

INLP Project Timeline & Implementation Plan

Team 30: Natural Stupidity

Prakhar Singhal (2022111025)

Mohak Somani (2022101088)

Prakhar Jain (2022115006)

February 17, 2025

1 Synopsis

The paper we are going to implement is *Titans: Learning to Memorize at Test Time*, which introduces a new family of deep learning architectures called **Titans**, integrating long-term memory mechanisms to enhance sequence modeling beyond the constraints of Transformers. While Transformers rely on attention mechanisms that model dependencies within a fixed-length context, Titans introduce a **neural memory module** capable of dynamically storing and retrieving long-past information, enabling efficient processing of extremely long sequences.

1.1 Key Ideas and Objectives

- **Memory Perspective:** Transformers act as short-term memory by attending to a fixed-length window, whereas Titans incorporate a deep neural long-term memory module for persistent context retention.
- **Neural Memory Module:** A novel memory mechanism that learns to store, update, and retrieve information at test time, enhancing the model's ability to recall distant past events.
- **Titan Architectures:** Three variants of Titans are proposed:
 - **Memory as Context (MAC)** – memory is appended to the current context for attention.
 - **Memory as Gating (MAG)** – memory and attention are combined using a gating mechanism.
 - **Memory as a Layer (MAL)** – memory is treated as an independent processing layer.
- **Efficient Training:** The architecture is designed for **parallelizable and scalable** training, making it computationally efficient for handling over 2 million tokens in a context window.

1.2 Relevance to Our Project

This paper is highly relevant to our project as it addresses the limitations of traditional sequence models in long-context tasks. The ability to incorporate long-term memory is crucial for applications such as:

- **Language Modeling:** Improving perplexity and contextual understanding in NLP tasks.
- **Commonsense Reasoning:** Enhancing logical reasoning in AI systems.
- **Long-Context Retrieval:** Efficiently recalling past information in extensive datasets.
- **Time-Series Forecasting and Genomics:** Processing long sequential data efficiently.

The experimental results show that Titans outperform both Transformers and state-of-the-art recurrent models across multiple benchmarks, making it a promising architecture for our implementation.

2 Project Timeline

2.1 Task Distribution

- **Titans Implementation (Basic Execution & Verification)**
- **Baseline Execution (Proper Analysis)**
- **Further Development Tasks:**
 - Modular Memory Implementation
 - Baseline Re-execution and Analysis
 - Latent Space Reasoning Implementation
 - Final Baseline Execution & Analysis
 - Presentation Preparation

2.2 Weekly/Monthly Plan

Week	Task
Week 1-2	Literature review: Understanding the research paper and related works. Setting up the development environment. Implementing a simple baseline model (e.g., Transformer or Linear RNN) for comparison.
Week 3-4	Implementing the core components of the Titans model, starting with the neural long-term memory module. Integrating the memory module into different Titans variants (MAC, MAG, MAL) and debugging.
Interim Submission	7 March(tentative)
Week 5-6	Running initial experiments, tuning hyperparameters, and comparing results with baseline models.
Week 7	Performance evaluation: Conducting tests on long-context tasks (e.g., language modeling, reasoning).
Week 8-9	Extending the Titans architecture: Exploring improvements such as integration of conceptual long-term memory and implementing latent space reasoning to further its applications.
Week 10	Final refinements, improving efficiency, and optimizing memory management. Finalizing results, documentation, and preparing the project report.

3 Plan of Action

Our approach to implementing the research paper will consist of the following steps:

1. **Understanding the Paper:** Analyze the paper in detail, focusing on the architecture and memory module.
2. **Setting Up Environment:** Install necessary dependencies (e.g., PyTorch, TensorFlow, CUDA, and required libraries).
3. **Baseline Implementation:** Implement a simple Transformer or RNN-based model for comparison.
4. **Core Model Implementation:** Develop the neural long-term memory module.
5. **Integrating Memory into Titans:** Implement different Titans variants (MAC, MAG, MAL) and test their functionality.
6. **Evaluation and Optimization:** Run tests, analyze results, and optimize model performance.
7. **Extending the Model:** Identify potential areas for improvement, currently considering integration of conceptual long term memory and implementing latent space reasoning to enhance its use case (subject to change as we progress).
8. **Implementation of Extensions:** Integrate enhancements and evaluate their impact on performance.

9. **Final Documentation:** Prepare a comprehensive report on findings, methodologies, and results.