Evaluating Network Reduction Techniques with Eye Gaze Data

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 the Network Reduction Technique

3.3 Implementation of Neural Network

Training Phase

Testing Phase

3.4 Data Set

Format and Preparation

(Not Really) Time-Series

Rationale

3.5 Hyperparameters and Experiments

Activation Function

Learning Rate

Hidden Layer Size

3.6 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. Zhu, D., Mendis, B.S., Gedeon, T.D., Asthana, A., & Gcke, R. (2008). A Hybrid Fuzzy Approach for Human Eye Gaze Pattern Recognition. ICONIP.
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.