Recurrent Neural Network on Bike-Sharing time siries analysis

Jose Mario Costa   
MSc Student Data Analytics  
CCT College DubinDublin, Ireland  
jmcloudpro@gmail.com

*Abstract*—

Traditional time series forecasting models like ARIMA remain popular due to their user-friendliness, despite the emergence of powerful Recurrent Neural Networks (RNN). This research bridges this gap by proposing a scalable RNN-based approach for daily bike-sharing trip prediction utilizing Apache Hadoop and PySpark for distributed computing.

A major challenge lies in the inherent irregularity of bike-sharing trip data, characterized by fluctuating demand patterns and missing values, which hinder the effectiveness of traditional forecasting methods.

This research addresses this challenge by presenting a study that involves data engineering techniques to transform the irregular dataset into a regular, daily format suitable for RNN analysis.

Keywords— Bike-Sharing,Prediction, Apache Hadoop, PySpark, Engineering, Scalable, Forecasting

# Introduction

The rise of bike-sharing programs in recent years has revolutionized urban transportation, offering a convenient and eco-friendly alternative to traditional modes of travel. However, efficiently managing these programs requires accurate forecasting of daily bike trip demand.

While established time series forecasting models like ARIMA have been widely used, they can struggle to capture the complexities inherent in bike-sharing data. This data is often characterized by:

Irregularity: Demand for bikes fluctuates significantly throughout the day, week, and year due to factors like weather, seasonality, and special events.

This research explores the potential of Recurrent Neural Networks (RNNs) for daily bike-sharing trip prediction. RNNs are a powerful class of machine learning models capable of learning complex patterns from sequential data. Their ability to handle temporal dependencies makes them well-suited for analyzing time series data like bike-sharing trips.

This project proposes a scalable RNN-based approach that leverages the distributed computing capabilities of Apache Hadoop and PySpark. This allows us to efficiently process large-scale bike-sharing datasets and explore different RNN architectures for accurate prediction.

## Objective:

By combining the power of RNN with scalable computing, this research aims to be a base for developing a highly accurate and efficient forecasting system to benefit bike-sharing companies by enabling them to optimize resource allocation, ensure bike availability, and ultimately, improve the user experience.

## **Research Question:**

Can Recurrent Neural Networks (RNN), utilizing Apache Hadoop and PySpark for distributed computing, outperform traditional time series forecasting models in predicting daily bike-sharing trip demand, particularly when considering the challenges of irregular data and missing values?

## State of the Art

# Neural networks for forecasting

Machine learning thrives on data, and supervised learning trains algorithms on data with both inputs and desired outputs. Neural networks, inspired by the brain's structure, are powerful supervised learning tools that excel at capturing complex relationships within data. This summary focuses on how neural networks, specifically recurrent neural networks (RNNs), are particularly well-suited for time series forecasting, which involves predicting future values based on a sequence of past data points.

## Why RNNs for Time Series?

Regular neural networks process information in a single forward pass. However, for time series data, the past has a significant influence on the future. RNNs address this by incorporating a feedback loop, allowing them to use not only the current input data but also the outputs from previous calculations. This enables RNNs to effectively "remember" past information and make predictions based on the entire sequence.

## Challenges and Solutions

While the concept of RNNs seems intuitive, training them can be challenging. Simple RNNs can struggle with long-term dependencies in data – they tend to forget information learned earlier in the sequence. To overcome this limitation, two popular and effective RNN models have emerged:

### Long Short-Term Memory (LSTM):

### LSTMs incorporate special gating mechanisms that allow them to selectively remember and forget information over long periods, making them ideal for capturing long-term dependencies in time series data, LSTMs can effectively learn and remember information over extended periods, making them highly popular for various sequential learning tasks.

One of the key advantages of LSTMs is their ability to overcome the vanishing gradient problem. This problem hinders traditional RNNs when dealing with long sequences. As the network processes information step-by-step, the gradient (a value used to adjust the network's weights during training) can become very small or large, hindering the learning process. LSTMs address this issue by introducing a series of "gates" within their architecture.

These gates act as intelligent filters, controlling the flow of information within the network. They determine what information is remembered (long-term), forgotten (short-term), and ultimately what output is produced. This selective memory allows LSTMs to learn what's important in the data sequence, focusing on relevant patterns and mitigating the vanishing gradient problem.

The specific structure of an LSTM cell, including the input gate, forget gate, and output gate, is a complex topic often visualized using diagrams. However, the key takeaway is that these gates empower LSTMs to excel at tasks involving sequential data with long-term dependencies.

Gated Recurrent Unit (GRU): Similar to LSTMs, GRUs use gating mechanisms to control information flow within the network. They are generally simpler and faster to train than LSTMs, making them a good choice for tasks where computational efficiency is a concern.

The combination of neural networks and RNN architectures allows for powerful time series forecasting in various domains, including stock market prediction, energy demand forecasting, weather forecasting, and sales and demand forecasting.