

# Deep Learning für Zeitreihen

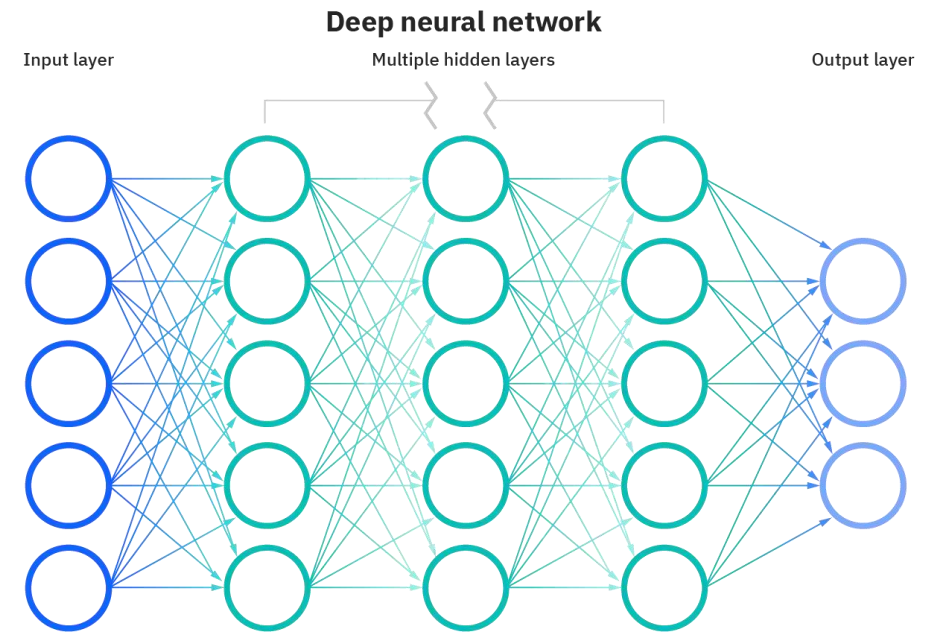
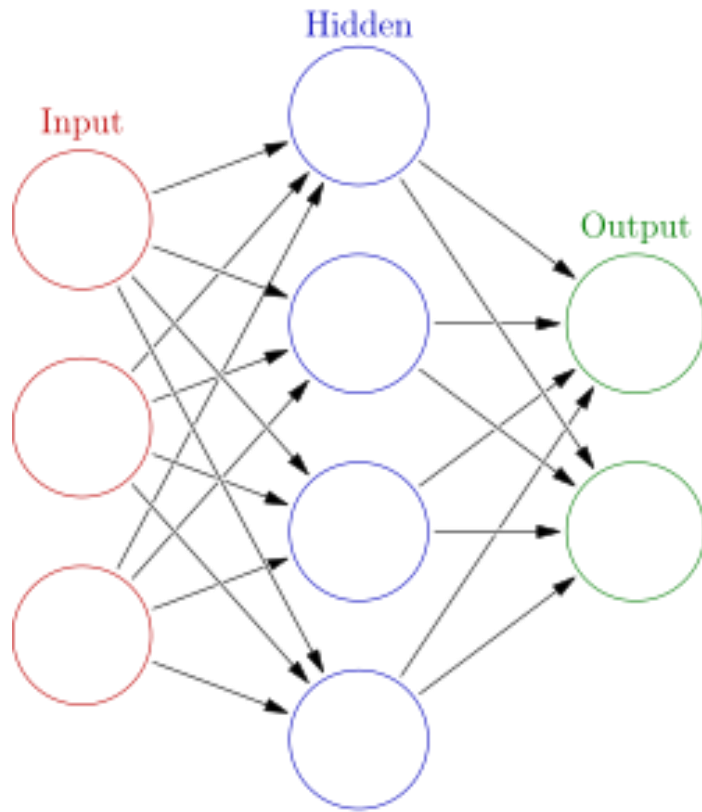
Session 8 (Dienstag 15:15 – 17:00)



# Deep Learning für Zeitreihen

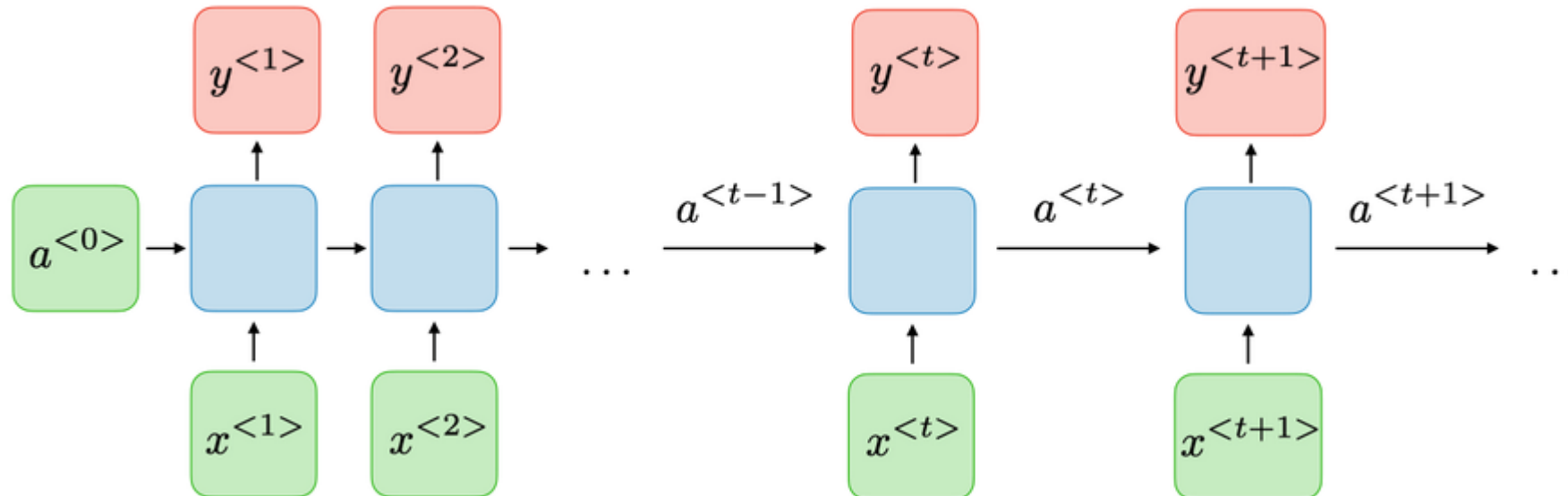
- Recurrent Neural Networks (RNN)
- LSTMs
- Bsp: Stromverbrauch vorhersagen mit mehreren Inputmerkmalen

# Neuronale Netze



# Recurrent neural network (RNN)

- RNNs sind eine Klasse von künstlichen neuronalen Netzen
- Verbindungen zwischen den Knoten bilden eine zeitlichen Abfolge
- Dadurch können sie ein zeitlich dynamisches Verhalten zeigen



Traditional neural network

Music generation

Sentiment classification

Name entity recognition

Machine translation

Welche Arten von Problemen haben wir bisher kennengelernt?

- Ein-Schritt Vorhersage
- Mehr-Schritt Vorhersage
- Clustering
- Klassifikation

# Vanishing/exploding gradient

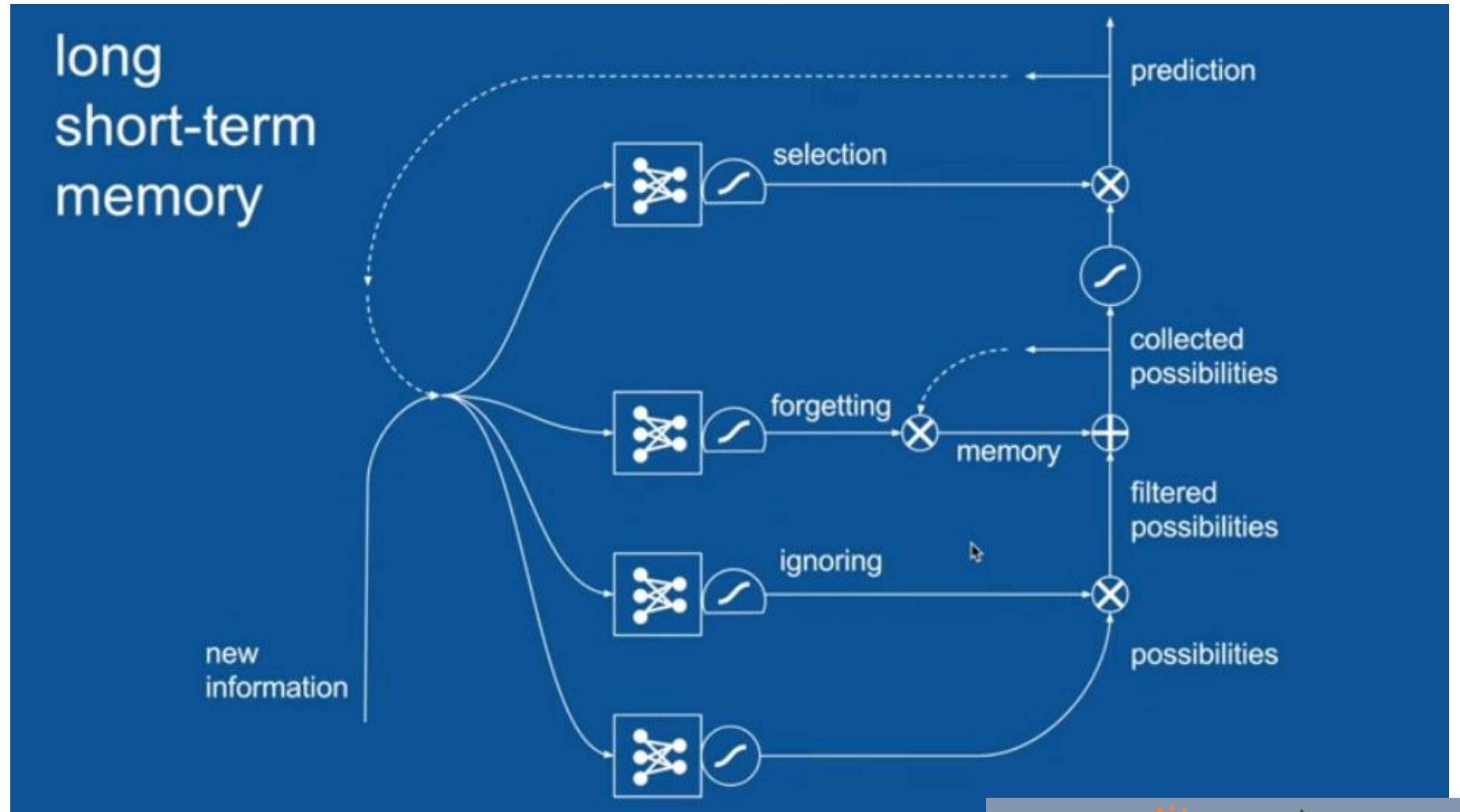
- Verschwindende und explodierende Gradienten häufig in RNNs
- Grund : schwierig, langfristige Abhängigkeiten zu erfassen, da der multiplikative Gradient mit der Anzahl der Schichten exponentiell ab- oder zunehmen kann
- Um das Problem zu lösen: Gates, die in der Regel einen genau definierten Zweck haben.



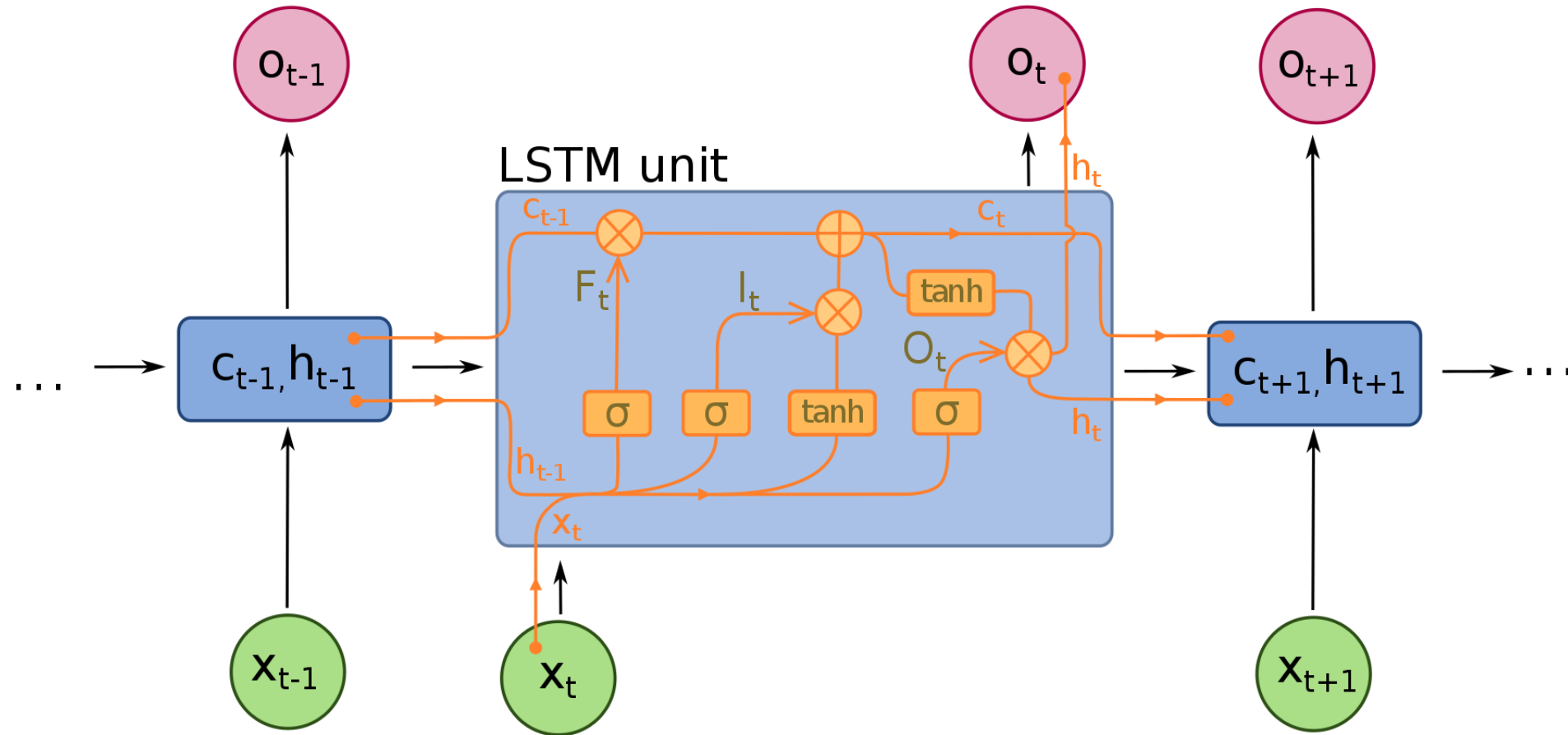
# Gates

Type of gate	Role	Used in
Update gate $\Gamma_u$	How much past should matter now?	GRU, LSTM
Relevance gate $\Gamma_r$	Drop previous information?	GRU, LSTM
Forget gate $\Gamma_f$	Erase a cell or not?	LSTM
Output gate $\Gamma_o$	How much to reveal of a cell?	LSTM

# LSTM



# LSTM



# Umsetzung

	<b>Keras</b>	<b>PyTorch</b>	<b>TensorFlow</b>
<b>Architecture</b>	Simple, concise, readable	Complex, less readable	Not easy to use
<b>Datasets</b>	Smaller datasets	Large datasets, high performance	Large datasets, high performance
<b>Debugging</b>	Simple network, so debugging is not often needed	Good debugging capabilities	Difficult to conduct debugging
<b>Does It Have Trained Models?</b>	Yes	Yes	Yes
<b>Popularity</b>	Most popular	Third most popular	Second most popular
<b>Speed</b>	Slow, low performance	Fast, high-performance	Fast, high-performance
<b>Written In</b>	Python	Lua	C++, CUDA, Python

Ab ins Jupyter  
Notebook



# Inhalte – Was haben wir gemacht?

## Tag 1

1. Einführung in Zeitreihendaten in Python
2. Zeitreihen und ihre Merkmale visualisieren
3. Zeitreihen vorhersagen (Statistik I): Exponentielle Glättung und Holt-Winters
4. Zeitreihen vorhersagen (Statistik II): ARIMA-Modelle

## Tag 2

5. Einblick in andere Zeitreihenmodelle
6. Machine Learning für Zeitreihen: Überblick, Vorbereitung und Klassifikation
7. Machine Learning für Zeitreihen: Clustering
8. Deep Learning für Zeitreihen (Einblick)



Feedback





# Vielen Dank

