Graph Neuronale Netze

Für Textklassifikation

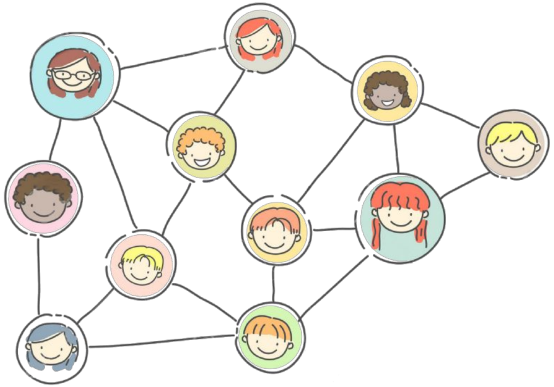
Nils Grünefeld

Inhalt

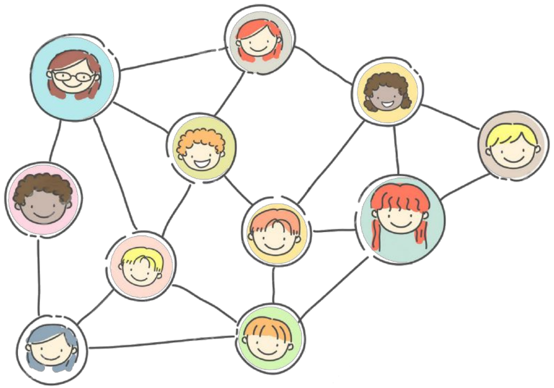
- Graphen und Neuronale Netze
- Graphen und Text
- GNNs und Text
- Results
- Fazit

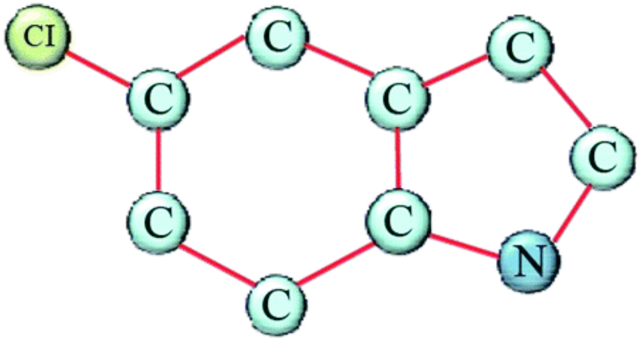
Graphen & Neuronale Netze

Graphen

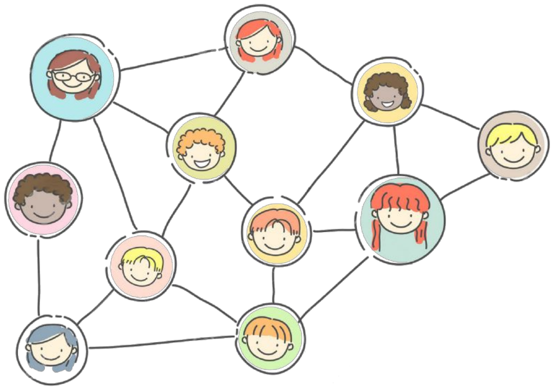


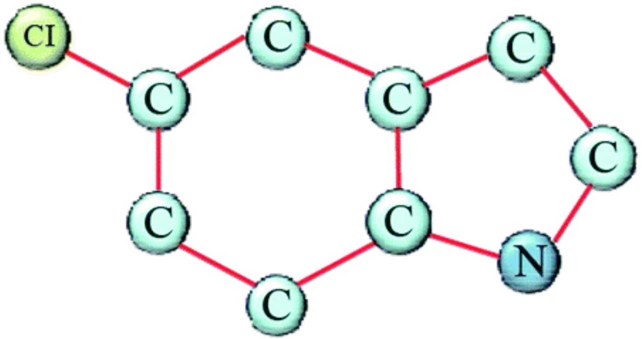
Graphen





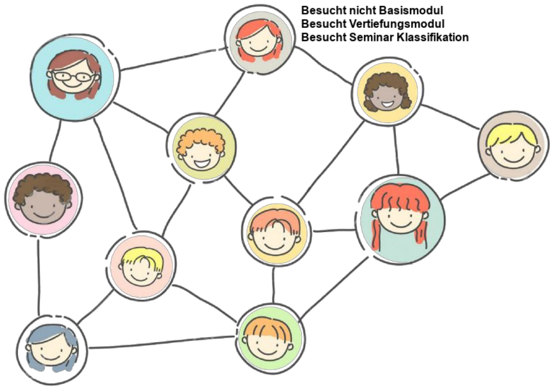
Graphen

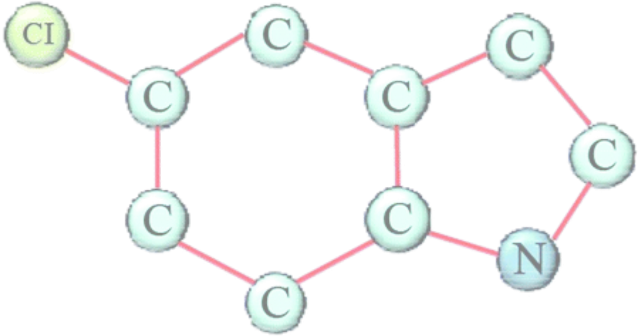




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten

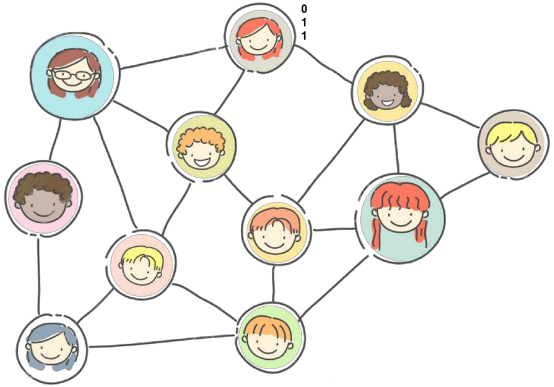
Graphen

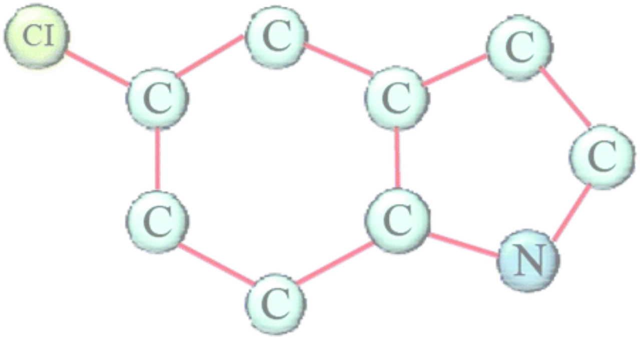




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben

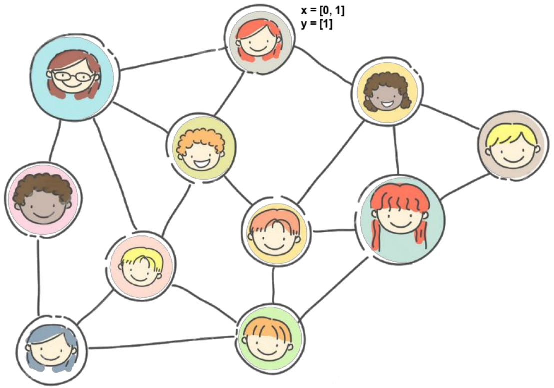
Graphen

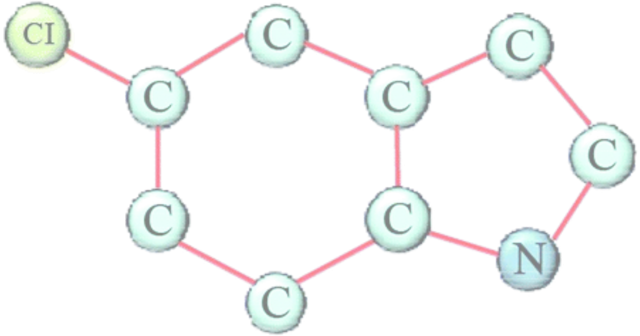




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben

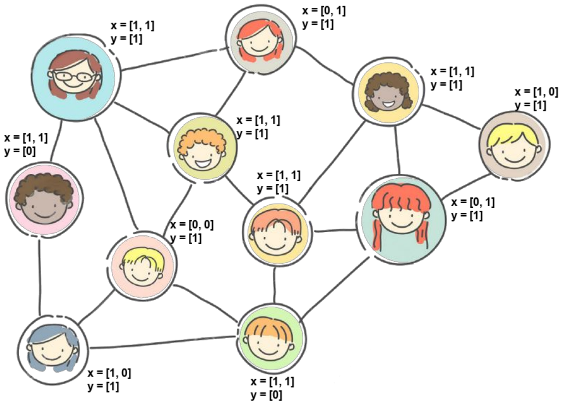
Graphen

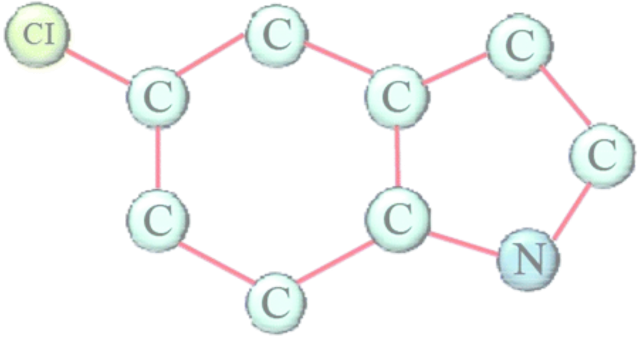




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben

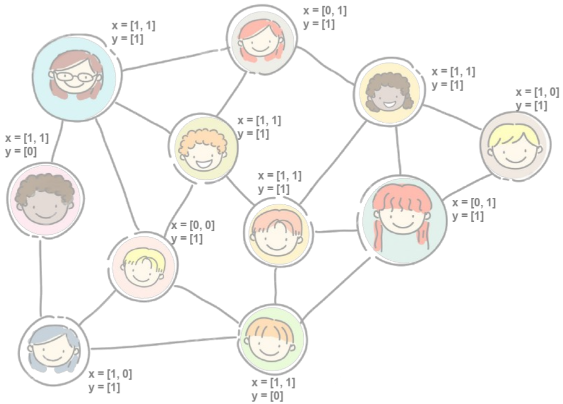
Graphen

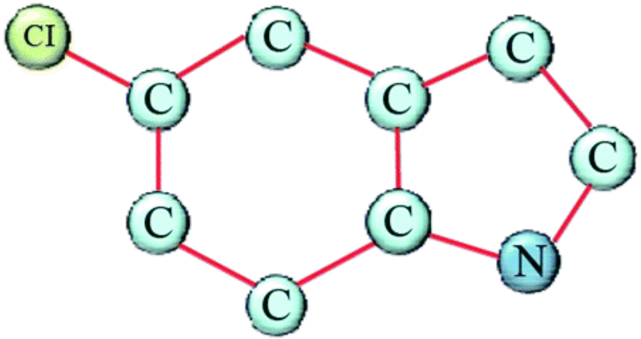




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben

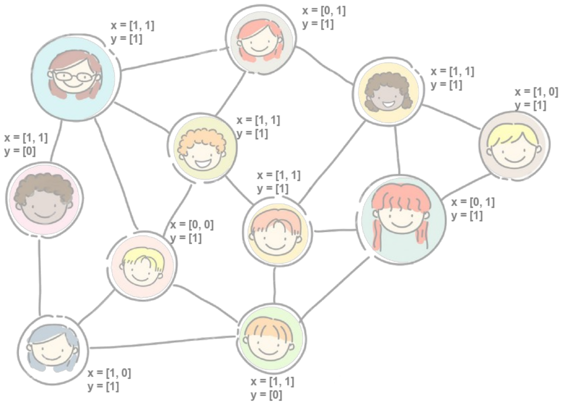
Graphen

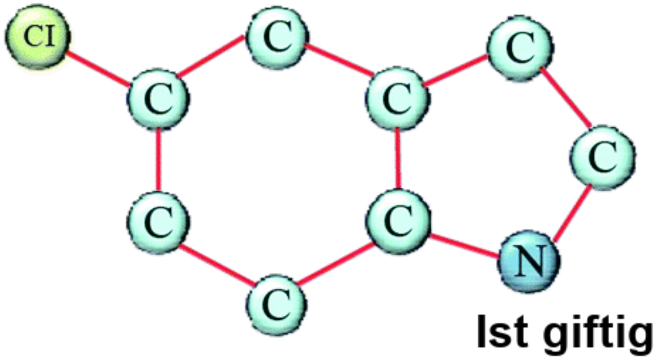




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben

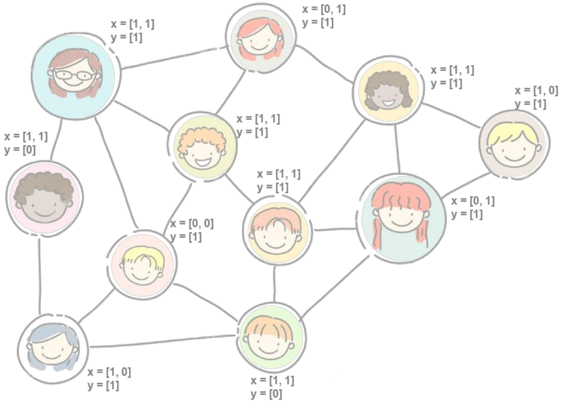
Graphen

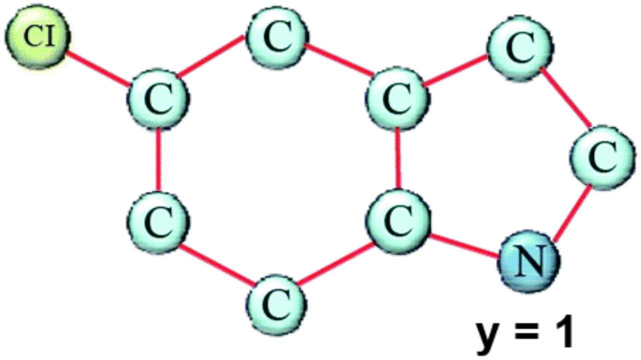




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben
- Graphen können Labels haben

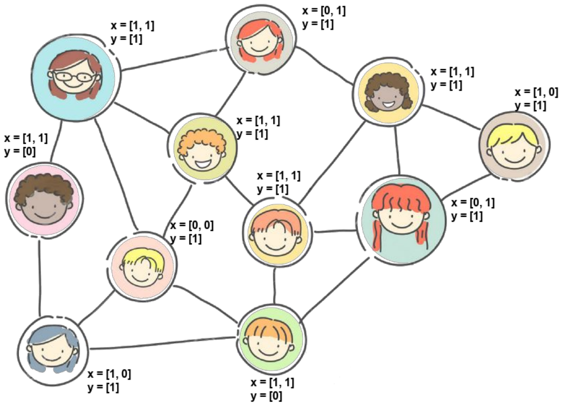
Graphen

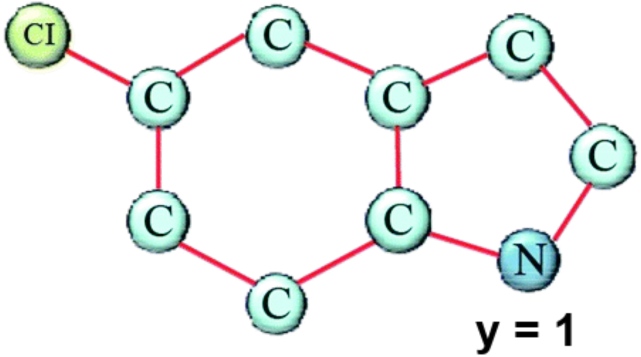




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben
- Graphen können Labels haben

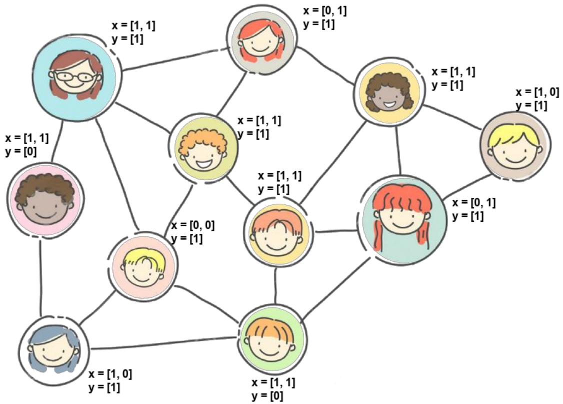
Graphen

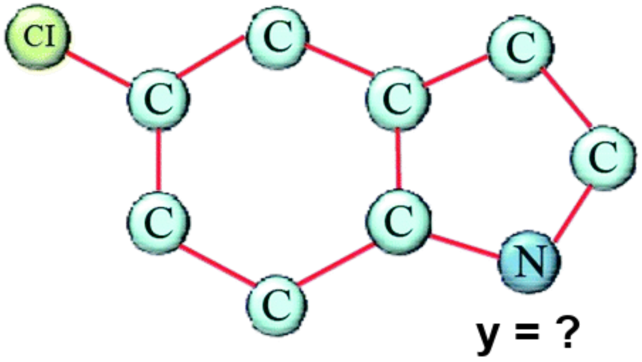




- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben
- Graphen können Labels haben

Graphen

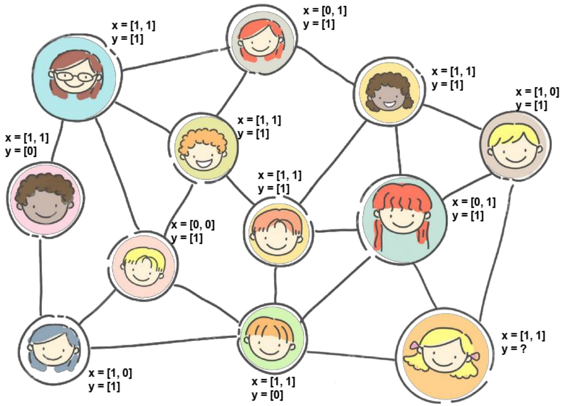




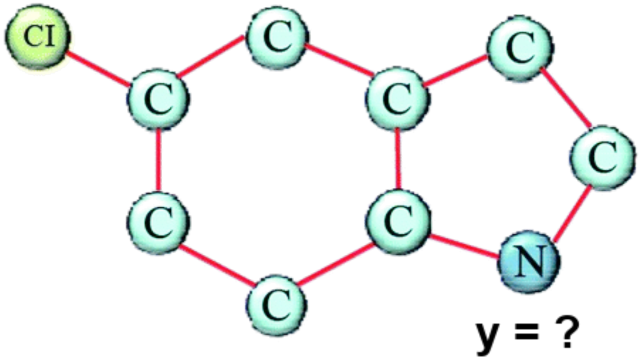
Graph Classification

- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben
- Graphen können Labels haben

Graphen



Node Classification



Graph Classification

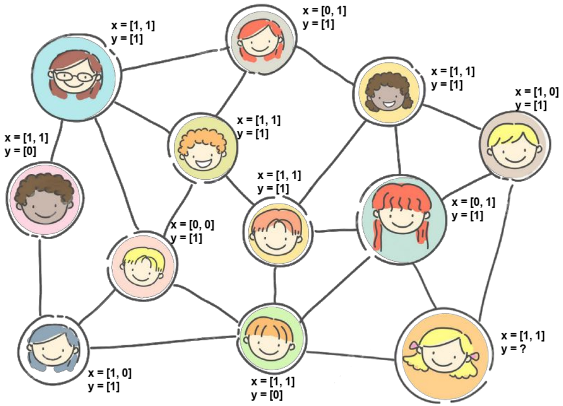
- Abstrakte Datenstruktur
- Knoten verbunden durch Kanten
- Knoten können Eigenschaften haben
- Knoten können Labels haben
- Graphen können Labels haben

Graphen

X	
1	1
0	1
1	1
0	0
1	1
1	1
1	1
1	1
1	0
1	0
0	1
1	1

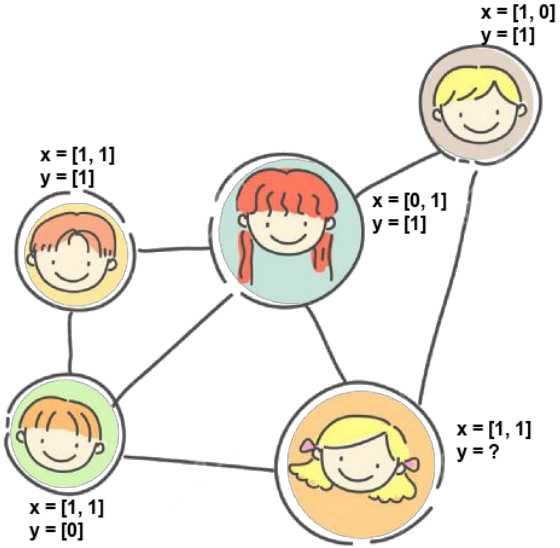
y
1
1
1
1
0
1
0
1
1
1
1
?

Adjazenzmatrix											
0	1	1	1	1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	1	1	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0	1	1	0
0	0	1	0	0	1	0	1	0	0	1	0
0	0	0	1	0	0	1	0	1	0	1	1
0	0	0	1	1	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	1	1	0	1	0	1
0	0	0	0	0	0	0	1	0	1	1	0



Graphen & NNs

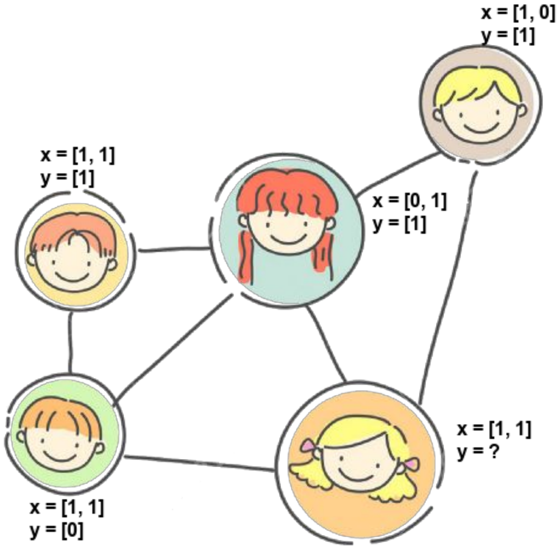
0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
1	1	0	0	1
0	1	1	1	0



Graphen & NNs

Adjazenzmatrizen sind permutationsinvariant

$$\begin{array}{|c|c|c|c|c|}\hline 0&1&0&1&0\\\hline 1&0&1&1&1\\\hline 0&1&0&0&1\\\hline 1&1&0&0&1\\\hline 0&1&1&1&0\\\hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 0&0&1&0&1\\\hline 0&0&1&1&1\\\hline 1&1&0&0&1\\\hline 0&1&0&0&1\\\hline 1&1&1&1&0\\\hline\end{array}$$



Graphen & NNs

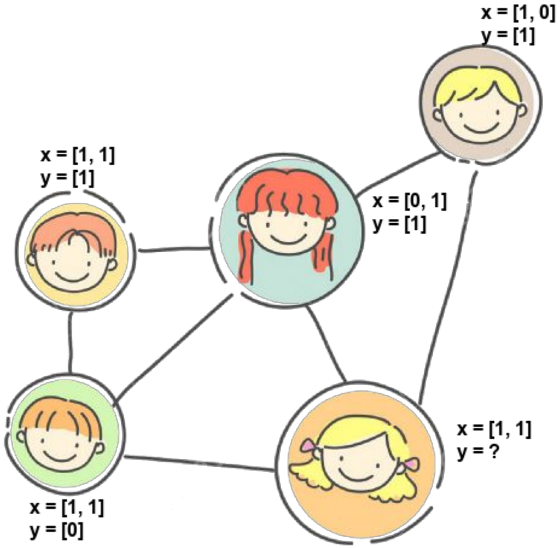
Adjazenzmatrizen sind permutationsinvariant

0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
1	1	0	0	1
0	1	1	1	0

=

0	0	1	0	1
0	0	1	1	1
1	1	0	0	1
0	1	0	0	1
1	1	1	1	0

- Variable Größen
    - Keine FFNNs



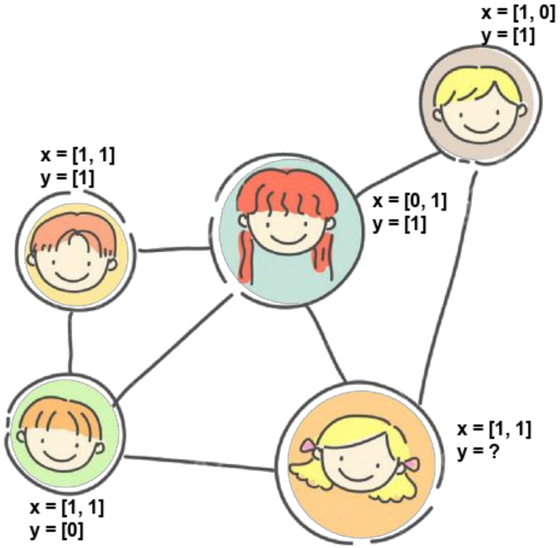
Graphen & NNs

Adjazenzmatrizen sind permutationsinvariant

0	1	0	1	0
1	0	1	1	1
0	1	0	0	1
1	1	0	0	1
0	1	1	1	0

=

0	0	1	0	1
0	0	1	1	1
1	1	0	0	1
0	1	0	0	1
1	1	1	1	0



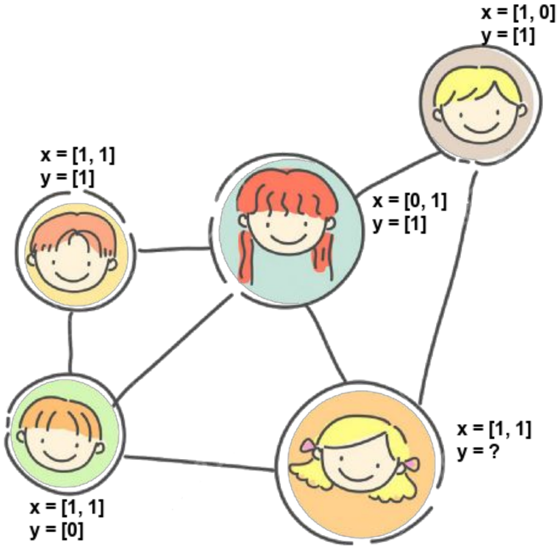
- Variable Größen
  - Keine FFNNs
- Keine Ordnung
  - Keine RNNs

Graphen & NNs

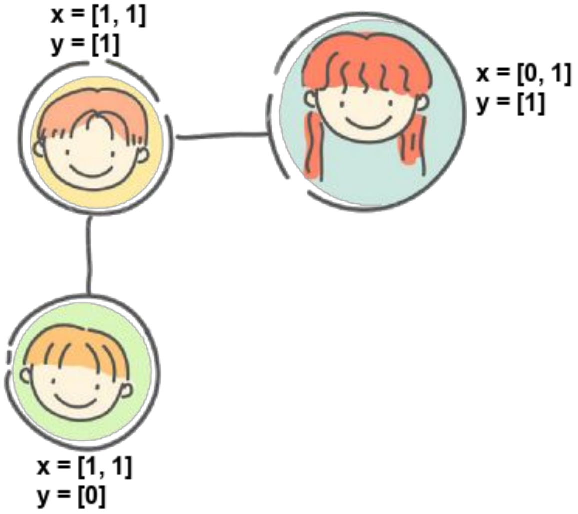
Adjazenzmatrizen sind permutationsinvariant

$$\begin{array}{|c|c|c|c|c|}\hline 0&1&0&1&0\\\hline 1&0&1&1&1\\\hline 0&1&0&0&1\\\hline 1&1&0&0&1\\\hline 0&1&1&1&0\\\hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 0&0&1&0&1\\\hline 0&0&1&1&1\\\hline 1&1&0&0&1\\\hline 0&1&0&0&1\\\hline 1&1&1&1&0\\\hline\end{array}$$

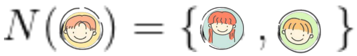
- Variable Größen
  - Keine FFNNs
- Keine Ordnung
  - Keine RNNs
- Keine Lokalität
  - Keine CNNs

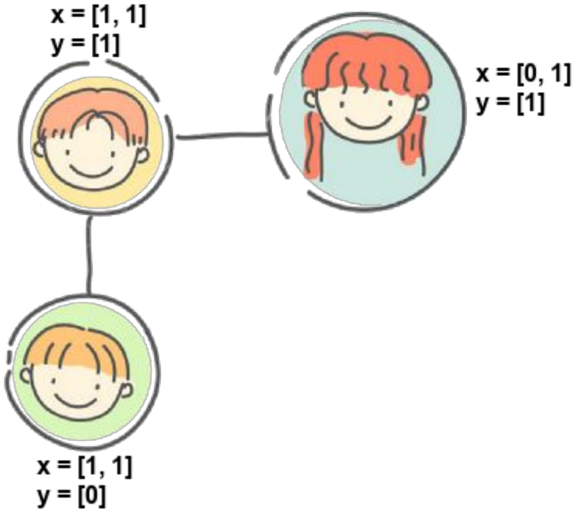


Nachbarschaft



Nachbarschaft

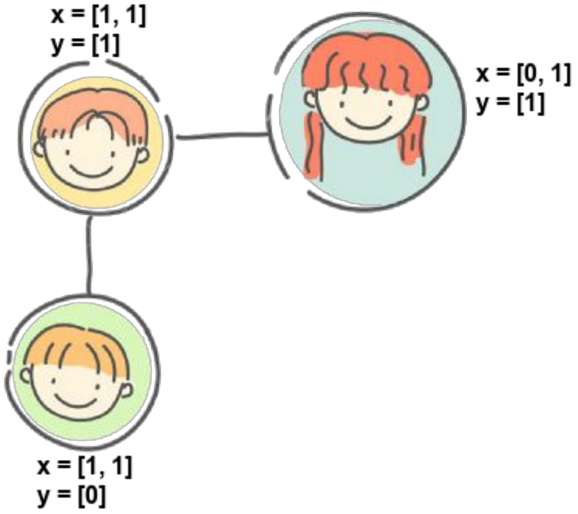




Message Passing

$$N(\;) = \{\;,\;\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

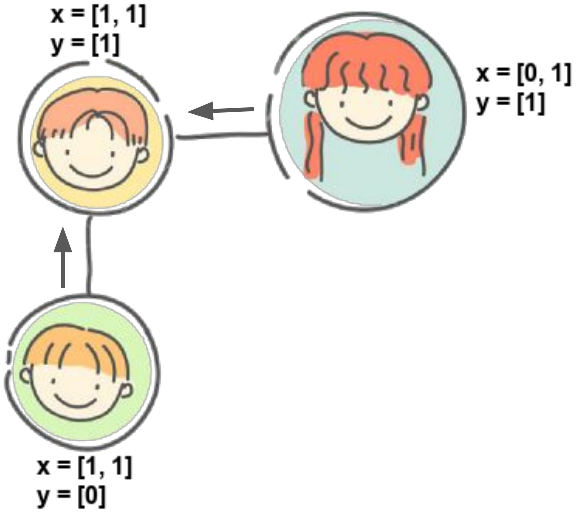


Message Passing

$$N(\;) = \{\;,\;\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x}^1 = ReLU(\mathbf{W}_1 \mathbf{x} + \mathbf{W}_2 \sum_{i \in \mathcal{N}(\;)} \mathbf{x}_i)$$



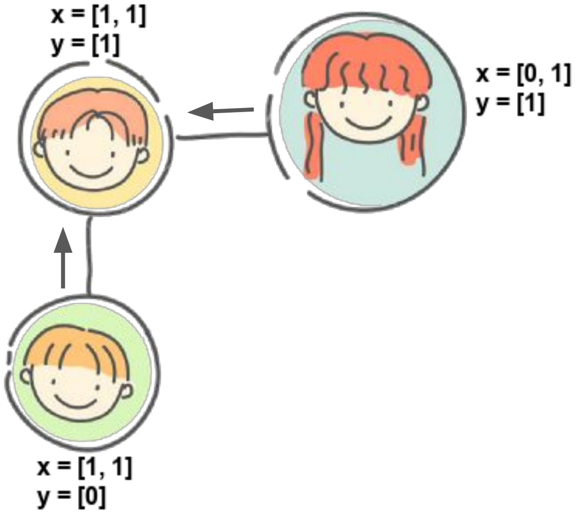
Graph Neuronale Netze

$$N(\;) = \{\;,\;\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x}^1 = ReLU(\mathbf{W}_1 \mathbf{x} + \mathbf{W}_2 \sum_{i \in N(\;)} \mathbf{x}_i)$$

$$\mathbf{x}^k = update(\mathbf{x}^{k-1}, aggregate(\{\mathbf{x}_j^{k-1} \forall j \in N(\;)\}))$$



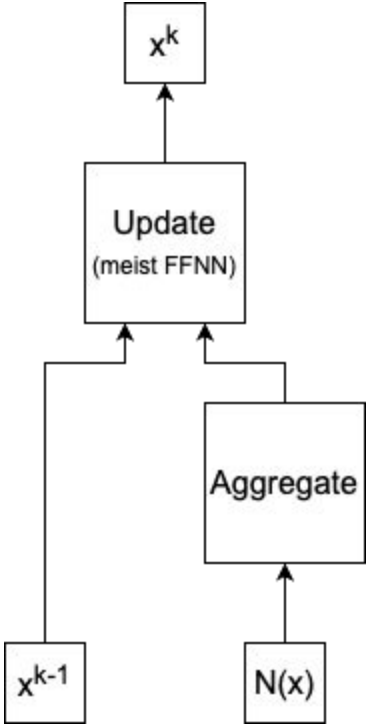
Graph Neuronale Netze

$$N(\ ) = \{\ ,\ \}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x}^1 = ReLU(\mathbf{W}_1 \mathbf{x} + \mathbf{W}_2 \sum_{i \in N(\ )} \mathbf{x}_i)$$

$$\mathbf{x}^k = update(\mathbf{x}^{k-1}, aggregate(\{\mathbf{x}_j^{k-1} \forall j \in N(\ )\}))$$



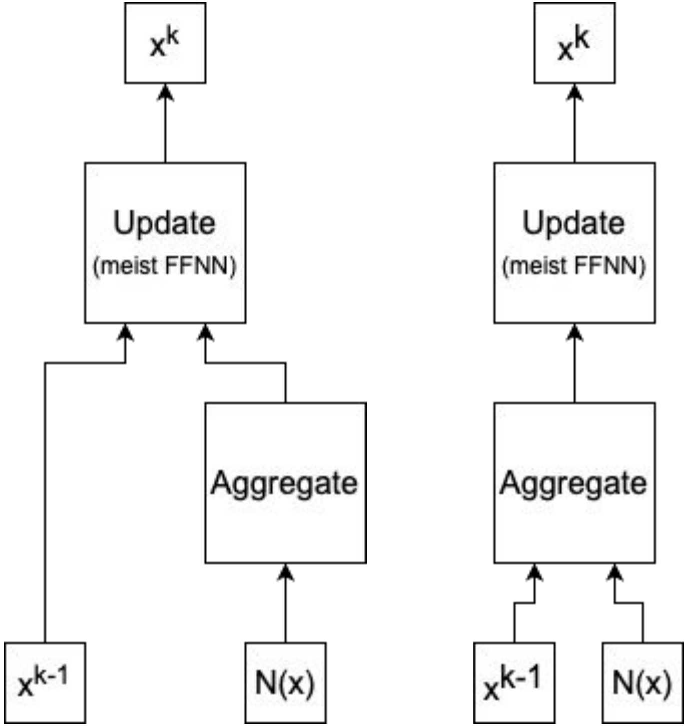
Graph Neuronale Netze

$$N(\;) = \{\;,\;\}$$

$$\mathbf{x}^0_{} = \begin{bmatrix}1\\1\end{bmatrix}$$

$$\mathbf{x}^1_{} = ReLU(\mathbf{W}_1\mathbf{x}_{} + \mathbf{W}_2 \sum_{i\in N(\;)} \mathbf{x}_i)$$

$$\mathbf{x}^k_{} = update(\mathbf{x}^{k-1}_{}, aggregate(\{\mathbf{x}^{k-1}_j \forall j \in N(\;)\}))$$
$$\mathbf{x}^k_{} = update(aggregate(\{\mathbf{x}^{k-1}_{} \cup \mathbf{x}^{k-1}_j \forall j \in N(\;)\}))$$



Graph Neuronale Netze

$$N(\;) = \{\;,\;\}$$

$$\mathbf{x}^0_{} = \begin{bmatrix}1\\1\end{bmatrix}$$

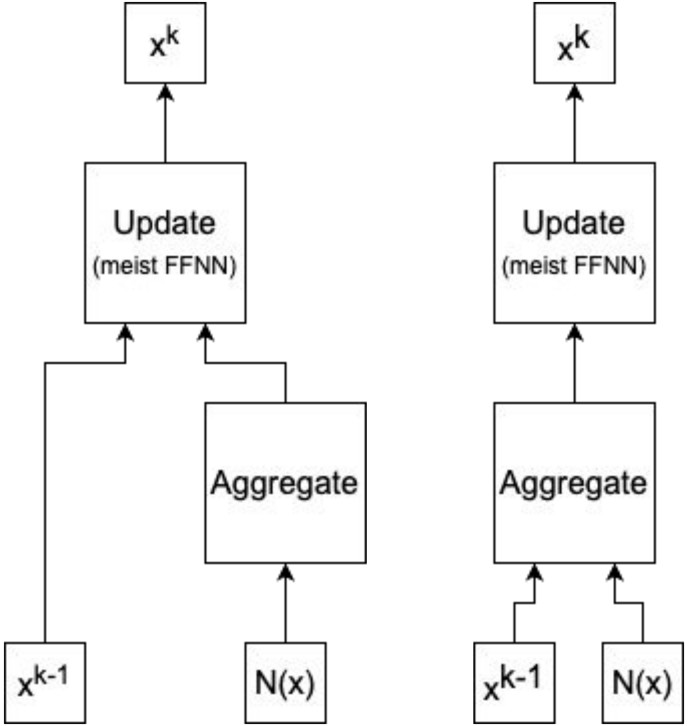
$$\mathbf{x}^1_{} = ReLU(\mathbf{W}_1\mathbf{x}_{} + \mathbf{W}_2 \sum_{i \in \mathcal{N}(\;)} \mathbf{x}_i)$$

$$\mathbf{x}^k_{} = update(\mathbf{x}^{k-1}_{}, aggregate(\{\mathbf{x}^{k-1}_j \forall j \in N(\;)\}))$$

$$\mathbf{x}^k_{} = update(aggregate(\{\mathbf{x}^{k-1}_{} \cup \mathbf{x}^{k-1}_j \forall j \in N(\;)\}))$$

$$\hat{y}_{} = softmax(\mathbf{x}_{})$$

$$\hat{y} = softmax(pool(\{\mathbf{x}_j \forall j \in G\}))$$



Message Passing

$$N() = \{ \, , \, \}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

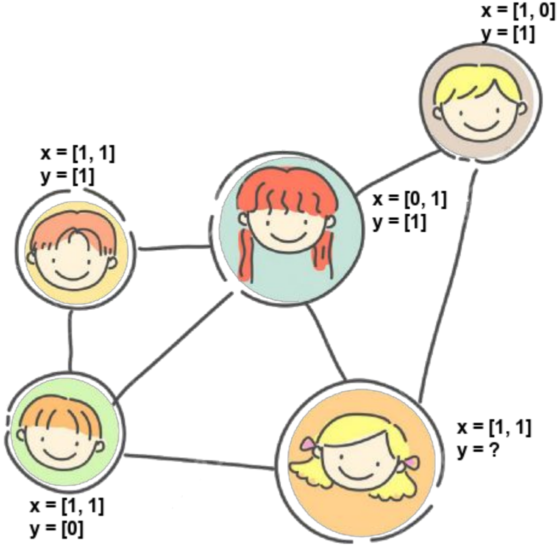
$$\mathbf{x}^1 = ReLU(\mathbf{W}_1 \mathbf{x} + \mathbf{W}_2 \sum_{i \in N()} \mathbf{x}_i)$$

$$\mathbf{x}^k = update(\mathbf{x}^{k-1}, aggregate(\{\mathbf{x}_j^{k-1} \forall j \in N()\}))$$

$$\mathbf{x}^k = update(aggregate(\{\mathbf{x}^{k-1} \cup \mathbf{x}_j^{k-1} \forall j \in N()\}))$$

$$\hat{y} = softmax(\mathbf{x})$$

$$\hat{y} = softmax(pool(\{\mathbf{x}_j \forall j \in G\}))$$



Message Passing

$$N(\ ) = \{\ ,\ \}$$

$$\mathbf{x}^0 = \begin{bmatrix}1\\1\end{bmatrix}$$

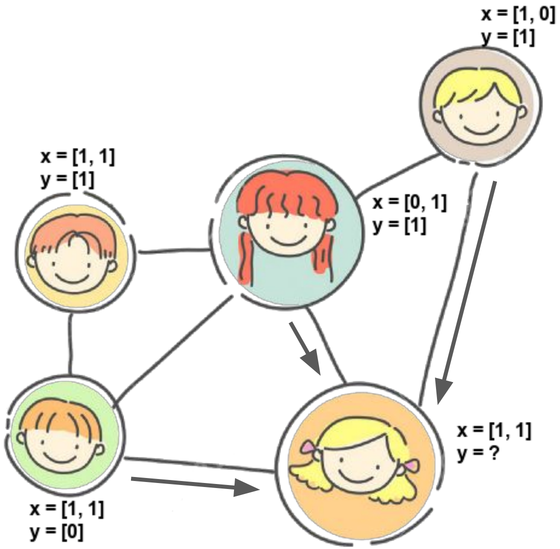
$$\mathbf{x}^1 = ReLU(\mathbf{W}_1\mathbf{x} + \mathbf{W}_2 \sum_{i \in N(\ )} \mathbf{x}_i)$$

$$\mathbf{x}^k = update(\mathbf{x}^{k-1}, aggregate(\{\mathbf{x}_j^{k-1} \forall j \in N(\ )\}))$$

$$\mathbf{x}^k = update(aggregate(\{\mathbf{x}^{k-1} \cup \mathbf{x}_j^{k-1} \forall j \in N(\ )\}))$$

$$\hat{y} = softmax(\mathbf{x})$$

$$\hat{y} = softmax(pool(\{\mathbf{x}_j \forall j \in G\}))$$



Message Passing

$$N(\;) = \{\;,\;\}$$

$$\mathbf{x}^0_{} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

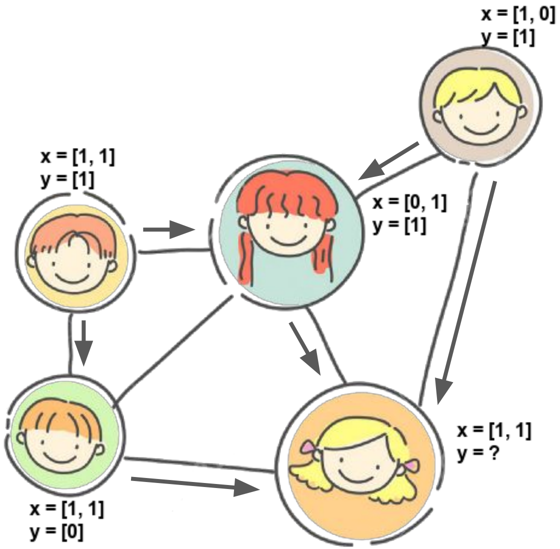
$$\mathbf{x}^1_{} = ReLU(\mathbf{W}_1 \mathbf{x}_{} + \mathbf{W}_2 \sum_{i \in \mathcal{N}(\;)} \mathbf{x}_i)$$

$$\mathbf{x}^k_{} = update(\mathbf{x}^{k-1}_{}, aggregate(\{\mathbf{x}^{k-1}_j \forall j \in N(\;)\}))$$

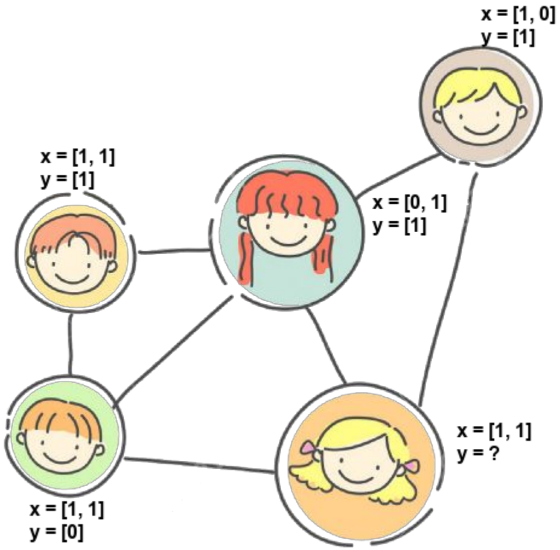
$$\mathbf{x}^k_{} = update(aggregate(\{\mathbf{x}^{k-1}_{} \cup \mathbf{x}^{k-1}_j \forall j \in N(\;)\}))$$

$$\hat{y}_{} = softmax(\mathbf{x}_{})$$

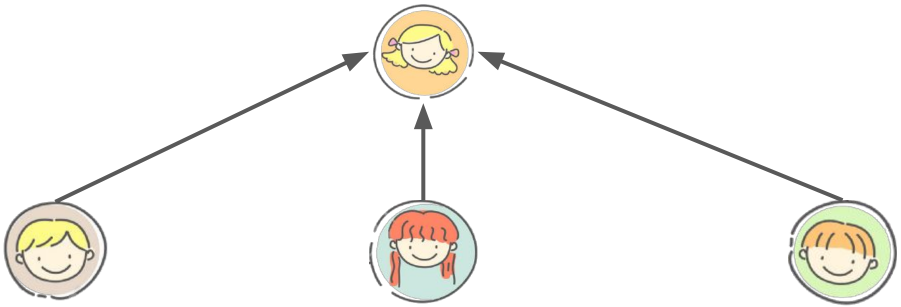
$$\hat{y} = softmax(pool(\{\mathbf{x}_j \forall j \in G\}))$$

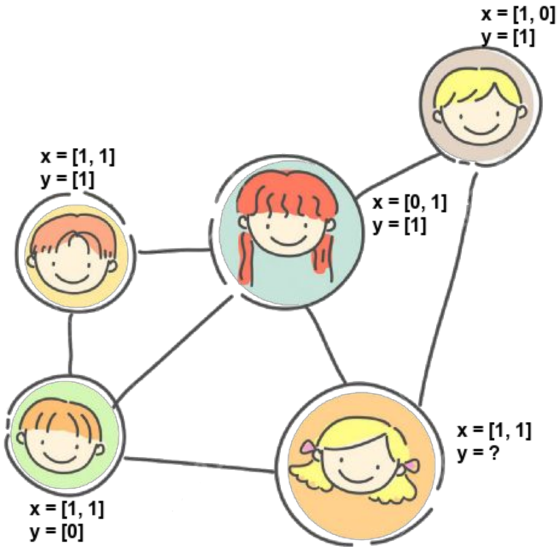


Computation Graphs

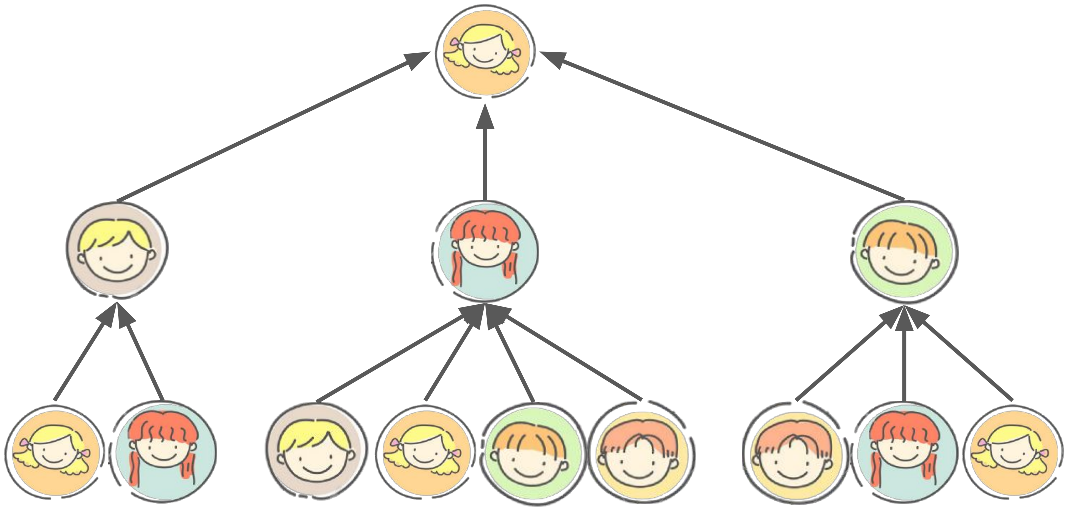


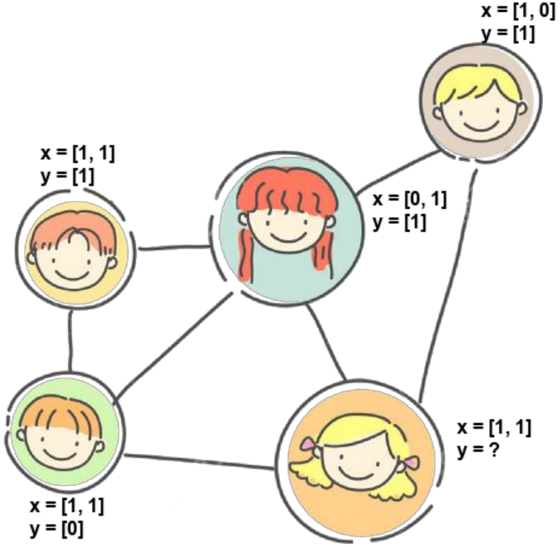
Computation Graphs





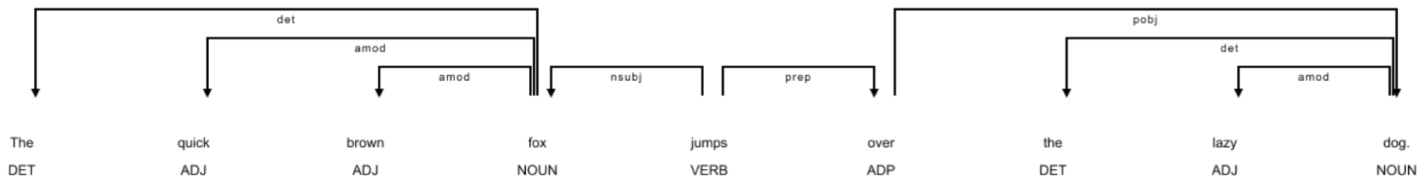
Computation Graphs



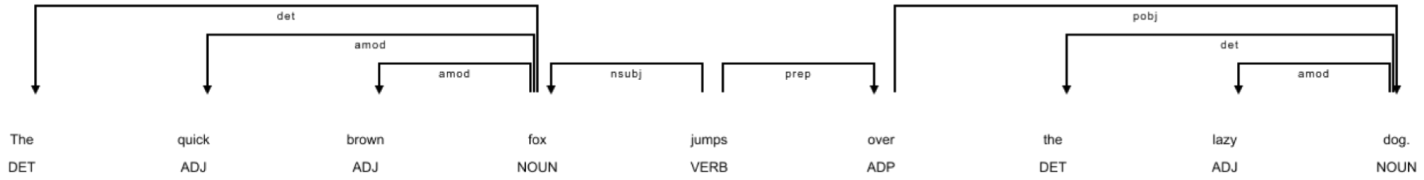


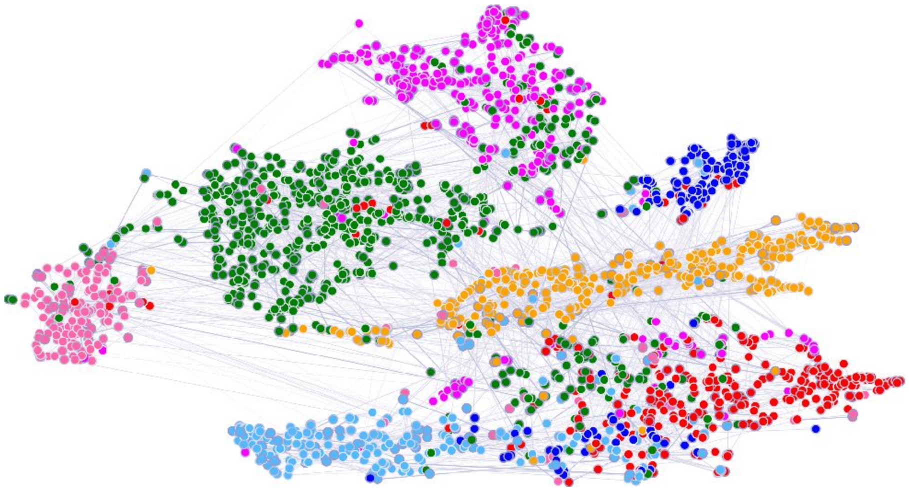
Graphen & Text

Graphen & Text



Graphen & Text

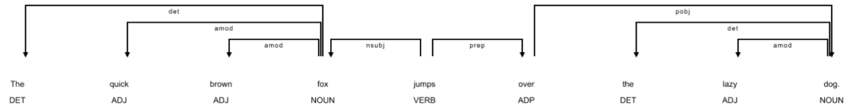




Graph Neuronale Netze & Text

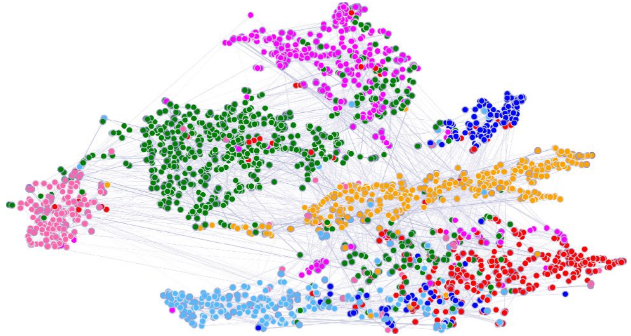
Textklassifikation durch
Graph Classification

- Texte repräsentiert als Graphen
    - Dependency Graphen
- Ein Graph pro Text
- Node Features sind Wort-Embeddings
- Downstream von Language Detection!



Textklassifikation durch
Node Classification

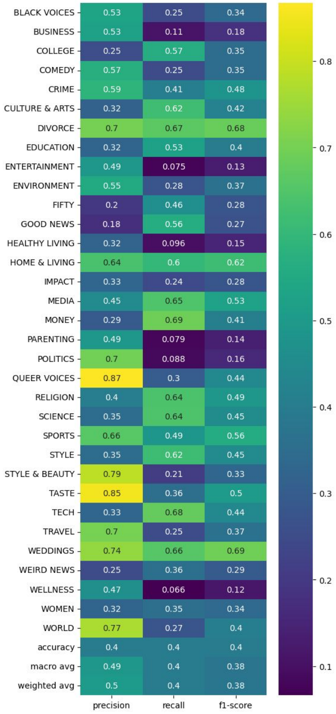
- Texte repräsentiert als Knoten
    - Zitationsnetzwerk
- Ein Graph pro Sammlung von Texten
- Node Features sind Text-Embeddings

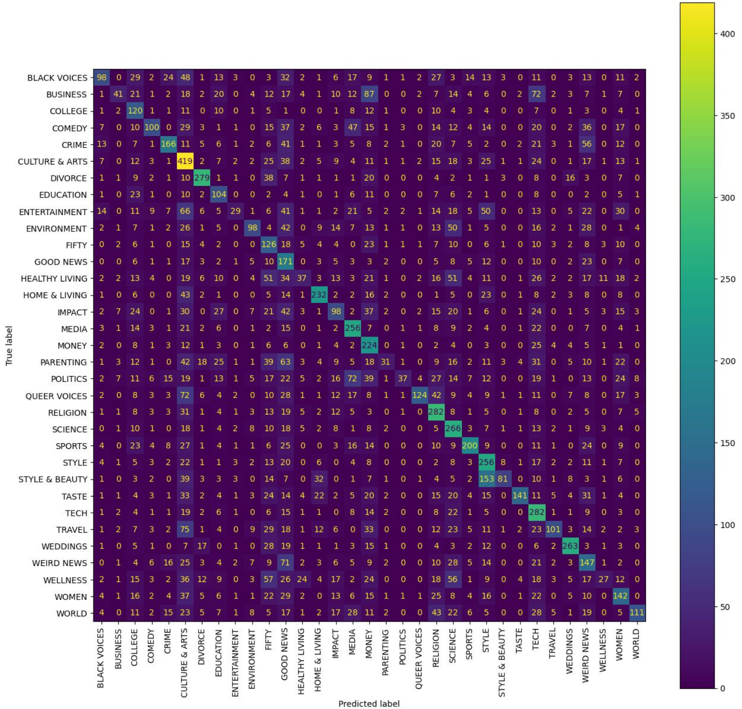


Results

News

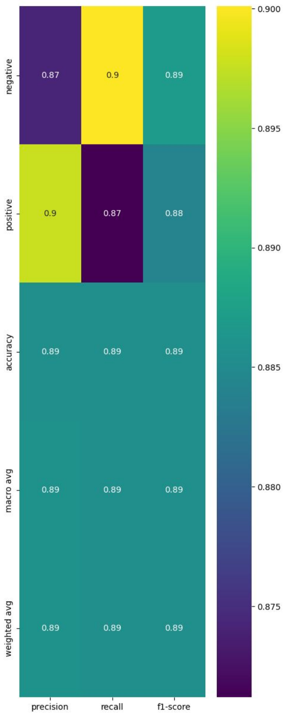
- Performance relativ schlecht
  - 40% Accuracy
- Relativ symmetrisch
  - Precision ~ Recall ~ F1 (Macro & Micro)
- Relativ breite Spanne zwischen Klassen (F1 von 0.12 bis 0.69)

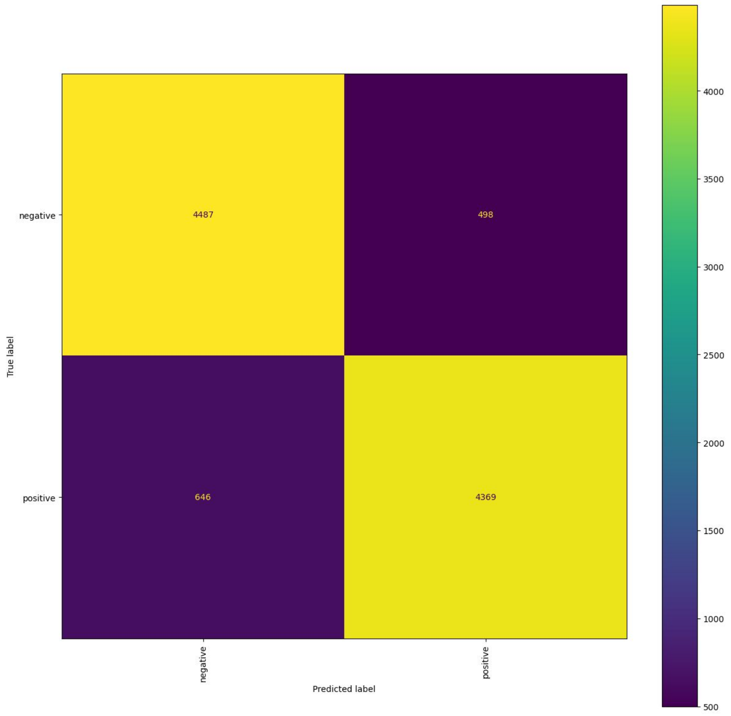




Sentiment

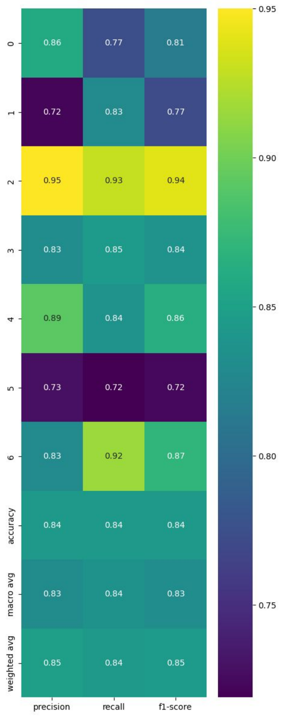
- Performance ganz gut
  - 89% Accuracy
- Relativ symmetrisch
  - Precision ~ Recall ~ F1 (Macro & Micro)
- Balanced über die Klassen

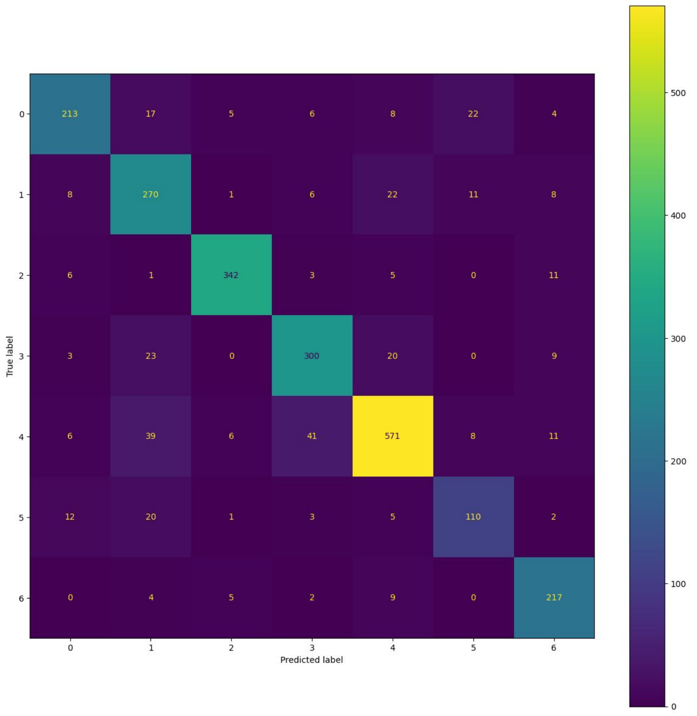




Cora

- Performance ganz gut
    - 84% Accuracy
- Relativ symmetrisch
    - Precision ~ Recall ~ F1 (Macro & Micro)
- Balanced über die Klassen





Fazit

Fazit

Sollte man GNNs für Text Klassifikation benutzen?

Fazit

Sollte man GNNs für Text Klassifikation benutzen?

In den allermeisten Fällen:

NEIN!

Fazit

Sollte man GNNs für Text Klassifikation benutzen?

In den allermeisten Fällen:

NEIN!

Wenn Beziehungen zwischen den Texten Herrschen:

Ja!

Literatur

- Hamilton 2020:

  Graph Representation Learning

- Kipf & Welling 2017:

  Semi-Supervised Classification with Graph Convolutional Networks

- Morris et al 2021:

  Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks

- PyTorch Geometric Documentation

Vielen Dank!

Fragen oder Anmerkungen?