Advanced Time Series

Lecture 6:
Time-to-event and PdM II

Gleb Ivashkevich

Today

Time-to-event and PdM $\longrightarrow$ transformer and t. s.:

- **Cox proportional hazards model**
- survival analysis and **deep learning**
- transformers for time series

Survival analysis and predictive maintenance

Survival analysis

Concept 1:

- **survival function:** probability of surviving past t

$$\underset{\uparrow \\ \text{time of failure}}{S(t) = P(T > t)}$$

Survival analysis

Concept 2:

- **hazard function:** conditioned event rate

$$\lambda(t) = -\frac{S'(t)}{S(t)}$$

Hazard function deciphered

Concept 2:

- **hazard function:** conditioned event rate

$$\lambda(t) = -\frac{P(T>t+dt)-P(T>t)}{S(t)dt}$$

Hazard function deciphered

Concept 2:

- **hazard function:** conditioned event rate

$$\lambda(t) = -\frac{P(T>t+dt)-P(T>t)}{S(t)dt}$$

Proportional hazards

Concept 3:

- **proportional** hazards model (Cox regression)

$$\lambda(t) = \lambda_0(t)\exp(a_i X_i)$$

covariates

Hazards and s. f.

Concept 4:

- **cumulative** hazard

$$\Lambda(t) = \int_0^t \lambda(\tau) d\tau \rightarrow S(t) = \exp(-\Lambda(t))$$

cumulative hazard

Survival models

non-parametric	**semi-parametric**	**parametric**	**ML**
Kaplan-Meier Nelson-Aalen	Cox PH	AFT models	survival trees, etc.

Survival data

Covariates X_i^k: a vector (i) per object (k)

Lifespan T^k

Event was observed? C^k

Notes:

- one vector of covariates **for entire lifespan**
- some events are **censored** (object **“died”**, but for a different reason)

Survival data: extensions

Covariates **may vary with time** $X_i^k(t)$

For example:

- each object is measured multiple times

Cox regression: likelihood

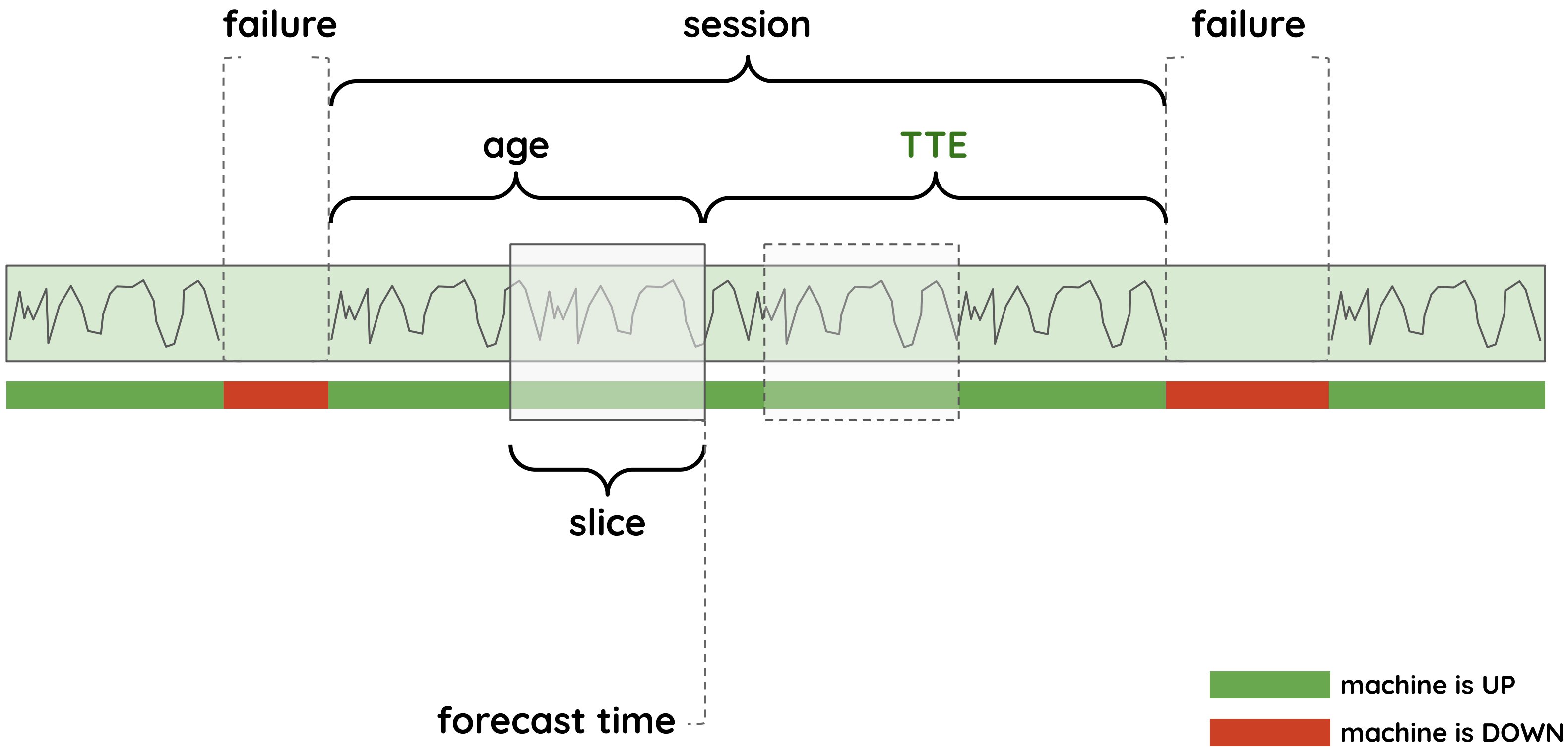
How can we train Cox PH?

Full likelihood **is not specified** (because of baseline hazard). Partial likelihood (for individual object and no ties):

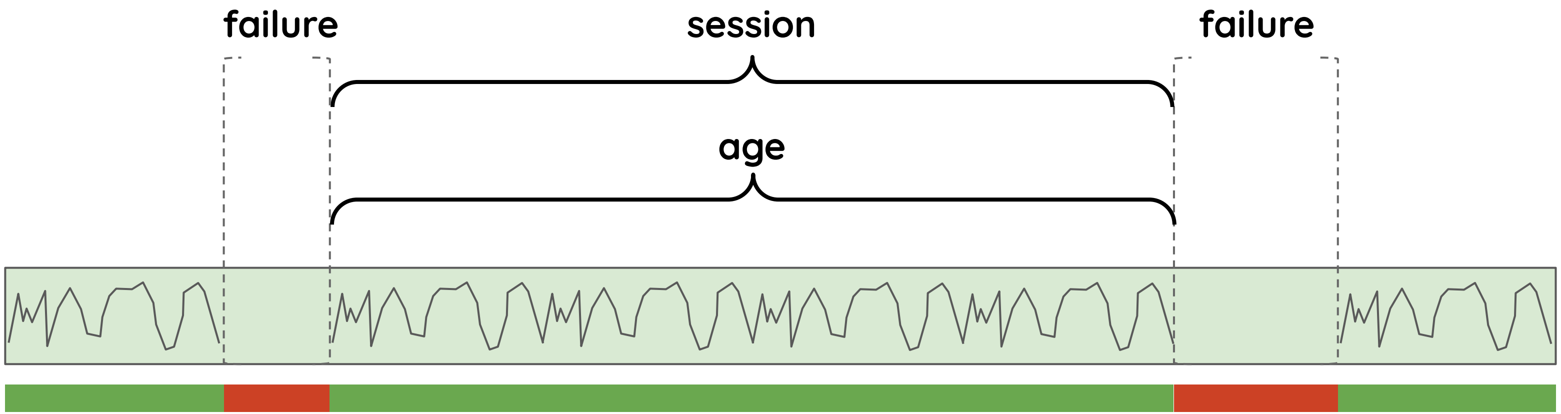
$$L_i = \frac{\lambda(t^i|X^i)}{\sum_{t^j \geq t^i} \lambda(t^i|X^j)} = \frac{\exp\left(a_\alpha X^i_\alpha\right)}{\sum_{t^j \geq t^i} \exp\left(a_\alpha X^j_\alpha\right)}$$

$$L = \prod L_i$$

PdM data setup: slices

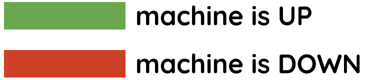


PdM data setup: sessions



One vector of covariates for entire session.

No need for time varying covariates.



Realistic PdM

Some considerations:

- model each type of failure **separately**
  (slices/sessions ended with a different failure are censored)
- session-based analysis for **post-mortem analysis**
- try session-based models for real-time predictions with **expanding windows** (may work for frequent failures)

DL and survival analysis

Some considerations:

- (partial) likelihoods are known for semi-parametric and parametric models
- encode time series: encoder
- push them into the appropriate model
- use Cox or something else as a baseline

DL and survival analysis

For example, for CoxPH:

- if you're lucky, you may train encodings, which work well in PH model:

$$\lambda(t) = \lambda_0(t) \exp(a_i X_i)$$

encoding

DL and survival analysis

Non-specific to time series:

- DeepSurv
- DeepHit

Transformer architectures for time series

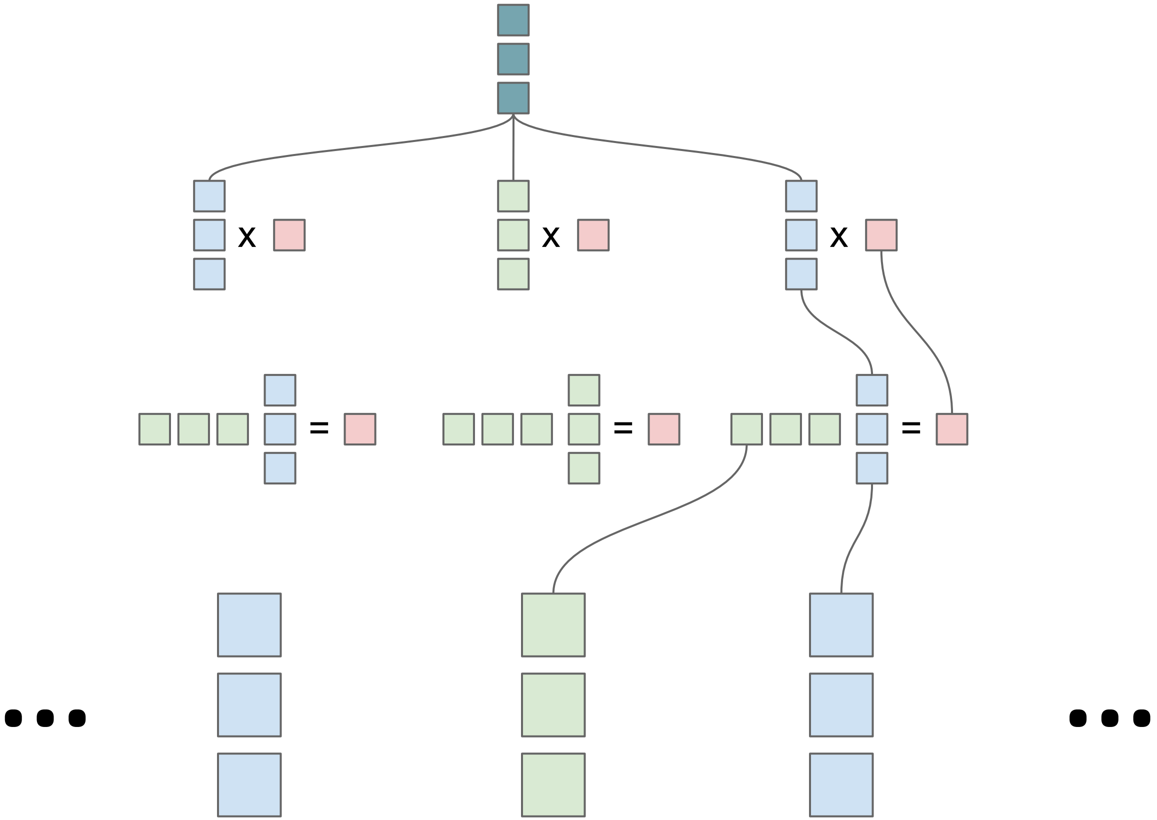
Transformers

Why?

- typical sequential models (RNNs) may still **not catch** temporal dynamics well
- **attention** layer may fix this
- but they are still **sequential**

Transformers are still **encoder-decoder**.

Attention



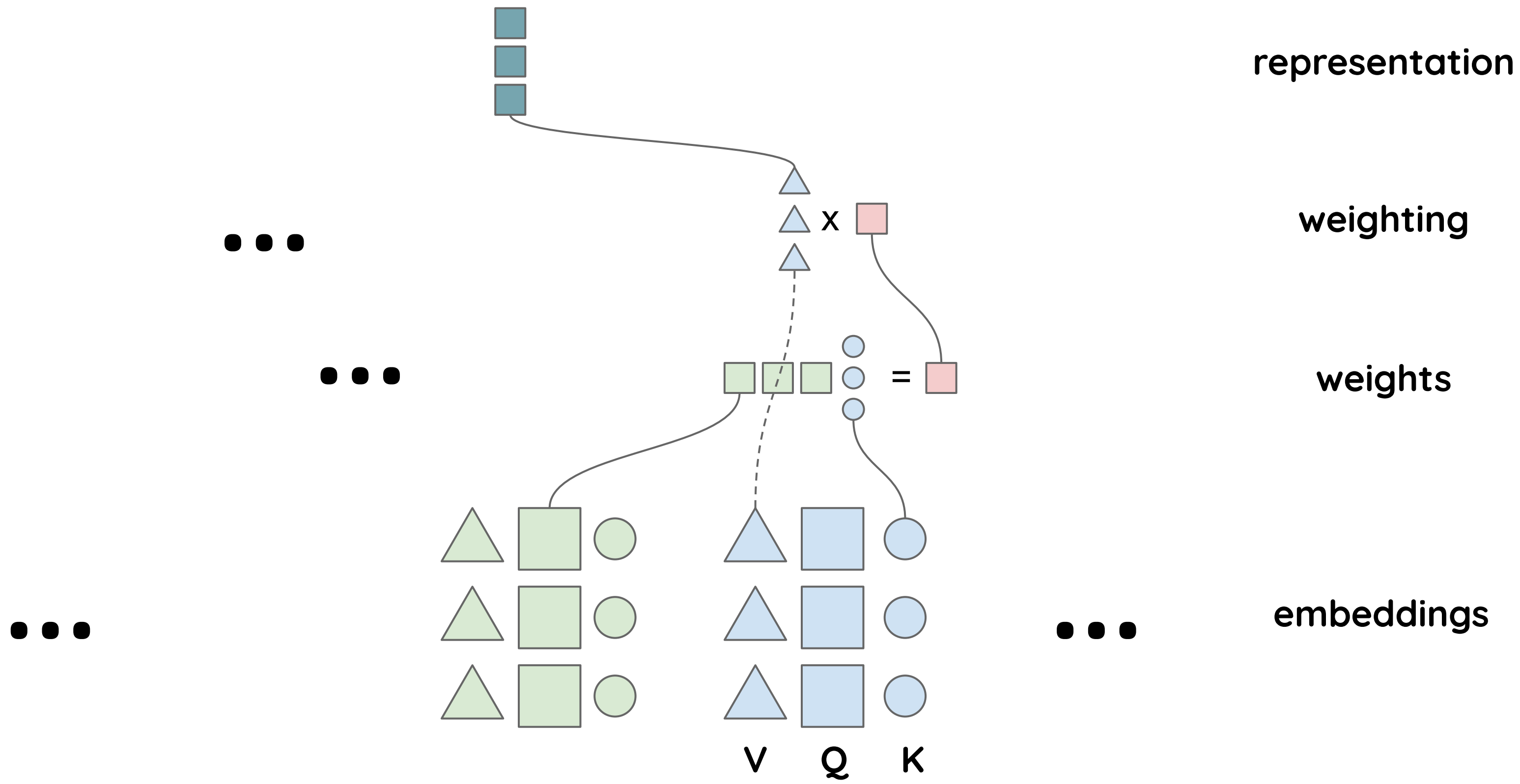
representation

weighting

weights

embedding

Multi-head attention



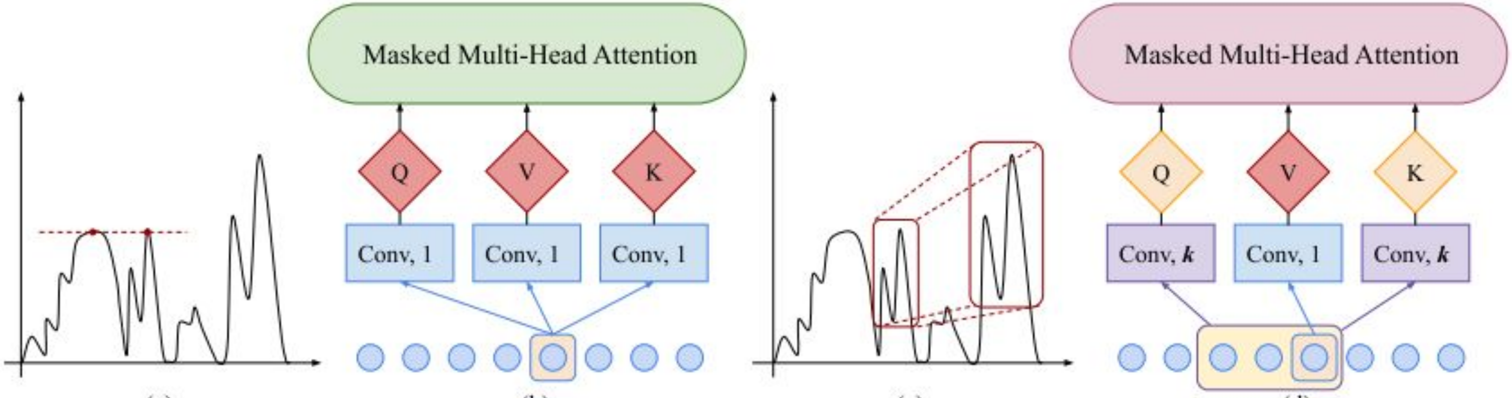
Transformers: forecasting

[Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting]

- innovation: **convolutional attention**
  (queries, keys and values are computed by conv layer)

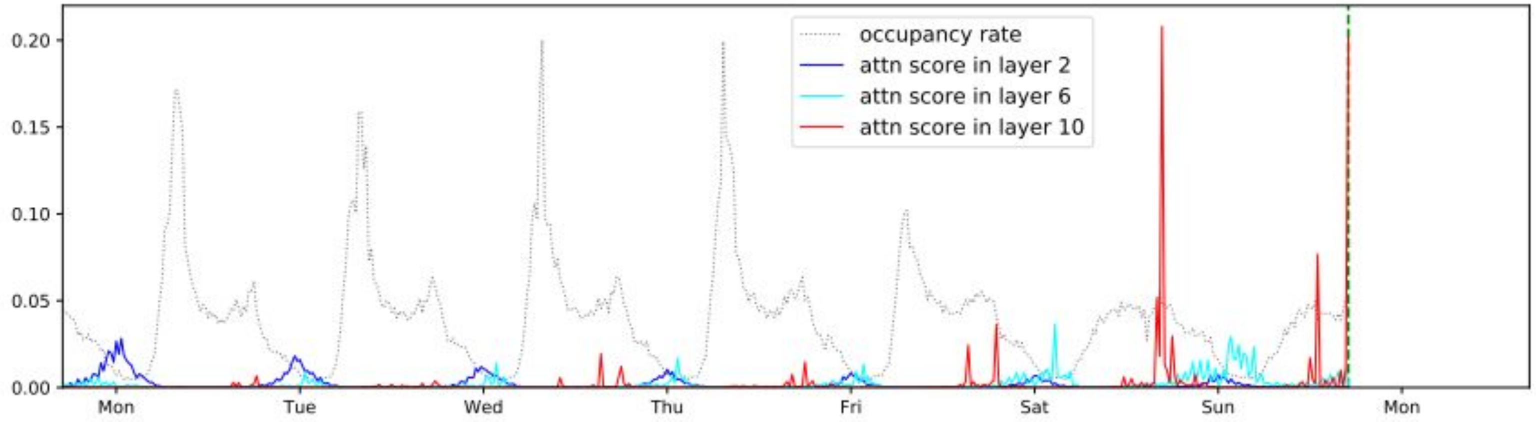
- very good performance compared to other architectures

Transformers: forecasting



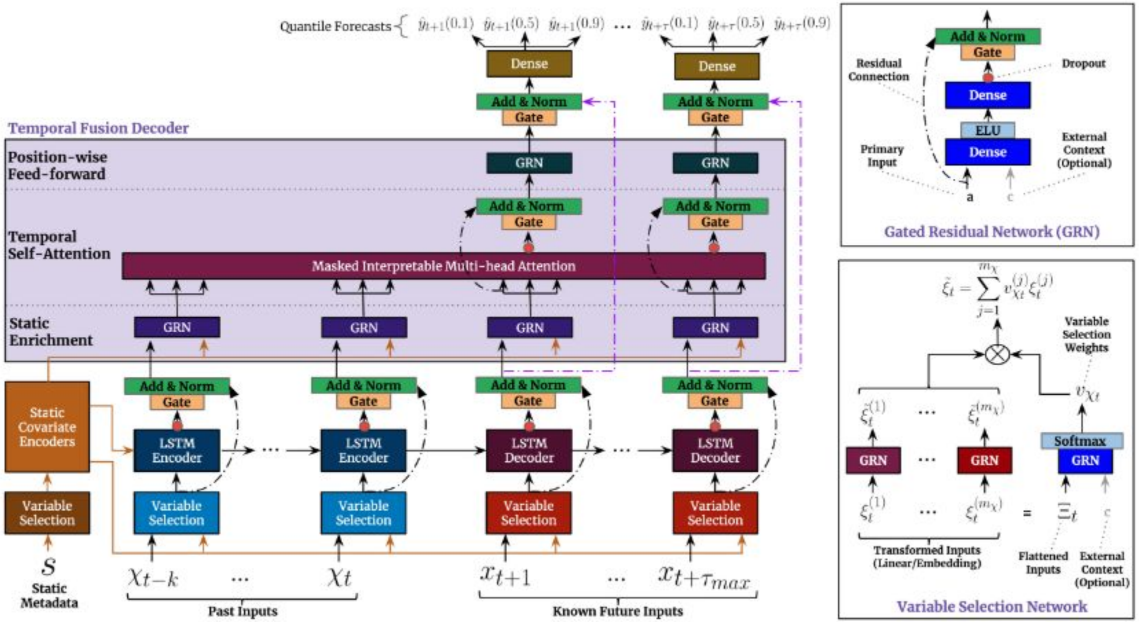
(a) (b) (c) (d)

Transformers: forecasting



Other papers

[Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting]()



Next time

- **representation** learning for time series
- **VAMPnets**
- various **autoencoder** architectures
- **wrap-up**

questions?