Advanced Time Series

Lecture 3:
Forecasting II

Gleb Ivashkevich

What we’ve learned so far

Basic elements:

- basic encoder-decoder design
- probabilistic forecasting
- mentioned some extensions: AR $\longrightarrow$ attention

Today

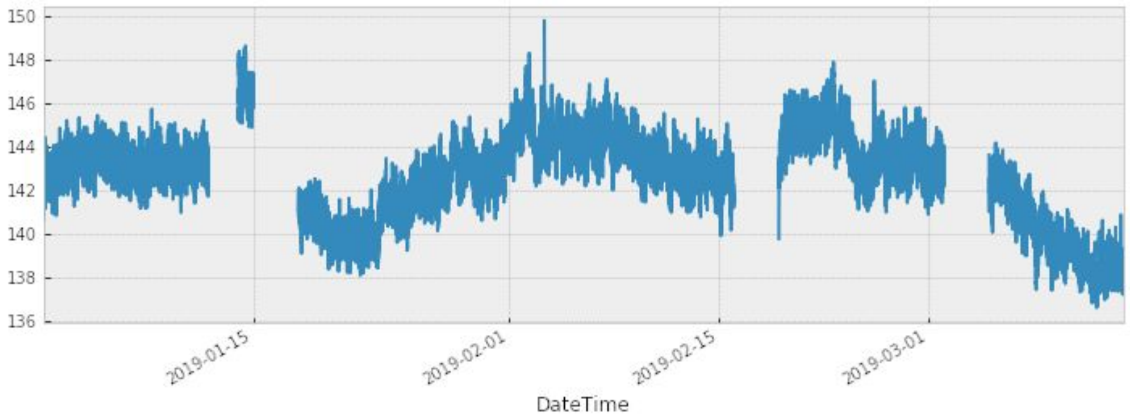
More advanced blocks:

- AR and attention in details
- various example architectures

AR block implementation

Time series scales

For **non-stationary** time series, **scale** may possess a problem: non-linear elements of deep learning models do not scale. **Variability** may be, in contrast, **nearly constant**.



Time series scales

Linear elements, in contrast, are **resilient** to scale changes. We may want to add a **direct AR component**:

$$\tilde{y}_{T+1} = \underbrace{y^{AR}_{T+1}}_{\substack{\textbf{autoregressive} \\ \text{part}}} + \underbrace{y^{ED}_{T+1}}_{\substack{\textbf{encoder-decoder} \\ \text{part}}}$$

$$y^{AR}_{T+1} = \sum_{k=0}^{k=h} W_k y_{T-k} + b$$

AR component

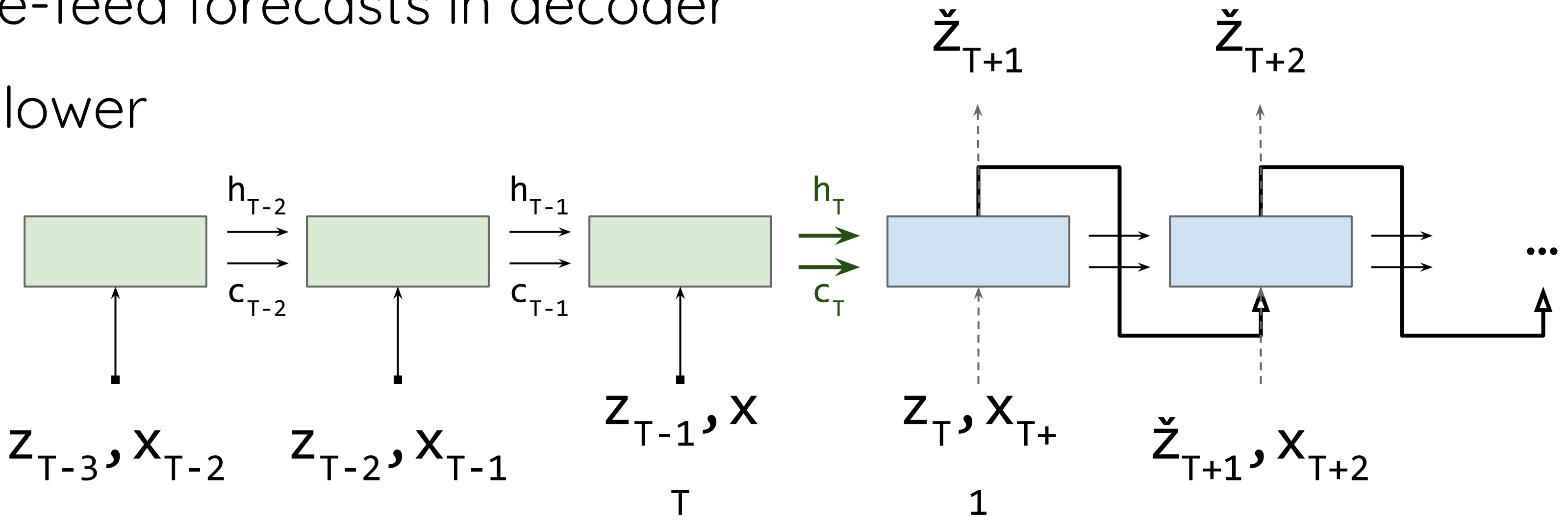
Direct AR component:

- handles changing (non-seasonal!) scale
- works nicely with other elements, objective functions and training procedures (it's just a dense layer after all)

Shared weights

Design technicalities:

- re-feed forecasts in decoder
- slower



LSTM recap

- input: (N, L, H_{in}) (batch_first==True)
- output: (N, L, H_{out})
- h: (n_layers, N, H_{out})
- c: (n_layers, N, H_{out})

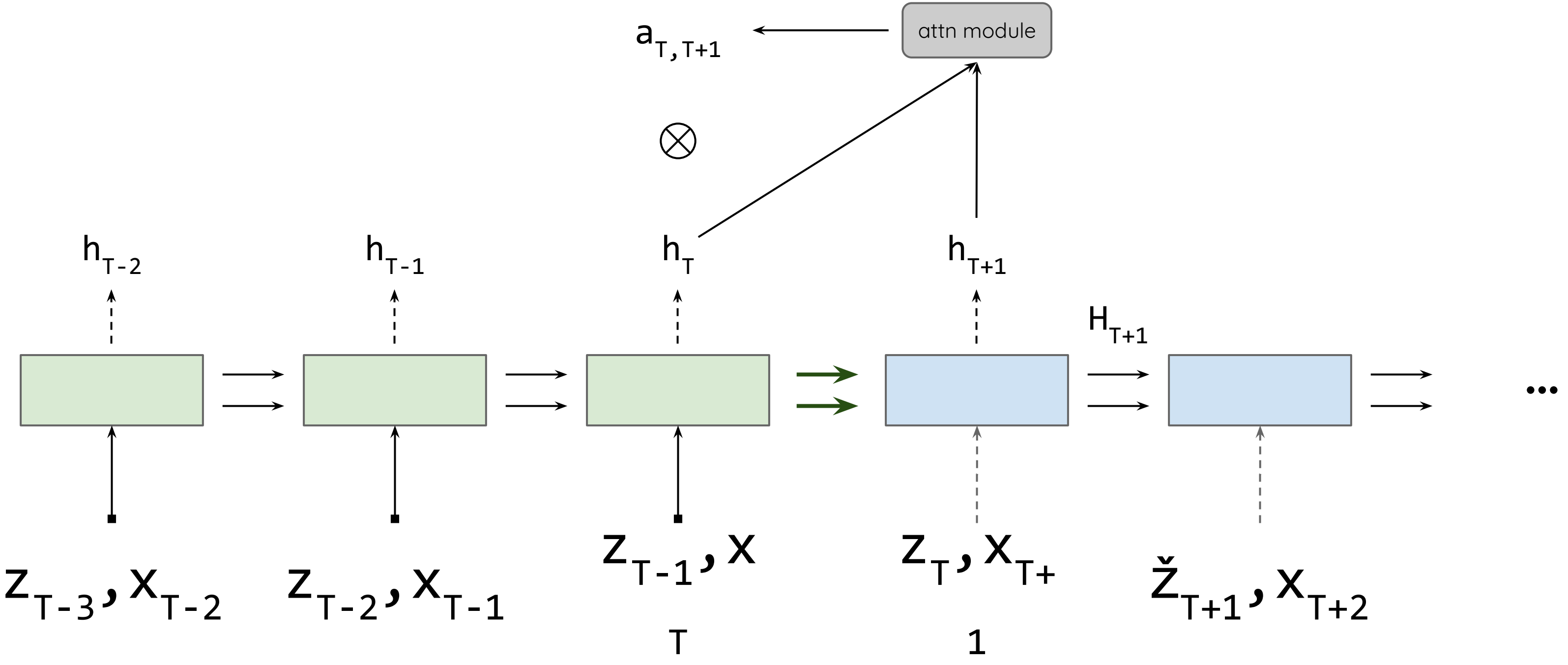
For power consumption (grid-level): $H_{in} = 1$

Temporal attention

Motivation

- last hidden state of the encoder may be not enough to encode the history properly
- some specific moments in history may be more important than others
- we want some mechanism able to look at those moments

Idea



attn module
$a_{T,T+1}$
$\otimes$
h_{T-2}
h_{T-1}
h_T
h_{T+1}
H_{T+1}
...
z_{T-3}, x_{T-2}
z_{T-2}, x_{T-1}
z_{T-1}, x_T
z_T, x_{T+1}
$\check{z}_{T+1}, x_{T+2}$

Idea

- hidden state $\mathbf{H}_{T+1}$ is a combination of $\mathbf{h}_{T+1}$ and a weighted sum of $(\ldots, h_{T-3}, h_{T-2}, h_{T-1}, h_T)$: $F(h_{T+1}; a_{k,T+1}h_k)$
- weights (**attention** scores) are some proper functions of $\mathbf{h}_k$ and $\mathbf{h}_{T+1}$: $A(h_k, h_{T+1})$
- this is basically a version of soft alignment, known in NLP

[Neural Machine Translation By Jointly Learning To Align And Translate]

Implementation

- power consumption dataset (for simplicity)
- GRU, not LSTM - single hidden vector
- **F** is a simple feedforward nets and **A** scalar product
- have to iterate manually in GRU (need all intermediate hidden states), hence will use **GRUCell**

LSTNet & others: all the blocks

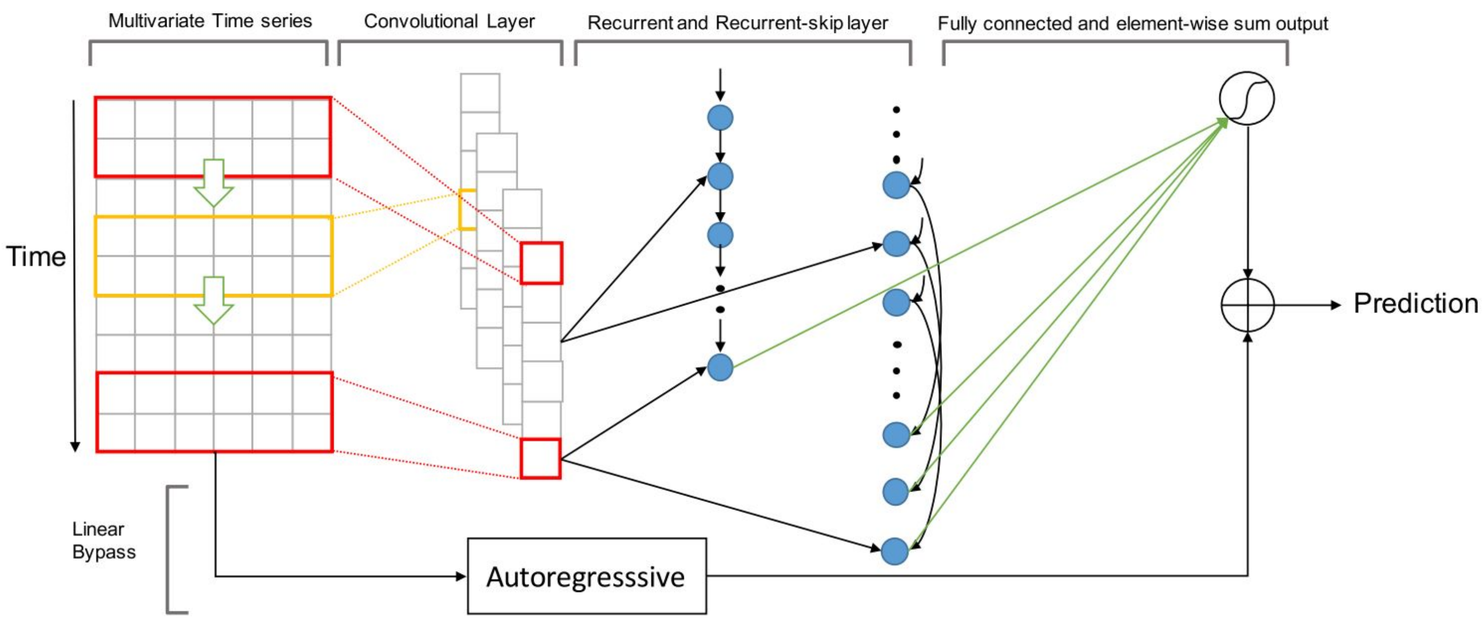
LSTNet

Setup:

- multivariate time series
- global + local patterns
- Conv + Recurrent + skip connections + AR loop

[Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks]

LSTNet



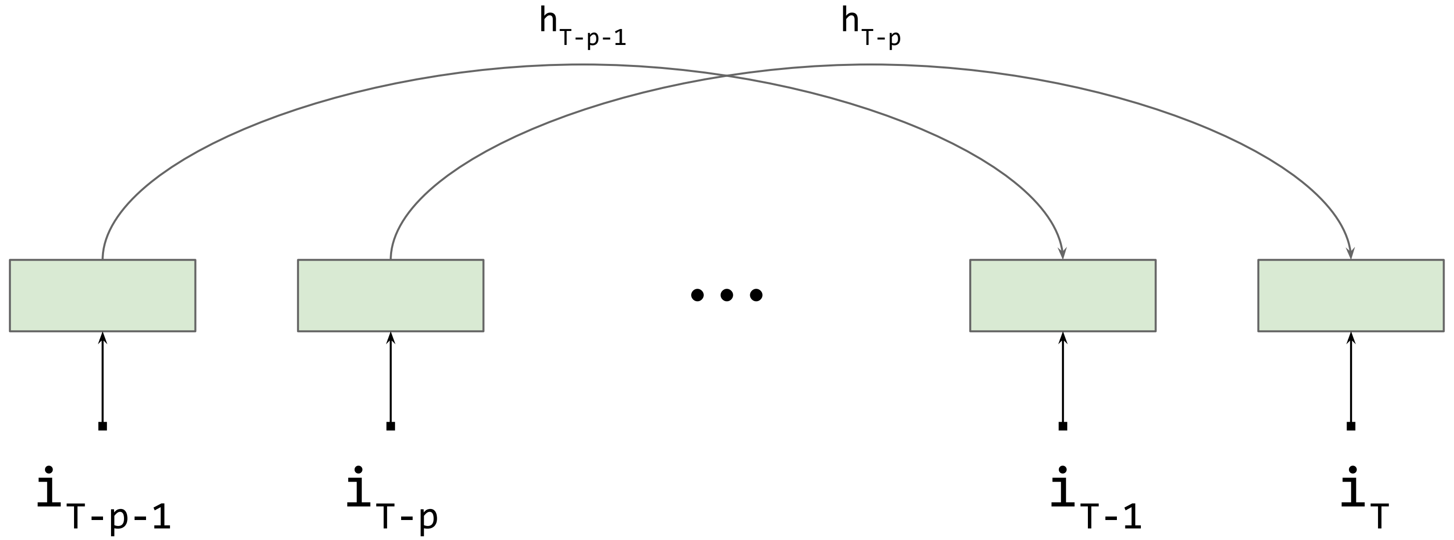
Multivariate Time series
Convolutional Layer
Recurrent and Recurrent-skip layer
Fully connected and element-wise sum output
Time
Prediction
Linear
Bypass
Autoregresssive

LSTNet

Features:

- **convolutions** look natural for multivariate t.s.: capture short range dependencies between variables
- **recurrent layer** (GRU) captures long range dependencies
- recurrent **skip-connections:** capture seasonality (longer range)

LSTNet: recurrent-skip



LSTNet

Features:

- **temporal attention**(instead of rec. skip)**:** no need to specify lag time
- **AR bypass:** discussed earlier
- available in GluonTS

LSTM-MSNet

Setup:

- input data: similar to DeepAR
- explicit handling of seasonality
- slicing: similar to W1 encoder-decoder

[LSTM-MSNet: Leveraging Forecasts on Sets of Related Time Series with Multiple Seasonal Patterns]

DeepGLO

Setup:

- **high-dimensional t. s.**
- **no normalization**: handled by initialization
- **temporal CNN**

[Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting]

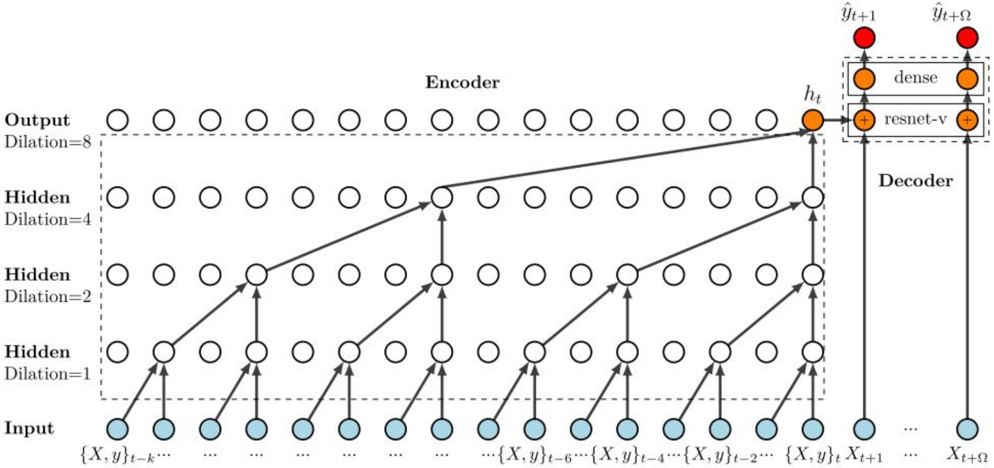
Other

[Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models]

[Probabilistic Forecasting with Temporal Convolutional Neural Network]

Other

[Probabilistic Forecasting with Temporal Convolutional Neural Network]



Next time

- N-BEATS
- transformer architecture
- convolutions in details
- setting up the classification problem

HW 2

A Multi-Horizon Quantile Recurrent Forecaster

https://arxiv.org/abs/1711.11053

Paper implemetation:

- deadline is June 10 24:00
- see instructions in Google Classroom

questions?