SIM-TO-REAL OPTIMIZATION OF COMPLEX REAL WORLD MOBILE NETWORK BY DEEP REINFORCEMENT LEARNING

Yongxi Tan[1],Jin Yang[1],Xin Chen[2],Qitao Song[2],Yunjun Chen[2],Zhangxiang Ye[2],Zhenqiang Su[1]
[1] Futurewei Tech. Inc., NJ Research Center, USA
{yongxi.tan,jin.yang,zhenqiang.su}@huawei.com
[2] Huawei Technologies Co. Ltd, P.R.China
{chenxin,songqitao,chenyunjun,yezhangxiang}@huawei.com

ABSTRACT

Mobile network that millions of people use every day is one of the most complex systems in real world. Optimization of mobile network to meet exploding customer demand and reduce CAPEX/OPEX poses greater challenges than in prior works. Actually, learning to solve complex problems in real world to benefit everyone and make the world better has long been ultimate goal of AI. However, it still remains an unsolved problem for deep reinforcement learning (DRL), given incomplete/imperfect information in real world, huge state/action space, lots of data needed for training, associated time/cost, interactions among multi-agents, potential negative impact to real world, etc. To bridge this reality gap, we proposed a DRL framework to direct transfer optimal policy learned from multi-tasks in source domain to unseen similar tasks in target domain without any further training in both domains. First, we distilled temporal-spatial relationships between cells and mobile users to scalable 3D image-like tensor to best characterize partially observed mobile network. Second, inspired by AlphaGo, we used a novel self-play mechanism to empower DRL agent to gradually improve its intelligence by competing for best record on multiple tasks. Third, a decentralized DRL method is proposed to coordinate multi-agents to compete and cooperate as a team to maximize global reward and minimize potential negative impact. Using 7693 unseen test tasks over 160 unseen simulated mobile networks and 6 field trials over 4 commercial mobile networks in real world, we demonstrated the capability of our approach to direct transfer the learning from one simulator to another simulator, and from simulation to real world. This is the first time that a DRL agent successfully transfers its learning directly from simulation to very complex real world problems with incomplete and imperfect information, huge state/action space and multi-agent interactions.

1 INTRODUCTION

Using deep neural network (LeCun et al., 2015) for a rich representation of high-dimensional visual input and as an universal function approximator, deep reinforcement learning (DRL) have achieved unprecedented success in some challenging domains, such as Atari game (Mnih et al., 2015), Go (Silver et al., 2016; Silver et al., 2017), Poker (Brown & Sandholm 2018). The ultimate goal of AI is creating agent that can not only learn like human, but also make the world better by solving complex problems in real world. However, application of DRL in complex real world problems still remains an unsolved problem due to imperfect information, huge state/action space, big gap between simulation and real world (Rusu et al., 2016; Tobin et al., 2017; Bousmalis & Levine 2017), multi-agent interactions (Vinyals et al., 2017), time/cost, negative impact, etc.

In this work, we use DRL for one-shot optimization of real world mobile network that millions of people use every day. Coverage & capacity optimization (CCO) of mobile network is crucial for mobile carrier to meet exploding customer demand and reduce CAPEX/OPEX (Fan et al., 2014), e.g., $11B CAPEX for Verizon in 2016 (Celentano, 2016), Cisco acquired Intucell for $475M (Marketwired, 2013). However, it poses much more difficult challenges than in prior works. First, mobile network is one of the most complex systems in real world since it is a multi-users, multi-cells (Macro, Small), multi-technologies (3G, 4G, 5G) heterogeneous network, in which mobile services (app, video, IOT) are consumed by billions of devices and many resource

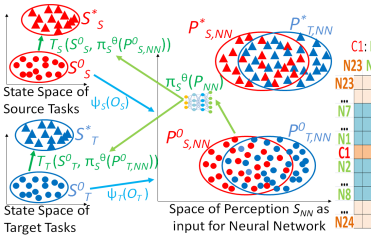

Figure 1: DRL framework

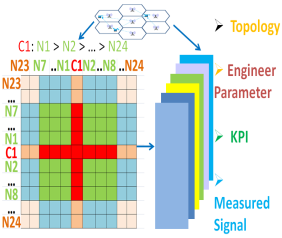

Figure 2: Distill tensor

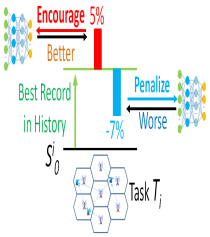

Figure 3: Self play

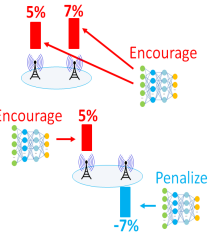

Figure 4: Cooperate

management decisions need to be made to provide seamless services. Second, it is critical to take actions only once (one-shot) since CCO involves time-consuming (1-3 days) and costly site visits to adjust vertical (Tilt) or horizontal (Azimuth) angle of cell antennas. Third, important state information is typically missing (e.g., user location, map and material of building) and erroneous (e.g., wrong Tilt or Azimuth). Fourth, action space is huge: (11 Tilt*13 Azim.)^50≈ 5.8e107 possible actions for 50 cells. At last, coordinating actions of multiple cells is crucial since cell action has significant impact to coverage of itself and interference to neighbor cells.

2 METHODS

2.1 DRL FRAMEWORK TO TRANSFER LEARNING FROM SOURCE DOMAIN TO TARGET DOMAIN

As in Figure 1, given the discrepancy between source and target domain, we use the same perception P_{NN} as input for DRL agent in both domains, by projecting observations O_S and O_T from source and target domain to P_{NN} via $\psi_S(O_S)$ and $\psi_T(O_T)$. Second, if tasks in source and target domain are similar, source and target task distributions can be thought of drawn from the same task population Ω, and direct transfer of policy can be treated as a generalization problem. Therefore, we design and generate sufficient amount of diversified tasks in source domain to minimize the difference between source task distribution and target task distribution from view of agent. Ideally, we want to learn optimal policy π_T^* in target domain to transit from initial state S^0_T to optimal state $S^*_T = T_T(S^0_T, \pi_T^*)$ in one-shot. In practice, we instead learn optimal policy π_S^* in source domain to approximate π_T^*, and further approximate π_S^* by a neural network $\pi_S^\theta(P_{NN})$ with weight θ: $\pi_S^\theta \approx \pi_S^* \approx \pi_T^*$, since S^0_T is partially observable.

2.2 DISTILL TEMPORAL-SPATIAL RELATIONSHIPS TO SCALABLE 3D IMAGE-LIKE TENSOR

Given the complex temporal-spatial relationships between cells and mobile users, discrepancy between simulator and real world, and capability of convolutional neural network (CNN) to exploit spatially local pattern (LeCun et al., 1998), we distill local observations of each agent/cell into scalable 3D tensor as field of view for DRL agent. As in Figure 2, for each cell C_1, we rank all neighbor cells N_i based on relationships between C_1 and N_i, e.g., inter-site distance (ISD), overall interference. We then select most important neighbors (e.g., 24) and put C_1 in center and arrange N_i around C_1 in X-Y axis of tensor based on its rank. At last, for each channel along Z-axis, we extract relevant information from temporal-spatial relationships between each pair of cells in X-Y axis, such as, topology (ISD), key performance indicator (cell load, throughput), measured signal (averaged signal strength, averaged interference), etc.

2.3 SELF-PLAY TO GRADUALLY IMPROVE INTELLIGENCE VIA COMPETITION

Inspired by AlphaGo, we use a novel self-play mechanism to encourage competition for best record on multi-tasks, just like athletes compete for world record in decathlon. As in Figure 3, for initial state S^i_0 of task T_i drawn from a distribution, if new actions achieve better immediate global reward over all cells R_{new} than the best record R_{best} in history by a threshold: $\Delta R_g = R_{new} - R_{best} >= Th_{ge}$, we encourage them by backpropagating a gradient, $g_e = T_e(R - B(s))*d\theta$, here T_e is a function (e.g., 2*Abs(x), Abs is absolute value function), R is expected total reward, $B(s)$ is baseline in REINFORCE (Williams 1992), $d\theta$ is gradient w.r.t. weights θ. If $\Delta R_g <= Th_{gp}$, we penalize them by gradient $g_p = T_p(R - B(s))*d\theta$, here T_p is a function, e.g, -1*Abs. If $Th_{gp} < \Delta R_g < Th_{ge}$, we use simulated annealing (SA) to decide if accepting them by comparing an uniform random number $\in [0,1]$ with acceptance probability $P_g = 1/(1+\exp(\Delta R_g / T_g))$, here T_g is global SA temperature annealed according to certain cooling schedule (e.g., exponential).

2.4 DECENTRALIZED SELF-PLAY, COMPETITIVE AND COOPERATIVE DRL (S2C)

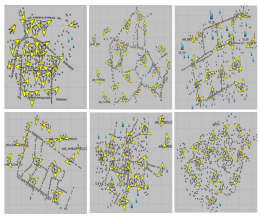

Figure 5: 5 Mobile Network

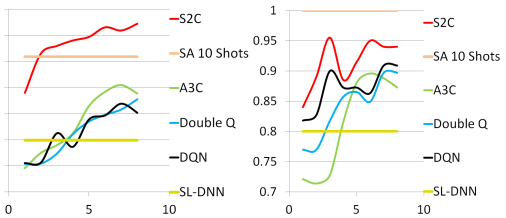

Figure 6: Validation results of DRL agents

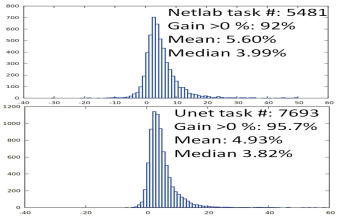

Figure 7: Test results of S2C agent

We proposed a decentralized self-play competitive/cooperative DRL method (S2C), to coordinate multi-agents to compete as a team for best global reward via self-play and cooperate with each other to minimize negative impact. As in Figure 4, each cell/agent takes action by its local view P_{NN}. When actions are accepted at global level, if local reward R_c for cell c_i is larger than a threshold $R_c >= Th_{ce}$, then accept action for c_i with gradient g_e; if $R_c <= Th_{cp}$, then reject it with gradient g_p; if $Th_{cp} < R_c < Th_{ce}$, we use SA with acceptance probability $P_c = 1/(1+\exp(R_c/T_c))$, here T_c is cell level SA temperature.

3 EXPERIMENTS AND RESULTS

3.1 DECENTRALIZED MULTI-AGENT MULTI-TASK DEEP REINFORCEMENT LEARNING IN SIMULATOR

First, we generated 2,380,000 CCO tasks T_i in Netlab simulator, with 10,000 random Tilt settings as initial states for each of 238 simulated mobile networks (<=60 cells, 400-620 users; 5 shown in Figure 5). We designed a SA agent to optimize each training task in 10 steps to generate labels (S^i_0 tensor for each cell in T_i, best Tilt action in 10 shots) for supervised learning (SL-DNN) by a depth-14 residual network (He et al., 2015) with 32×32×12 input and 11 output Tilt ∈ [-5,5]. Using 146k training data, we achieved 78.4% accuracy <=1 degree and 91.5% <=2 degree for 16k validation data. We then use weights of SL-DNN to initialize CNN for 4 DRL agents, DQN (Mnih et al., 2015), Double Q (Hasselt et al., 2016), A3C (Mnih et al., 2016), and S2C. We use 160k-320k training tasks over 80-160 mobile networks (distributed over 640 simulators on 80 VMs) to train agents for one-shot CCO in 8 epochs with 4-16 threads, and 300-500 validation tasks (15-25 mobile networks) per epoch. As in Figure 6, S2C achieved better result, in terms of immediate global reward averaged over all validation tasks (Left, 6.46% for S2C), and ratio of validation tasks with positive global reward (Right, 94% for S2C). We also tested the same S2C policy for 5481 unseen tasks over 238 mobile networks in Netlab without retraining. As in Figure 7 (Upper), it achieved 5.60% average global reward and 92% ratio of test tasks with positive gain. We further verified cross-domain generalization power by testing the same S2C policy for 7693 unseen tasks over 160 unseen mobile networks (100-140 cells, 2480-19840 users) in another simulator Unet, without any further training in both simulators. As in Figure 7 (Lower), it also achieved good results with 4.93% average global reward and 95.7% ratio of test tasks with positive gain.

3.2 DIRECT TRANSFER LEARNING FROM SIMULATION TO REAL WORLD MOBILE NETWORK

To verify the generalization capability of our approach to direct transfer learning from simulation to unseen CCO tasks in unseen very complex real world mobile network without any further training in both domains, we performed 6 field trials over 4 commercial mobile networks that have never been simulated in both simulators, and are very different from all simulated mobile networks, e.g., multi-frequency (MF) or carrier aggregations (CA) has never simulated before, user distribution/number in real world mobile network is temporal-spatial dynamic and very different from static distribution/number in simulators, very different cell/building layouts and radio propagation. We separated commercial mobile network A into 2 neighboring clusters C_1/C_2 (66/47 cells, MF), and performed a trial for each one, with 2.03% RSRP (coverage indicator) and 5.62% RSRQ (interference/capacity indicator) improvement in C_1, and 3.17% RSRP, 4.86% RSRQ improvement for C_2. The 3rd trial was done for whole mobile network A (113 cells, MF), and no significant improvement was observed since most gain has been achieved in first 2 trials. In 4th trial, we achieved 10.79% RSRP and 6.74% RSRQ improvement for mobile network B (151 cells, MF). In 5th trial for mobile network C (131 cells, MF/CA), no significant improvement was observed due to either little room for optimization or significant difference between mobile network C and task distributions in simulation. In 6th trial, we achieved 9.55% RSRP and 12.42% RSRQ improvement for commercial mobile network D (159 cells, MF/CA).

REFERENCES

Noam Brown, Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359:418-424, 2018.

Konstantinos Bousmalis, Sergey Levine. Closing the Simulation-to-Reality Gap for Deep Robotic Learning. *Google Research Blog*, https://research.googleblog.com/2017/10/closing-simulation-to-reality-gap-for.html, 2017

John Celentano. Verizon Wireless: The Big Spender in 2016. *AGL Media Group,* http://www.aglmediagroup.com/verizon-wireless-the-big-spender-in-2016/, 2016

Shaoshuai Fan, Hui Tian, Cigdem Sengul. Self-optimization of coverage and capacity based on a fuzzy neural network with cooperative reinforcement learning. *EURASIP Journal on Wireless Communications and Networking*, 2014:57, 2014.

Hado van Hasselt, Arthur Guez, David Silver. Deep Reinforcement Learning with Double Q-learning. *AAAI'16 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2094-2100, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. *Preprint* at https://arxiv.org/abs/1512.03385, 2015

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278-2324, 1998.

Yann LeCun, Yoshua Bengio and Geoffrey Hinton. Deep Learning. *Nature*, 521:436-444, 2015.

Marketwired. Cisco Announces Intent to Acquire Intucell. http://www.marketwired.com/press-release/cisco-announces-intent-to-acquire-intucell-nasdaq-csco-1748745.htm, 2013

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *Preprint* at https://arxiv.org/abs/1602.01783, 2016

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. *IROS*, 2017

Andrei A. Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, Raia Hadsell. Sim-to-Real Robot Learning from Pixels with Progressive Nets. *Preprint* at https://arxiv.org/abs/1610.04286, 2016.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550:354–359, 2017.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, Rodney Tsing. StarCraft II: A New Challenge for Reinforcement Learning. *Preprint* at https://arxiv.org/abs/1708.04782, 2017.

Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for. Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 1992