
RECURRENT NEURAL NETWORKS
Paul Speaker

Recurrent Neural Networks

- Recurrent Neural Networks are extremely important with **sequential** data
  - Sequential Data → data that has a (1-dimensional) order, with a start and a finish
- We have already seen sequential data previously
  - Time series data is sequential data
- Another type of sequential data is text
  - Order matters in text data
- Early RNN's did not work very well
  - Modeled data with many hidden layers, with each hidden layer modeling previous values
  - Problems with many layers → optimization difficulties
    - Problems with errors in optimization increase geometrically with the number of layers
    - Either errors greatly compound, or convergence is too slow

Multiscale Sequential Data

- The simplest RNN's have just a couple of layers
  - Very similar in structure to AR(n) models for small n
  - Can sometimes outperform AR(n) models
- More complicated models are **multiscale**
  - Multiscale means the modeling looks at separate levels back
- Example—text generation with ChatGPT
  - ChatGPT generates words sequentially. To generate a word, ChatGPT uses features built off of
    - The previous word
    - The previous sentence
    - The paragraph up until that point
    - The entire document
  - These are all different scales for modeling

Recurrent Neural Networks for Time Series Data

- Problem for using RNN's on time series data
  - The inputs for a RNN are sequences
  - You train on some sequences
  - You test on others
  - For a time series, we only have 1 sequence!
- Solution: create mini-series of length L which subset the time series data
- If the time series is of length n, we have n – L + 1 sequences as inputs
  - Data will have to be lagged and reshaped manually before we get to neural network

Recurrent Neural Networks for Time Series Data

- We can use keras again for the neural network
- Pre-specified number of lags from inclusion
  - Layers not sequential—single layer
  - The neural network architecture has the nodes pointing to themselves
    - Each prior state in a sequence feeds into the next state in the hidden layer
    - Even though sequences of different length have different structures, they can be handled by the same RNN
    - You can also have connections from further back
      - Example of Multiscale RNN
      - Time series with seasonality can be modeled with a multiscale RNN
- How do RNN do in forecasting vs. regular time series methods?
  - For short series, there is not enough for RNN's to outperform standard time series methods
  - RNN's can work with much more data—high frequency trading can make use of RNN's

LSTM

- LSTM stands for Long (and) Short Term Memory
- LSTM overcome the gradient convergence problem by stabilizing modes of keeping longer-term memory
  - Has a multi-scale structure
- In LSTM, there are nodes in the hidden layers called **cells**
  - Connected to the cells are **gates** which control the flow of information through the cells. There are
    - **Input gates**
    - **Output gates**
    - **Forget gates**
  - The forget gate is the key feature which contains the gradient convergence problem