An Empirical Study of the Lottery Ticket Hypothesis: Replication and Analysis

Soline Mignot and Leena Boyina

October 2025

1 Introduction

1.1 General Topic

Training deep neural networks is notoriously resource-intensive. However, numerous studies have shown that final trained models are often highly over-parameterized and can be drastically "pruned" post-training without a loss in performance [3].

The Lottery Ticket Hypothesis, introduced by Frankle and Carbin [1], states a counter-intuitive idea: within a dense, randomly-initialized network, there exists a subnetwork (a "winning ticket") that, when trained in isolation from scratch using its original initial weights, can match the test accuracy of the original dense network in a similar number of iterations [1, 2].

Identifying such subnetworks could revolutionize network training, enabling the development of smaller, more efficient models from the outset. Recent work has even sought to theoretically prove the existence of these tickets [2].

1.2 Project Overview

Our project aims to first empirically replicate the foundational results of the original paper [1]. Following this validation, we will extend the analysis to answer a more pragmatic question: how do these "winning tickets" compare to networks pruned using standard fine-tuning techniques?

1.3 Core References

Our work will be based on the following key resources:

1. **The Original Hypothesis:** Frankle, J., & Carbin, M. (2019). *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks* [1].

2. **Theoretical Proof:** Malach, E., Yehudai, G., Shalev-Shwartz, S., & Shamir, O. (2020). *Proving the Lottery Ticket Hypothesis: Pruning is All You Need* [2].

3. **Pruning techniques:** Hongrong Cheng, Miao Zhang & Javen Qinfeng Shi (2024). *A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations* [4].

2 Project Plan

Our project is divided into two distinct phases: replication and extension.

2.1 Phase 1: Replication of Core Findings

The first objective is to validate our implementation by reproducing the key experiments from Frankle & Carbin (2019).

2.1.1 Identifying Winning Tickets

- **Experiment:** Implement the "iterative magnitude pruning" algorithm to find winning tickets. We will train dense networks, prune them, and reset the surviving weights to their original initial values (θ_0).
- **Measurement:** Test these tickets at various sparsity levels (e.g., 50%, 20%, 10%, 5% of remaining weights).
- **Expected Outcome:** These winning tickets should train effectively, matching or exceeding the original network's accuracy, often while learning faster.

2.1.2 The Importance of Initialization

- **Experiment:** Take the structure (mask m) of the winning tickets found above, but re-initialize their weights with new random values (θ'_0).
- **Measurement:** Compare the convergence speed and final accuracy of these "randomly re-initialized" tickets against the true winning tickets.
- **Expected Outcome:** Randomly re-initialized tickets are expected to train less effectively, confirming that the original initial weights (θ_0) are critical.

2.2 Phase 2: Project Extension (Original Analysis)

The second objective is to compare the implementation with the performances of the other pruning techniques. Using the last core paper [4], we will try pruning techniques of different types : unstructured, structured, and semi-structured pruning, and pruning before/during/after training.

2.2.1 Comparison: Standard Pruning (Fine-Tuning) vs. Winning Tickets

- **Hypothesis:** Resetting to θ_0 (winning ticket) versus keeping trained weights θ_j (standard pruning).
- **Experiment:** Train several models with the same mask m:
  1. Winning Ticket: Reset weights to θ_0 and train from scratch.
  2. Pruned Models: Keep trained weights θ_j and continue training (fine-tuning). Store the different performances for each model.
- **Measurement:** Compare final accuracy, convergence speed, and model size at identical sparsity levels.

3 Methodology

3.1 Datasets

- **MNIST / Fashion-MNIST:** Ideal for rapid testing of Multi-Layer Perceptron (MLP) architectures. Training takes only a few minutes on a basic CPU/GPU. [5]
- **CIFAR-10:** Used to validate findings on more complex CNN architectures. Requires a GPU for efficient training.

Both were used in the initial paper.

3.2 Implementation Framework

We will implement this project using **PyTorch**, which provides flexibility for weight manipulation and application of pruning masks.

3.2.1 Key Implementation Steps

1. Train a dense reference network (e.g., LeNet for MNIST, simple CNN for CIFAR-10) and save its initial weights (θ_0).
2. After full training, save the final weights (θ_j).
3. Apply unstructured, magnitude-based pruning to θ_j to create the binary mask m.
4. Train and evaluate models for comparison:
   - Winning Ticket (Replication): Load θ_0, apply mask m, and train.
   - Random Ticket (Replication): Generate new random weights θ'_0, apply mask m, and train.
   - Other standard pruning methods (Extension): Load θ_j, apply mask m, and continue training (fine-tune).
5. Visualize and compare results using `matplotlib`: learning curves, duration of the training, final accuracy vs. sparsity, etc.

References

[1] J. Frankle and M. Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[2] E. Malach, G. Yehudai, S. Shalev-Shwartz, and O. Shamir. Proving the Lottery Ticket Hypothesis: Pruning is All You Need. *arXiv preprint arXiv:2002.00585*, 2020.

[3] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag. What is the State of Neural Network Pruning? *arXiv preprint arXiv:2003.03033*, 2020.

[4] Hongrong Cheng, Miao Zhang & Javen Qinfeng Shi. A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations. *arXiv preprint arXiv:2308.06767*, 2024.

[5] LeCun, Yann and Cortes, Corinna and Burges, CJ. MNIST handwritten digit database, vol. 2 *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010.