

Einführung in Unüberwachtes Lernen mit Graph Neural Networks (GNNs)

- Ziel: Verstehen, wie GNNs aus unbeschrifteten Graphdaten lernen
- Anwendungsbereiche: Community-Erkennung, Link-Vorhersage, Embeddings von Nodes

Was ist Unüberwachtes Lernen?

- Lernen ohne bekannte Labels
- Ziel: Muster, Strukturen oder Gruppen erkennen
- Beispiele: Clustering, Dimensionsreduktion, Anomalie-Erkennung

Was ist ein Graph?

- Nodes = Objekte
- Edges = Beziehungen
- Beispiele: Soziale Netzwerke, Moleküle, KGs

Was ist ein GNN?

- Ein neuronales Netz für Graphdaten
- Lernprozess:
 - Nachrichtenaustausch zwischen Nachbarn
 - Aktualisierung der Node repräsentationen
- Funktioniert mit und ohne Labels

GNN - Struktur

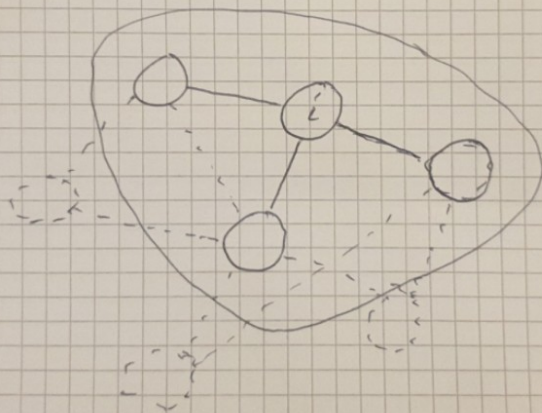
Für Node i werden die Feature Vektor h_j^{k-1} von jedem seiner Nachbarn aggregiert (SUM, MEAN, MAX) usw. und dann zusammen mit seinem eigenen Feature Vektor h_i^{k-1} in einer Aktualisierungsfunktion (UPDATE) verwendet. UPDATE kann auf unterschiedliche Weise erfolgen, ist aber häufig ein MLP. Das heißt, wir können das GNN mit Backpropagation trainieren.

* k ist die Anzahl der Sprünge (Hops) von Knoten i bei der Nachrichtenübermittlung. Beispiel: $k-2$ wären die Nachbarn j von i , die zwei Hops entfernt sind.

GNN - Struktur

Graph Isomorphism Network (GIN)

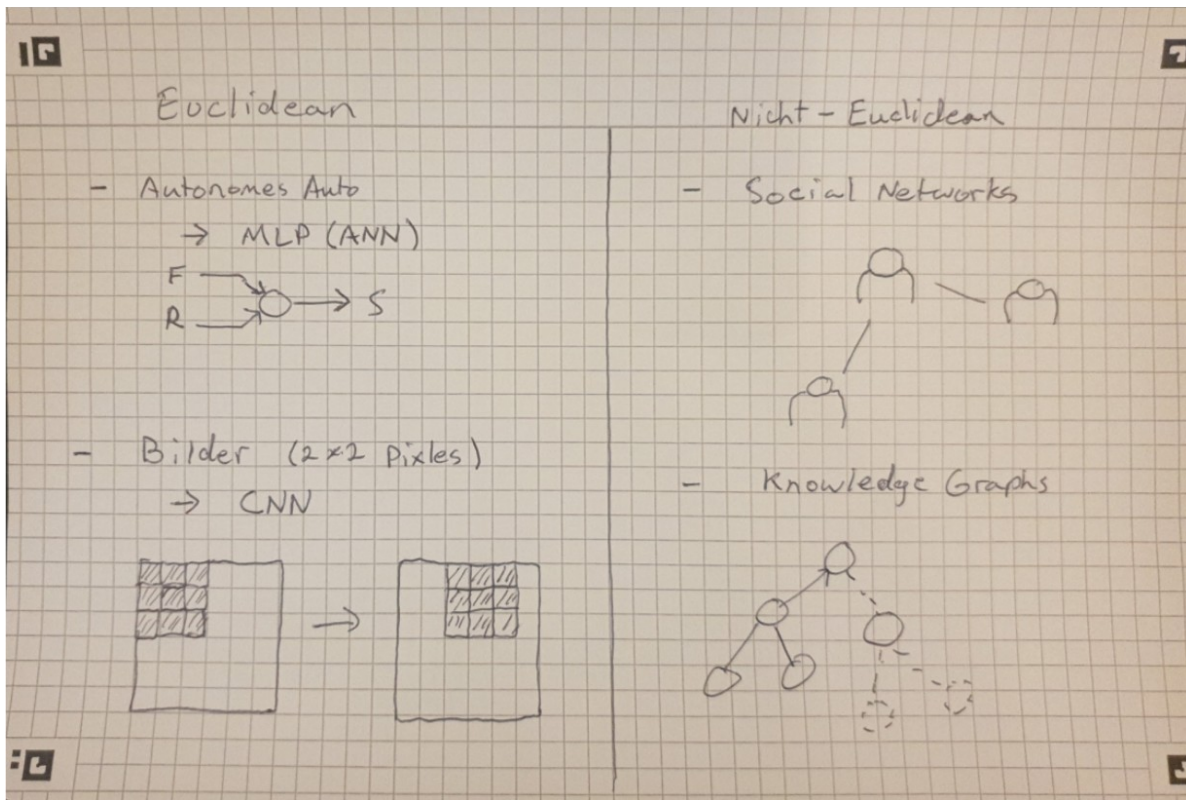
$$h_i^{(k)} = \text{UPDATE} \left(h_i^{(k-1)}, \text{AGG} \left(\{ h_j^{(k-1)} \mid j \in N(i) \} \right) \right)$$



Warum GNN?

Normale ANNs eignen sich gut zum Lernen auf euklidischen Daten, die eine einheitliche Struktur haben: Bilder, Tabellen usw. GNNs sind gut für das Lernen von non-euklidischen Daten, die eine unregelmäßige oder graphenartige Struktur haben: Wissensgraphen, soziale Netzwerke, Moleküle.

Warum GNN?



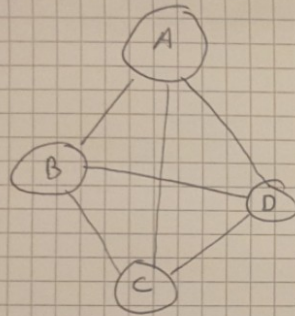
Unüberwachte Aufgaben für GNNs

- Ziel: Lerne nützliche Node-Einbettungen
- Methoden:
 - Graph Attention Networks (GAT)
 - Graph Convolutional Network (GCN)
- Anwendungen:
 - Link-Vorhersage, Anomalie-Erkennung

GAT – Link-Vorhersage

Bei zusammenhängenden Knoten A, B, C und D und Merkmalsvektoren der Dimension $d = 3$ könnten wir auf der Grundlage eines **Attention**-Mechanismus (z. B. Self-Attention) bestimmen, welche Knoten mit anderen verbunden werden sollten. Im einfachsten Fall wird ein Ähnlichkeitsmaß (z. B. Cosinus-Ähnlichkeit) zwischen den Merkmalsvektoren verwendet, und unnötige Verbindungen werden entfernt. Dann erhalten wir eine neue **Adjacency** Matrix A .

GAT – Link-Vorhersage



$$A = \begin{matrix} & A & B & C & D \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

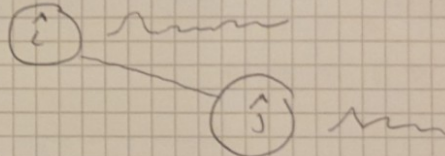
$$A \rightsquigarrow [A_1, A_2, A_3]$$

$$B \rightsquigarrow [B_1, B_2, B_3]$$

$$C \rightsquigarrow [C_1, C_2, C_3]$$

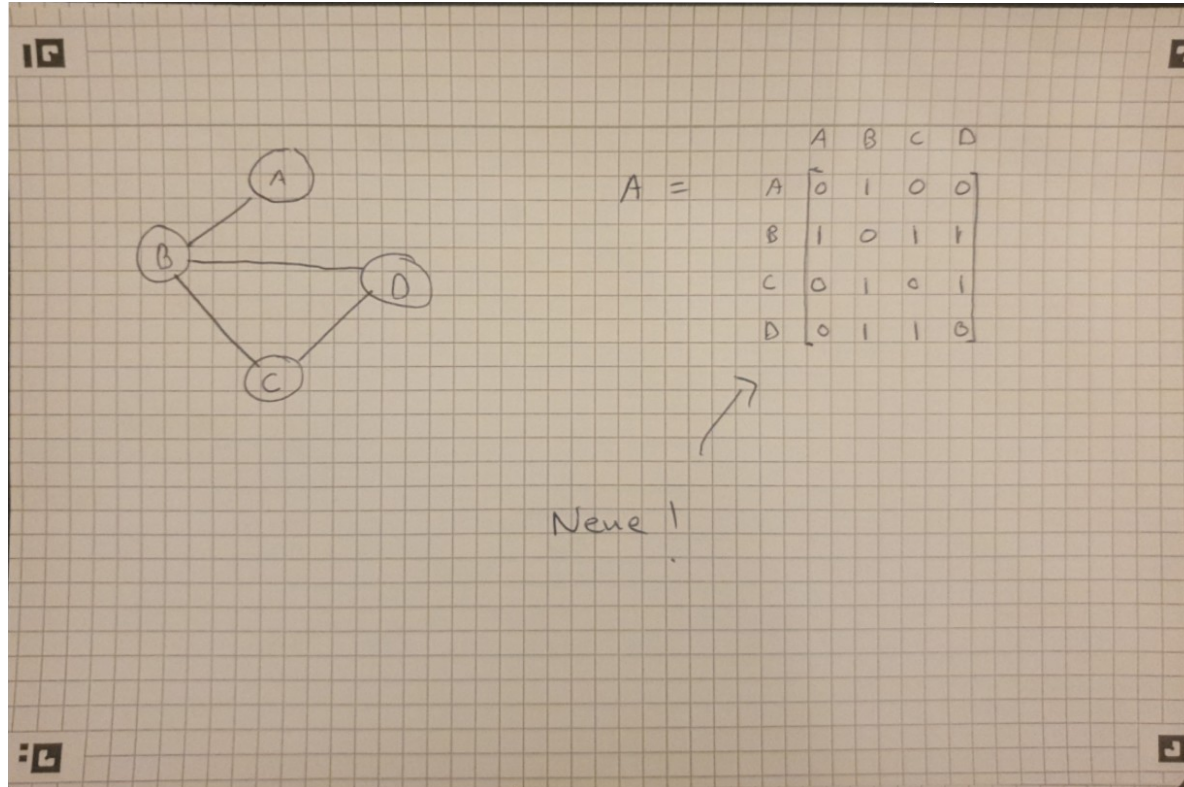
$$D \rightsquigarrow [D_1, D_2, D_3]$$

GAT – Link-Vorhersage

$$S_c(\hat{i}, \hat{j}) = e_{ij}$$

$$S_c = \frac{\hat{i} \cdot \hat{j}}{\|\hat{i}\| \|\hat{j}\|}$$

Cosine Similarity

GAT – Link-Vorhersage



GCN – Anomalie-Erkennung

Ein GCN verwendet trainierbare Gewichtungsmatrizen \mathbf{W} und eine normalisierte Adjazenzmatrix \mathbf{A} , um die Nachrichtenübermittlung über k Schichten durchzuführen, wobei jede Schicht Informationen von einem zusätzlichen Hop im Graphen enthält. Wie bei einem MLP wird nach jeder Schicht eine Aktivierungsfunktion σ (z. B. ReLU) angewendet, um Nichtlinearität einzuführen. In unserem Beispiel eines Graphen mit den Knoten A, B, C und D, jeder mit Merkmalsvektoren der Dimension $d=3$, würde ein 1-Schicht-GCN berechnen:

GCN – Anomalie-Erkennung

Handwritten mathematical derivation and matrix diagram illustrating the GCN layer computation for anomaly detection.

Dimensions:

$$\begin{aligned} A &\in \mathbb{R}^{N \times N} &\Rightarrow \mathbb{R}^{4 \times 4} \\ H^{(k)} &\in \mathbb{R}^{N \times d_k} &\Rightarrow \mathbb{R}^{4 \times 3} \\ W^{(k)} &\in \mathbb{R}^{d_k \times d_{(k+1)}} &\Rightarrow \mathbb{R}^{3 \times 3} \\ H^{(k+1)} &\in \mathbb{R}^{N \times d_{(k+1)}} &\Rightarrow \mathbb{R}^{4 \times 3} \end{aligned}$$

Equation:

$$H^{(k+1)} = \sigma(A H^k W^k)$$

Matrix Diagram:

The diagram illustrates the matrix multiplication $A H^k W^k$ using a sequence of boxes:

- A box labeled A is multiplied by a box labeled H^k to produce a box labeled $A H^k$.
- The box labeled $A H^k$ is then multiplied by a box labeled W^k to produce a box labeled $A H^k W^k$.
- The final result is shown as $H^{(k+1)} = \sigma \left[A H^k W^k \right]$, where the matrix $A H^k W^k$ is enclosed in brackets.

GCN – Anomalie-Erkennung

Im Jupyter Notebook finden Sie ein Beispiel für die Erkennung von Anomalien in unserem Wissensgraphen für Lieferketten (nächste Seite). Es kann hier gefunden werden:

https://github.com/chuckgish/Modul_AKIB10_Grundlagen_KI/blob/main/4_Unsupervised_Learning/GNN.ipynb

GCN – Anomalie-Erkennung

