Introduction to State Space Models

UG BTech - 2nd Year

Vinay , Vaibhav Mahore , Shubhadeep Sing , Snehal Biswas

Indian Institute of Science, Bangalore

March 16, 2025

Outline

The Need for Sequential Data Modeling

- **What is Sequential Data?**
  - **Ans :** It's the data that comes in a specific order, where the arrangement of the data points matters.
- **Example(NLP):** The dog bites the man vs The man bites the dog.
- **Need for Sequential Data Modeling:** It's crucial because many datasets have an inherent order e.g Language, Time series in which the sequence and context between data points are essential for accurate analysis and predictions.
- **Challenge:** Traditional models(e.g. Feedforward Networks) treat inputs independently and fail to capture such temporal dependencies.

Recurrent Neural Networks(RNNs)

- **This is an introduction to the topic**
  - Process inputs in isolation.
  - Suitable for static pattern recognition.
- **Recurrent Neural Networks (RNNs):**
  - Incorporate loops to retain information across time steps.
  - Use hidden states to capture temporal dependencies.
- **Key Difference:** RNNs are designed to handle sequential data, while FNNs lack an inherent mechanism for managing context over time.

From RNN to LSTM

- **Limitations of Vanilla RNNs:**
  - Suffer from vanishing and exploding gradients during training.
  - Struggle with capturing long-range dependencies.
- **Introduction of LSTMs:**
  - LSTMs introduce gating mechanisms (input, forget, output) to better manage memory.
  - They effectively mitigate gradient issues and improve long-term dependency learning.

Limitations of LSTMs

- **Scalability and Computational Complexity:**
  - Despite improvements, LSTMs remain computationally intensive for very long sequences.
- **Long-Range Dependency Challenges:**
  - Research such as Bengio et al. (1994) and Pascanu et al. (2013) has demonstrated that LSTMs can still struggle with vanishing gradients when modeling very long sequences.
- **Sequential Processing Bottleneck:**
  - The inherent sequential nature of LSTM processing limits parallelism, resulting in slower training compared to more modern architectures.

Methodology

- Data collection and preprocessing
- Model architecture
- Training approach
- Evaluation metrics

Implementation

- Technologies used
- Key algorithms
- Technical challenges
- Solutions implemented

Results

- Model performance
- Key findings
- Comparative analysis
- Visualizations

Future Work

- Potential improvements
- Scalability considerations
- Additional features
- Research directions

Thank You

Questions?