Advanced Time Series

Lecture 3:
Forecasting - II

Gleb Ivashkevich

HW 1 review

Finalization of HW 1:

- solutions were **released**
- full-mode for **all who submitted**
- **review** will be ready soon (screencast)

HW 2 review

A Multi-Horizon Quantile Recurrent Forecaster

https://arxiv.org/abs/1711.11053

Paper review:

- deadline is Feb 28 24:00
- see instructions in full-mode Google Classroom
- compare with DeepAR (today)

Today

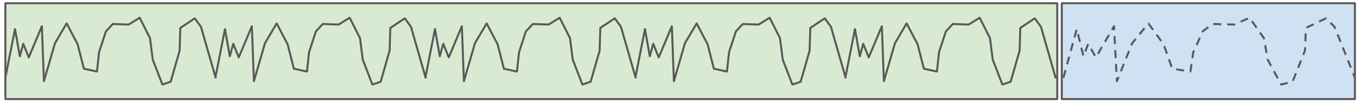
Time series forecasting:

- naive **LSTM encoder-decoder** model implementation
- **probabilistic** forecasting: DeepAR model
- AR connections

Encoder-decoder architecture

Encoder-decoder setup

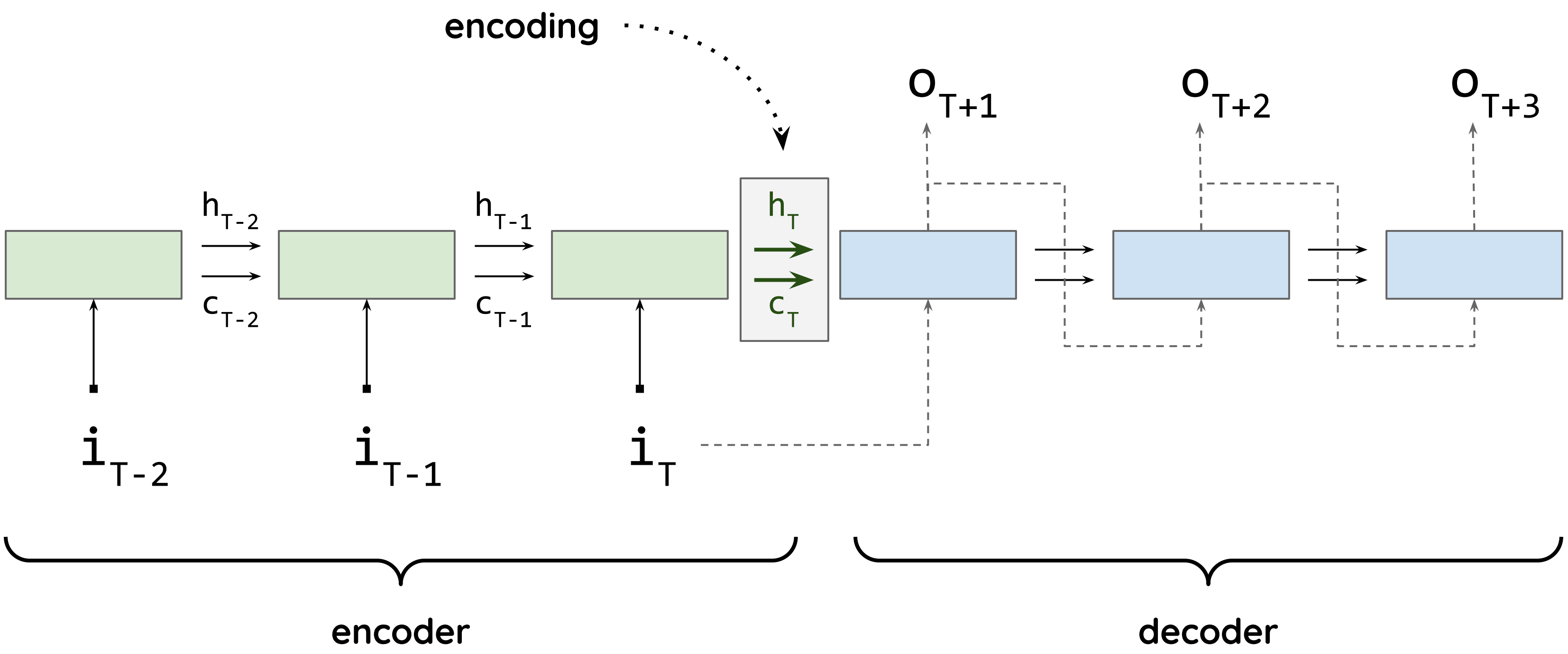
endogenous only weather time series, power consumption, sales



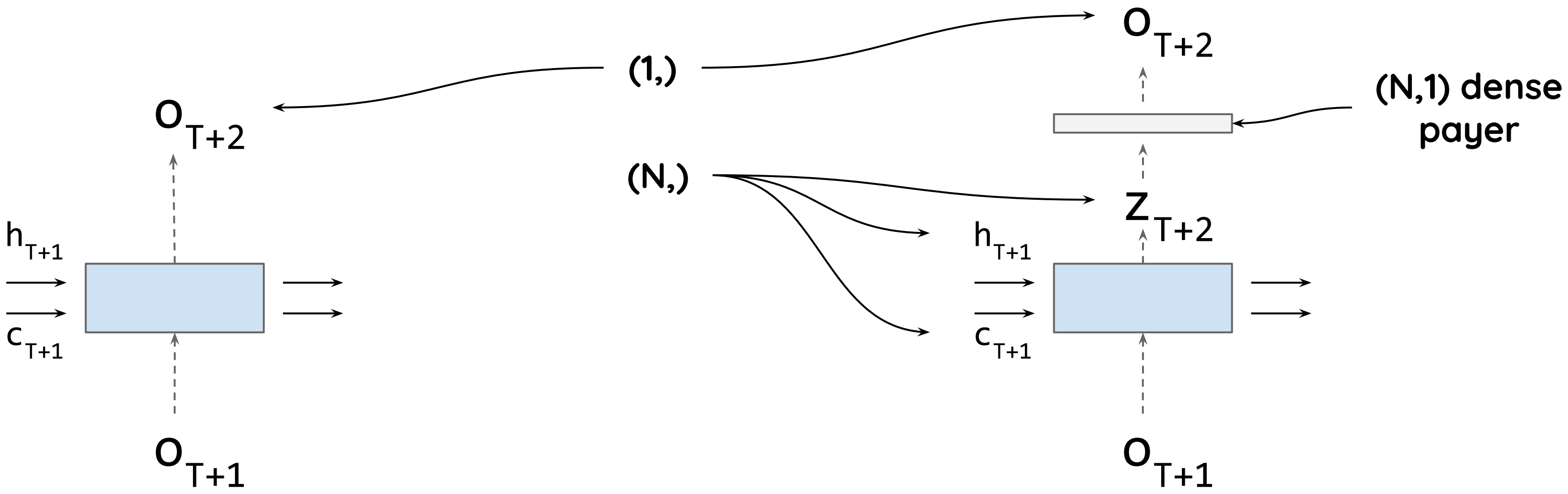
The most basic setup:

- **only target** itself is used
- **calendar** information may be added

Encoder-decoder arch



Encoder-decoder: implementation details



*1D target, N hidden units

Encoder-decoder arch

Variations:

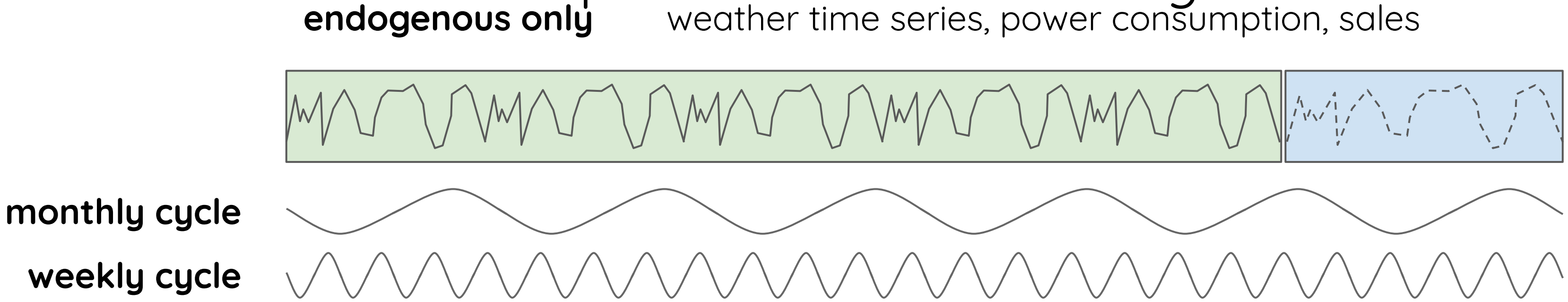
- encoder and decoder may **share weights**
- encoder and decoder may have **different architectures**
- **calendar** information, **exogeneous** variables may be added
- **categoricals** may be added (one-hot or embeddings)

Calendar information

Two main options:

- **one-hot encoded** (month, day of week, weekend/weekday, holidays, sale)

- **Fourier features:** explicit multi-seasonality

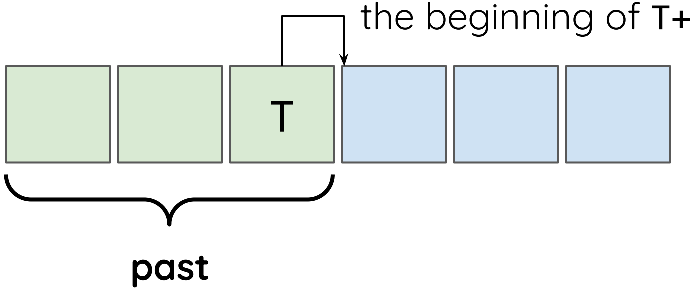


Production considerations

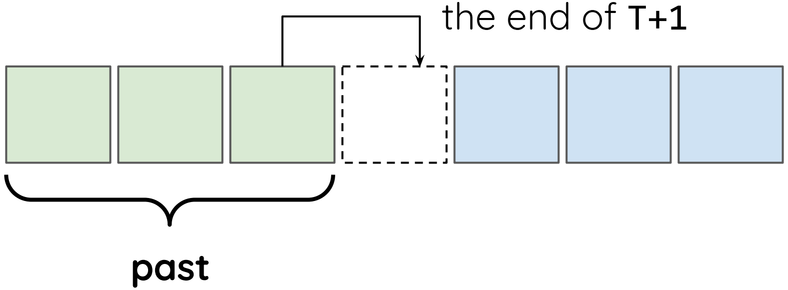
Forecasting windows:

- it is usually desirable to forecast on **regular intervals**
- based on data availability, you may need to **skip a window**

Data for **T** is available at the beginning of **T+1**



Data for **T** is available at the end of **T+1**



Probabilistic forecasting

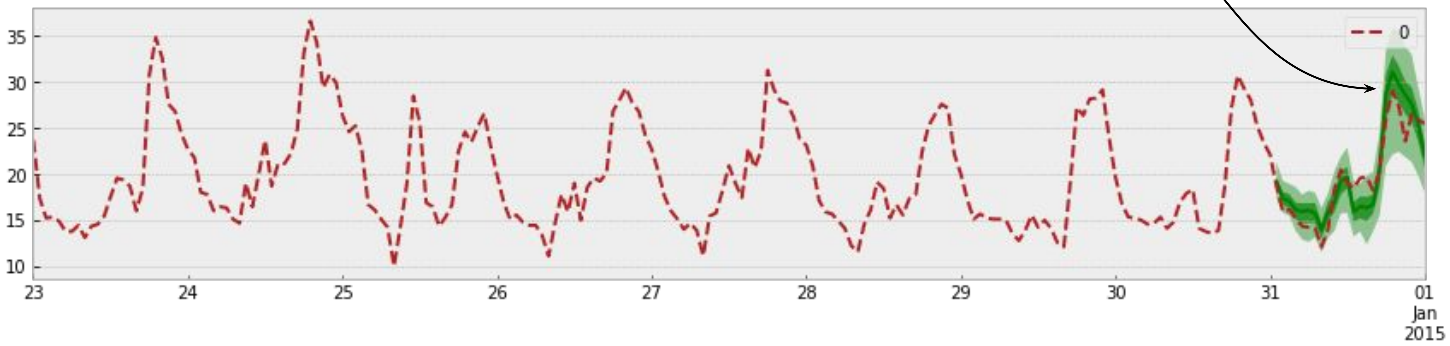
Probabilistic forecasting

Point forecasts provide only one sample of a **random process**. Very often it is **not useful**.

Probabilistic forecasts provide entire distribution over future values. Very plausible feature, as it allows for uncertainty estimation and fine-grained management of extreme scenarios.

Probabilistic forecasting

50% and 90%
uncertainty bands



Two flavors of probabilistic forecasting

- **Bayesian**
- forecast **probability distribution** directly

Bayesian

likelihood

prior
distribution
over model
parameters

$$\underbrace{p(\theta|D)} = \frac{\overbrace{p(D|\theta)}\,\overbrace{p(\theta)}}{p(D)}$$

posterior
distribution
over model
parameters

family
of models → **inference**

Bayesian

Bayesian ML is a **domain on its own**. We need only **likelihood**.

To compute likelihood, we need to set **probability distribution**.

Likelihood

Gaussian distribution:

$$p(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$p(D|\theta)$$

Direct

Predict **probability distribution parameters**:

- impose a **proper** distribution (say, Gaussian for continuous, Poisson for counts)
- **predict parameters** of that distribution with your model
- **maximize likelihood** of your data given the predicted parameters

at inference time: **sample** from the predicted distribution

DeepAR model

[DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks]

DeepAR model implements the idea:

- first introduced in 2017
- one of the goals of handling multiple related t. s.
- improves over the benchmarks

DeepAR model

DeepAR:

- despite “autoregressive” in its name, it’s an **encoder-decoder** architecture with **shared weights**
- trained on **electricity load** dataset and others
- **pre-processing** to handle **dramatically different ranges** for different time series

DeepAR model

Pre-processing, training sampling, features:

- **scale** inputs by **per-item average** (works for continuous, but not counts)
- sample time series with large scale **more frequently**
- calendar features + age
- **embeddings** for categoricals (product category for sales/ identity for others)

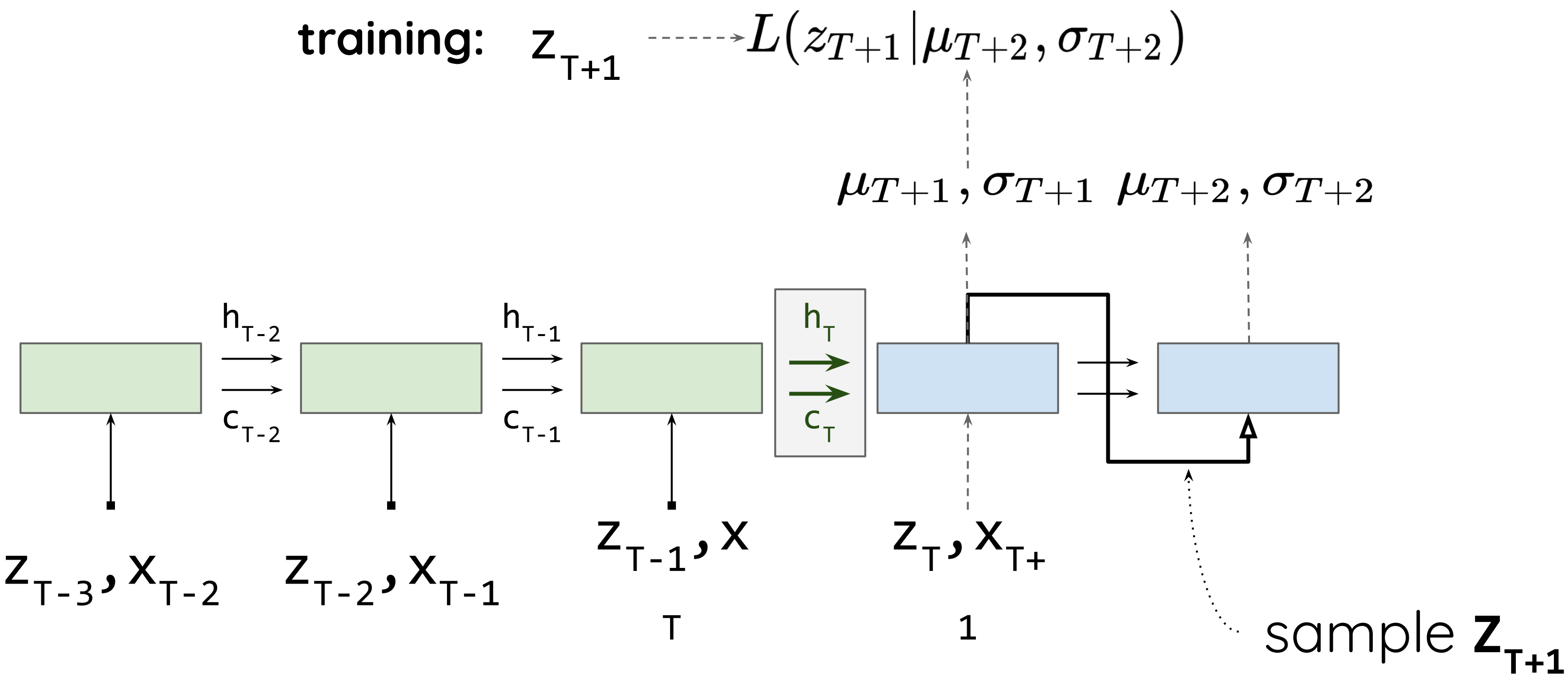
DeepAR model

Design allows for:

- **leveraging information** from related time series: cold start problem resolved
- straightforward **probabilistic forecasting**
- **additional covariates** are not a problem

Implementations: GluonTS - MXNet-based

DeepAR design in details



DeepAR model

linear dense layer

$$\mu(h_{i,t}) = W_\mu h_{i,t} + b_\mu$$

softplus dense layer

$$\sigma(h_{i,t}) = \log(1 + \exp(W_\sigma h_{i,t} + b_\mu))$$

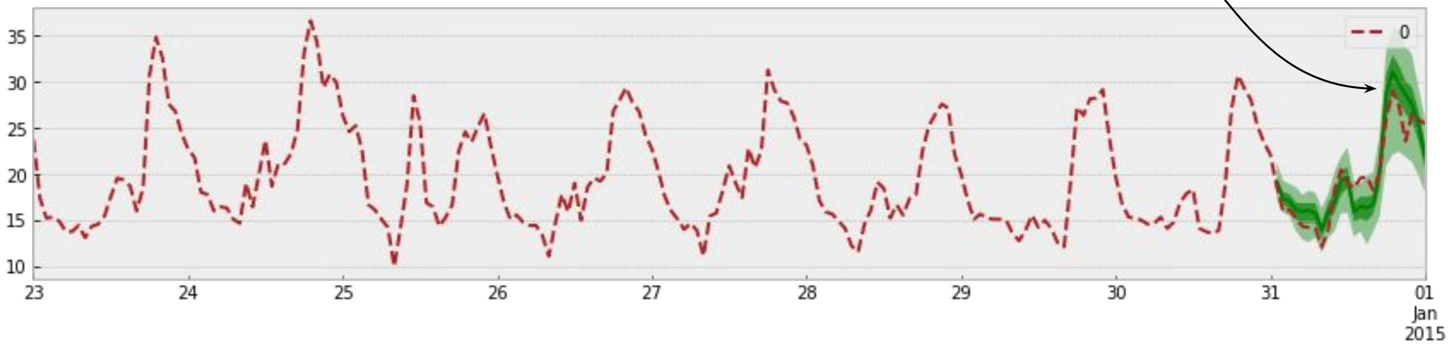
DeepAR model

Design technicalities:

- hourly data, 168 hours of historical data, 24 hours forecasting horizon
- 3 LSTM layers, 40 hidden units
- batch size: 64

DeepAR model

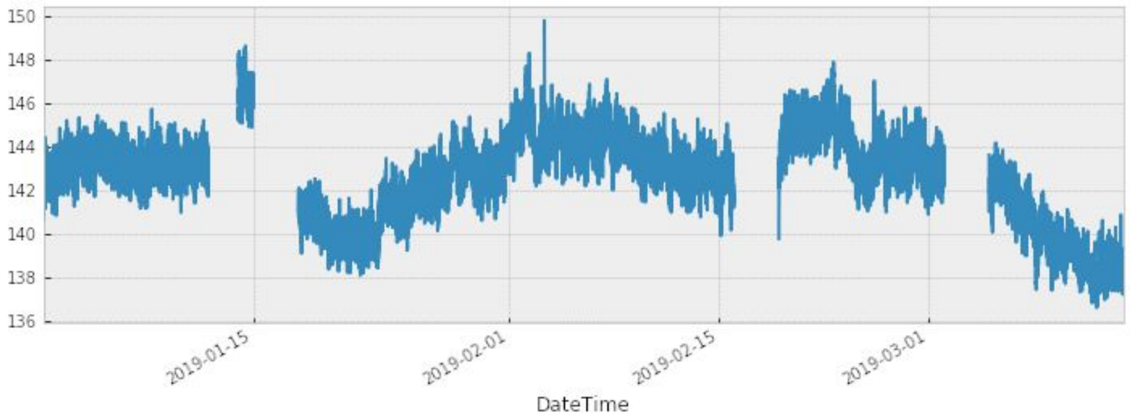
50% and 90%
uncertainty bands



Explicit autoregressive component

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)

Next steps

Hybrid architectures

Usable components:

- **recurrent**
- **AR**
- convolutional
- attention

link to t. s. **classification**

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

Assignment

- implement Quantile Forecaster (just a prototype) paper from HW 2

questions?