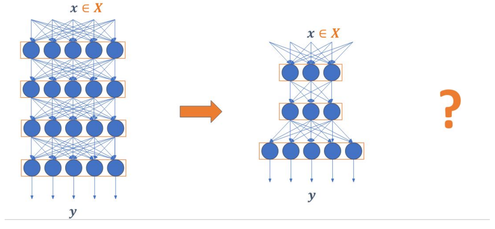
Lossless Compression of Deep Neural Networks

Thiago Serra[1], Abhinav Kumar[2], Srikumar Ramalingam[3]

[1]Bucknell University; [2]Michigan State University; [3]The University of Utah

1. Problem Statement

Can we find a smaller neural network that models the same function x→ y for the inputs that are relevant for a given application?

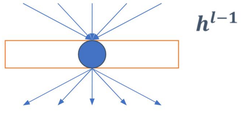


For networks trained on MNIST, we only need equivalence for $x \in [0,1]^{784}$

Josef Steppan - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/wiki/File:MnistExamples.png

2. Scope

We study feedforward rectifier networks, where the activation functions of all the artificial neurons are Rectified Linear Units (ReLUs)



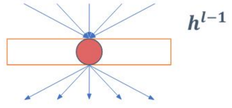
$$h_i^l = max\{0, W_i^l h^{l-1} + b_i^l\}$$

Inactive Active

If some units are always inactive or always active for any valid input, we might be able to compress the neural network without incurring any losses

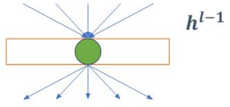
3. Stability of Units

Stably inactive units never yield positive outputs



$$h_i^l = 0$$

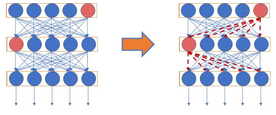
Stably active units always yield positive outputs



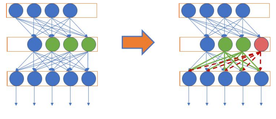
$$h_i^l = W_i^l h^{l-1} + b_i^l > 0$$

4. Lossless Compression Operations

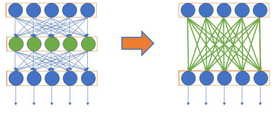
The output of stably inactive units can be ignored and those units can be safely removed



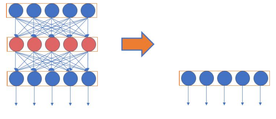
If a set of stably active units S has rank(W_S^l) < |S|, the output of some units can be defined using the output of others and those units can be removed by adjusting the parameters accordingly



A stably active layer can be folded by adjusting the weights to directly connect the preceding and succeeding layers of the network



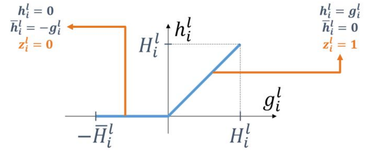
A stably inactive layer implies that we are modeling a constant function and the neural network can be collapsed to a single layer



5. Identifying Stable Units

We can model the inputs and output of each unit of the neural network by using a Mixed-Integer Linear Programming (MILP) formulation:

$$\begin{aligned} W_i^l h^{l-1} + b_i^l &= g_i^l \\ g_i^l &= h_i^l - \bar{h}_i^l \\ h_i^l &\geq 0 \\ \bar{h}_i^l &\geq 0 \\ z_i^l &\in \{0,1\} \\ h_i^l &\leq H_i^l z_i^l \\ \bar{h}_i^l &\leq \bar{H}_i^l (1 - z_i^l) \end{aligned}$$



A unit is stably active if min{ g_i^l | (input/output mappings of previous units), $x \in X$ } > 0

A unit is stably inactive if max{ g_i^l | (input/output mappings of previous units), $x \in X$ } < 0

5. Experimental Results

Experimental Setup:

- We trained 2-hidden-layer rectifier networks with widths 25, 50, and 100 on the MNIST benchmark
- To induce stability, we applied L1 regularization to the weights of the units
- For each width, we identified an amount of regularization L that improves accuracy in preliminary experiments
- To induce more stability, we also trained neural networks with more regularization ($5L$)
- For each width, we report result on training 31 networks using L1 weights 0, L, and $5L$

Computational Results:

- Neural networks are more stable and compressible if trained with more weight regularization
- The compression obtained in the experiments is due to the removal of stably inactive units
- **For the amount of regularization that improves accuracy, we also observe compressibility**

Layer Width	L1 Weight	Accuracy (%)	Compression (%)
25	0.001	95.76 ± 0.05	22 ± 1
25	0.0002	97.24 ± 0.02	8.3 ± 0.7
25	0	96.68 ± 0.03	0 ± 0
50	0.001	96.05 ± 0.04	29.4 ± 0.7
50	0.0002	97.81 ± 0.02	15.1 ± 0.6
50	0	97.62 ± 0.02	0 ± 0
100	0.0005	97.14 ± 0.02	30.8 ± 0.5
100	0.0001	98.14 ± 0.01	14.9 ± 0.4
100	0	98.00 ± 0.01	0 ± 0

6. Conclusion and Future Work

- To the best of our knowledge, we introduced the first method for exact compression of rectifier networks
- We may interpret L1 regularization as inducing a subnetwork to represent the function associated with the neural network
- Future work may explore the connection between these subnetworks identified by compression and lottery tickets, bounding techniques to identify stable units faster, and other forms of inducing posterior compressibility while training

Citing This Work:

Serra, T., Kumar, A., Ramalingam, S.: Lossless Compression of Deep Neural Networks. In *Proceedings of the 17th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research* (CPAIOR 2020), LNCS (to appear).

Preprint Link:

https://arxiv.org/abs/2001.00218