Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
Московский государственный технический университет
имени Н.Э. Баумана
Факультет «ИУ»

Курсовая работа по дисциплине «Численные Методы» на тему «Поиск оптимальной архитектуры графовой нейронной сети для сложных наборов данных на основе генетического алгоритма»

Исполнитель: Студент группы ИУ9-72Б
Балтаева М.
Научный руководитель: Каганов Ю. Т.

Москва, 2024 г.

- Цель: Исследование и применение генетического алгоритма для поиска оптимальной архитектуры графовых нейронных сетей на сложных наборах данных.
- Задачи:
  - реализовать генетический алгоритм для оптимизации GNN;
  - выявление наиболее эффективных комбинаций гиперпараметров для GNN;
  - выявление характеристик и особенностей отдельных архитектур GNN на основе полученных результатов;
  - сохранение и представление результатов работы в виде графиков, метрик и таблиц.

Методология исследования

Генетический алгоритм - метод поиска оптимального решения, вдохновленный механизмами естественного отбора и эволюции.

Кодирование архитектуры GNN через набор генов:

1. Количество скрытых слоев (hidden_num): диапазон 2-10 слоев
2. Размерность скрытого слоя (hidden_dim): диапазон: 16-128 нейронов
3. Тип свёрточного слоя (conv_type): GCN, SAGEConv, GATConv, GraphConv
4. Функция активации: SiLU, ReLU, Sigmoid, tanh
5. Метод пула (pooling): global_mean_pool, global_max_pool
6. Скорость обучения: диапазон: 0.0001 - 0.01

Принципы генетического алгоритма:

- Случайная initial популяция
- Оценка fitness каждой особи
- Селекция лучших
- Скрещивание
- Мутация
- Эволюция к оптимальному решению

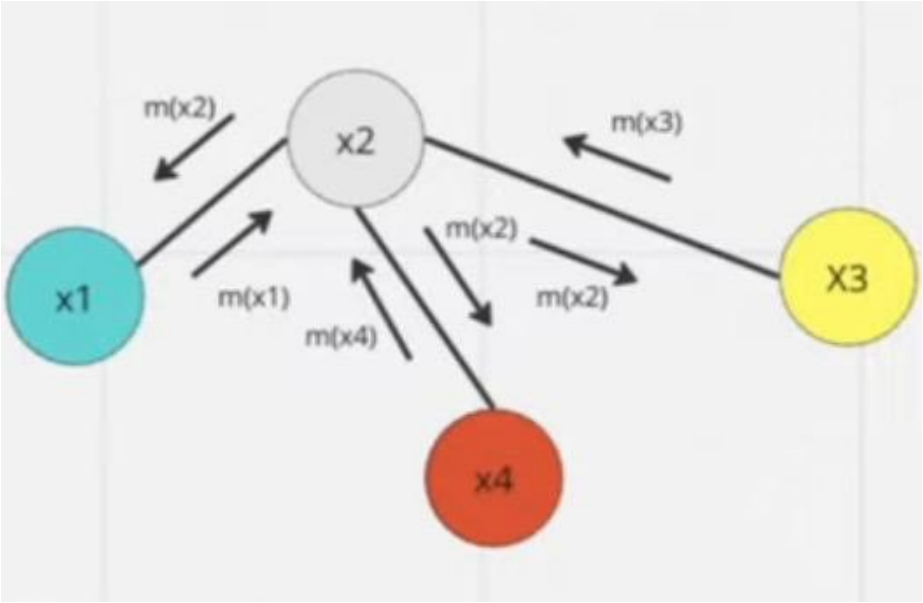
Что из себя представляют графовые нейронные сети

Грубо говоря:

- Метод обогащения узла информацией о его связях
- Метод учета связей между узлами
- Обработка данных, представленных в виде графа

Основная идея графовых нейронных сетей: Message passing

Каждый шаг все вершины отправляют сообщения своим соседям, после чего обновляют свое состояние в соответствие с полученными новыми сообщениями



Архитектура графовой нейронной сети. Графовые свёрточные слои.

- **GCN**

GCNConv использует спектральное представление графа для выполнения свертки. В этом методе матрица смежности графа используется для агрегации информации от соседних узлов.

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

где $H^{(l)}$ – преставление узлов на l-ом слое,

$\hat{A}$ – нормализованная матрица смежности,

$W^{(l)}$ – обучаемые веса слоя,

σ – функция активации.

- **GraphConv**

GraphConv использует более гибкую форму агрегации информации от соседних узлов. Основная формула для обновления представления узла i выглядит следующим образом:

$$h_i^{(l+1)} = \sigma(\sum_{j \in \mathcal{N}(i)} h_j^l W^{(l)} + b^{(l)})$$

где $b^{(l)}$- вектор смещения слоя l.

Архитектура графовой нейронной сети. Графовые свёрточные слои.

- **SAGEConv (GraphSAGE)**

GraphSAGE разработан для работы с большими графами и позволяет модели обучаться на подмножествах соседей. На каждом шаге он выбирает случайное подмножество соседей и применяет функции агрегации, такие как среднее или максимум. Эта архитектура подходит для задач индуктивного обучения, где модель может обрабатывать графы, неизвестные во время обучения. Обновление узла записывается как:

$$h_v^{(l+1)} = \sigma(W^{(l)} h_v^{(l)} + W^{(l)} AGGREGATE^{(l)}(\{h_u^{(l)} : u \in \mathcal{N}(v)\}))$$

- **GATConv (Graph Attention Network)**

GATConv использует взвешенную сумму представлений соседних узлов, где веса определяются механизмом внимания. Основная формула для обновления представления узла i выглядит следующим образом

$$h'_v = \sigma(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} W h_u)$$

где α_{vu} – коэффициент внимания, который определяется на этапе обучения

Генетический алгоритм оптимизации GNN

1) Инициализация популяции:

Случайная генерация первоначального набора архитектур

2) Оценка особей (fitness-функция):

- Критерий оценки: точность классификации на тестовой выборке

3) Селекция лучших особей:

- Принцип "естественного отбора"
- Сохранение top-особей для следующей генерации

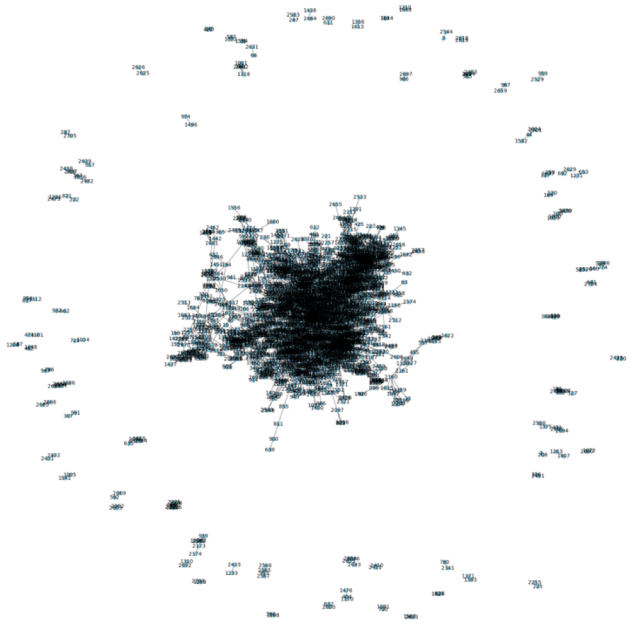
4) Скрещивание (кроссинговер):
Обмен генетической информацией между родителями
Стратегии скрещивания:
• Случайный выбор параметров от родителей
• Усреднение числовых параметров
• Сохранение лучших характеристик

5) Мутация:
Внесение случайных изменений в особи.
Вероятность мутации: 10%

Циклограмма генетического алгоритма:
[Популяция] → [Оценка] → [Селекция] → [Скрещивание] → [Мутация] → [Новая популяция]

Экспериментальные наборы данных

- Cora

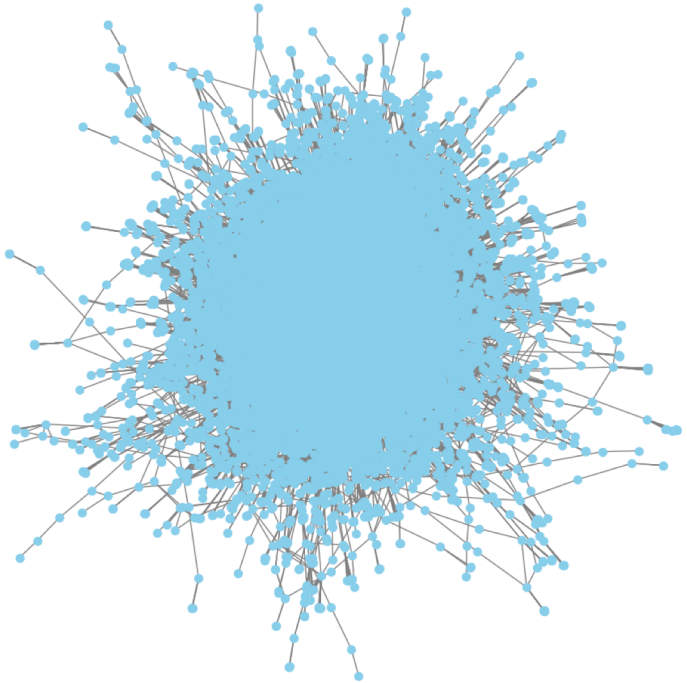


Характеристики:

- Количество узлов: 2708
- Количество ребер: 5429
- Количество классов: 7
- Количество атрибутов узла: 1433

Экспериментальные наборы данных

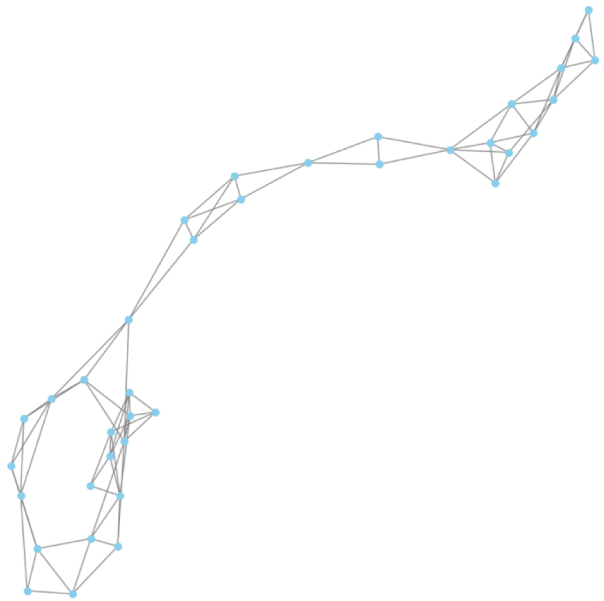
- Pubmed



Характеристики:
- Количество узлов: 19717
- Количество ребер: 44338
- Количество классов: 4
- Количество атрибутов узла: 500

Экспериментальные наборы данных

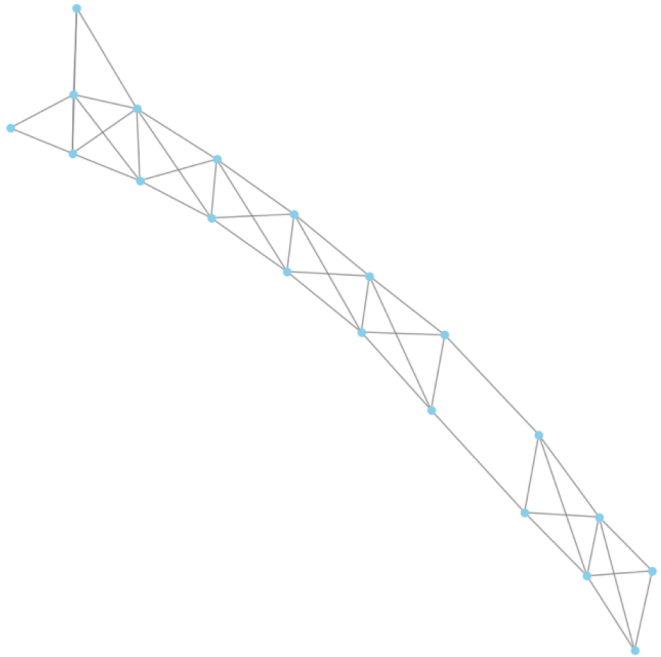
- ENZYMES



Характеристики:
- Количество графов: 600
- Количество классов: 6
- Среднее количество узлов в графе: 32.63
- Среднее количество ребер в графе: 62.14

Экспериментальные наборы данных

- PROTEINS



Характеристики:
- Количество графов: 1113
- Количество классов: 2
- Среднее количество узлов в графе: 39.06
- Среднее количество ребер в графе: 72.82

Результаты экспериментов для каждого набора данных

Dataset	hidden_num	hidden_dim	conv_type	activation	pooling	lr
Cora	8	74	SAGEConv	silu	global_max_pool	0.007
Pubmed	5	107	GATConv	silu	global_max_pool	0.0033
Enzymes	2	44	SAGEConv	relu	global_max_pool	0.0084
Proteins	3	35	SAGEConv	silu	global_mean_pool	0.0078

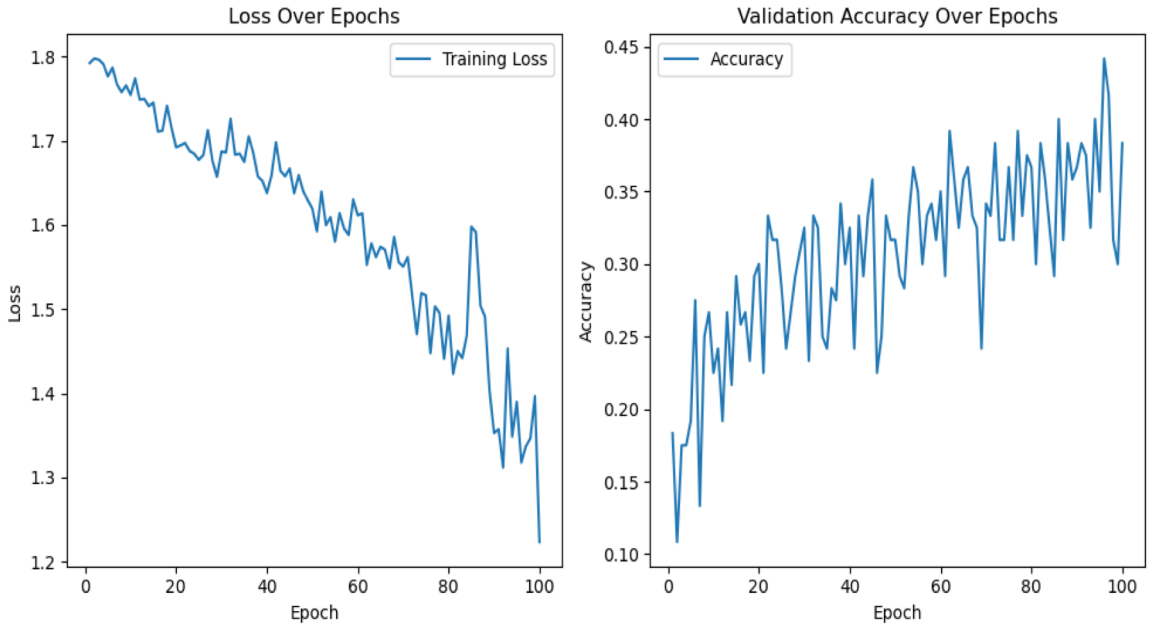


График зависимости точности и потерь от эпох для датасета ENZYMES

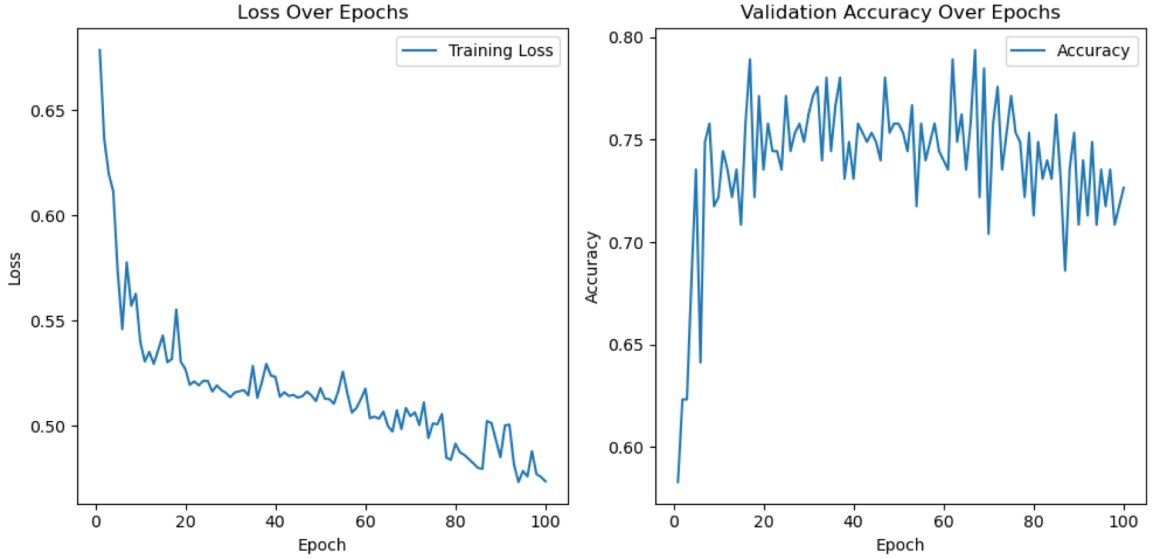


График зависимости точности и потерь от эпох для датасета PROTEINS

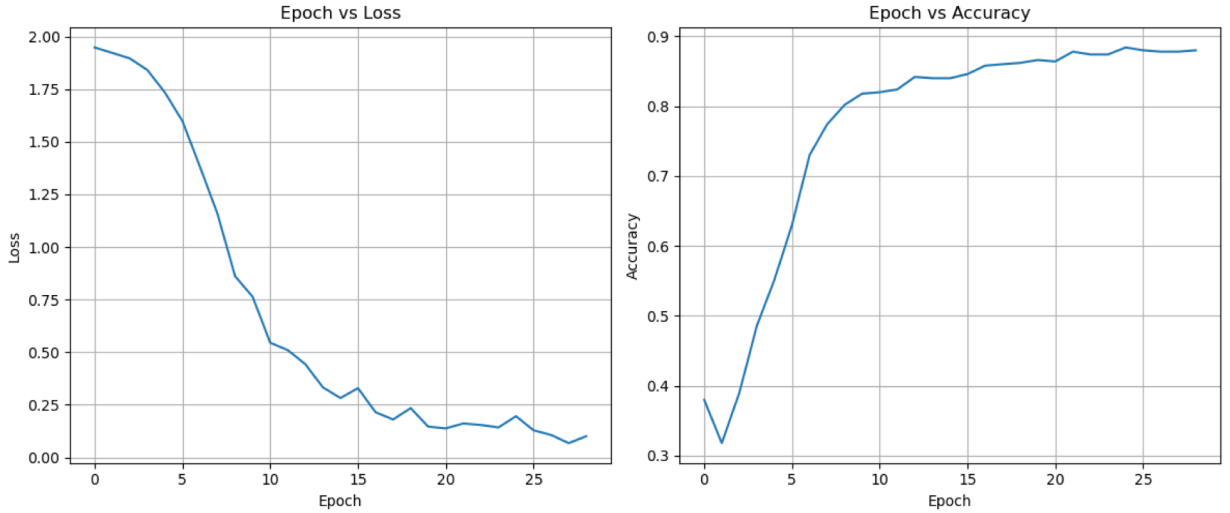


График зависимости точности и потерь от эпох для датасета CORA

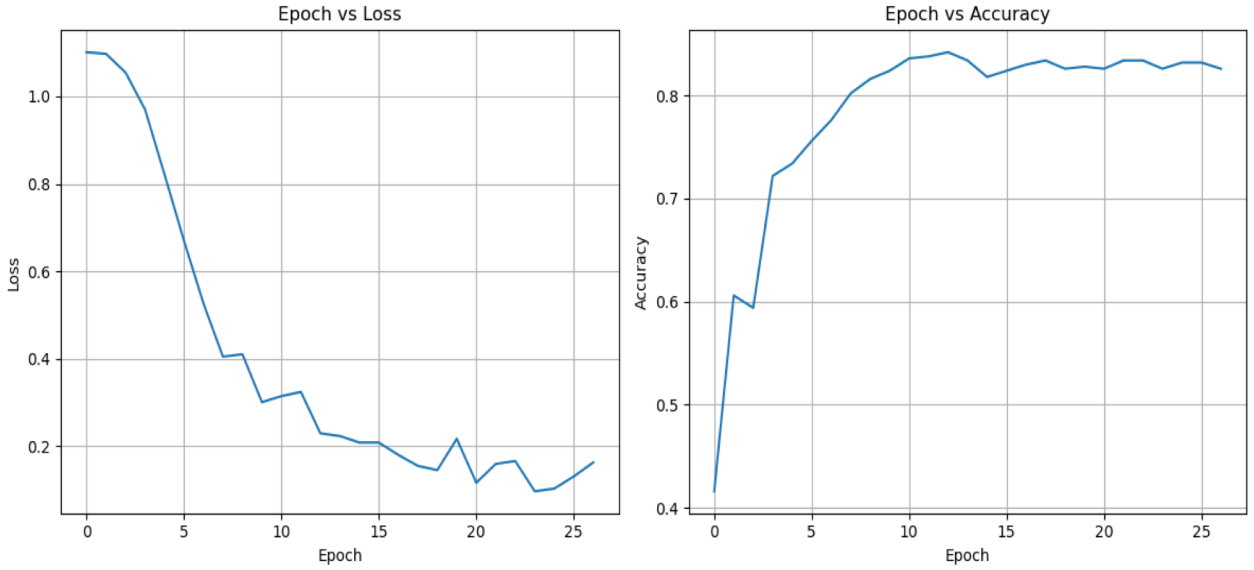


График зависимости точности и потерь от эпох для датасета PUBMED

ВЫВОДЫ

- Наиболее универсальными типами свёрточных слоёв оказались SAGEConv и GCNConv
- GCNConv и GraphConv появляются наиболее часто и обеспечивают стабильные результаты, включая некоторые из самых высоких значений fitness.
- SAGEConv также показывает хорошие результаты, в сочетании с активацией relu и методом объединения global_max_pool.
- GATConv иногда показывает хорошие результаты, но наблюдается большая вариативность, возможно, из-за чувствительности к другим гиперпараметрам.
- Функции активации Silu и ReLU показали себя эффективными, способствуя хорошей производительности моделей и стабильному обучению.
- Sigmoid и Tanh показывают менее стабильные результаты.
- Методы пуллинга Global Mean Pooling и Global Max Pooling оказались наиболее эффективными, при этом Global Mean Pooling был более универсальным выбором.
- Большинство моделей используют скорость обучения около 0,007-0,0084. Более низкие значения, такие как 0,002, иногда приводят к снижению производительности из-за слишком медленной сходимости.
- Архитектуры с 3 скрытыми слоями и размерностью от 44 до 50 нейронов часто достигают высоких значений fitness. Более глубокие сети (до 11 слоев) не обязательно показывают лучшие результаты, что может указывать на переобучение или затруднённую тренировку глубоких моделей.