

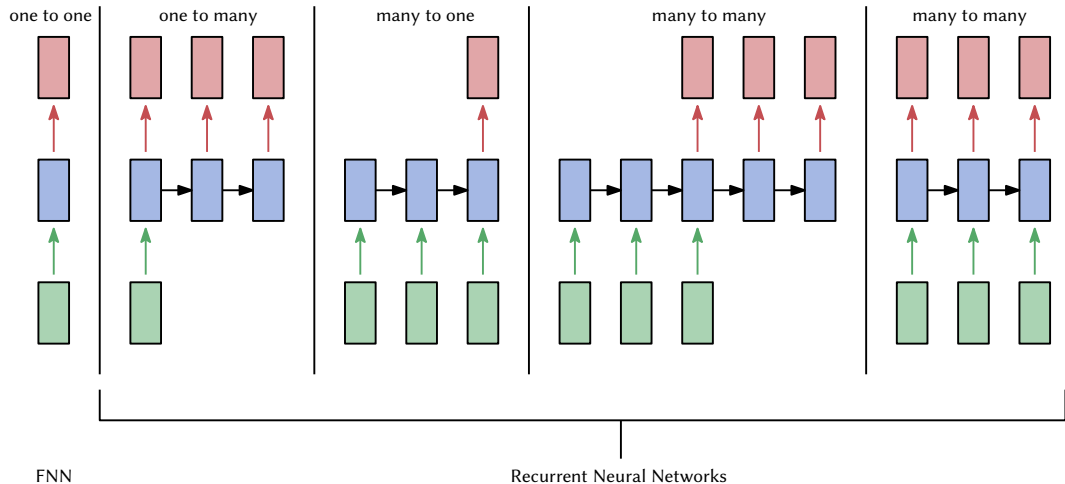
# Recurrent Neural Networks

**Benedikt Zönnchen**

22. Februar 2023

Motivation

# Motivation

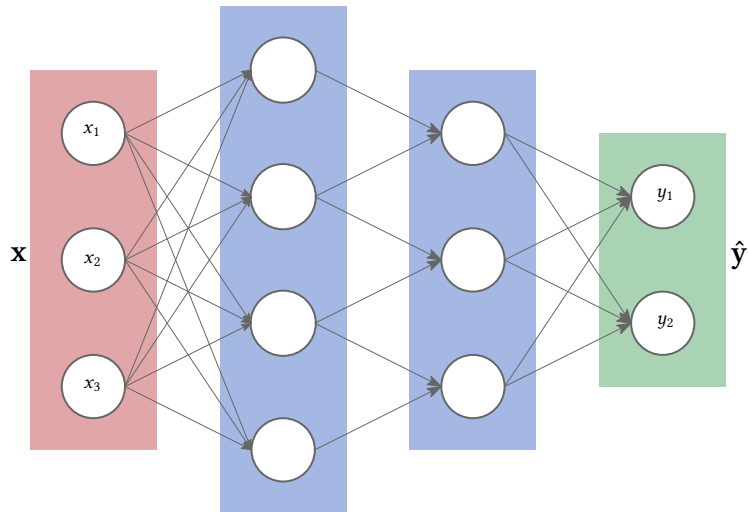


## Von Feedforward zu Recurrent

# Von Feedforward zu Recurrent

Die Eingabe durchläuft im Falle von **Feedforward Neuronale Networks** das Netz von vorne nach hinten ohne Zyklus. Das Netz bildet einen gerichteten azyklischen Graphen.

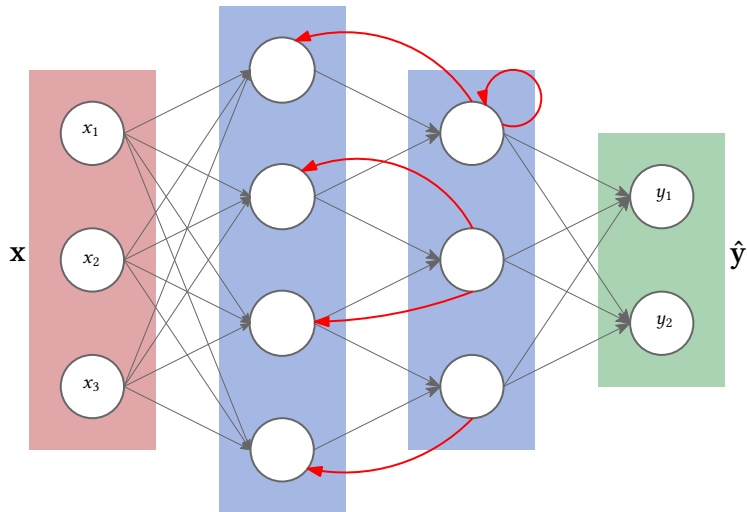
# Von Feedforward zu Recurrent



# Von Feedforward zu Recurrent

Führen wir Zyklen in das Netz ein, sprechen wir von einem sog. **Recurrent Neural Network (RNN)**.

# Von Feedforward zu Recurrent

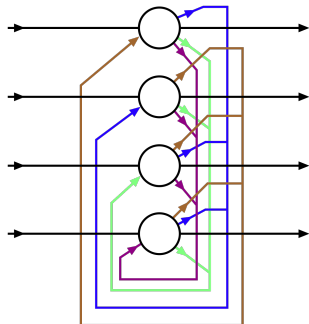




# Von Feedforward zu Recurrent

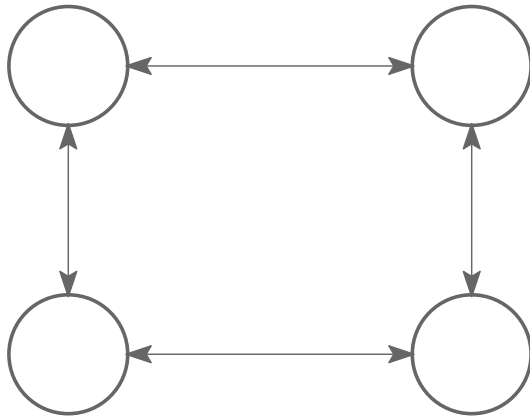
## Hopfield Networks

- Erste Einführung durch John J. Hopfield [1] im Jahr 1982
- Motiviert durch:
  - Automatische Fehlerkorrektur
  - Informationsvervollständigung
  - Dynamische Systeme mit stabilen Zuständen (Beispiel: Planetensysteme)
- Fehlerhafte oder unvollständige Information 'konvergiert' zu korrekter Information (Assoziation)



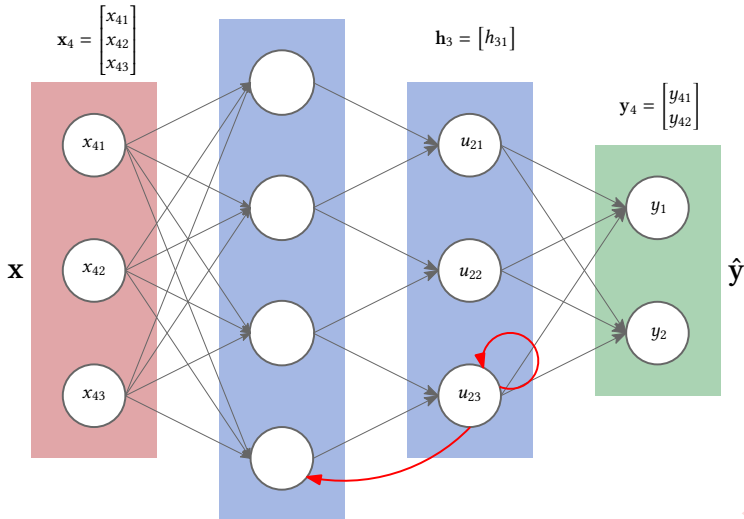
# Von Feedforward zu Recurrent

Hopfield Networks am Beispiel der Platzwahl



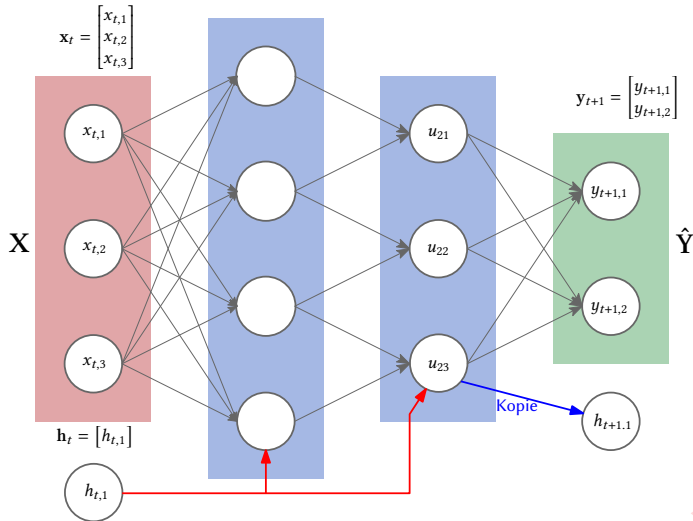
# Von Feedforward zu Recurrent

- Eingabe: Sequenz  
 $\mathbf{X} = \mathbf{x}_0, \dots, \mathbf{x}_n$
- Ausgabe: Sequenz:  
 $\mathbf{Y} = \mathbf{y}_0, \dots, \mathbf{y}_n$
- Hidden States:  
 $\mathbf{H} = \mathbf{h}_0, \dots, \mathbf{h}_n$

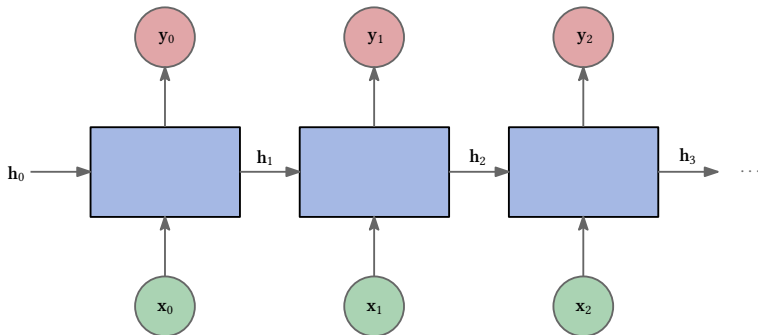
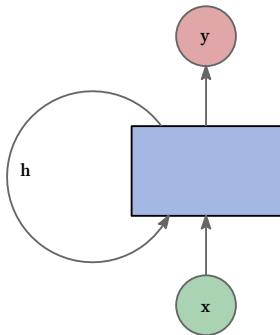


# Von Feedforward zu Recurrent

- Eingabe: Sequenz  
 $\mathbf{X} = \mathbf{x}_0, \dots, \mathbf{x}_n$
- Ausgabe: Sequenz:  
 $\mathbf{Y} = \mathbf{y}_0, \dots, \mathbf{y}_n$
- Hidden States:  
 $\mathbf{H} = \mathbf{h}_0, \dots, \mathbf{h}_n$



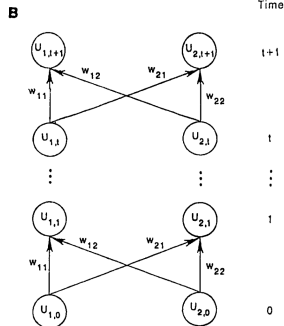
# Von Feedforward zu Recurrent



# Von Feedforward zu Recurrent

## Theorem

Für jedes *RNN* gibt es ein *Feedforward Network* mit dem gleichen Verhalten über eine endliche Zeit [3].



# Recurrent Neural Networks

## Training (Gradient Descent)

### Probleme:

- Black Box Problem
- Berechnungsaufwand (besonders beim Training)  $\Rightarrow$  hoher Ressourcenverbrauch
- Je komplexer das Netz desto mehr Daten werden fürs Training gebraucht
- Keine Garantie für Korrektheit
- Explodierende oder verschwindende Gradienten
- Information tendiert sich “auszuwaschen”
- Schlechtes Langzeitgedächtnis

# Referenzen I

- [1] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.
- [2] McLaughlin, B. P. (2008). Computationalism, connectionism, and the philosophy of mind.
- [3] Rumelhart, D. E. and McClelland, J. L. (1987). *Learning Internal Representations by Error Propagation*, pages 318–362.