Neural Network Transfer extended with Multiple Style Images

Xi Du

Australian National University, Australia
u6559090@anu.edu.au

Abstract. [2]

Keywords: Neural network, Deep learning, Small data, Interpretability, Network pruning, Network reduction.

1 Introduction

2 Literature Review

3 Method

3.1 Preliminary

3.2 Implementation of Neural Network

Since we are not training the neural network, the model is just a sequence of functions on arrays.

Testing Phase

3.3 Data Set

Format and Preparation

(Not Really) Time-Series

Rationale

3.4 Hyperparameters and Experiments

For the vanilla style-transfer algorithm, there were a few hyper-parameters that affected the result.

- Number of iterations in the optimization. We did not have to expriment with different numbers of iterations though. It was more sensible to run the algorithm for more-than-enough iterations and store the intermediate results.
- Weights of style loss k_S and content loss k_C in the optimisation target. Although these appeared to be two parameters, actually only their ratio k_S/k_C mattered. We simply fixed $k_C = 1$ and varied k_S.
- Choice of initial values. Some sensible choices were:
  - White noise.
  - White.
  - Black.
  - The model mean.
  - The content image.
  - The style image.
  - The average of the content image and the style image.

Learning Rate

Hidden Layer Size

3.5 Evaluating Prediction

Binarisation and Choice of Loss

Amount of Network Reduction

Mechanism

Execution The program was developed under and are compatible with:

- Python 3.6
- PyTorch 1.0.1.post2
- CentOS 7 x86_64

To run the code, execute shell commands like

```
python36 0.py 5000
```

where 5000 is the number of desired training cycles to reach. The different hyperparameters will be automatically covered. The program automatically picks up stored models. To start fresh, clear the stored models and outputs by

```
rm out/*/*.*
```

but do not remove the directories.

4 Results and Discussion

5 Conclusion and Future Work

References

1. Gatys, L., Ecker, A., & Bethge, M. (2016). A neural algorithm of artistic style. Journal of Vision, 16(12), 326. doi:10.1167/16.12.326
2. Gedeon, T.D. and Harris, D. (1991) "Network Reduction Techniques," Proceedings International Conference on Neural Networks Methodologies and Applications, AMSE, San Diego, vol. 1: 119-126.
3. Penington, Jocelyn & Dow, R.J.F.. (1988). Neural net pruningWhy and how. Proceedings of the IEEE Conference on Neural Networks. 1. 325 - 333 vol.1. 10.1109/ICNN.1988.23864.
4. Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning.
5. Han, S., Mao, H., & Dally, W.J. (2016). Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. CoRR, abs/1510.00149.
6. Sun, X., Ren, X., Ma, S., Wei, B., Li, W., & Wang, H. (2018). Training Simplification and Model Simplification for Deep Learning: A Minimal Effort Back Propagation Method. CoRR, abs/1711.06528.
7. Cheng, Yu & Wang, Duo & Zhou, Pan & Zhang, Tao. (2017). A Survey of Model Compression and Acceleration for Deep Neural Networks.
8. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in PyTorch.