Refining Deep Reinforcement learning in Image Restoration

Zhi Ji[1], Boyuan Sun[2]

Abstract

We introduce our approach to refine deep reinforcement learning used in image restoration. The image restoration network uses a tool box of many small scale convolutional neural networks for very specific tasks instead of using a large scale deep convolutional neural train on a wide range of samples and different levels of severities. The agent needs to choose the appropriate tool of the right type and severity of the distortion. This process can be done using Deep Q-learning Network. We provide multiple approaches to improve the DQN. The methods we represent combining three main perspectives: Double Deep Q-learning Networks, Dueling Deep Q-learning Network and prioritized experience replay. All three methods have been formulated and combined to the image restoration network. Their potentials and advantages are studied and explored.

1. Introduction

Introduced since 2014, the deep Q-learning networks (DQN) have been widely apply to solve Markov Decision Processes (MDPs) for the optimal solution. Since then many improvements has been made including double DQN[3], dueling deep Q-network (DDQN)[4], and prioritized experience replay (PER)[5]. We introduce our approach to this image restoring task using a reinforcement learning based CNN tool selection LSTM network with three improvements in deep Q-learning. Firstly, Deep Q-network is replaced by the Double DQNs. Double DQNs provide our network robustness towards overestimations of action values caused by traditional maximum action value estimation in DQN.

Secondly, dueling DQN is added to our network architecture. While the output of original DQN is the Q value of each action taken by agent, dueling DQN output both state values and state advantages of each actions. However, this method does not guarantee the uniqueness of the state values and state advantage for single action we take. To further improve our dueling DQN, we designed to replace the state advantage value with single averaged advantage value to increase the stability and ensure that is the single action solution for greedy policy.

Lastly, we add prioritized replay to our DQN. Instead of updating the batch stochastically, we introduced the TD-error to our system and update batches based on the value of TD-error. The larger the TD-error of a batch is, the more likely the prediction of that batch can be improved. Priorities of each states are also being stored in a tree structure for easy search and update.

2. Background/Related Work

Image restoration has been well studied in the field of deep learning using convolutional neural networks. Images that are corrupted by noise and JPEG artifacts can be restored to a high-quality output. Many studies present a single, deep complex, human made network trained for certain tasks. Tasks include deblurring [14, 15, 18], denoising [6, 13], JPEG artifacts reduction [7, 9] and super-resolution [8, 9, 10, 11, 12, 16, 17]. Our project works on another route to the similar problem.

Using a deep Q-learning Network, we are able to achieve the same or better result for various tasks comparing to studies mentioned above. With addition of three different ways to improve the deep reinforcement learning network, we accomplish the network performance on the next level providing with extra robustness and speed to our architecture.

There are many works that have been done prior to ours and many of them use deep CNN to handle image restoration related problems. Kim et al. [10] developed a 20-lay CNN network to solve multi-scale image super-resolution tasks. Zhang et al. [19] used a 20-layer deep CNN to deal with multiple restoration tasks by multi-threading. In addition, one of them innovatively used Deep Q-learning Network [1] that successfully reduced the complexity of deep networks in prior works to a small scale with multiple shallower CNNs.

Our works were developed on the top of [1] and introduced three improvements: Double Q-learning [3], Prioritized Experience Replay [2] and Dueling Q-learning [5].

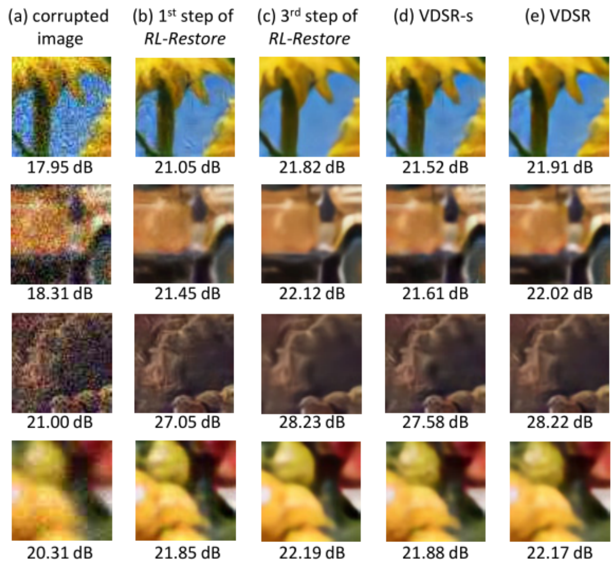

Figure 1. (a) images corrupted by various distortions. (b-c) Each step a specific tool is selected by the agent to improve the image quality. (d-e) CNN-based results.

3. Approach: algorithm development/theoretic results

0. To restore a clear image I_1 from a given distorted image I_2 to the ground truth image I_3. The toolchain shown in the figure 3 is introduced. At each step t, the agent observes the current state S_t , the current restored image I_2 and the output of the agent at the previous step v_t. Note that I_1 represents the input image and v_1 is a zero vector. Based on the maximum value of the agent's output v_t , an action a_t is selected and the corresponding tool is used to restore the current image. Steps of restoration iteratively replays until the stopping action is selected.

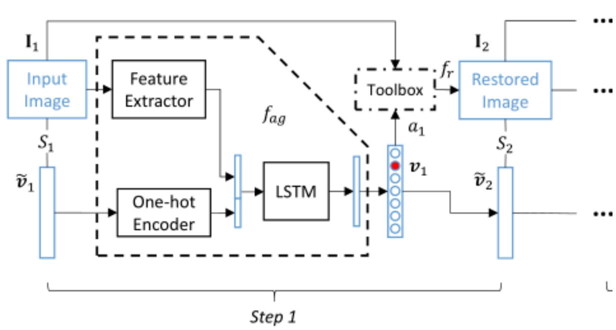

Figure 3, a single iteration of toolchain selection to restore one kind of distortion.

1. Double Deep Q-learning Network

In the Deep Q-learning Network, we use double q network structure, q-prediction network and q-target network. The q-prediction network is updated by the q-target network each step. In another word, the q-target network is frozen in time. Since the data structure that have been stored in the memory is (s,a,r,s') and the q-target is calculated base on the reward R and maximum output Q(s',a') that obtained by inputting s' to our q-target network. During this process, the selection of action a' using s' and predict Q(s',a') are using the same Q value. As a result, overestimation is expected to occur in this case. To avoid the overestimation, Double DQN update the action a' using Q value from another architecture and this could balance the overestimation brought by using the same Q value.

2. Prioritized Experience Replay

Beside the q-target and q-prediction network we mentioned above in our approach one, memory replay also plays an important role in DQN. It solves the relativity and dynamic distribution problem during reinforcement learning process. It stores the transaction sample (st,at,rt,st+1) to memory replay storage in minibatch. For small reward causing slow training speed, stochastic replay usually can not generate an efficient transition in humongous errors trials. To deal with this problem, we sample batch based on their priorities calculated using TD-error. We calculate the priority as

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{1}$$

To efficiently sample from distribution (1), the complexity cannot depend on N. We save and sample the memory through sum-tree [2] data structure as shown in Figure 4.

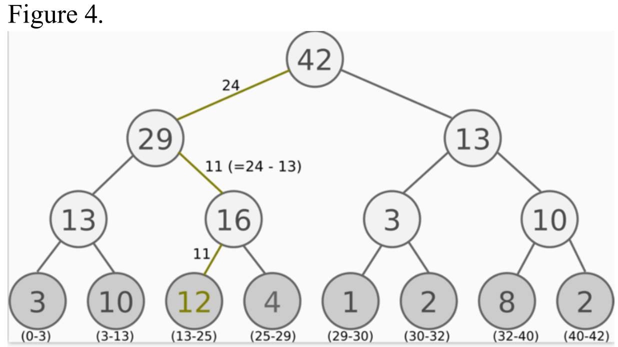

Figure 4, a single example of SumTree structure where every node is the sum of its children.

3. Dueling Deep Q-learning Network

In DQN, the output of network is the q-value of each actions. On the other hand, Dueling DQN outputs state value combined with the advantage of each action in this state to generate the q-value.

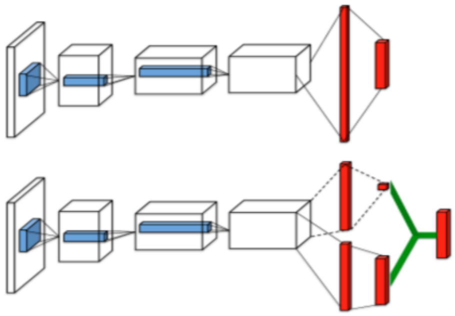

Figure 5, demonstration of dueling DQN

4. Experiment results

Datasets and Evaluation Metrics. We perform experiments on the DIV2K dataset [20], which is the most recent large-scale and high-quality dataset for image restoration. The 800 DIV2K training images are divided into two parts: 1) the first 750 images for training and 2) the rest 50 images for testing. The DIV2K validation images are used for validation. Training images are augmented by down-scaling with factors of 2, 3 and 4. The images are then cropped into 63×63 sub-images, forming our training set and testing set with 249,344 and 3,584 sub-images, respectively. We trained it on one Nvidia Tesla P100 GPU.

As for other detailed training implementation, we inherent the work from [1] but run on an improved algorithm.

Reward Function. The reward drives the training of the agent as it learns to maximize the cumulative reward. The agent is sup- posed to learn a good policy so that the final restored image is satisfactory. We wish to ensure that the image quality is enhanced at each step, therefore a stepwise reward is de- signed as follows:

$$r_t = P_{t+1} - P_t, \qquad (2)$$

where rt is the reward function at step t, P_{t+1} denotes the PSNR between $I_{t+1}$and the reference image I_{gt} at the end of the t-th step restoration, and Pt represents the input PSNR at step t. The cumulative reward is the overall PSNR gain during the restoration procedure, and it is maximized to achieve optimal enhancement.

Besides the proposed stepwise reward based on PSNR, we also investigate other reward functions: 1) stepwise SSIM [4] where the reward is the SSIM gain at each step; 2) final PSNR where the reward is the final PSNR gain given at the last step; 3) final MSE as in [21] where the reward is the negative MSE in the end

Results. As shown in the Figure 6, the reward sum increased significantly over iterations which proves that our training was successful. The test results of image restoration are shown in Figure 9

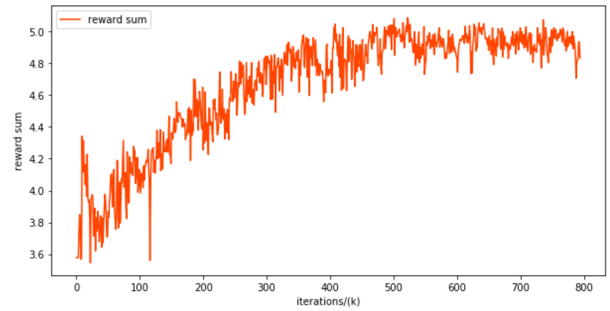

Figure 6, sum reward over iterations.

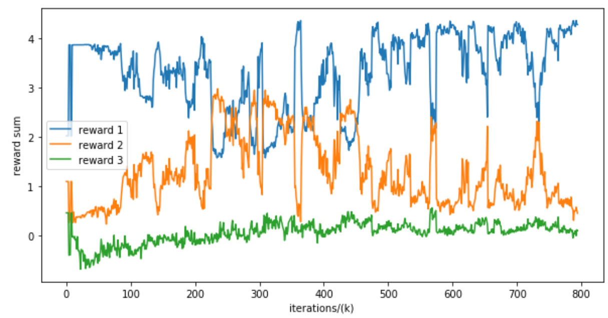

Figure 7, three types of rewards over iterations.

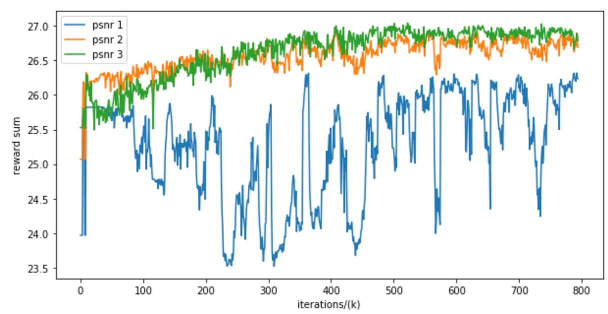

Figure 8, three types of PSNRs over iterations.

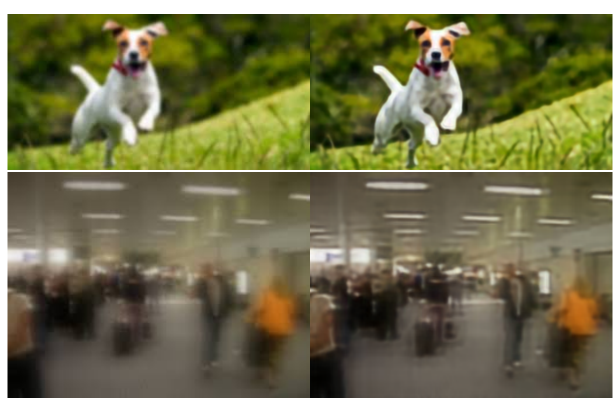

Figure 9, examples of image restoration.

5. Conclusion

We have successfully improved the performance of DQN in image restore using Deep Q-learning Network. As mentioned in abstract three improvements, double DQN, dueling deep Q-network, and prioritized experience replay, have been successfully implemented.

6. References

[1] K. Yu, C. Dong, L. Lin, C. Change Loy. Crafting a Toolchain for Image Restoration by Deep Reinforcement Learning. In CVPR, 2018.
[2] T. Schaul, J. Quan, I. Antonoglou and D. Silver. Prioritized Experience Replay. In ICLR, 2016.
[3] Hado van Hasselt, Arthur Guez, David Silver. Deep Reinforcement Learning with Double Q-learning. In AAAI, 2015.10.
[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.
[5] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt , Marc Lancto, Nando De Freitas. Dueling network architectures for deep reinforcement learning. In ICML, 2016.
[6] Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. In CVPR, 2015.
[7] C. Dong, Y. Deng, C. C. Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In ICCV, 2015.
[8] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. TPAMI, 38(2):295–307, 2016.
[9] T.-W. Hui, C. C. Loy, and X. Tang. Depth map super-resolution by deep multi-scale guidance. In ECCV, 2016.
[10] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. InCVPR, 2016.
[11] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In CVPR, 2016.
[12] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super resolution. In CVPR, 2017.
[13] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In CVPR, 2017.
[14] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In CVPR, 2017.
[15] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In CVPR, 2015.
[16] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In CVPR, 2017.
[17] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In ICCV, 2017.
[18] L. Xu, X. Tao, and J. Jia. Inverse kernels for fast spatial deconvolution. In ECCV, 2014.9
[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. TIP, 2017.
[20] E. Agustssonand, R. Timofte.Ntire2017. Challenge on single image super-resolution: Dataset and study. In *CVPR Work- shop*, 2017.
[21] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li. Attention-aware face hallucination via deep reinforcement learning. In *CVPR*, 2017.

7. Contributions

1. Zhi Ji
zj2242@columbia.edu
Columbia University, Electrical Engineering Department

Programmed and tested on double q-learning network, sum tree data structure, prioritized replay and dueling q network.
Trained on different networks, tuned hyper parameters, adjusted algorithm and data structure accordingly and get the final results.

2. Boyuan Sun
bs3113@columbia.edu
Columbia University, Electrical Engineering Department

Built training environments with GPU Tesla P100, CUDA and cuDNN.
Found the dataset and generated dataset into h5 file as the format of training set in MATLAB.