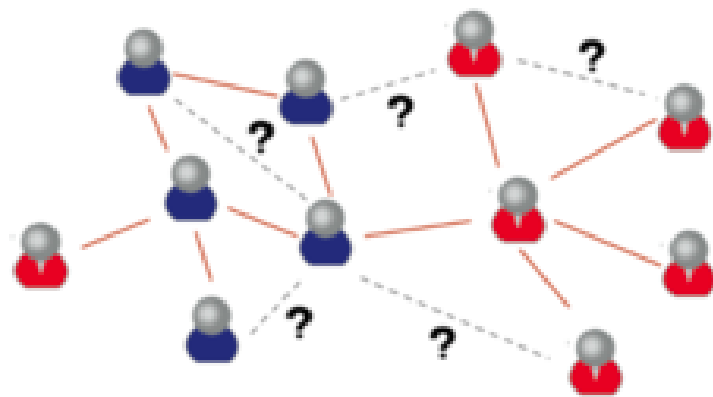
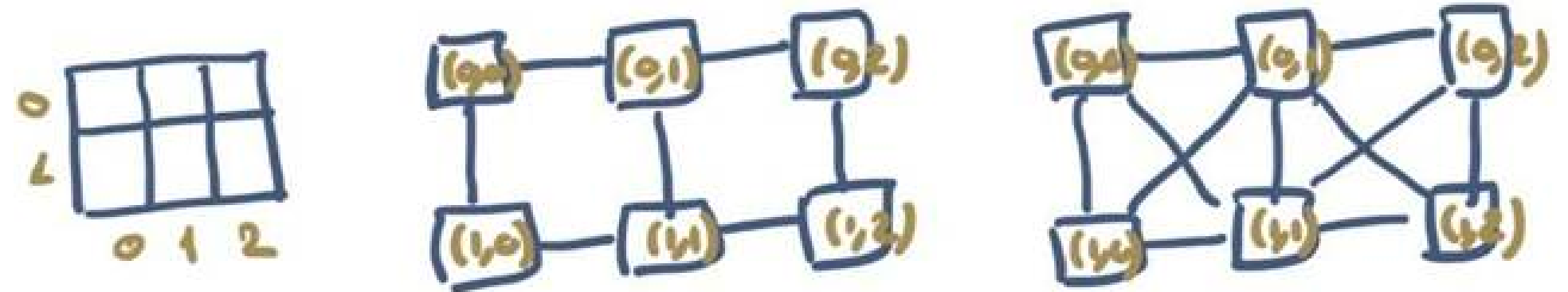


Классификация узлов сети с помощью графовой нейронной сети, использующей атрибуты узлов

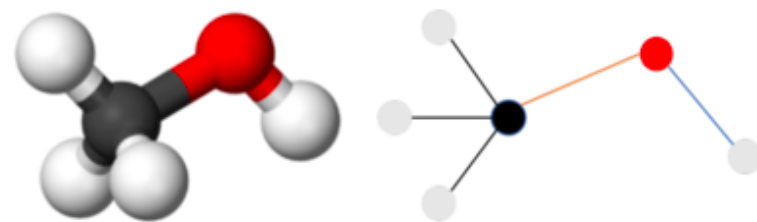
Введение



Представление социальной сети



Представление матрицы пикселей



Представление молекулы



Представление текста

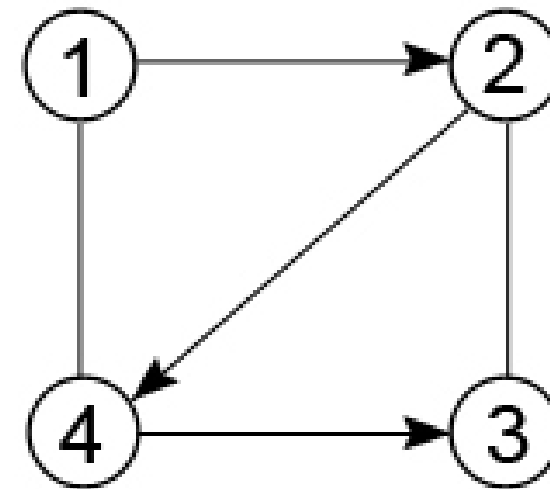
Три типа задач

- Классификация вершин
- Классификация графа
- Предсказание ребер графа

Что такое граф

Граф - структура данных, которая может быть использована для описания сложных систем. Формально граф $G = (V, E)$ определяется как множество вершин V и ребер между ними E .

	1	2	3	4
1	0	1	0	1
2	0	0	1	1
3	0	1	0	0
4	1	0	1	0



Задача классификации узлов

Задачей классификации узлов является предсказание метки объекта

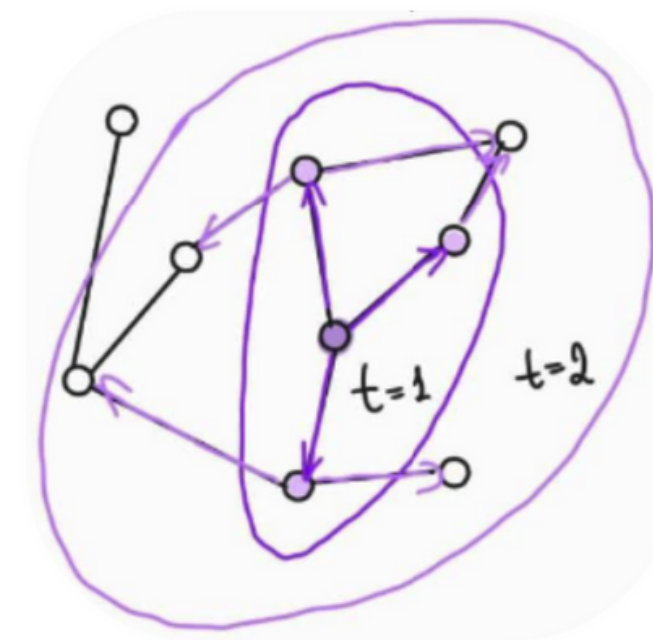
Используется идея гомофилии - вершины, которые находятся близко друг к другу обладают схожими атрибутами

Метод распространения сообщений

Формула для обновления состояния вершины:

$$\begin{aligned}\mathbf{h}_u^{(k+1)} &= \text{UPDATE}^{(k)} \left(\mathbf{h}_u^{(k)}, \text{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}) \right) \\ &= \text{UPDATE}^{(k)} \left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)} \right),\end{aligned}$$

Иллюстрация двух итерация метода распространения сообщений:



Простейшая GNN

Формула для обновления состояния вершины:

$$\mathbf{h}_u^{(k)} = \sigma \left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)} \right)$$

Эквивалентная запись через функции UPDATE и AGGREGATE:

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v, \quad \mathbf{m}_{\mathcal{N}(u)} = \text{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\})$$

$$\text{UPDATE}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) = \sigma (\mathbf{W}_{\text{self}} \mathbf{h}_u + \mathbf{W}_{\text{neigh}} \mathbf{m}_{\mathcal{N}(u)})$$

Упрощенный вариант модели с добавлением ребер из вершин в самих себя:

$$\mathbf{h}_u^{(k)} = \text{AGGREGATE}(\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u) \cup \{u\}\})$$

Нормализация и GCN

Нормирование функции агрегации по числу соседей:

$$\mathbf{m}_{\mathcal{N}(u)} = \frac{\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v}{|\mathcal{N}(u)|},$$

Симметричная нормализация:

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)| |\mathcal{N}(v)|}}.$$

Графовая сверточная сеть:

$$\mathbf{h}_u^{(k)} = \sigma \left(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)| |\mathcal{N}(v)|}} \right).$$

Обучение сети и функция потерь

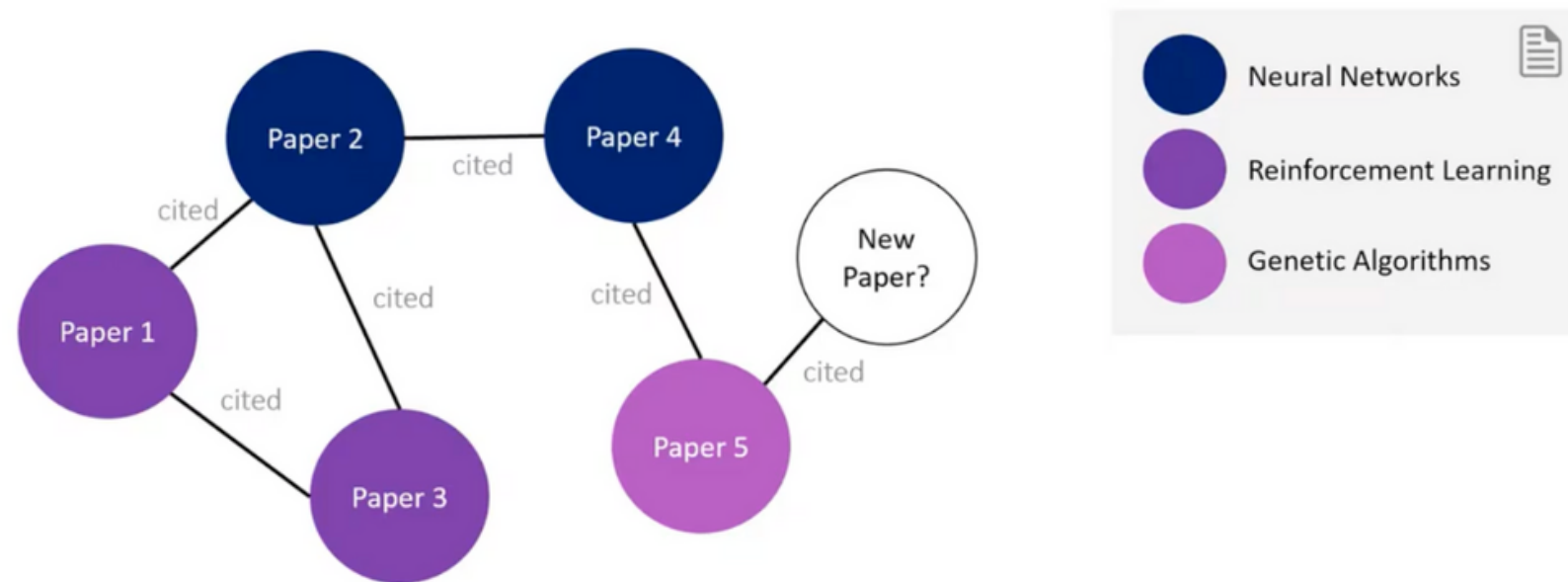
Для обучения используется логарифмическая функция потерь:

$$\mathcal{L} = \sum_{u \in \mathcal{V}_{\text{train}}} -\log(\text{softmax}(\mathbf{z}_u, \mathbf{y}_u)).$$

Функция softmax определяется следующим образом:

$$\text{softmax}(\mathbf{z}_u, \mathbf{y}_u) = \sum_{i=1}^c \mathbf{y}_u[i] \frac{e^{\mathbf{z}_u^\top \mathbf{w}_i}}{\sum_{j=1}^c e^{\mathbf{z}_u^\top \mathbf{w}_j}}$$

Обзор датасета



Небольшая иллюстрация Cora датасета

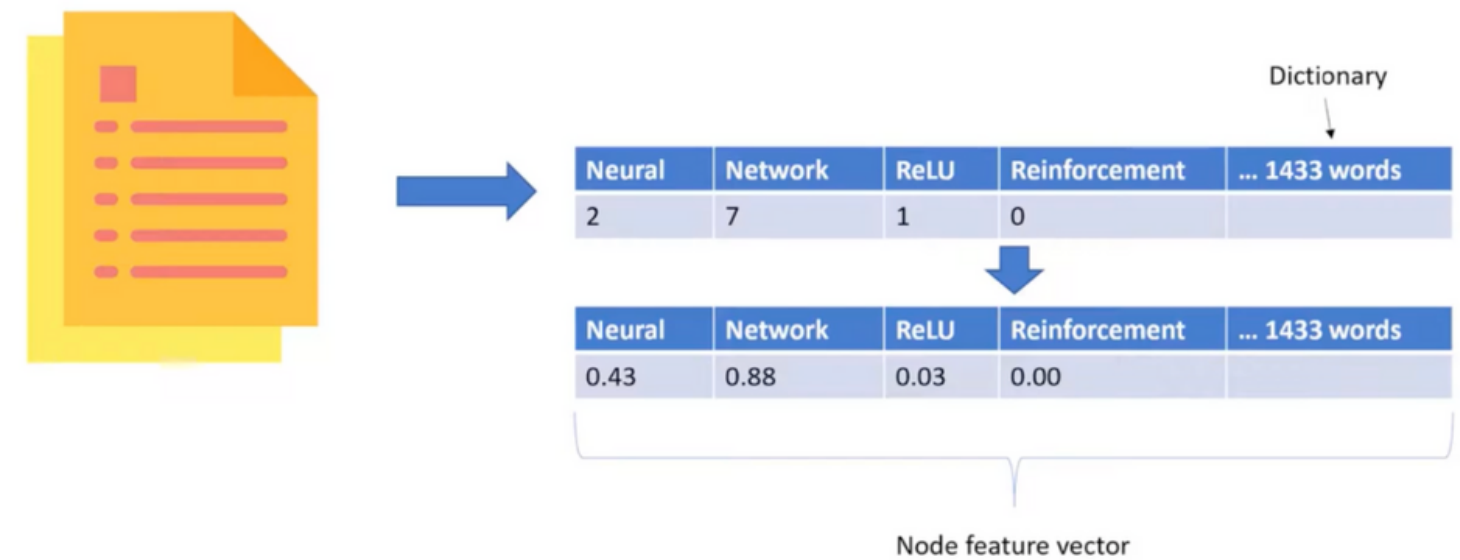
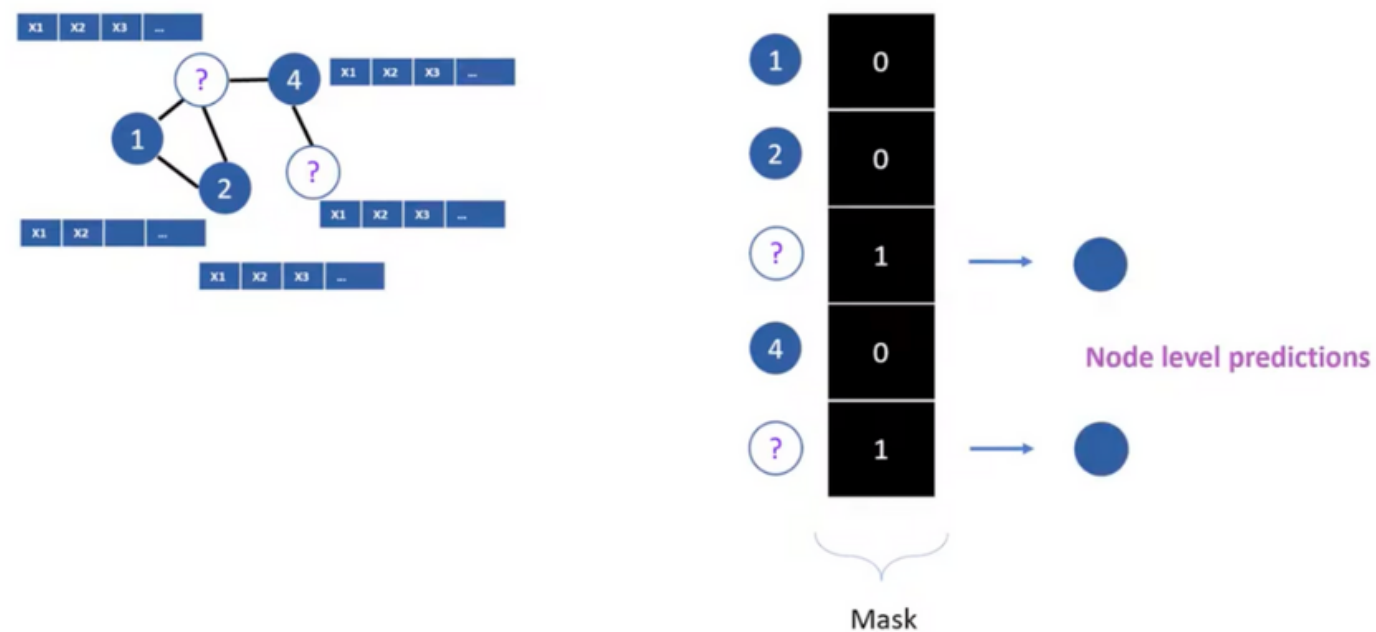


Иллюстрация набора признаков вершины

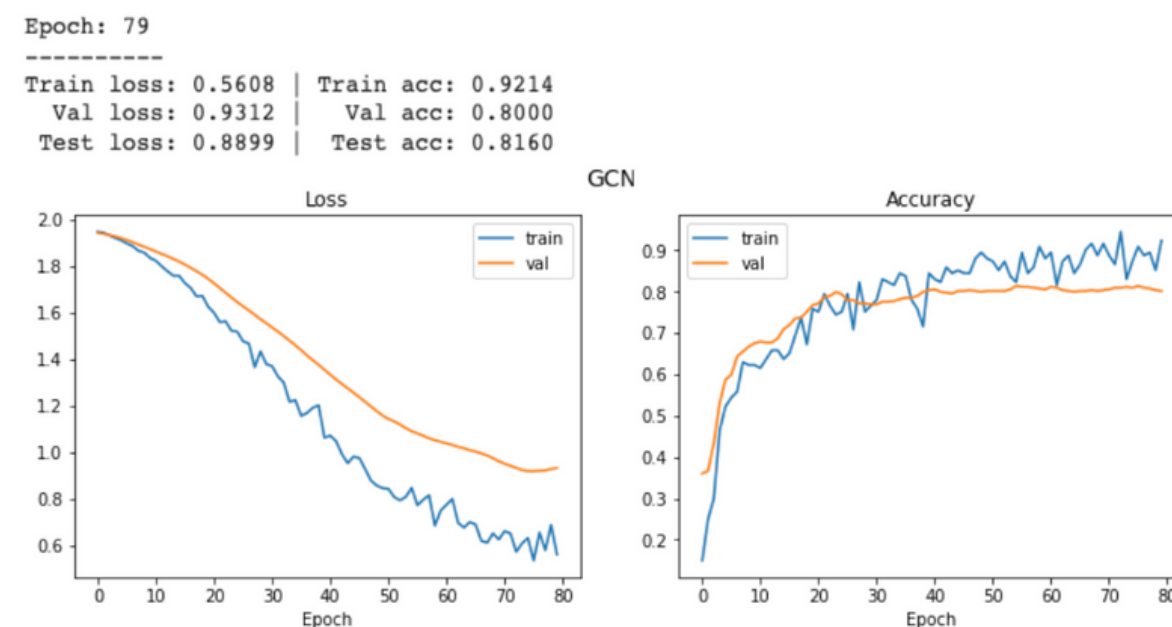


Бинарная маска для подсчета ошибки

Построение сети и результаты

```
Graph Convolutional Network (GCN):  
GCN(  
  (dropout1): Dropout(p=0.5, inplace=False)  
  (conv1): GCNConv(1433, 64)  
  (relu): ReLU(inplace=True)  
  (dropout2): Dropout(p=0.5, inplace=False)  
  (conv2): GCNConv(64, 7)  
)
```

Построенная сверточная нейронная сеть



Результаты работы сети

Заключение

В целом графовые нейронные сети набирают популярность в последнее время, поскольку появляются все новые виды задач, которые можно решить с помощью графовых структуры данных и данный вид архитектур в ближайшем будущем будет только все больше развиваться и находить новые приложения в практической сфере деятельности.