Distilled Bandit for Real-time Online Optimization

Ziying Liu
Indeed.com
Sunnyvale, USA
ziyingl@indeed.com

Haiyan Luo
Indeed.com
Sunnyvale, USA
hluo@indeed.com

Jianjie Ma
Indeed.com
Sunnyvale, USA
jma@indeed.com

Yu Sun
Indeed.com
Sunnyvale, USA
sunyu@indeed.com

Yujing Wu
Indeed.com
Sunnyvale, USA
yujingw@indeed.com

Elizabeth Lattanzio
Indeed.com
Sunnyvale, USA
elattanzio@indeed.com

ABSTRACT

In online learning and decision making problems such as contextual bandit, a crucial trade-off is about the complexity of the model: while a complex model can potentially deliver better performance, the slower inference speed that comes with it would often lead to violations of the real-time requirements. On the other hand, a simple model can have the advantage of fast inference speed, but its performance is usually less desirable.

We tackle this problem by leveraging knowledge distillation technique. In particular, we propose to rely on a simpler model for real-time decision making, in the meantime we use a more complex teacher model to 'guide' the student model towards better performance. To address the mismatch of inference speeds between the teacher model and the student model, we introduced a replay buffer to cache the training data. Experimental results on two public data sets confirmed that our approach is able to significantly improve the inference speed in online decision making, and greatly enhances the performance of the student model.

ACM Reference Format:
Ziying Liu, Haiyan Luo, Jianjie Ma, Yu Sun, Yujing Wu, and Elizabeth Lattanzio. 2020. Distilled Bandit for Real-time Online Optimization. In *Proceedings of DLP-KDD 2020.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Contextual multi-arm bandit (contextual bandit for short) is an extension of classic multi-arm bandit (MAB) where at each iteration an context vector x is observed. This context vector, along with historical actions and rewards, can be used by a policy to choose the best arm to play. As a natural formulation for most real-life online decision making problems, it fits well in many sequential decision making applications, including recommender system [18, 20, 27, 30], ads creative optimization [14, 23, 31], information retrieval [3, 9, 16] etc. Different algorithms were proposed to solve contextual bandit problem, including LinUCB and LinTS [1, 20], etc., where a linear relation between an arm's expected reward and the context is typically assumed.

With the recent advances in deep learning and approximate Bayesian methods, neural networks have been used to model the context-reward relation [5, 34]. Although convergence bounds are hard to derive for many cases [24], applying deep neural networks to contextual bandit has shown competitive performance on a variety of data sets.

One problem with deep contextual bandit algorithms is that they tend to be slow at inference time comparing to traditional linear approaches. This problem becomes more severe when the number of arms becomes large. For a contextual bandit problem with N arms, the network needs to be evaluated N times at each inference step. Even in the case when arms are parameterized and heuristic optimization is performed (e.g. [14]), multiple evaluations of the neural network is unavoidable. For applications such as ads and recommendation systems, decisions have to be made in real-time with latency requirement around tens of milliseconds. Thus, slow evaluation time would not be acceptable.

Driven by the need to perform inference faster, or on memory limited devices, there has been a lot of work in the literature focusing on model compression of deep neural networks. A variety of techniques are proposed, including quantization or binarization [6, 7, 10, 12], pruning [13, 28], factorization [8, 26], knowledge distillation [15, 25, 32], etc.

Inspired by the idea of teacher-student knowledge distillation [15], we propose a novel approach for solving contextual bandits problem, as shown in Figure 1. In this schema, we train a compact neural network to model the relation between context, action and reward. This is done with the help of a pre-trained, full-complexity 'teacher' model. The compact model, also referred to as 'student' model, is then used for online learning and decision making in contextual bandit, since it can perform inference much faster. Parameters of the compact model are updated online based on the observed feedback as well as knowledge from the teacher model.

We evaluate the proposed approach offline, on CIFAR-10 and Criteo's display ads CTR prediction data set. Its performance is compared to two groups of baselines: the compact model with the student model architecture while not learning from the teacher model outputs, as well as the deep contextual bandits with full-complexity models. Experiment results show that our approach out performs the former in terms of regret, and latter in terms of speed.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DLP-KDD 2020, August 24, 2020, San Diego, California, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

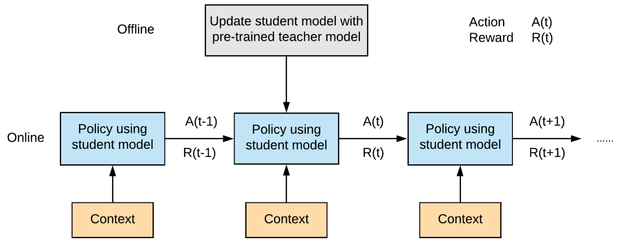
Offline
Update student model with pre-trained teacher model
Action A(t)
Reward R(t)
Online
Policy using student model
A(t-1)
R(t-1)
Policy using student model
A(t)
R(t)
Policy using student model
A(t+1)
R(t+1)
......
Context
Context
Context

Figure 1: Knowledge distillation bandit

2 PRELIMINARIES

2.1 Contextual multi-armed bandit

In the basic contextual bandit setup, an agent needs to make sequential decisions at time steps $\{1, 2, \cdots, T\}$, based on past observations of the world. At each time step t,

- The agent observes a current context vector x_t and is given a set of arms A_t or actions to choose from.
- Based on observed payoffs in previous time steps, the agent chooses an arm $a_t \in A_t$, and receives payoff r_{t,a_t} whose expectation depends on x_t and a_t.
- The agent can improve its arm-selection strategy with the newly collected observation (x_t, a_t, r_{t,a_t}). The objective is to minimize the 'regret', which is defined as

$$R_\Omega(T) = E[\sum_{t=1}^{T} r_{t,a_t^*}] - E[\sum_{t=1}^{T} r_{t,a_t}] \tag{1}$$

where Ω is the agent's arm-choosing policy and a_t^* is the arm with maximum expected payoff at time t. Equivalently, the performance of contextual bandit algorithm can also be evaluated by cumulative rewards.
- For arms that are not chosen at time t, no reward is observed for that time step.

Many algorithms can be found in existing literature that solves the contextual bandit problem and its variations. In recent years, combining deep neural networks with contextual bandit to solve online decision making problems has become an attractive idea [5, 24]. While popular techniques such as LinUCB and Thompson sampling are extended to deep neural networks, it was realized that the high complexity and slow inference time of deep networks had become an obstacle to adopting deep contextual bandit algorithms in real-time applications [24].

2.2 Knowledge distillation

Knowledge Distillation (KD) can be considered as one type of model compression which trains a smaller model (student) with low resource requirements and hopefully small performance degradation comparing to an original, more complex model (teacher). Hinton et al. [15] first proposed the concept of KD in the teacher–student framework by using the teacher's softened output to guide the student model's learning. The student model is trained with a distillation loss in addition to the task loss.

After this seminal work, many modifications were proposed [22, 25, 32]. Some transfer signals other than output, e.g. intermediate layer weights, other work applies the idea to large models such as BERT [29, 33]. There is one line of research that focuses on "online knowledge distillation" [2, 4, 19]. The idea is more about using large scale parallel training in the absence of pre-trained powerful teacher model. Some can deliver better performance than state-of-the-art approaches without sacrificing training or inference complexity. Our approach is different from this because we aim at improving inference time speed with KD.

3 DISTILLED DEEP CONTEXTUAL BANDIT

3.1 Algorithm overview

To leverage the high performance of the teacher model while achieving low latency of online inference for contextual bandit algorithm, we propose a simple yet powerful approach for performing contextual bandits using knowledge distillation. We assume binary rewards in the remaining of the paper. However, our approach can be easily adapted to contextual bandit problems with continuous rewards. The algorithm assumes starting with pre-trained teacher models, one for each arm (or one single pre-trained model when arms are parameterized) that was trained offline. While having high accuracy, the teacher model takes long time to predict due to its complexity and thus is infeasible to be used for online decision making. We use a light weight student model to make decisions online. The student model learns from the behaviors of both the teacher model as well as the ground truth, which is partially observed rewards associated with chosen actions.

To allow the teacher model to provide guidance for the student model during online updates, we introduce an experience replay buffer similar to that used in neural network based reinforcement learning [17, 21]. Our goal here is not to remove undesirable temporal correlations (because there is no dynamics in our case) in training sample, but to accommodate the speed gap between the teacher model and the student model.

With this experience replay buffer, online updates of the student model are still performed in mini-batches, following the procedure described below:

- Upon receiving a context vector x_i, the student model makes a decision a_i using Thompson sampling algorithm with dropout (Algorithm 1). It then receives a reward r_i and stores a tuple (x_i, a_i, r_i) into the replay buffer.
- The teacher model samples data points from the replay buffer, and augments each tuple (x_i, a_i, r_i) with $(x_i, a_u, \hat{r}_i)$, where $a_u \in \mathcal{A} \setminus a_i$ is unselected, counterfactual action, and $\hat{r}_i$ the softened reward generated by the teacher model.
- At the end of each time interval, the student model randomly draws mini-batches from the replay buffer, and updates its own parameters by minimizing cost function described in Section 3.2.

A schematic plot is given in Figure 2. Note that we can also update the teacher model in mini-batches using the data in the replay buffer. Its newly gained knowledge is also transferred through distillation to the student model.

The replay buffer implements a FIFO replacement policy and we choose the size of the buffer to decide how much stale data points

Algorithm 1 Thompson Sampling with Dropout

```
Coefficients for student models at step $t$ $\theta_t$; Student Models $M_s^a(C;\theta)$; Drop out rate $\epsilon$; Pre-trained Teacher Model $M_t^a(C)$;
for t = $1, 2, \ldots, T$ do
    Receive context $C_t$
    Generate $M_{s\,\epsilon}^a(C;\theta)$ by randomly setting neuron to 0 with probability $\epsilon$
    Selection action $A_t = BestAction(C_t;\theta_t)$ based on $M_{s\,\epsilon}^a(C;\theta)$
    Take action $A_t$
    Observe reward $R_t$
    Append $(C_t, A_t, R_t)$ to the replay buffer (partial list of historical data) $H$
    Update $M_s^a(C;\theta)$ with $H$ and $M_t^a(C)$ using adam optimizer, with loss function described in Section 3.2
end for
```

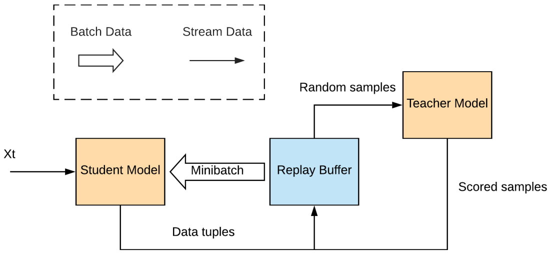

Figure 2: The student model online update with experience replay buffer

to keep. Because the teacher model is much slower than the student model, it cannot score all the data points in real-time, so we only use the teacher model to augment random samples (of unscored points) in the replay buffer. The down-sample rate ϵ is decided based on its processing capacity.

3.2 Loss function

In contextual bandit problems, only the rewards of chosen arms can be observed. For a given arm a, when the ground truth reward is available, the loss function that is computed from ground truth for a given sample i is given by Equation 2:

$$L^a_{KD}(i) = (1-\alpha)L(r^{a(i)}, \hat{y}_s^{a(i)}) \tag{2}$$

where L is the cross-entropy loss for student model, $r^{a(i)}$ the ground truth reward arm a receives for sample i, $\hat{y}_s$ is the predicted reward computed by the student model when a given arm a is chosen. Parameter $\alpha \in [0, 1]$ controls the relative contribution between ground truth label and teacher model predictions.

When scores from teacher model is available for the given sample, an additional loss function term is given by the following equation

$$L^a_{KD}(i) = \alpha D_{KL}(\sigma(\frac{\hat{y}_t^{a(i)}}{T}), \lambda(\frac{\hat{y}_s^{a(i)}}{T}))T^2 \tag{3}$$

where $\hat{y}_t$ is the output of teacher models for the given arm, T is the temperature, σ is the softmax function and λ is the log-softmax function. Both T and α are hyperparameters that can be tuned.

As describe in Section 3.1, when a data sample is scored by the teacher model, we compute the teacher model's output for not only the arm that is pulled, but also for those arms that are not actually pulled, given the same context. This additional information helps the student model to learn the reward function of less-pulled arms more efficiently, thereby speeds up the distillation training.

When learning in batches, aggregating Equation 2 and Equation 3, loss function for arm a for a given batch is given by:

$$\begin{aligned} L^a_{KD} =&(1-\alpha)\sum_{i=1}^{N} I_g(a,i)L(r^{a(i)}, \hat{y}_s^{a(i)}) \\ &+\alpha\sum_{i=1}^{N} I_t(i)D_{KL}(\sigma(\frac{\hat{y}_t^{a(i)}}{T}), \lambda(\frac{\hat{y}_s^{a(i)}}{T}))T^2 \end{aligned} \tag{4}$$

Here, N is the batch size, $I_g(a,i)$ is the indicator function for whether arm a is selected for sample i, $I_t(i)$ is the indicator function for whether sample i is scored by teacher model.

Let N_a be the number of times arm a is selected by the policy, N_t be the number of times the output from the teacher model is available in the given batch, then the above loss function is equivalent to:

$$L^a_{KD} = (1-\alpha)\frac{N_a}{N}L(r^{a(i)}, \hat{y}_s^{a(i)}) + \alpha\frac{N_t}{N}D_{KL}(\sigma(\frac{\hat{y}_t^{a(i)}}{T}), \lambda(\frac{\hat{y}_s^{a(i)}}{T}))T^2 \tag{5}$$

The contribution of loss from ground truth label for arm a is thus not only driven by α, but also number of times the arm is chosen, as well as the number of times scores from teacher model is available. Since the number of times the arm is chosen varies for different arms, the relative contribution of ground truth and teacher models in the loss functions for different arms will vary, causing bias for this naive approach. To alleviate this problem, we modify the contribution to loss function from ground truth for the distilled bandit algorithm as the following:

$$L^a_{KD} = (1-\alpha)\frac{1}{p_a^t}L(r^a, \hat{y}_s^a) \tag{6}$$

where $p_a^t = \frac{N_a}{N}$ is the probability of arm a being chosen up to time t. We apply pseudo counts to the calculation of p_a^t in order to avoid numerical issues in cold start situations. We also apply lower thresholds to p_a^t so that for rarely pulled arms the probabilities do not become too small, otherwise they'll dominate the loss function. The p_a^t is thus calculated as

$$p_a^t = \frac{N_a + \beta_0}{N + \beta_1} \tag{7}$$

In all our experiments presented in the next section, we set the parameters $\beta_0 = \beta_1 = 100$.

4 EXPERIMENTS

4.1 Image classification on CIFAR-10

Any fully-labeled classification modeling data set can be turned into contextual bandit data set, which can further be used as benchmark data to evaluate contextual bandit algorithms. We follow the same approach as Beygelzimer and Langford (2009) to simulate a contextual bandit data set using the CIFAR-10 data. Specifically, given any image C_i, we train a 5-layer convolutional neural network as the

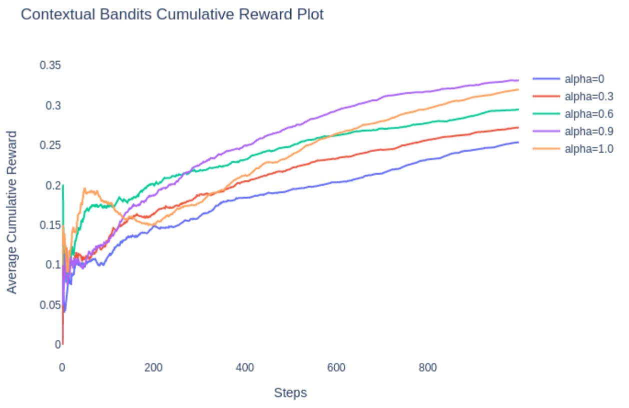

Figure 3: Cumulative Rewards Plot for CIFAR-10 Data

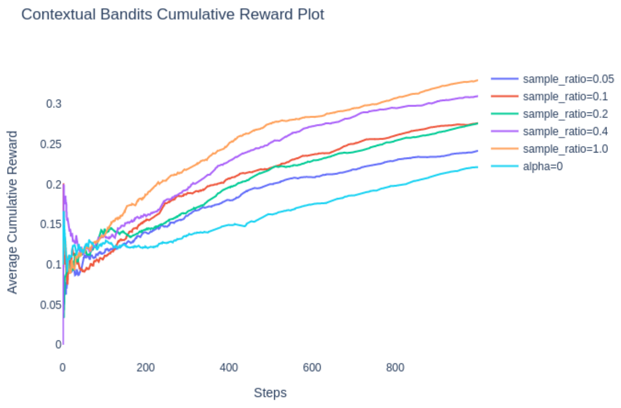

Figure 4: Effect of Different sample ratio ϵ

student model to predict the label. We then compare the prediction results with the ground truth label of the same image. If prediction is correct, we receive reward 1. Otherwise, the received reward is 0. Note that with this setting, we only know whether the prediction of an image's label is correct or not, while having no information about labels that wasn't chosen by our policy. The student model learns from both the ground truth of rewards it has received, as well as a teacher model, which is a pre-trained MobileNet. We use this as a proof-of-concept to test the effectiveness of our approach.

We first compare the inference time of our model. From Table 1, we can see that the student model dramatically speeds up decision making, dropping inference time from 7.7 ms to 0.9 ms.

Models	Inference Time (ms)
Teacher Model	7.7
Student Model	0.9

Table 1: Inference time evaluation comparing the student and teacher models on the CIFAR-10 classification problem.

We then evaluate the performance of our student model by checking the average cumulative rewards over time. We compare the performance of the student model trained using both the teacher model and the ground truth. In this experiment, we compare the effect of different weights α on the performance of the student model. From Figure 3, we can see that student model trained with teacher model outputs ($\alpha > 0$) performs significantly better than the model trained solely on ground truth ($\alpha = 0$). In addition, including ground truth enhances the performance of the student model compared to training the student model solely using outputs from the teacher model ($\alpha = 0.9$ compared to $\alpha = 1.0$). This is consistent with previous observations that including the ground truth enhances the performance in knowledge distillation [15].

As discussed in section 3.1, in online learning situations, due to the limitation of inference speed from the teacher model, it is unable to score every sample in the replay buffer. Instead, it samples on average ϵ ($0 < \epsilon < 1$) data from the replay buffer to score, where ϵ varies depending on relative speed between teacher and student models. We compared the effect of different values ϵ on the performance of the student model. From Figure 4, we can see that our algorithm is able to tolerate very large inference speed gap from the teacher model. We have observed significant performance lift even when $\epsilon = 0.05$.

In summary, our approach is effective on achieving both high performance and low latency, as tested on the CIFAR-10 data.

4.2 Display ads click through rate prediction

We also evaluate the performance of our approach on Criteo's display ads data set, which is a well-known benchmark for CTR prediction tasks. Criteo's dataset contains 45 million samples and each sample has 13 integer features and 26 categorical features. Since the meanings of the features columns are undisclosed, to convert the data set to be used in contextual bandit experiment we have to pick one categorical feature and treat if as adjustable parameter (i.e. arms). In the following we picked feature C17 as arms. The reward is binary (click or not click). We pre-trained a DeepFM model [11] as our teacher model using a subset of the data. We then ran distillation bandit algorithm on the remaining data, training a factorization-machine as student model with both the ground truth and the teacher model as described above. The inference time of the student model and the teacher model are compared in Table 2.

Models	Inference Time (ms)
Teacher Model	58.4
Student Model	15.2

Table 2: Inference time evaluation comparing the student and the teacher model on Criteo data.

Figure 5 shows the average rewards of distillation bandit algorithm compared to the baseline algorithm, which is student model

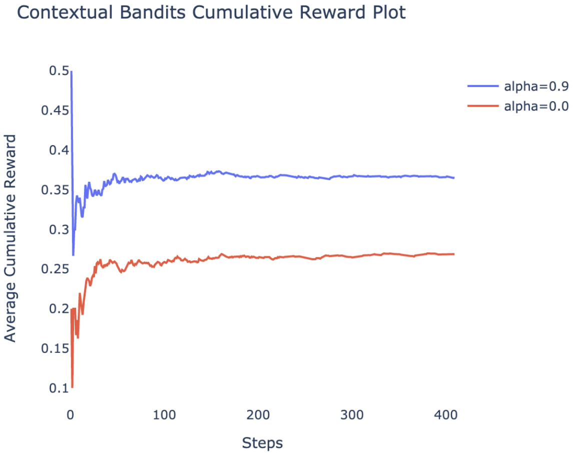

Figure 5: Cumulative Reward Result for Criteo Data

trained only on the ground truth. As shown in the figure, including teacher model significantly enhanced the performance of the algorithm.

5 CONCLUSIONS

Motivated by the need to trade-off model complexity against inference time in online optimization, we proposed to apply knowledge distillation to, in particular, deep contextual bandit problems. We demonstrated that our approach is able to significantly reduce inference time when comparing to complex models, and outperforms compact models that are not leveraging knowledge distillation. To our knowledge, this is the first time that knowledge distillation technique is applied in a contextual bandit setting. The approach we described here enables additional complex features such as image and high dimensional embeddings to be incorporated into online optimization models in various applications, including display ads ranking and recommender systems.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2013. Thompson Sampling for Contextual Bandits with Linear Payoffs. *ArXiv* abs/1209.3352 (2013).

[2] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Róbert Ormándi, George E. Dahl, and Geoffrey E. Hinton. 2018. Large scale distributed neural network training through online distillation. *ArXiv* abs/1804.03235 (2018).

[3] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Gançarski. 2013. Contextual Bandits for Context-Based Information Retrieval, Vol. 8227. 35–42. https://doi.org/10.1007/978-3-642-42042-9_5

[4] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. 2020. Online Knowledge Distillation with Diverse Peers. *ArXiv* abs/1912.00350 (2020).

[5] Mark Collier and Hector Urdiales Llorens. 2018. Deep Contextual Multi-armed Bandits. *ArXiv* abs/1807.09809 (2018).

[6] Matthieu Courbariaux and Yoshua Bengio. 2016. BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. *ArXiv* abs/1602.02830 (2016).

[7] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. *ArXiv* abs/1511.00363 (2015).

[8] Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. 2014. Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation. In *NIPS*.

[9] Dorota Glowacka. 2017. Bandit Algorithms in Interactive Information Retrieval. *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval* (2017).

[10] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. 2014. Compressing Deep Convolutional Networks using Vector Quantization. *ArXiv* abs/1412.6115 (2014).

[11] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *In Proceedings of the IJCAI* (2017).

[12] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep Learning with Limited Numerical Precision. In *ICML*.

[13] Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both Weights and Connections for Efficient Neural Network. *ArXiv* abs/1506.02626 (2015).

[14] Daniel N. Hill, Houssam Nassif, Yi Liu, Anand Iyer, and S. V. N. Vishwanathan. 2017. An Efficient Bandit Algorithm for Realtime Multivariate Optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 1813–1821. https://doi.org/10.1145/3097983.3098184

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML]

[16] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2011. Contextual Bandits for Information Retrieval.

[17] Long ji Lin. 1993. Reinforcement learning for robots using neural networks. In *Technical report DTIC Document*.

[18] Anísio Lacerda. 2015. Contextual Bandits for Multi-objective Recommender Systems. *2015 Brazilian Conference on Intelligent Systems (BRACIS)* (2015), 68–73.

[19] xu lan, Xiatian Zhu, and Shaogang Gong. 2018. Knowledge Distillation by On-the-Fly Native Ensemble. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7517–7527. http://papers.nips.cc/paper/7980-knowledge-distillation-by-on-the-fly-native-ensemble.pdf

[20] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A Contextual-Bandit Approach to Personalized News Article Recommendation. *CoRR* abs/1003.0146 (2010). arXiv:1003.0146 http://arxiv.org/abs/1003.0146

[21] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *CoRR* abs/1509.02971 (2015).

[22] Asit K. Mishra and Debbie Marr. 2018. Apprentice: Using Knowledge Distillation Techniques To Improve Low-Precision Network Accuracy. *ArXiv* abs/1711.05852 (2018).

[23] Daisuke Moriwaki, Komei Fujita, Shota Yasui, and Takahiro Hoshino. 2019. Fatigue-Aware Ad Creative Selection. arXiv:1908.08936 [cs.CY]

[24] Carlos Riquelme, George Tucker, and Jasper Snoek. 2018. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SyYe6k-CW

[25] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. *CoRR* abs/1412.6550 (2015).

[26] Tara N. Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. 2013. Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), 6655–6659.

[27] Yilin Shen, Yue Deng, Avik Ray, and Hongxia Jin. 2018. Interactive recommendation via deep neural memory augmented contextual bandits. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018).

[28] Suraj Srinivas and R. Venkatesh Babu. 2015. Data-free Parameter Pruning for Deep Neural Networks. In *BMVC*.

[29] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient Knowledge Distillation for BERT Model Compression. *ArXiv* abs/1908.09355 (2019).

[30] Liang Tang, Yexi Jiang, Lei Li, Chunqiu Zeng, and Tao Li. 2015. Personalized Recommendation via Parameter-Free Contextual Bandits. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2015).

[31] Liang Tang, Rómer Rosales, Alok Kumar Singh, and Deepak Agarwal. 2013. Automatic ad format selection via contextual bandits. In *CIKM*.

[32] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 7130–7138.

[33] Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. 2019. Extreme Language Model Compression with Optimal Subwords and Shared Projections. *ArXiv* abs/1909.11687 (2019).

[34] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural Contextual Bandits with UCB-based Exploration. *arXiv: Learning* (2020).