Défis en Intelligence Artificielle

Défi 3 : L'IA pour l'analyse et la prévision de séries temporelles (III/III)

Souhaib Ben Taieb

University of Mons

December 22, 2022

UMONS
Université de Mons

Faculté
des Sciences

Overview

Fully Connected Neural Networks (FCNN)

Convolutional Neural Networks (CNN)

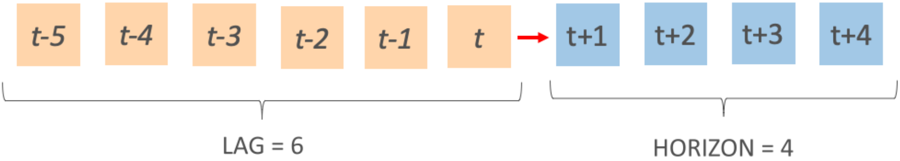
Recurrent Neural Networks (RNN)

Key Takeaways

Kaggle competition

Multi-step ahead forecasting

Assuming we are at time t, we want to predict the values at time $t+1, t+2, \ldots, t+\text{HORIZON}$ conditional on the previous LAG values of the time series.



Outline

Fully Connected Neural Networks (FCNN)

Convolutional Neural Networks (CNN)

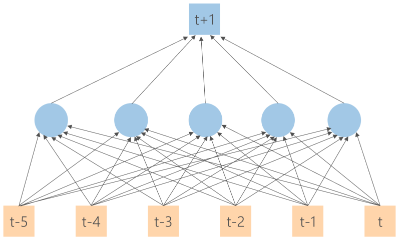
Recurrent Neural Networks (RNN)

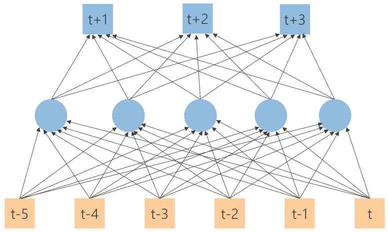
Key Takeaways

Kaggle competition

Fully Connected Neural Networks

Also called Multilayer perceptron (MLP)





Multi-step ahead forecasting strategies: recursive, direct and multi-output.

Outline

Fully Connected Neural Networks (FCNN)

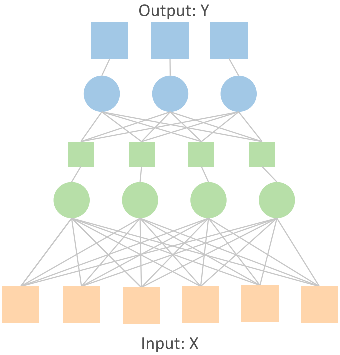
Convolutional Neural Networks (CNN)

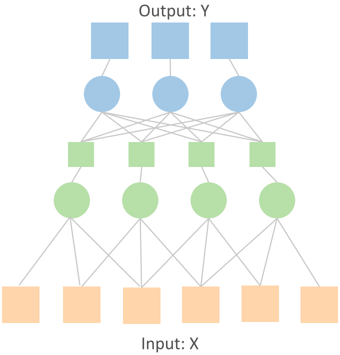
Recurrent Neural Networks (RNN)

Key Takeaways

Kaggle competition

Fully connected layers vs Convolution layers

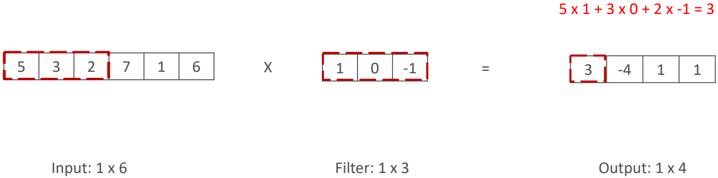




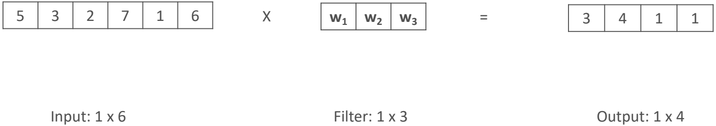
- **Fully connected layers**: units in hidden layers are connected to *every* unit in previous layer
- **Convolution layers**: units in hidden layers operate on a *field* of the input and weights are *shared* across input

1D Convolutions

- Apply a 1D filter to all elements of a 1D input vector
- Result is the sum of the element-wise product

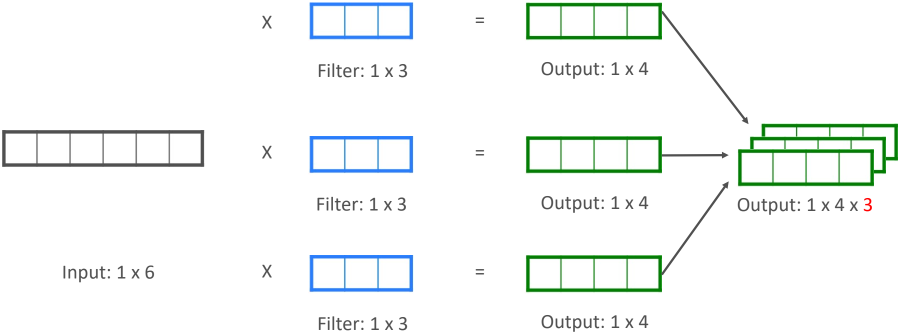


- Filters are trained to detect features in the input sequence



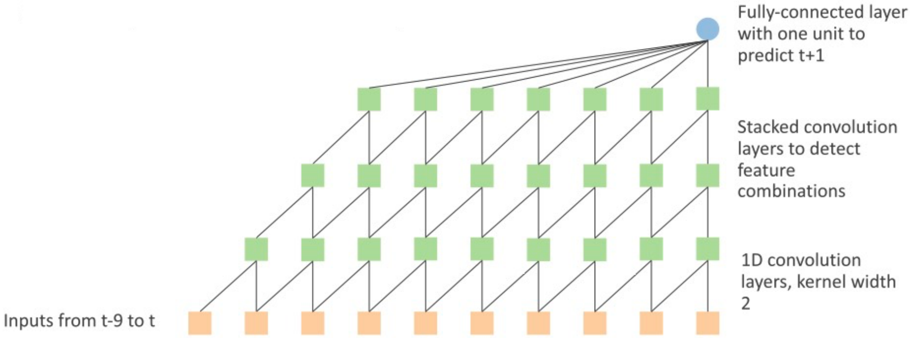
1D Convolutions

- Apply multiple filters to the input data to detect multiple features



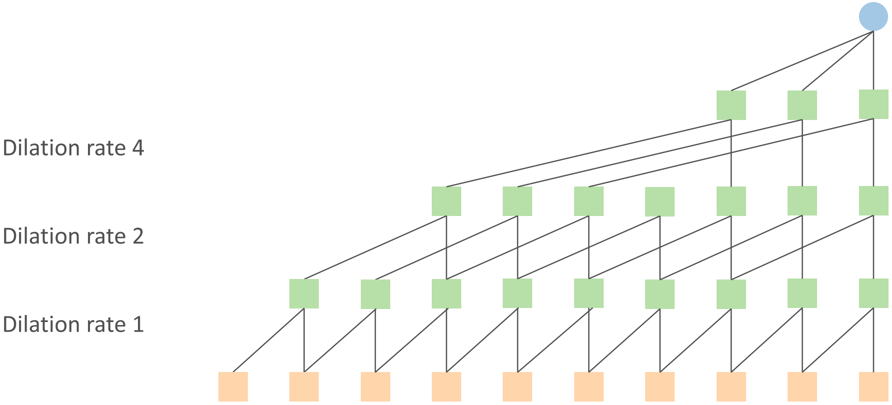
Causal 1D Convolutions for forecasting

- Causal convolutions: the output at each time step does not depend on future time steps



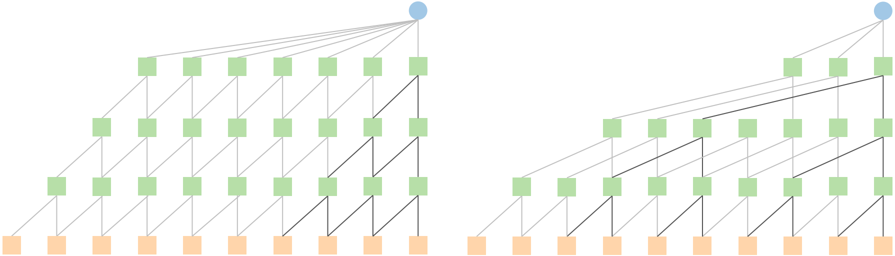
Dilated convolutions

- Skip outputs from previous layers



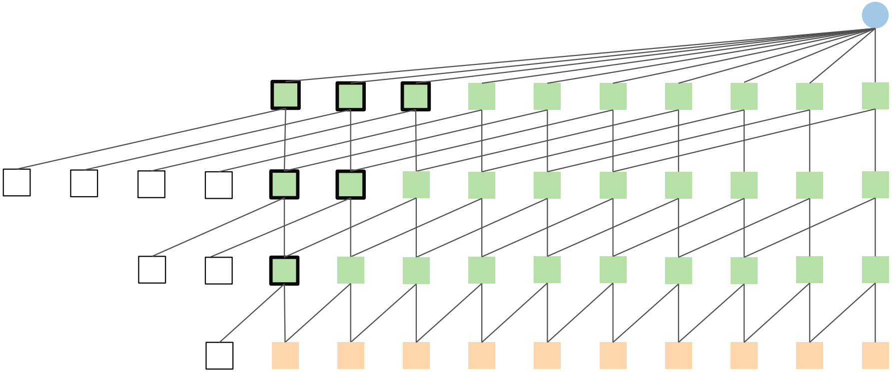
Why dilated convolutions?

- Normal convolutions = more connections = more weights to train
- Reach information from more distant values in the time series



Causal padding

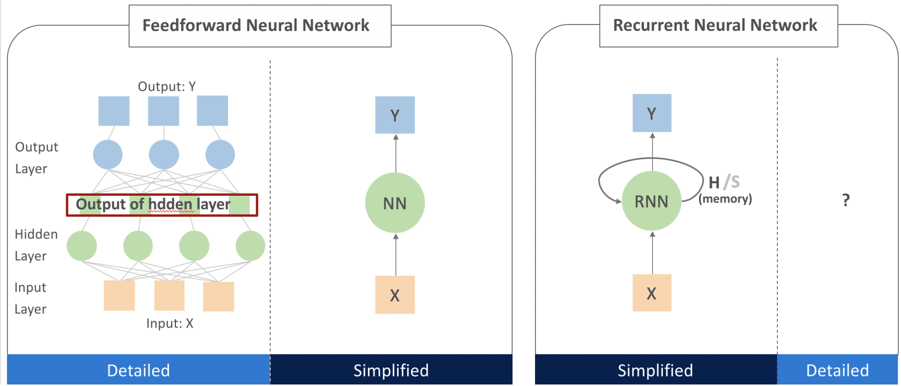
- Padding is necessary to preserve output dimension
- Allows all filter weights to apply to all inputs



Outline

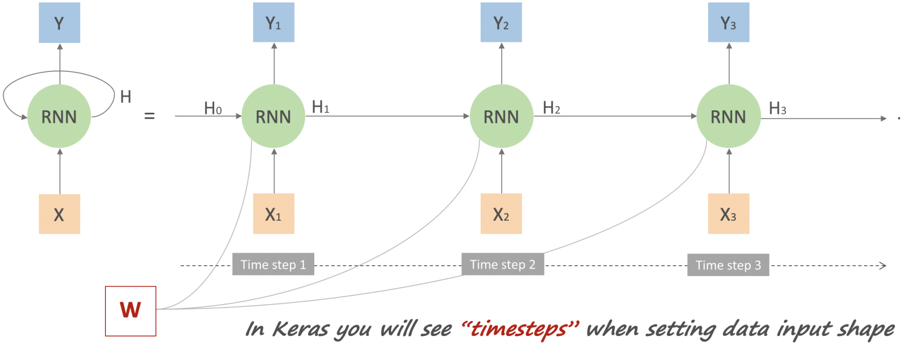
What are RNNs?

- RNNs have internal hidden states which can be fed back to the network



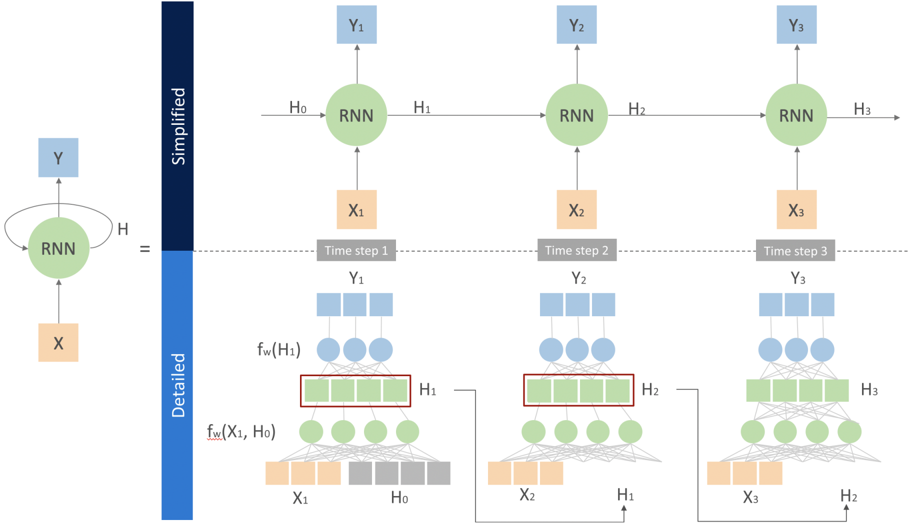
Unrolled RNN

- The same weights are shared across all the steps



In Keras you will see "timesteps" when setting data input shape

Unrolled RNN

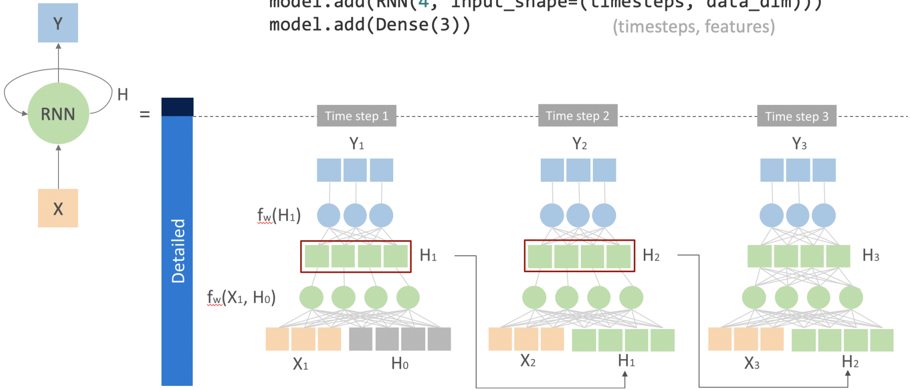


Unrolled RNN

In Keras, the parameter "units" is dimensions of hidden state.
(Think of it as feedforward neural network number of units in hidden layer.)

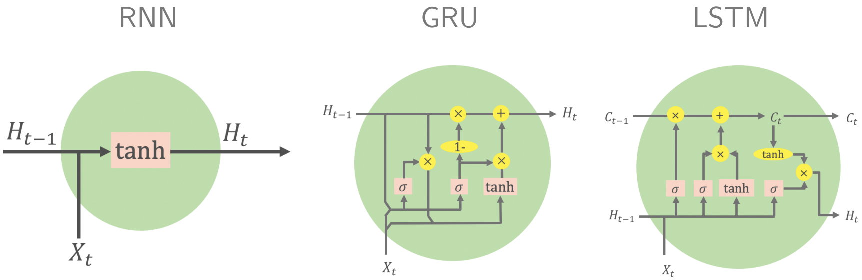
```
model = Sequential()
model.add(RNN(4, input_shape=(timesteps, data_dim)))
model.add(Dense(3))
```

(timesteps, features)

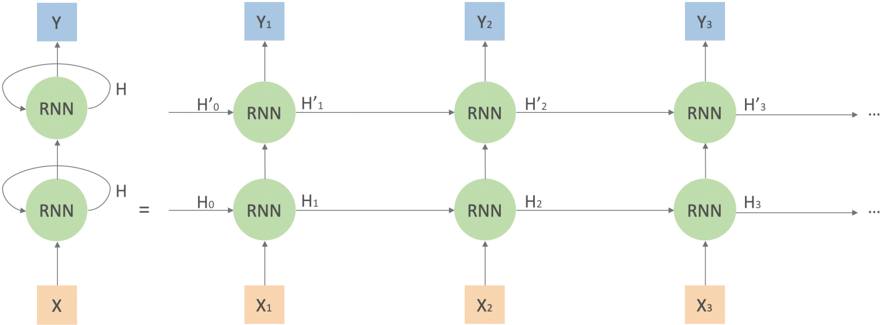


Other architectures

- Exploding and vanishing gradients
  - Change architecture to
    - Gated Recurrent Unit (GRU) or Long Short Term Memory (LSTM)

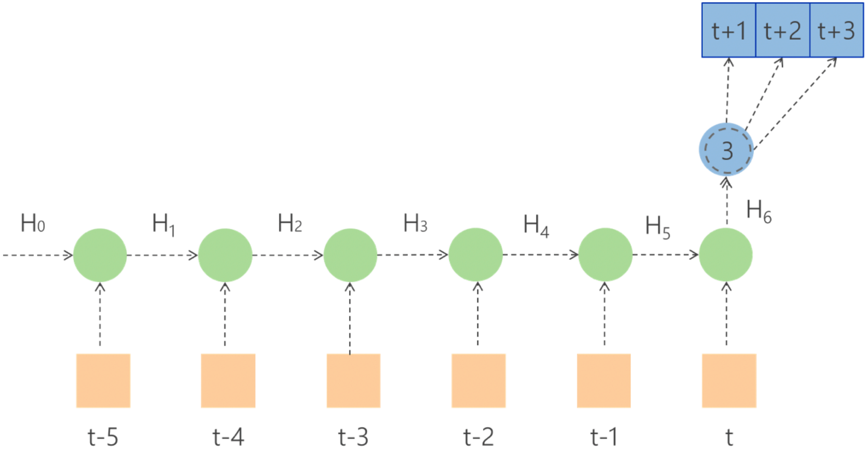


Stacking layers

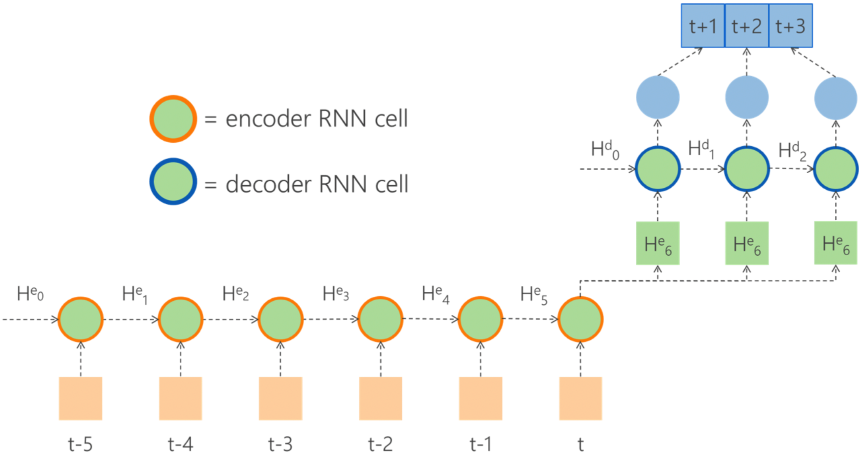



```
model = Sequential()
model.add(RNN(4, return_sequences=True, input_shape=(timesteps, data_dim)))
model.add(RNN(4))
```

Vector output



Encoder-decoder



Outline

Key Takeaways

- **Specific features of time series data**
  - Time series patterns (trend-cycle, seasonality, etc)
  - Temporal dependences
  - Non-stationarity (in mean and variance)
- **Statistical/time series models (e.g. SARIMA)**
  - Good performance for many real-world applications
  - Generally, less computationally demanding than AI methods
  - Poor performance on large datasets with highly nonlinear patterns
  - Often used as baselines for AI models
- **AI forecasting methods**
  - Reduce the problem to one or multiple regression problems
  - MLP with various strategies for multi-step ahead forecasting
  - Specific neural network architectures for sequential data: 1D CNN, RNN (GRU, LSTM), encoder-decoder, etc
  - Time series (cross-)validation
  - Tuning hyperparameters and other tricks are essential for creating an accurate forecasting model

Key Takeaways

- **Ideas to explore for the competition**
  - Global models
  - Forecast combination
  - ...
- **Topic not covered**
  - Multivariate time series
  - Probabilistic forecasting
  - Exponential smoothing methods
  - Other architectures: attention/transformers, GANs, etc.
- **Software for time series forecasting** (not for the competition!)
  - Darts (Python) (`https://unit8co.github.io/darts/`)
  - Prophet (R, Python) (`https://facebook.github.io/prophet/`)
  - NeuralProphet (PyTorch) (`https://neuralprophet.com/`)
  - GluonTS (PyTorch, Gluon) (`https://ts.gluon.ai/`)
  - StatsForecast (Python) (`https://nixtla.github.io/statsforecast/`)
  - Fable (R) (`https://fable.tidyverts.org/`)

Outline

Kaggle competition

- Kaggle Forecasting Competition
  - 111 daily time series observed over two years ($T = 735$)
  - The forecast horizon is the next four weeks ($H = 28$)
  - `https://www.kaggle.com/t/f8c9ccb1233e4cb491bc1ce0fd3d1f17`
- Rules
  - Group of **two** students
  - Max. five submissoins per day

Task	**Due Date**	**Value**
Project		100%
$\rightarrow$ Kaggle first submission	9 January 2023, 11:55pm	
$\rightarrow$ Kaggle final submission	22 January 2023, 11:55pm	35%
$\rightarrow$ Report	25 January 2023, 11:55pm	65%