1 Experiments

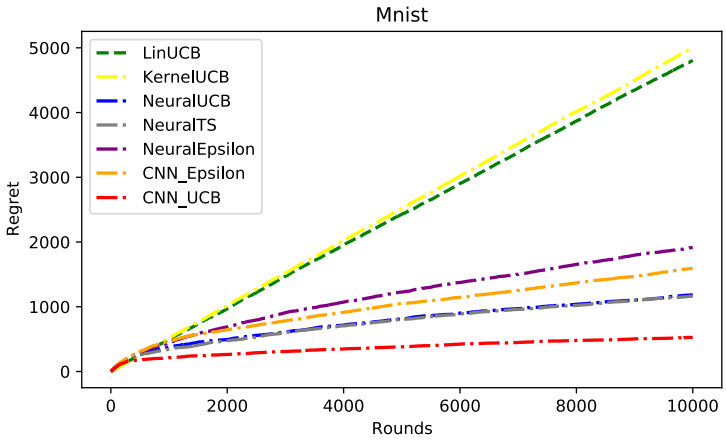

Figure 1: Regret comparison on Mnist.

Image data sets. We choose three well-known image data sets: Mnist [LeCun et al., 1998], Notmnist, and Cifar-10 [Krizhevsky et al., 2009]. All of them are 10-class classification data sets. Following the evaluation setting of existing works [Zhou et al., 2020, Valko et al., 2013, Deshmukh et al., 2017], transform the classification into bandit problem. Consider an image $\mathbf{x} \in \mathbb{R}^{c \times p}$, we aim to classify it from 10 classes. Then, in each round, 10 arms is presented to the learner, formed by 10 tensors in sequence $\mathbf{x}_1 = (\mathbf{x}, \mathbf{0}, \ldots, \mathbf{0}), \mathbf{x}_2 = (\mathbf{0}, \mathbf{x}, \ldots, \mathbf{0}), \ldots, \mathbf{x}_{10} = (\mathbf{0}, \mathbf{0}, \ldots, \mathbf{x}) \in \mathbb{R}^{10 \times c \times p}$, matching the 10 classes. The reward is defined as 1 if the index of selected arm equals the index of $\mathbf{x}$' ground-truth class; Otherwise, the reward is 0. For example, an image with number "6" belonging to the 7-th class on Mnist data set will be transformed into 10 arms in a round and the reward will be 1 if selecting the 7-th arm; Otherwise, the reward is 0. For **Mnist** and **Notmnist**, we transform them into a 10-arm bandit problem. For **Cifar-10**, we tranform it into a 3-arm bandit problem to alleviate the huge computation cost caused by the input dimensions. Specifically, the arm 0 $(\mathbf{x}, \mathbf{0}, \mathbf{0})$ matches the image classes $0 - 3$; the arm 1 $(\mathbf{0}, \mathbf{x}, \mathbf{0})$ matches the image classes $4 - 7$; the arm 1 $(\mathbf{0}, \mathbf{0}, \mathbf{x},)$ matches the image classes $8 - 9$.

Yelp data set [1]. Yelp is a data set released in the Yelp data set challenge, which consists of 4.7 million rating entries for 1.57×10^5 restaurants by 1.18 million users. We build the rating matrix by choosing the top 2000 users and top 10000 restaurants and use singular-value decomposition (SVD) to extract the 10-dimension feature vector for each user and restaurant. In this data set, the bandit algorithm is to choose the restaurants with bad ratings. We generate the reward by using the restaurant's gained stars scored by the users. In each rating record, if the user scores the restaurant less than 3 stars (5 stars totally), the reward is 1; Otherwise, the reward is 0. In each round, we set 10 arms as follows: we randomly choose one rating with reward 1 and randomly pick the other 9 restaurants with 0 rewards; then, the representation of each arm is the concatenation of corresponding user feature vector and restaurant feature vector.

Configurations. For LinUCB, following [Li et al., 2010], there is a exploration constant α (to tune the scale of UCB) and we do a grid search for α over $(0.01, 0.1, 1)$. For KernelUCB [Valko et al., 2013], we use the radial basis function kernel and stop adding contexts after 2000 rounds. There are regularization parameter λ and exploration parameter ν in KernelUCB and we do the grid search for λ over $(0.1, 1, 10)$ and for ν over $(0.01, 0.1, 1)$. For NeuralUCB and NeuralTS, following setting of [Zhou et al., 2020, Zhang et al., 2020], we use a 2 fully-connected layer with the width 100 and conduct the grid search for the exploration parameter ν over $(0.001, 0.01, 0.1, 1)$ and for the regularization parameter λ over $(0.1, 1, 10)$. For NeuralEpsilon, we use the same neural network with NeuralUCB/TS and do the grid search for the exploration probability ϵ over $(0.01, 0.1, 0.2)$. For CNN-UCB, we use two convolutional layers connected with two fully-connected layers, where the first convolutional layer has 32 channels and the second have 64 channels. For image data sets, we

[1] https://www.yelp.com/dataset

use the 2-dimension CNN while using 1-dimension CNN for Yelp data set. And we conduct the grid search for the exploration parameter ν over $(0.001, 0.01, 0.1, 1)$ and for the regularization parameter λ over $(1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5})$. For CNN-Epsilon, we use the same CNN with CNN-UCB and do the grid search for the exploration probability ϵ over $(0.01, 0.1, 0.2)$. For the neural bandits including NeuralUCB/TS and CNN-UCB, as it has expensive computation cost to store and compute the whole matrix $\mathbf{A}_t$, we use a diagonal matrix which consists of the diagonal elements of $\mathbf{A}_t$ to approximate $\mathbf{A}_t$. For all grid-searched parameters, we choose the best of them for the comparison and report the averaged results of 5 runs.

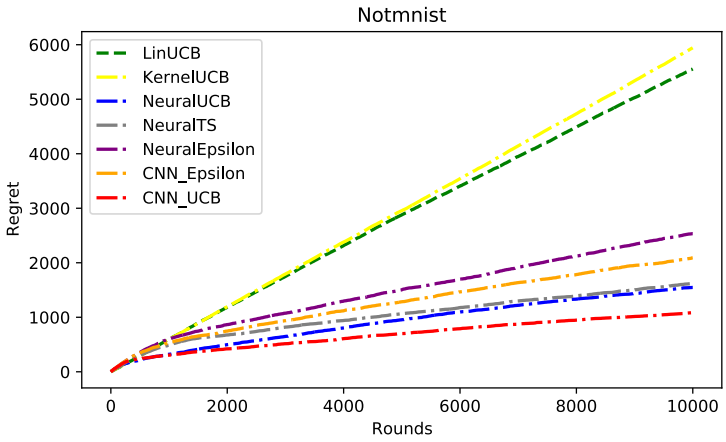

Figure 2: Regret comparison on Notmnist.

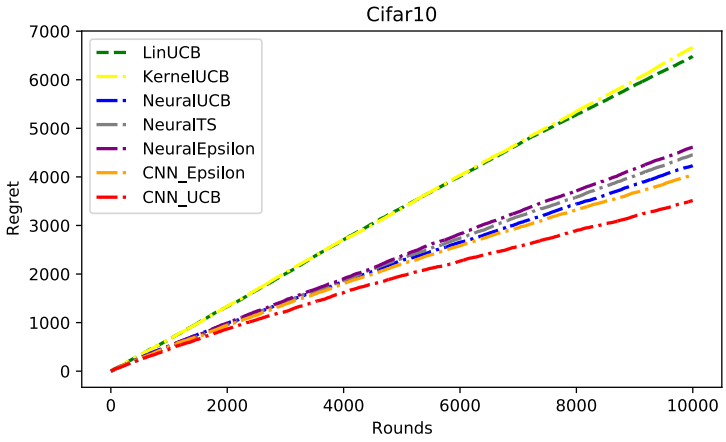

Figure 3: Regret comparison on Cifar10.

References

A. A. Deshmukh, U. Dogan, and C. Scott. Multi-task learning for contextual bandits. In *Advances in neural information processing systems*, pages 4848–4856, 2017.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

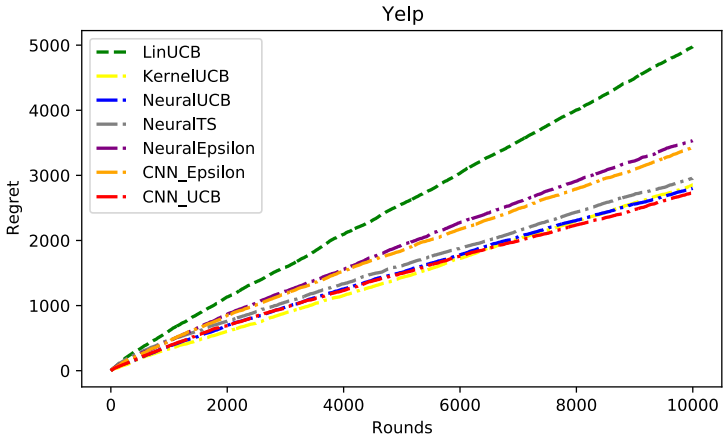

Figure 4: Regret comparison on Yelp.

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

W. Zhang, D. Zhou, L. Li, and Q. Gu. Neural thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

D. Zhou, L. Li, and Q. Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.