_{\mathrm{ViT}}(x_{t,i};\theta^{t-1}_{\mathrm{LoRA}})$

  Compute gradient: $g_{t,i} = \nabla_{\theta_{\mathrm{LoRA}}} f_{\mathrm{ViT}}(x_{t,i};\theta^{t-1}_{\mathrm{LoRA}})$

  Compute exploration bonus: $b_{t,i} = \alpha \| \frac{g_{t,i}}{\sqrt{d_{LoRA}}} \|_{A^{-1}_{t-1}}$

  Compute UCB: $U_{t,i} = \hat{r}_{t,i} + b_{t,i}$

 end for

 Select arm $a_t = \arg\max_i U_{t,i}$

 Observe reward r_t for arm a_t

 Update Gram matrix: $A_t = A_{t-1} + g_{t,a_t} g^{\top}_{t,a_t}$

 Update LoRA parameters: θ^t_{LoRA} with Gradient Descent on past rewards $\{(x_{i,a_i}, r_i)\}_{i=1}^{T}$

end for

Mathematical Proof

Assumption 1 (Representability). *We assume the reward function is well-approximated by the first-order tangent model in the LoRA subspace. In other words, there exists* $\boldsymbol{\theta}^*_{LoRA} \in \mathbb{R}^{d_{LoRA}}$ *such that for any* $\boldsymbol{x}$ *in the training set:*

$$f^*(\boldsymbol{x}) = \left\langle \boldsymbol{g}_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{LoRA}), \boldsymbol{\theta}^*_{LoRA} - \boldsymbol{\theta}^0_{LoRA} \right\rangle + \varepsilon(\boldsymbol{x})$$

where $\boldsymbol{g}_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{LoRA}) \triangleq \nabla_{\boldsymbol{\theta}_{LoRA}} f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{ViT},\boldsymbol{\theta}_{LoRA})$*, and for some* $\bar{S} > 0$*:*

$$\left\| \boldsymbol{\theta}^*_{LoRA} - \boldsymbol{\theta}^0_{LoRA} \right\|_2 \leq \bar{S}, \quad |\varepsilon(\boldsymbol{x})| \leq \varepsilon_{\max}$$

Assumption 2 (Bounded Gradients). *There exists* $G_{\max} > 0$ *such that for all* $\boldsymbol{x}$ *with* $\|\boldsymbol{x}\|_F = 1$ *and all* $\boldsymbol{\theta}_{LoRA}$*:*

$$\|\boldsymbol{g}_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{LoRA})\|_2 \leq G_{\max}$$

Assumption 3 (Smoothness). *The ViT-LoRA function is L-smooth in* $\boldsymbol{\theta}_{LoRA}$*: for all* $\boldsymbol{\theta}_{LoRA}, \boldsymbol{\theta}'_{LoRA}$ *and* $\boldsymbol{x}$ *with* $\|\boldsymbol{x}\|_F = 1$*:*

$$\left\| \boldsymbol{g}_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{LoRA}) - \boldsymbol{g}_{ViT}(\boldsymbol{x};\boldsymbol{\theta}'_{LoRA}) \right\|_2 \leq L \left\| \boldsymbol{\theta}_{LoRA} - \boldsymbol{\theta}'_{LoRA} \right\|_2$$

$$|f^*(\boldsymbol{x}) - f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{ViT},\boldsymbol{\theta}_{LoRA})| \leq \alpha \left\| \frac{\boldsymbol{g}_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{LoRA})}{\sqrt{d_{LoRA}}} \right\|_{\mathbf{A}_t^{-1}} + \beta$$

$$\alpha = \sqrt{\log\left(\frac{\det(\boldsymbol{A}_t)}{\det(\lambda \boldsymbol{I})}\right) - 2\log\delta + \sqrt{\lambda}\bar{S}}$$

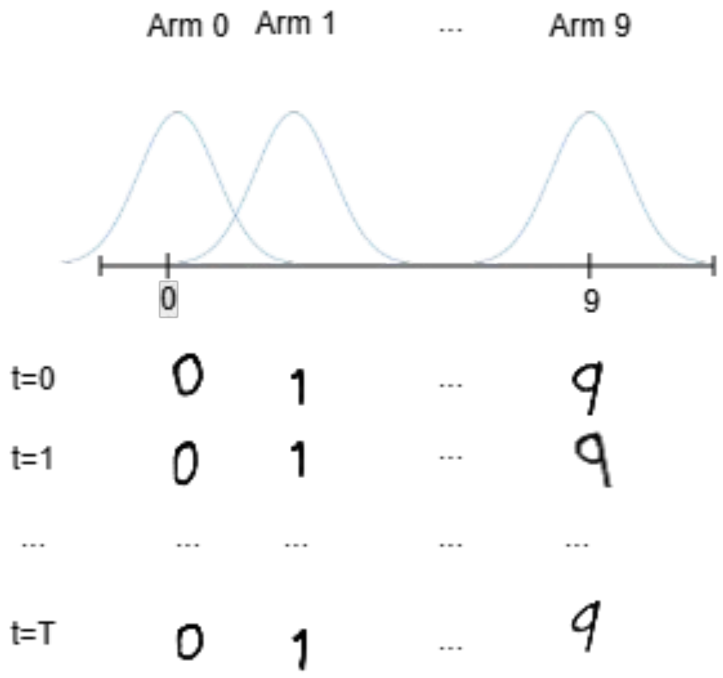
$$\beta = \varepsilon_{\max} + G_{\max}\Delta_t + \beta_{linear}$$

$$\beta_{linear} = |f_{ViT}(\boldsymbol{x};\boldsymbol{\theta}_{ViT},\boldsymbol{\theta}^0_{LoRA})| + \frac{3L}{2}\left\|\boldsymbol{\theta}_{LoRA} - \boldsymbol{\theta}^0_{LoRA}\right\|_2^2 + \frac{C_{ViT}}{6}\cdot\left\|\boldsymbol{\theta}_{LoRA} - \boldsymbol{\theta}^0_{LoRA}\right\|_2^3$$

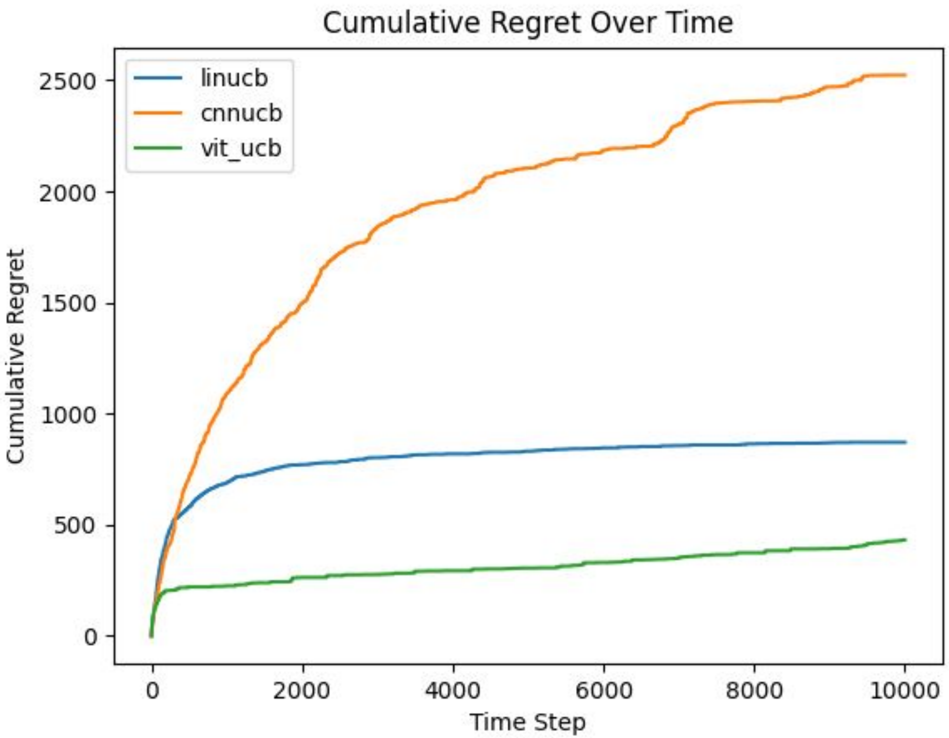
- This shows that the actual reward is within the confidence bound with high probability 1-δ
- β should be negligible for a well trained ViT
- (unproved) this should yield a regret bound of $O(\sqrt{T})$

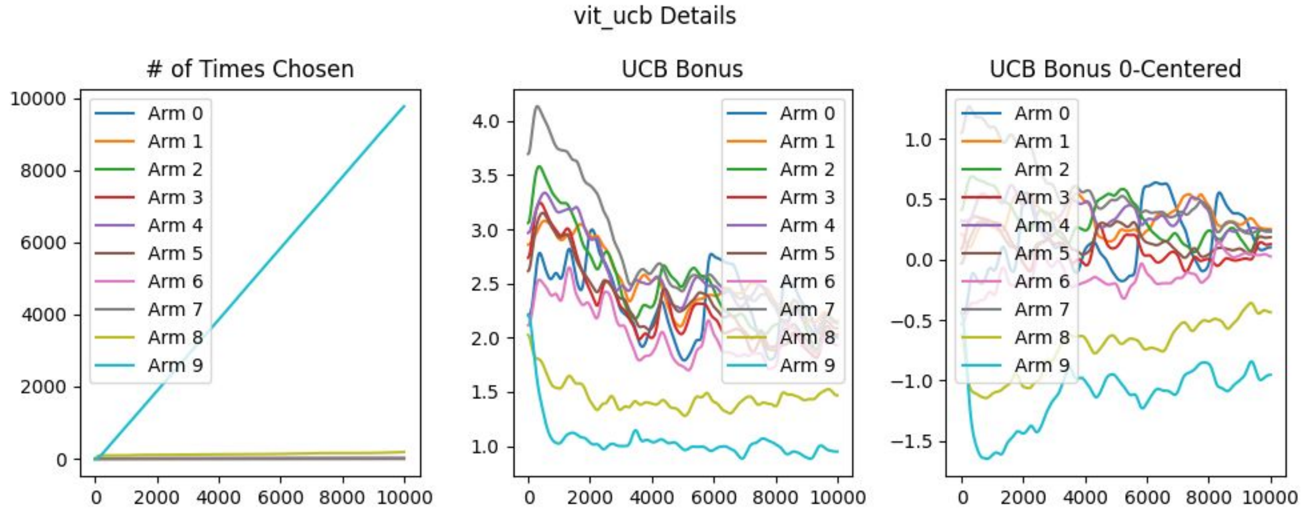
Experiment 1 (handwritten digits)

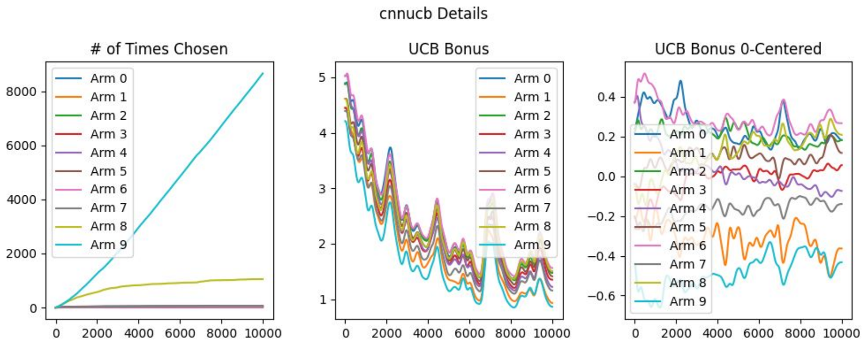
- Each arm is a digit (10 arms total from 0-9)
- Arm i has contexts of handwritten digit i
  - Sampled uniformly random at each t
- Arm i samples reward from N(i, 9)
- T=10000
- Baselines
  - CNN UCB
  - LinUCB (On embedding from a pretrained model)
- ViT UCB
  - Model: WinKawaks/vit-tiny-patch16-224 (5.7M param)
  - LoRA rank = 20, LoRA alpha = 24, alpha=85
  - Tuned using 50 trials of Bayesian Optimization

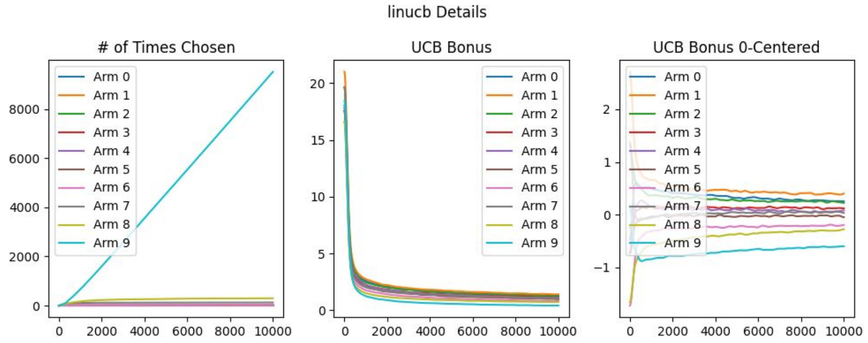


Experiment 1 Results




vit_ucb Details
# of Times Chosen
UCB Bonus
UCB Bonus 0-Centered
Arm 0
Arm 1
Arm 2
Arm 3
Arm 4
Arm 5
Arm 6
Arm 7
Arm 8
Arm 9

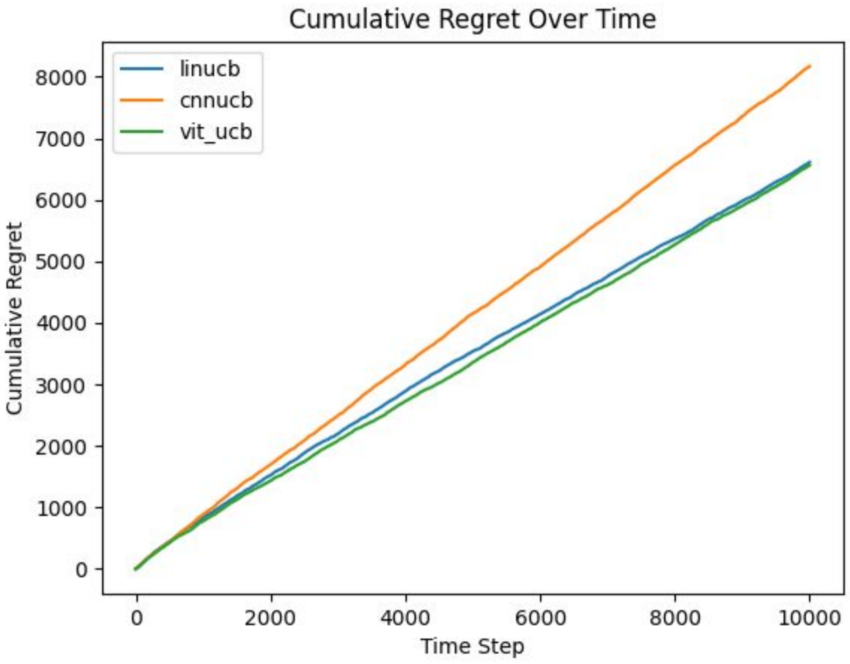

cnnucb Details
# of Times Chosen
UCB Bonus
UCB Bonus 0-Centered
Arm 0
Arm 1
Arm 2
Arm 3
Arm 4
Arm 5
Arm 6
Arm 7
Arm 8
Arm 9

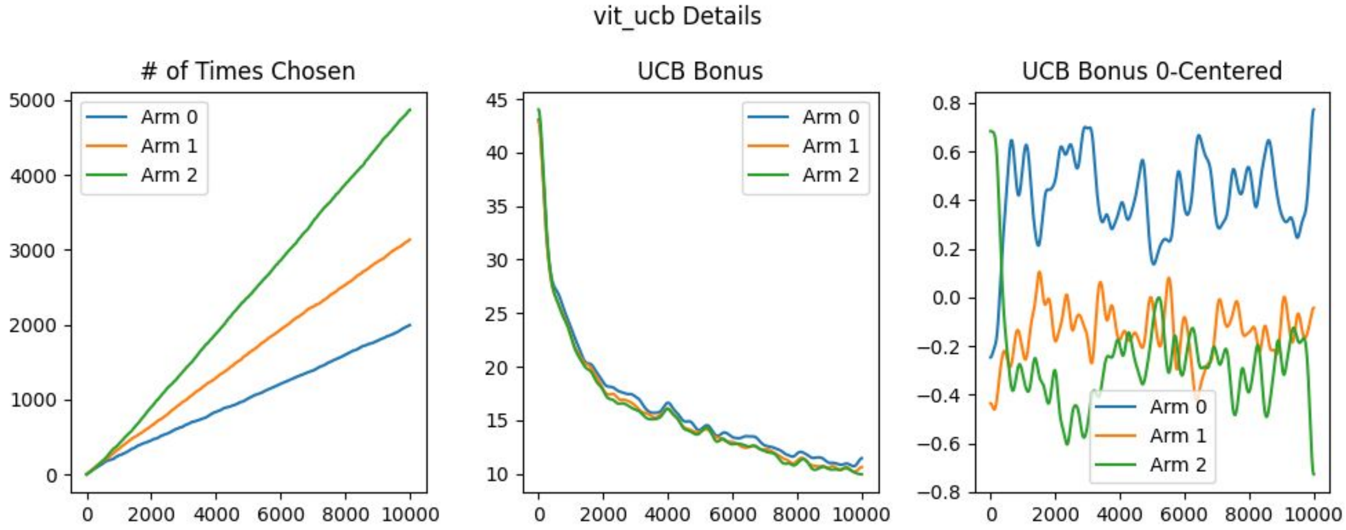

linucb Details
# of Times Chosen
UCB Bonus
UCB Bonus 0-Centered
Arm 0
Arm 1
Arm 2
Arm 3
Arm 4
Arm 5
Arm 6
Arm 7
Arm 8
Arm 9

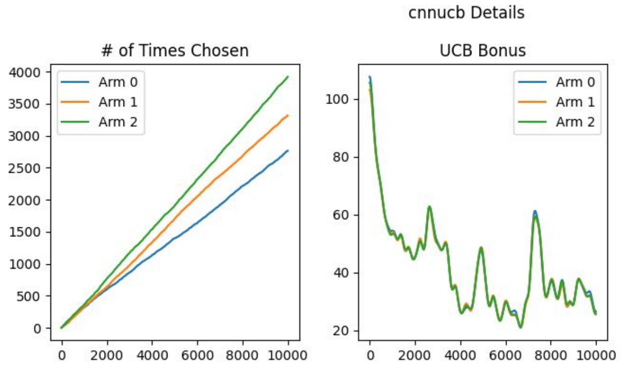
Experiment 2 (Anime)

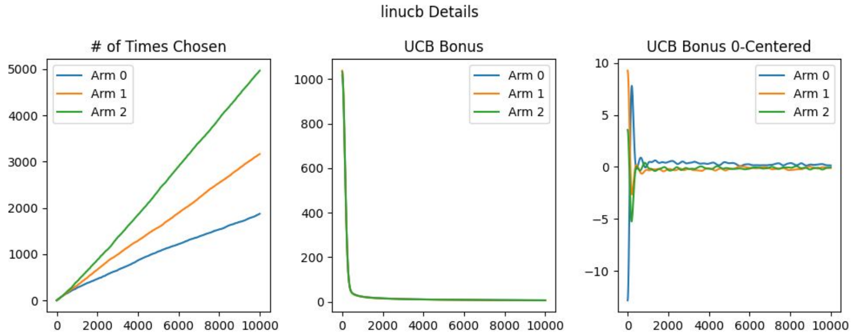
- Contexts are 10000 anime thumbnails pulled from MyAnimeList
- Sort animes by rating and divide into 3 groups of equal size
  - E.g. if animes have ratings {7.2, 2.6, 4.4, 5.3, 9.1, 8.3}
  - Group 0 = {2.6, 4.4}
  - Group 1 = {5.3, 7.2}
  - Group 2 = {8.3, 9.1}
- Arm i has contexts from group i
  - Sampled uniformly random at each t
- Arm i samples reward from N(mean rating of group i, 1)
- In the experiment
  - Group 0 mean ≈ 5.5
  - Group 1 mean ≈ 6.5
  - Group 2 mean ≈ 7.4

Experiment 2 Results




vit_ucb Details
# of Times Chosen
UCB Bonus
UCB Bonus 0-Centered
Arm 0
Arm 1
Arm 2


cnnucb Details
# of Times Chosen
UCB Bonus
UCB Bonus 0-Centered
Arm 0
Arm 1
Arm 2


linucb Details
# of Times Chosen
UCB Bonus
UCB Bonus 0-Centered
Arm 0
Arm 1
Arm 2

Limitations

- Even when using LoRA, there are too many parameters to keep track of A_t
    - Only keep track of parameters from MLP head and a fraction of parameters from ViT LoRA
- Used the smallest ViT I could find
- Architecture of CNN was not tuned
    - Copied architecture of Ban 2021
    - Input image size: 50x50 (once again limited by memory)
    - Two convolutional layers connected with two fully-connected layers, where the first convolutional layer has 32 channels and the second have 64 channels
- Performing gradient descent on all past context-reward pair takes too long
    - Each step, only train on the most recent 50 data points

Next steps

- Refine proof
- First draft of paper
- Final draft based on feedback
- Polish GitHub repository

Thank You