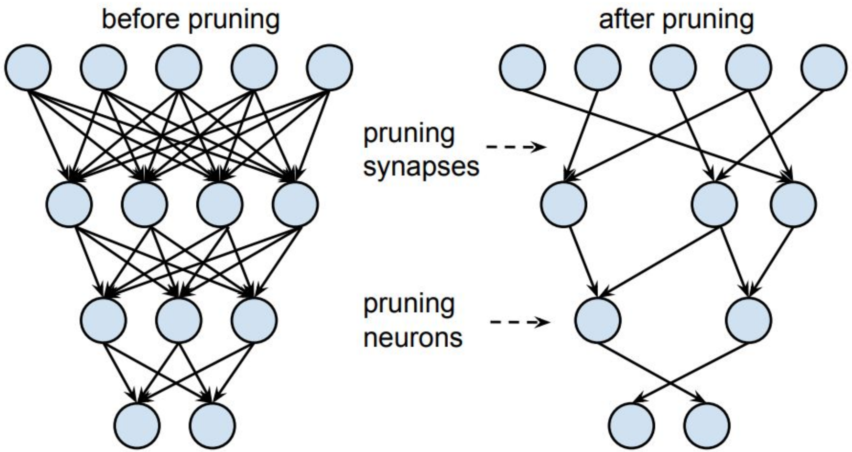
The Lottery Ticket Hypothesis: Training Pruned Neural Networks

Jonathan Frankle, Michael Carbin

MACHINE
INTELLIGENCE
COMMUNITY

Devin de Hueck
Mar. 29, 2018

Pruning Neural Networks



[1]

The Hypothesis

"Training succeeds for a given network if one of its subnetworks (a "winning ticket") has been randomly initialized such that it can be trained in isolation to high accuracy in at most the number of iterations necessary to train the original network."

Research Questions

“How effectively do winning tickets train in comparison to the original network and to randomly sampled networks of similar size?”

“How big are winning tickets relative to the size of the original network?”

“How sensitive are our results to particular pruning strategies?”

Extracting a “Winning Ticket”

1. Randomly Initialize a neural network
2. Train the network until convergence
3. Prune
4. Reset weights of the remaining portion to original initialization

Learning the XOR function - Results

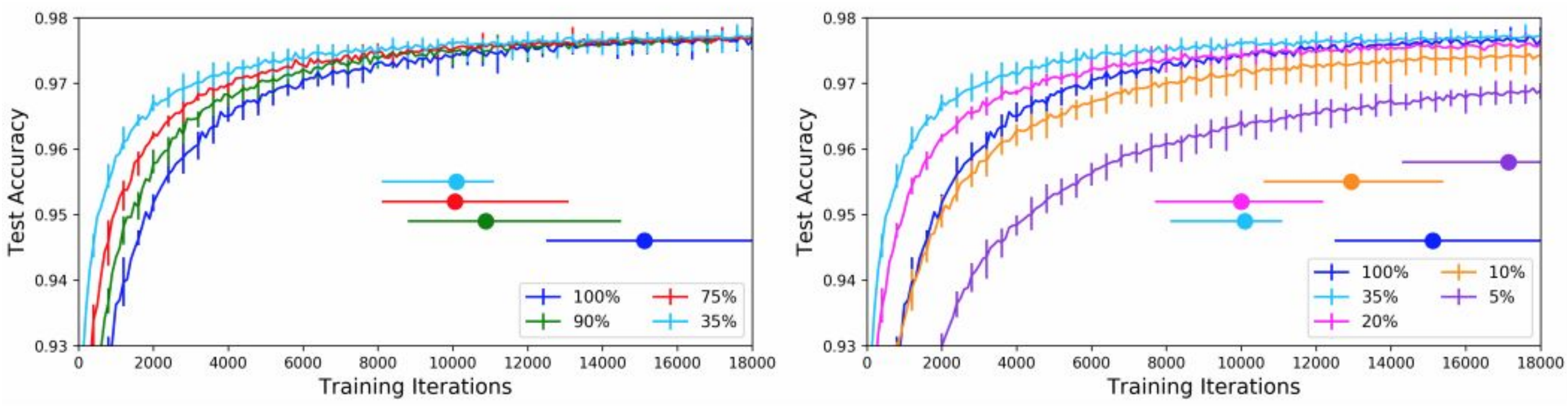
10 Units		8 Units		6 Units		4 Units		2 Units	
DB	ZL	DB	ZL	DB	ZL	DB	ZL	DB	ZL
98.5	92.9	96.8	87.5	92.5	76.8	78.3	55.3	49.1	17.6

***DB**: Decision Boundary*
***ZL**: Zero loss*

Pruning Strategy	10 Units		4 Units (Pruned)		2 Units (Pruned)	
	DB	ZL	DB	ZL	DB	ZL
One-shot Product	99.2	93.3	98.0	90.3	82.4	65.3
Input Magnitude	98.9	93.5	97.9	92.2	83.8	76.5
Output Magnitude	99.0	93.3	96.9	85.9	78.6	56.1
Product	98.5	92.9	97.6	90.3	91.5	79.4

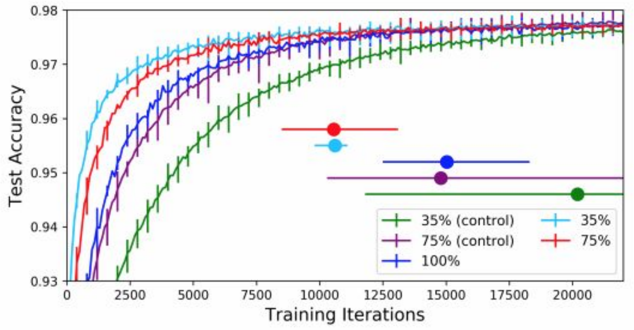
Iterative Pruning

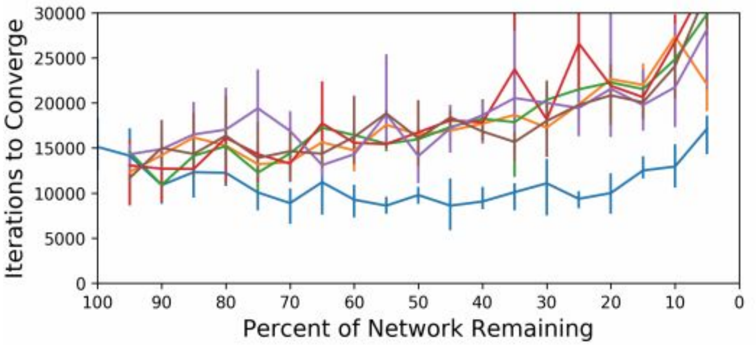
MNIST - One Shot Pruning

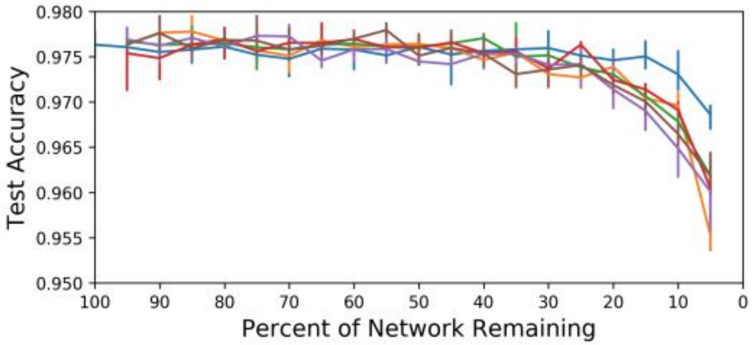


Each curve represents the average performance of five trials in which networks were pruned to the stated size

MNIST - One Shot Pruning - Control Experiment 1

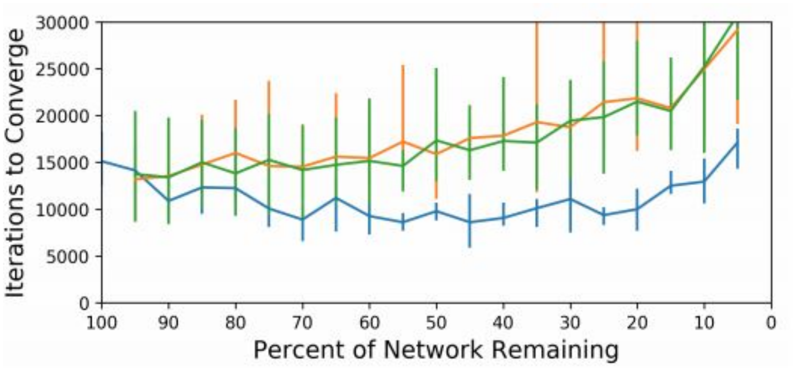


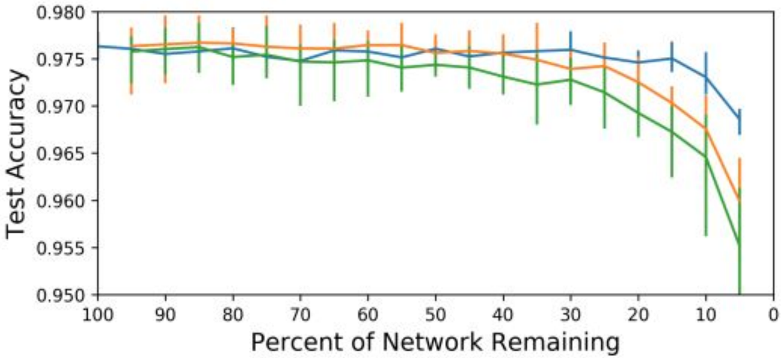




Blue curve: *average of 5 winning tickets*
Multi-colored curves: *the controls for each of the 5 trials*

MNIST - One Shot Pruning - Control Experiment 2



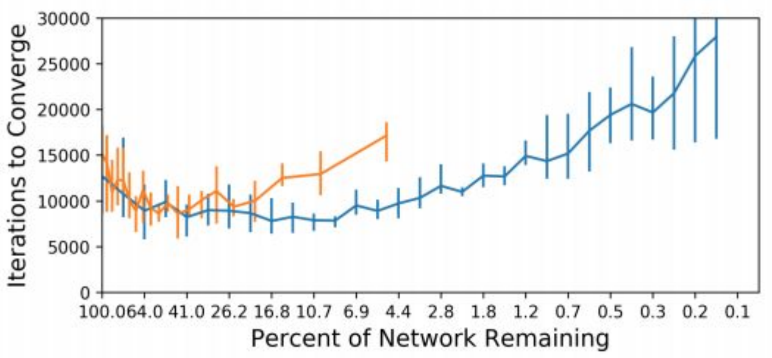


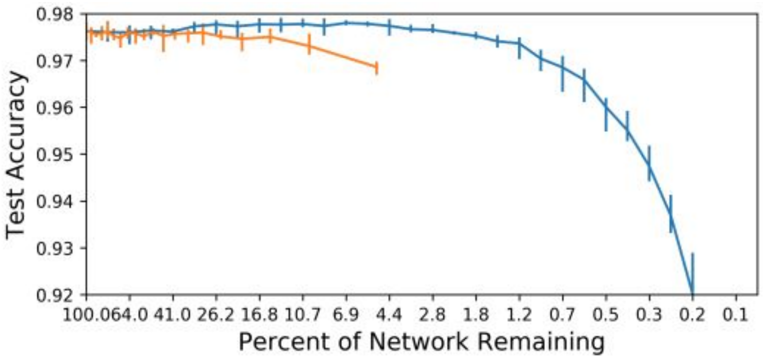
Blue Curve: *average of 5 winning tickets*
Orange Curve: *results from control experiment 1*
Green Curve: *results from control experiment 2*

MNIST - Iterative Pruning

1. Randomly Initialize the network
2. Train the network until convergence
3. Prune 20% from each hidden layer
4. Reset weights to original initialization
5. Repeat steps 2 through 4 until network is pruned to desired size

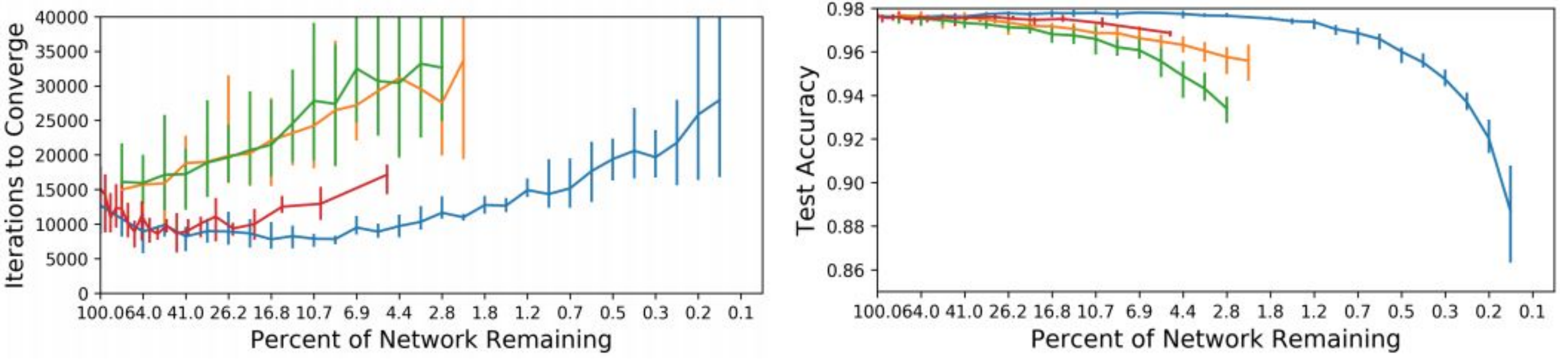
MNIST - Iterative Pruning - Results





Orange Curve: *One-shot pruning results*
Blue Curve: *Iterative pruning results*

MNIST - Iterative Pruning - Control Experiments

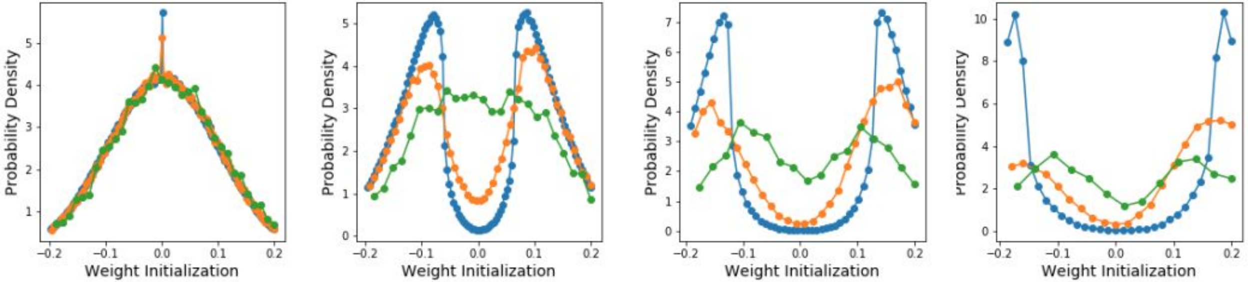


Blue Curve: *average of five winning tickets pruned iteratively*
Red Curve: *one-shot pruning*
Orange Curve: *results of control experiment 1*
Green Curve: *results of control experiment 2*

Examining Winning Tickets - Initializations

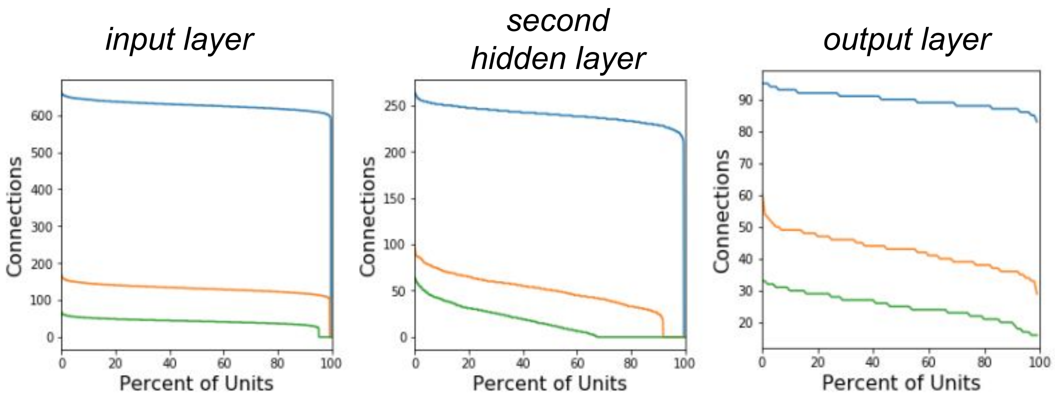
% of network remaining:

100% *51.2%* *16.8%* *5.5%*



Blue curve: *first hidden layer*
Orange curve: *second hidden layer*
Green curve: *output layer*

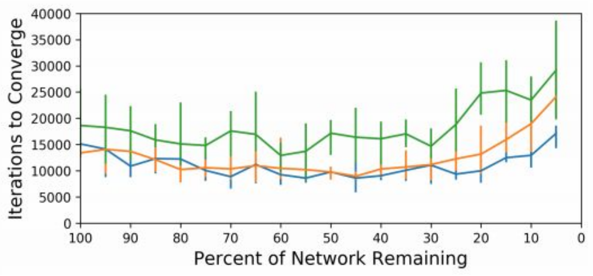
Examining Winning Tickets - Architecture

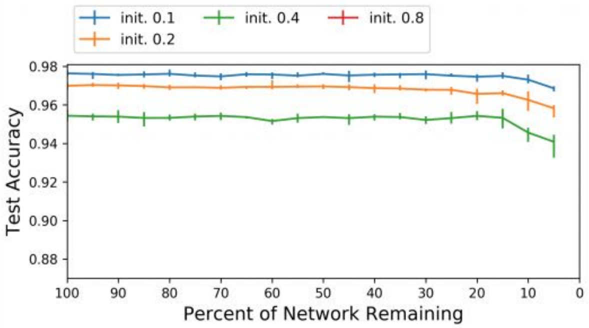


Blue Curve: *iteratively pruned to 80%*
Orange Curve: *iteratively pruned to: 16.8%*
Green Curve: *iteratively pruned to 5.5%%*

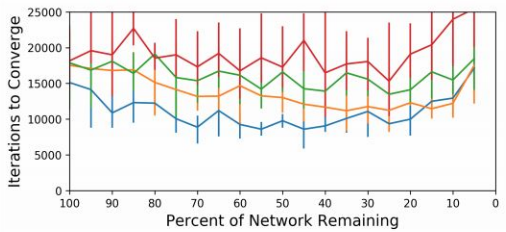
Exploring MNIST Parameters - Initialization

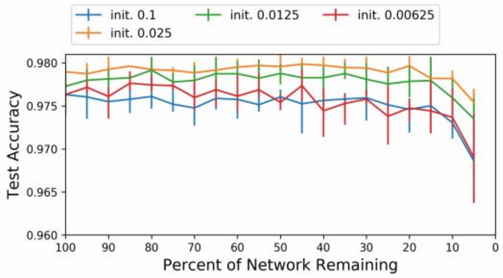
STD >= 0.1





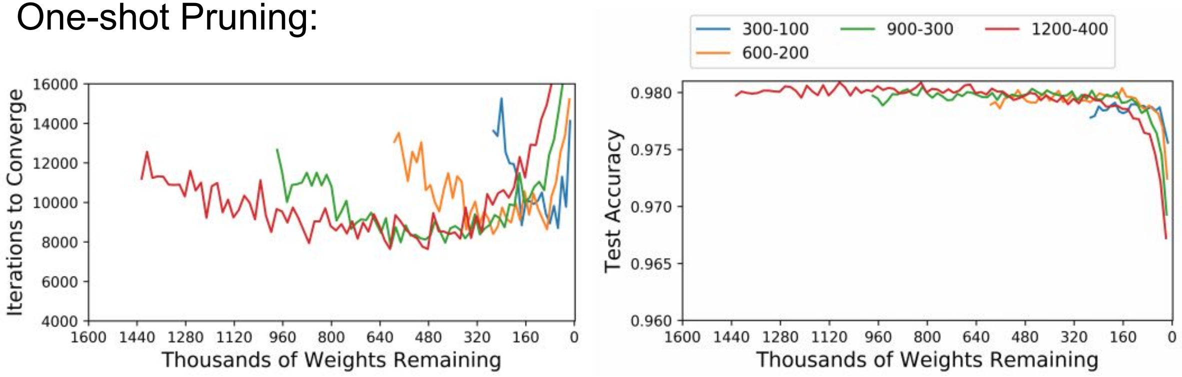
STD <= 0.1



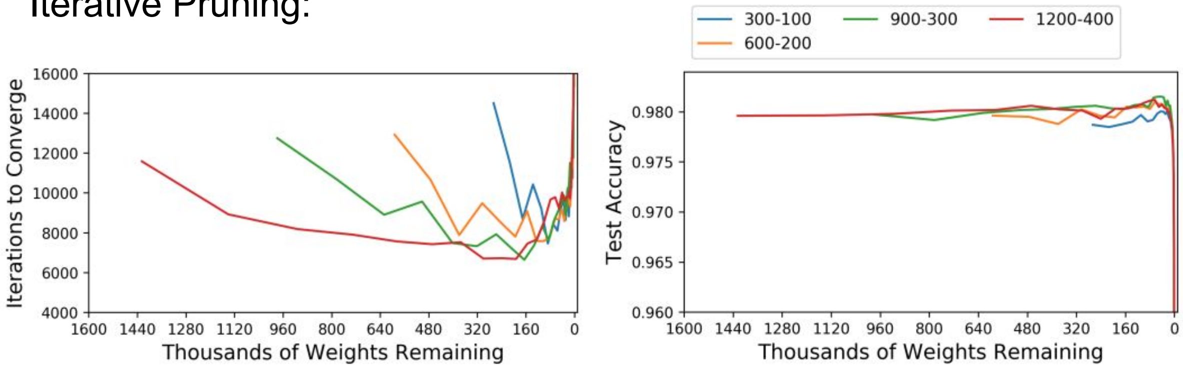


Exploring MNIST Parameters - Network Size

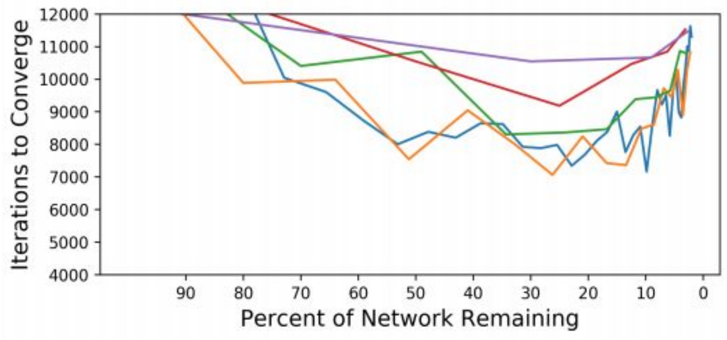
One-shot Pruning:

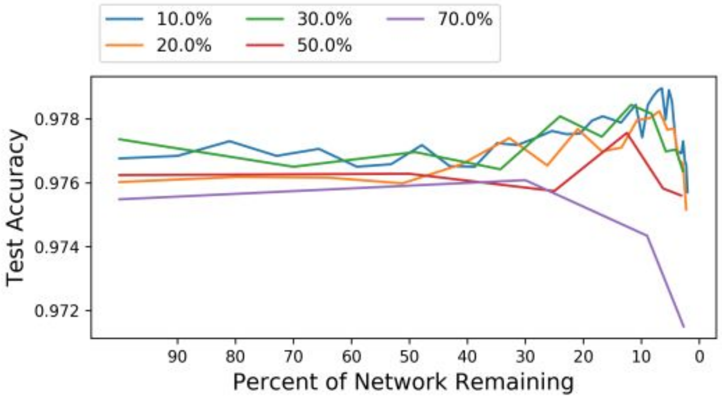


Iterative Pruning:

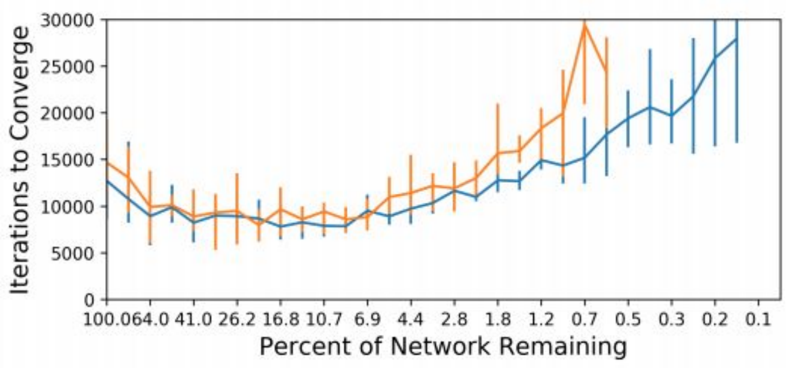


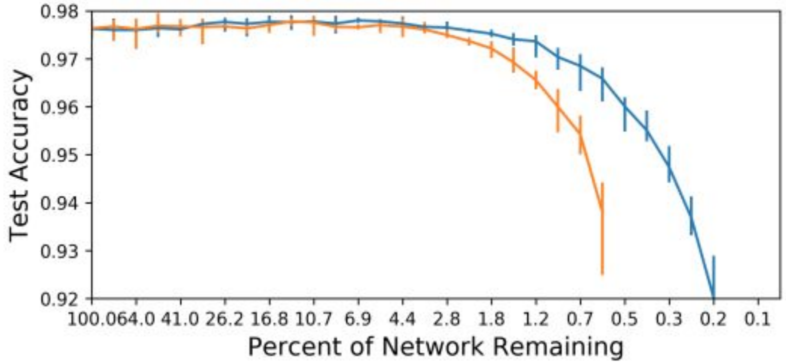
Exploring MNIST Parameters - Iterative Pruning Rates





Exploring MNIST Parameters - Weight Resetting





Blue: *This papers strategy*
Orange: *Previous research strategy*

Limitations

- Small Examples
- No Theoretical Analysis
- No Useful Strategies Devised

M I C

Conclusion and What's Next

- XOR
- MNIST
- Future Research Directions
    - Larger Examples
    - Understanding Winning Tickets
    - New Strategies

References and Further Reading

1. Han, Song, et al. **"Learning both weights and connections for efficient neural networks."** Advances in neural information processing systems. 2015.
2. Srivastava, Nitish, et al. **"Dropout: A simple way to prevent neural networks from overfitting."** The Journal of Machine Learning Research 15.1. 2014.
3. Han, Song, et al. **"Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding."** arXiv preprint arXiv:1510.00149. 2015.
4. Hinton, Geoffrey E., et al. **"Improving neural networks by preventing co-adaptation of feature detectors."** arXiv preprint arXiv:1207.0580. 2012.
5. Lin, Henry W., et al. **“Why does deep and cheap learning work so well?”** arXiv preprint arXiv:1608.08225. 2017.