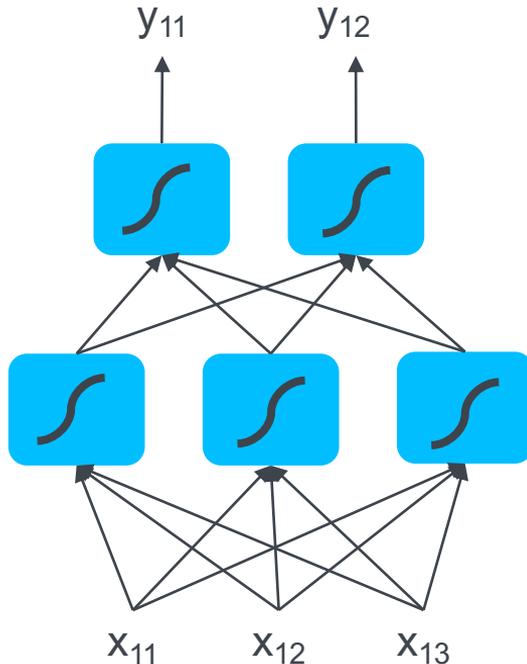


Netzwerkarchitekturen

Teil 1 – Rekurrente neuronale Netze

Einfachste Architektur: FNNs



- Alle Outputs sind mit **allen** Inputs der Schichten **darüber** verknüpft
 - D.h. **Fully connected**
 - **Feed forward neural network (FNN)**
 - Multilayer Perceptron (MLP), mehrlagiges Perzeptron
- Alle Daten gehen gleichzeitig in das Netzwerk
- Es ist vorher bekannt, wie viele Daten man bekommt (hier: Vektor mit drei Werten)

Aber:
sequentielle
Daten?

Neuronale Netze mit mehreren Schichten, bei denen Outputs nur mit Inputs der darüberliegenden Schicht verbunden sind, werden als Feed Forward Networks (FNNs) bezeichnet. Oft spricht man dann auch von mehrlagigen Perzeptronen (MLPs).

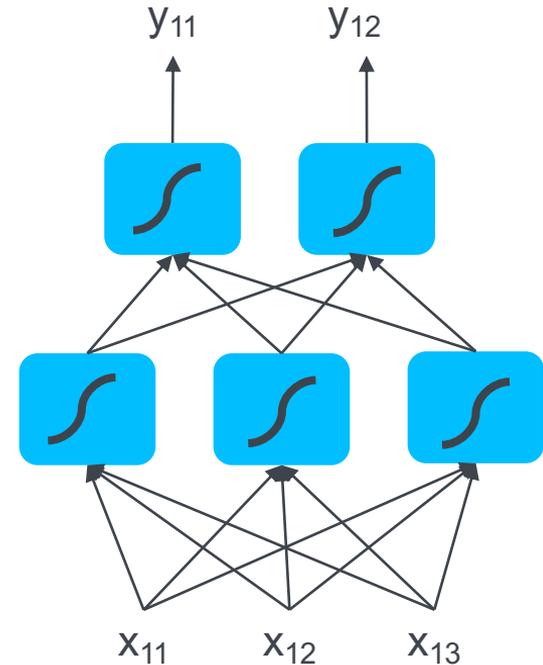
Wenn dabei alle Neuronen aufeinanderfolgender Schichten miteinander verbunden sind, spricht man von „fully connected“ FNNs.



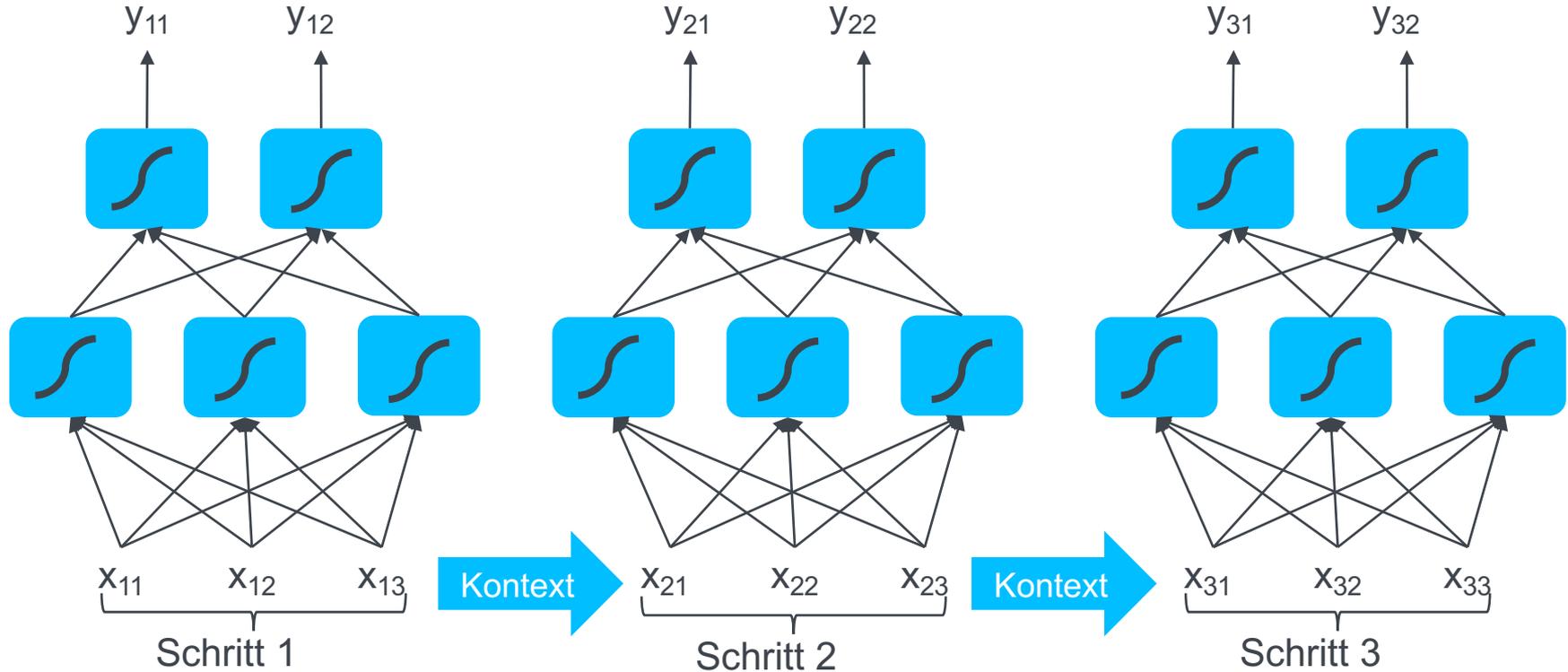
**Single Choice: FNNs -
Netzwerkarchitekturen (Teil 1)**

Sequentielle Daten

- Folgen von Sensordaten, Folgen von Wörtern in Texten, ...
- Meist ist die Reihenfolge wichtig
- Oft vorher nicht bekannt, wie viele Elemente die Folge haben wird
- Idee: Netzwerk verarbeitet in vielen Schritten je ein Element der Folge
- Element: einfache Werte oder Vektoren (hier Vektor mit drei Werten)



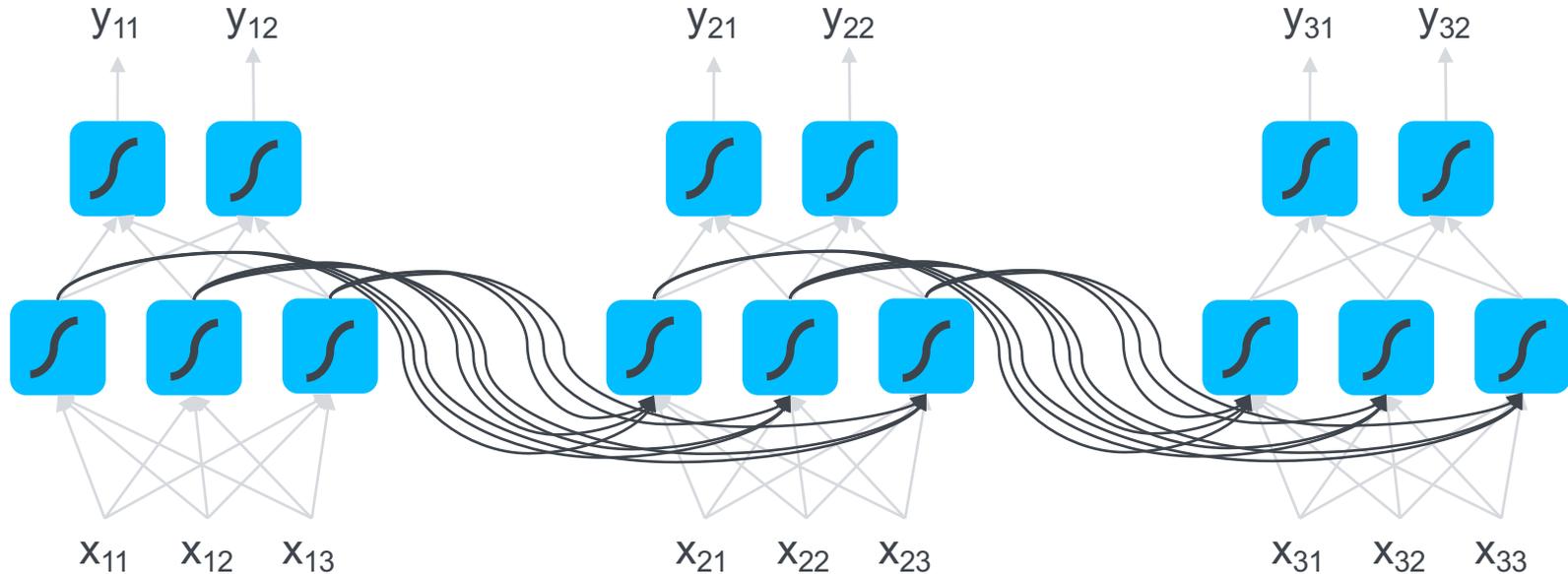
Rekurrente Neuronale Netze



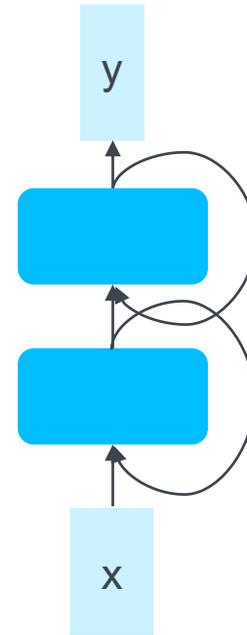
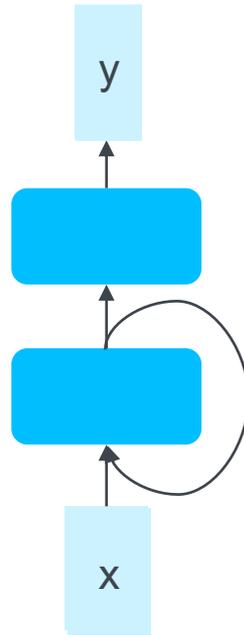
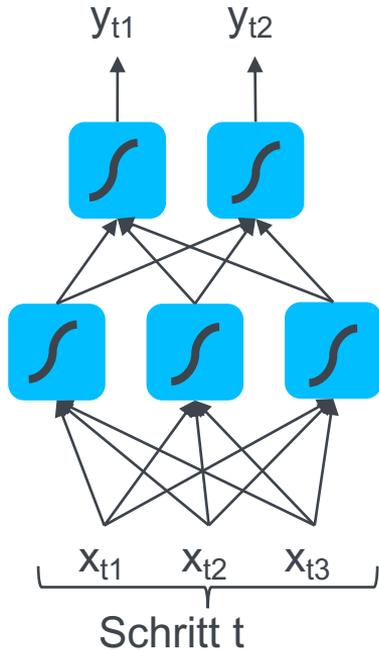
In jedem Zeitschritt
dasselbe Netzwerk
(dieselben Gewichte)

Kontext bei sequentiellen Daten

- Vorangehende (oder auch folgende) Daten liefern wichtige Kontextinformationen



Vereinfachte Darstellung für ein RNN



Rekurrente Verbindungen in mehreren Schichten erlaubt

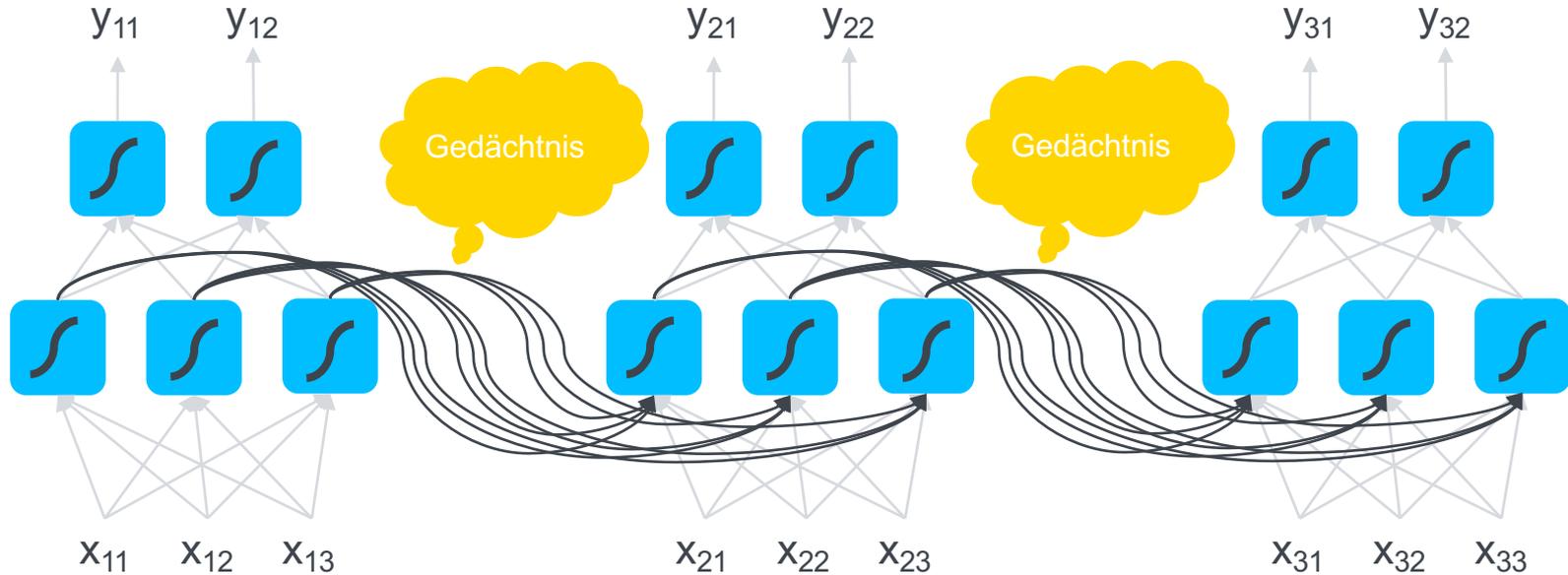
Rekurrente neuronale Netze (RNNs) sind Netzwerke, in denen Output von Neuronen einer Schicht als Input von Neuronen derselben oder einer vorangehenden Schicht erlaubt ist.

Es gibt also zirkuläre Verbindungen.

RNNs eignen sich zur Modellierung von sequentiellen Daten: in jedem Schritt bekommt das RNN ein Element der Sequenz als Input.

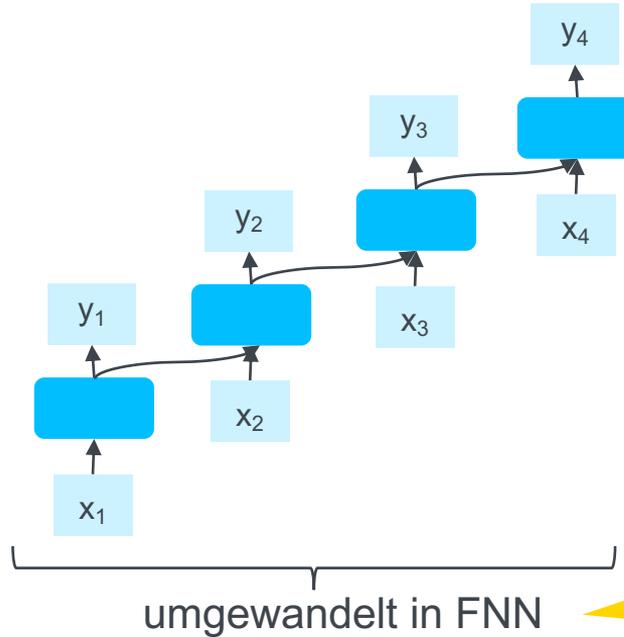
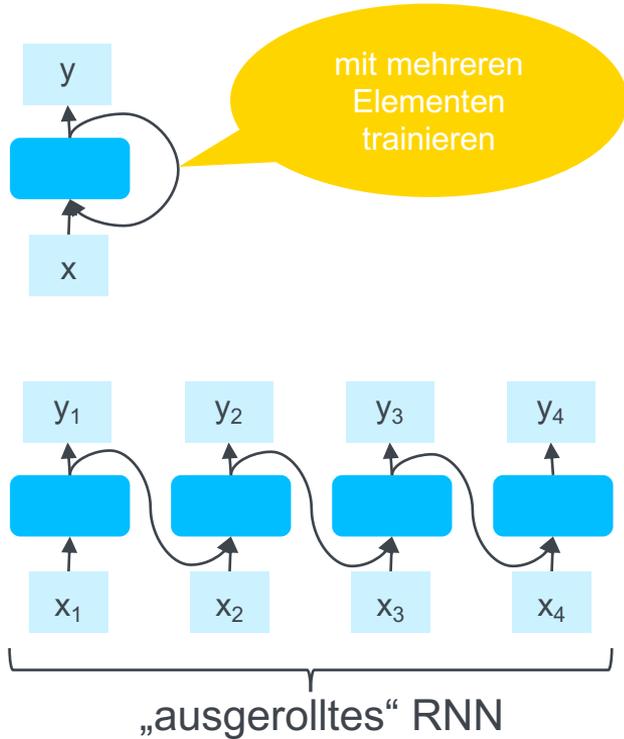
Meist nutzt man dabei den Output einer Schicht des vorangehenden Schritts als Input für dieselbe Schicht im aktuellen Schritt.

RNNs zur Modellierung eines einfachen Gedächtnisses



**RNNs erlauben die Modellierung
(sehr kurzer) zeitlicher
Abhängigkeiten, weil der Input aus
vorangehenden Schritten ein
(sehr simples) Gedächtnis simuliert.**

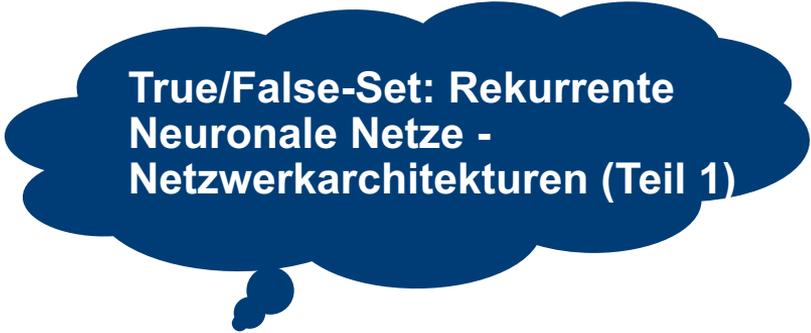
Training von RNNs



- Training mithilfe von „Back Propagation durch die Zeit“
- Gleiche Gewichte werden dabei gleich angepasst

kann sehr tiefe FNNs ergeben!

**RNNs werden beim Training
ausgerollt. Dadurch ergeben sich
potentiell sehr tiefe Netzwerke. Aus
diesem Grund werden meist RNNs
verwendet, die im nicht ausgerollten
Zustand eher flach sind.**



**True/False-Set: Rekurrente
Neuronale Netze -
Netzwerkarchitekturen (Teil 1)**

Dr. Antje Schweitzer

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung



Universität Stuttgart

Institut für Maschinelle Sprachverarbeitung
Institut für Software Engineering



IHK Industrie- und Handelskammer
Reutlingen

Reutlingen | Tübingen | Zollernalb



IHK Region Stuttgart



IHK Industrie- und Handelskammer
Karlsruhe



LMU LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Lizenzbestimmungen

“Netzwerkarchitekturen – Teil 1: Rekurrente neuronale Netze“ von Antje Schweitzer, KI B³ / Uni Stuttgart

Das Werk - mit Ausnahme der folgenden Elemente:

- Logos der Verbundpartner und des Förderprogramms
- im Quellenverzeichnis aufgeführte Medien

ist lizenziert unter:

 [CC BY 4.0 \(https://creativecommons.org/licenses/by/4.0/deed.de\)](https://creativecommons.org/licenses/by/4.0/deed.de)

(Namensnennung 4.0 International)

Quellenverzeichnis

Titelfoto: [Lars Kienie \(https://unsplash.com/@larskienie\)](https://unsplash.com/@larskienie), „fibre optic cable rack“, auf [Unsplash \(https://unsplash.com/de/fotos/llxX7xnbRF8\)](https://unsplash.com/de/fotos/llxX7xnbRF8), lizenziert unter [Unsplash-Lizenz \(https://unsplash.com/license\)](https://unsplash.com/license).

Bildausschnitt verändert.