



دانشگاه صنعتی امیرکبیر
(پلی تکنیک)



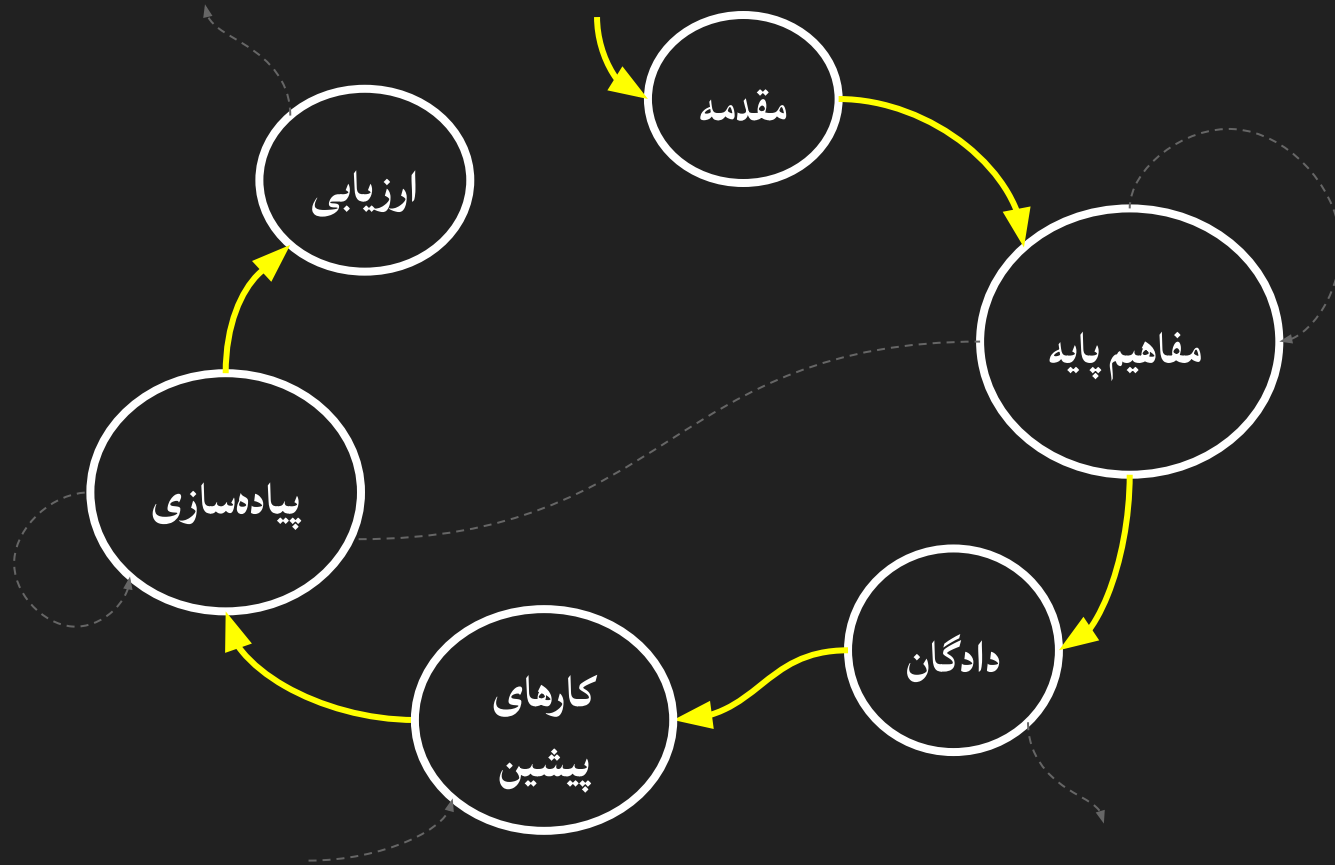
دانشکده مهندسی کامپیوتر

بررسی عملکرد شبکه‌های عصبی گرافی در رگرسیون گره‌ها

گردآورنده: امیرمهدی زرین‌نژاد
استاد راهنما: جناب آقای دکتر حقیر چهرقانی
استاد داور: جناب آقای دکتر میبیدی

بهار ۱۴۰۲

مسیر ارائه



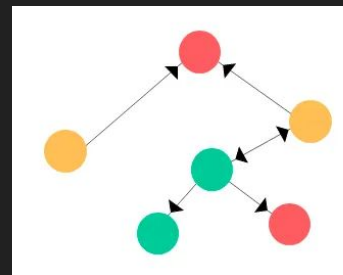
١. مقدمه

شبکه‌های عصبی گرافی

نوعی شبکه عصبی که داده‌های با ساختار گراف را پردازش می‌کنند

ابزاری قدرتمند در حوزه علم داده، هوش مصنوعی، یادگیری ماشین و داده‌کاوی

پاسخ به وظایف متعدد



اهداف پروژه

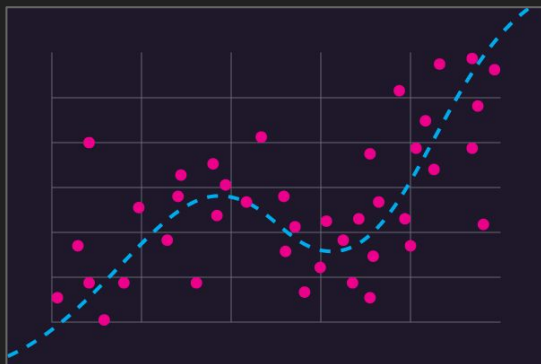
تهیه دادگان مناسب

پیاده‌سازی مدل‌های شبکه عصبی گرافی

ارزیابی و مقایسه عملکرد مدل‌ها در برخورد با دادگان

رگرسیون

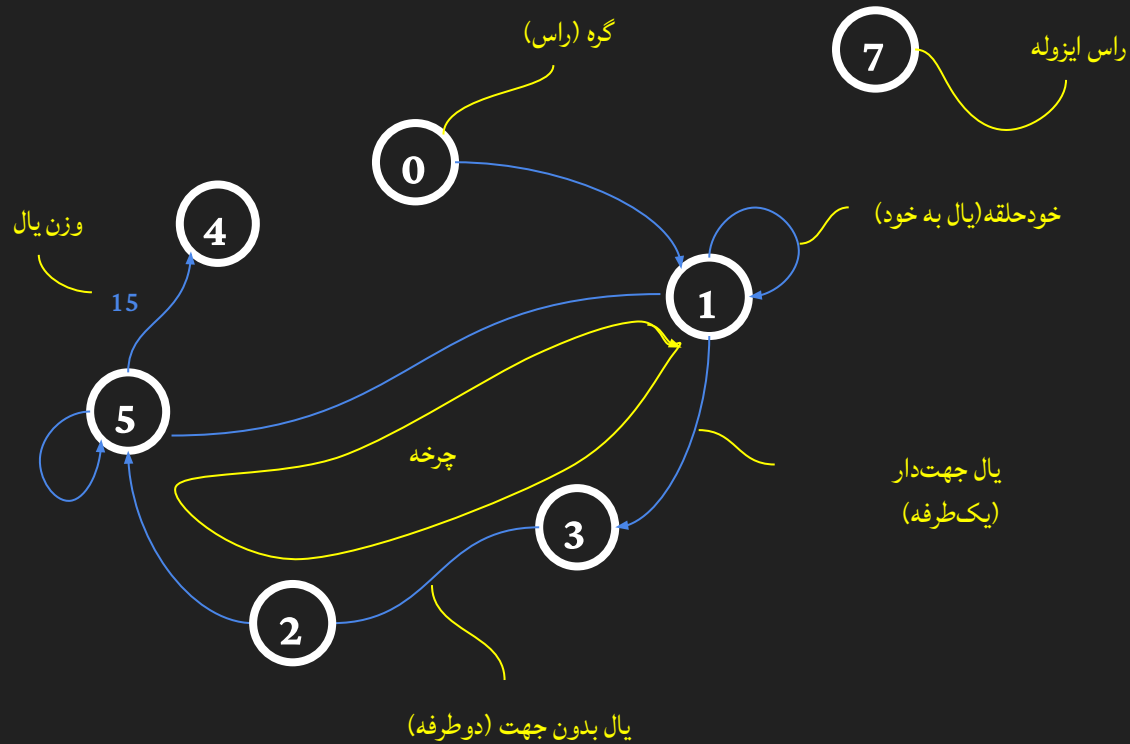
یکی از مسائل پرکاربرد و مهم در حوزه هوش مصنوعی

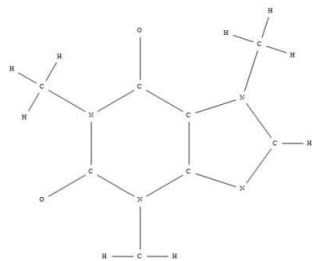


۲. ادبیات مسئله و مفاهیم پایه

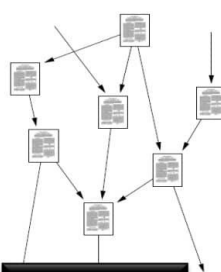
گراف

گراف $G = G(V, E)$

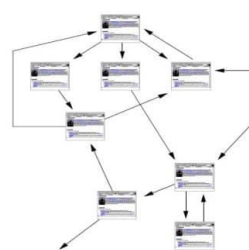




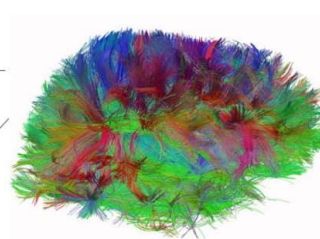
Molecules



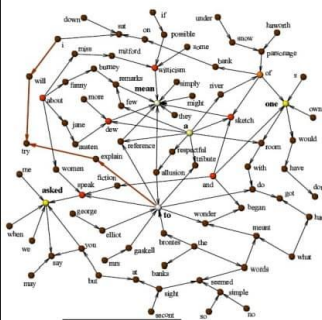
Knowledge



Information



Brain/neurons



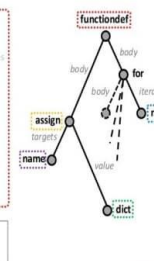
Genes



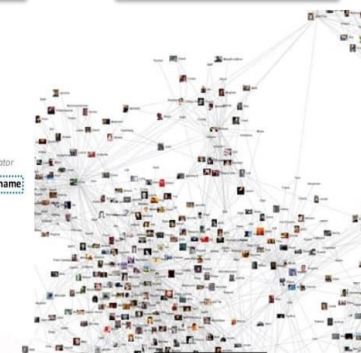
Communication

```
def encode(obj):
    """
    Encode a (possibly nested)
    dictionary containing complex values
    into a form that can be serialized
    using JSON.
    """
    e = {}
    for key, value in obj.items():
        if isinstance(value, dict):
            e[key] = encode(value)
        elif isinstance(value, complex):
            e[key] = ('type': 'complex',
                    'r': value.real,
                    'i': value.imag)
    return e

import ast
tree = ast.parse('')
```



Software



Social

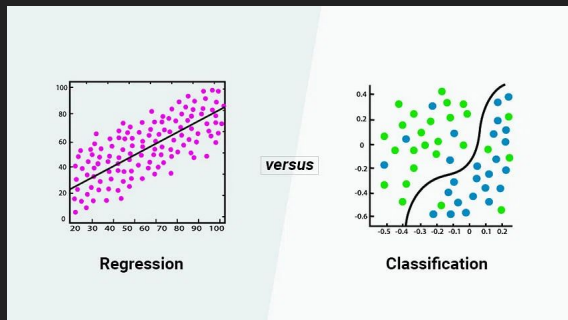
رگرسیون

پس از برآورد رابطه بین متغیرهای مستقل و وابسته

می توان مقادیر را پیشبینی کرد.

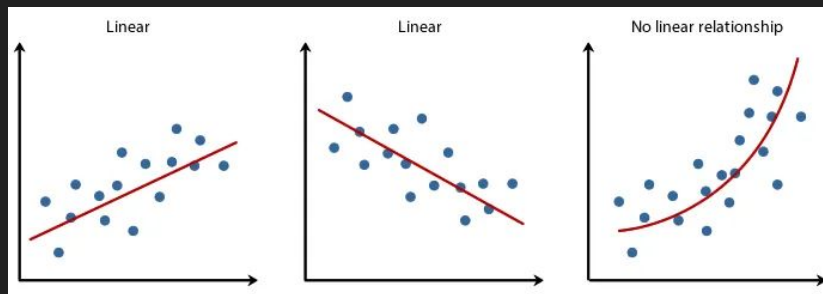
خطی:

$$y = \beta_0 + \beta_1 x + \varepsilon$$



غیر خطی:

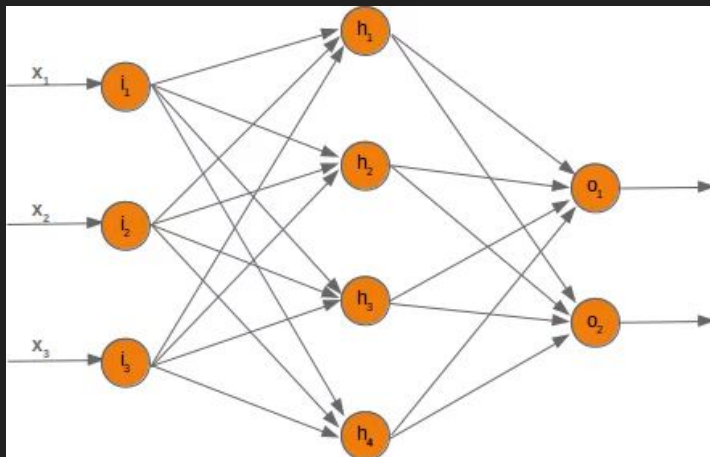
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_p x^p + \varepsilon$$



شبکه عصبی

مدل‌های محاسباتی که از ساختار اعصاب مغز انسان الهام گرفته شده‌اند و می‌خواهند تابعی برای حل مسئله پیدا کنند.

لایه ورودی لایه (های) پنهان لایه خروجی



نورون

$$y = f(w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n + b)$$

یادگیری

$$\theta = \theta - \alpha \cdot \nabla J(\theta)$$

(روش کاهش گرادیان)

کاهش گرادیان تصادفی (SGD)

در پروژه‌های ما: بهینه‌ساز آدام؛ Momentum + RMSProp

شبکه عصبی گرافی

مفاهیم جدید

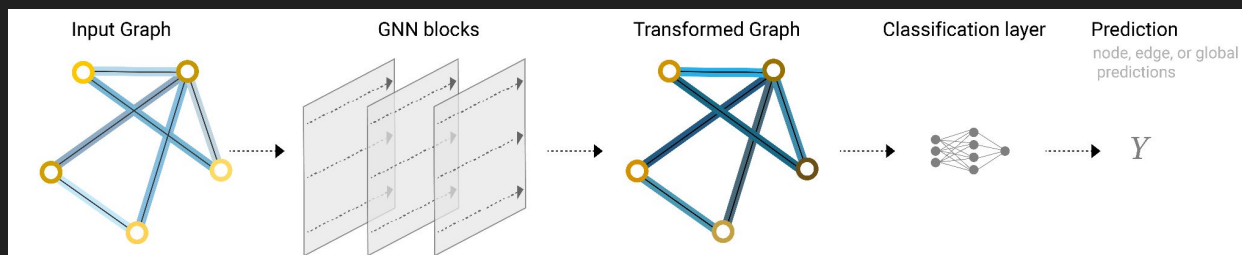
ورودی: یک گراف شامل راس‌ها، یال‌ها (ماتریس مجاورت) و ماتریس ویژگی‌ها

مقداردهی اولیه‌ی جاسازی‌ها (embeddings): بازنمایی از گره، لبه یا گراف

تبادل پیام: انتقال اطلاعات بین گره‌های همسایه

جمع‌آوری و بروزرسانی: بروزرسانی جاسازی‌ها با اطلاعات جمع‌آوری شده از مرحله تبادل پیام

بازخوانی و ادغام: جمع‌آوری اطلاعات تمام گره‌ها در یک بازنمایی سطح گراف



خروجی: با توجه به وظیفه، مقادیر پیوسته،

احتمال تعلق به کلاس، وجود یا عدم وجود یال

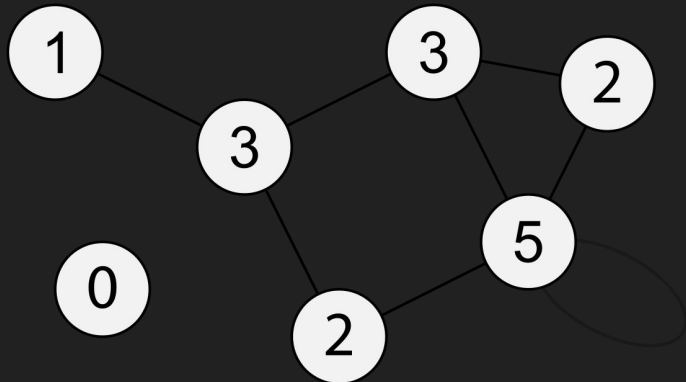
جاسازی‌های نهایی (یا بازنمایی در سطح گراف)

شبکه عصبی گرافی

وظایف:

سطح گره، سطح یال، سطح گراف، دسته‌بندی گره، دسته‌بندی گراف، پیش‌بینی پیوند، ...، رگرسیون گراف و

رگرسیون گره مانند: پیش‌بینی قیمت یک خانه براساس ویژگی‌ها و ارتباطات محله‌ای



۳. کارهای پیشین

روش‌های سنتی

رگرسیون خطی روی گراف‌ها: یک رابطه ساده و خطی بین مقادیر گره

روش‌های هسته: گراف‌ها از طریق هسته گراف به بردارهای ویژگی تبدیل می‌شوند. سپس از این بردارها به عنوان ورودی برای مدل‌های رگرسیون سنتی استفاده می‌شود.

روش انتشار برچسب: برچسب‌های گره‌های شناخته شده را از طریق گراف منتشر می‌کنند تا برچسب‌ها یا مقادیر گره‌های بدون برچسب را پیش‌بینی کنند.

نظریه گراف طیفی: با استفاده از مقادیر ویژه و بردارهای ویژه ماتریس لاپلاسی گراف، گراف‌ها را به فضاهای با ابعاد پایین‌تر می‌برد.

و ...

این رویکردهای سنتی ابزارهای ارزشمندی را برای مدلسازی پیش‌بینی کننده در داده‌های ساختاریافته گرافی ارائه می‌کنند. با این حال، آن‌ها اغلب بر فرضیات ساده‌سازی تکیه می‌کردند یا به مهندسی ویژگی‌های دستی گسترده نیاز داشتند.

۴. دادگان، آماده‌سازی و مراحل اولیه

انتخاب دادگان

داده‌ها بستری را تشکیل می‌دهند که تحلیل‌های ما بر اساس آن شکل می‌گیرند.

ویژگی‌های مجموعه دادگان مناسب پروژه:

۱- ساختار شبکه‌ای

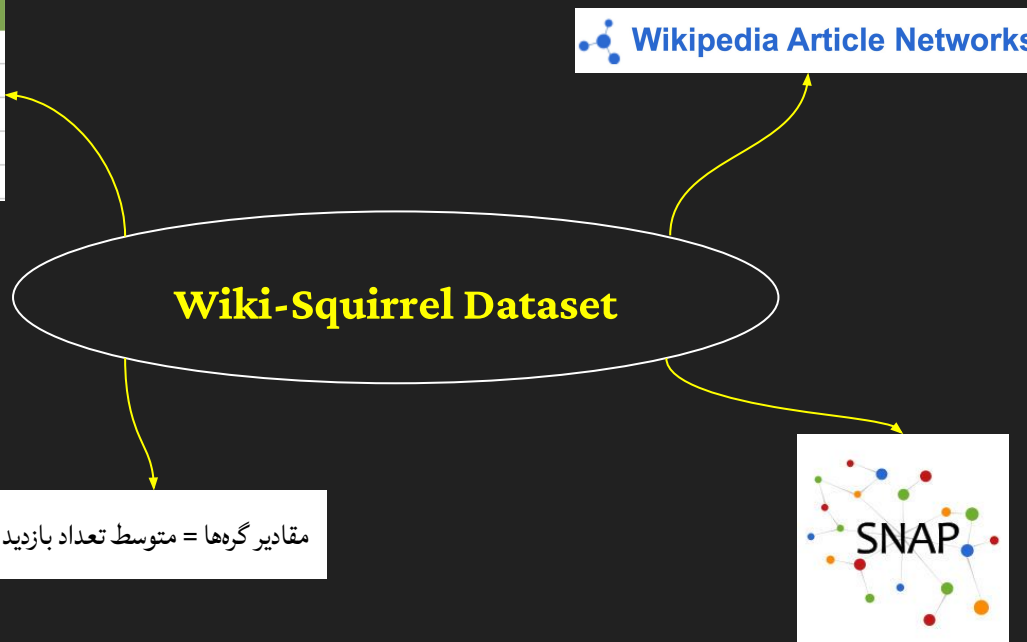
۲- مقادیر هدف پیوسته

۳- ابعاد مناسب

۴- معتبر بودن

۵- بکر بودن

Dataset statistics			
	Chameleon	Crocodile	Squirrel
Nodes	2,277	11,631	5,201
Edges	31,421	170,918	198,493
Density	0.012	0.003	0.015
Transitivity	0.314	0.026	0.348



Wiki-Squirrel Dataset

```
{
  "2098": [1765, 3047, 630, 1308, 2455, 2623, 1883, 1810, 934, 2111, 288, 2077, 642, 2405, 3025, 437, 86, 649, 860, 2784,
    1231, 430, 830, 216, 1051, 2980, 1787, 1324],
  "47": [3001, 2690, 2299, 1312, 1109, 721, 1721, 2887, 1794, 2297, 2753],
  "2539": [415, 598, 302, 1846, 2090, 1297, 1488, 1674, 1316, 1834],
  "1674": [2114, 2518, 1794, 768, 2479, 997, 849],
  "1207": [1855, 1073, 251, 1000, 1800, 953, 1794, 2297, 2753, 2539, 598, 528, 1319, 847, 2899],
  "1340": [2357, 2453, 2468, 73, 584, 1001, 165, 372, 988, 178, 666, 2201, 1300, 2441, 370, 971, 2158, 1130, 1884, 2645, 2038, 2364],
  "196": [1079, 2802, 2468, 73, 584, 165, 2826, 599, 655, 2070, 923, 191, 207, 1677, 2645, 2948, 2116, 1669, 2716],
  "1586": [1157, 245, 2620, 1347, 73, 584, 372, 1008, 2492, 1161, 1026, 1733, 3065, 1098],
  "1581": [1765, 3047, 1634, 73, 3118, 584, 165, 2562, 1057, 2720, 734, 2812, 3104, 1495, 1195, 514, 2462, 2246, 2350, 1632, 847, 2090, 512, 2055, 1212, 1272, 40, 3033, 498, 3031,
    1354, 2546, 2879],
  "1224": [1765, 3047, 1634, 2300, 1749, 399, 2160, 1727, 1482, 1570, 508, 2041, 900, 1656],
  "1277": [434, 2153, 630, 2636, 892, 1308, 65, 2141, 3051, 1199, 3058, 253, 787, 1595],
  "664": [2357, 2463, 3014, 2049, 2254, 874, 224, 321, 73, 3118, 584, 1001, 165, 2562, 372, 1734, 1399, 775, 332, 617, 43, 84, 3022, 2118, 2745, 2172, 2504, 710,
    2970, 2002, 1841, 1695, 47, 2502, 1261, 1197, 1551, 1991, 2070, 939, 2386, 1668, 1136, 924, 2222, 522, 715, 2691, 2444,
    2211, 542, 1545, 2442, 2038, 95, 781, 2550, 3056, 1529, 1998, 622, 654, 2908, 2094, 2338, 1030, 1009, 1661, 1171, 2889,
    1183, 2384, 2650, 3111, 1676, 123, 974, 2879],
  "408": [2198, 1503, 3012, 1880, 1308, 2297, 2753, 2354, 598, 1944, 1688, 1983, 1377, 219, 2817, 643, 1846, 1291,
    "2195": [2114, 2068, 1794, 2753, 415, 528, 1367, 617],
    "1496": [1505, 208, 2299,
```

Features: JSON file

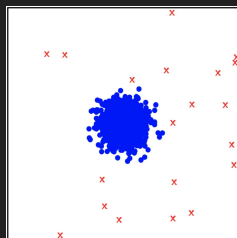
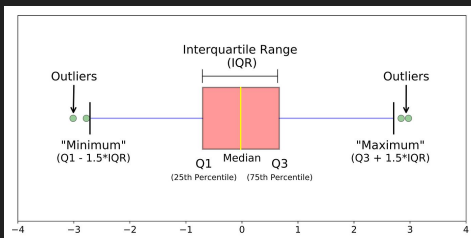
	id	target
1	0	171
2	1	8089
3	2	8568
4	3	47178
5	4	3634
6	5	215
7	6	1054
8	7	1106
9	8	109
10	9	3863
11	10	598
12	11	175
13	12	1399
14	13	39
15	14	444
16	15	64492
17	16	62284
18	17	154527
19	18	673
20	19	33
21	20	14480

targets: csv file

	id1	id2
1	2034	1939
2	2034	2263
3	2034	1976
4	2034	2234
5	2034	804
6	2034	1474
7	2034	2110
8	2034	652
9	2034	1350
10	2034	1860
11	2034	2243
12	2034	1889
13	2034	1356
14	2034	536
15	1747	1554
16	1747	1043

edges: csv file

پیش پردازش و آماده سازی



$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

```
features:
tensor([[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

```
Train, Validation, and Test Masks:
tensor([False, True, True, ..., True, True, True])
tensor([ True, False, False, ..., False, False, False])
tensor([False, False, False, ..., False, False, False])
```

۱- خوانش دادگان

۲- داده های پرت

۳- نرمال سازی مقادیر هدف

۴- کدگذاری وان-هات

آماده سازی نهایی

مشکلات: - تحلیل نادرست

- افزایش واریانس

- تخریب کارایی و یادگیری الگوهای اشتباه

مدیریت: - تشخیص داده های پرت

- حذف یا تصحیح

- تبدیل به داده عددی

- حذف ارزش ترتیبی

- سهولت پردازش و تحلیل مدل

۱- ایجاد گراف

۲- ایجاد ماسک ها

- یادگیری

- اعتبارسنجی

- تست

۵. پیاده‌سازی و ساختار مدل‌ها

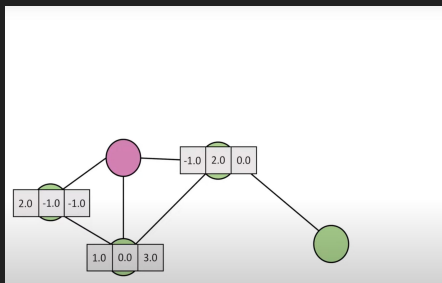
Graph Convolutional Networks (GCN) یا شبکه‌های پیچشی گرافی

ساختار

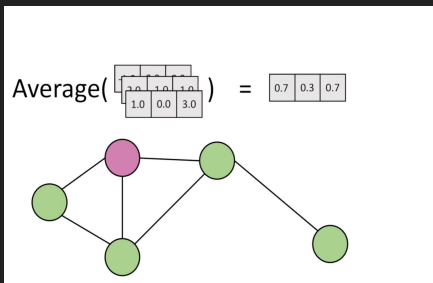
- انتشار ویژگی و مکانیزم تجميع
- میانگین وزنی
- اعمال شبکه عصبی
- بروزرسانی

با مقدار خودش با خود حلقه

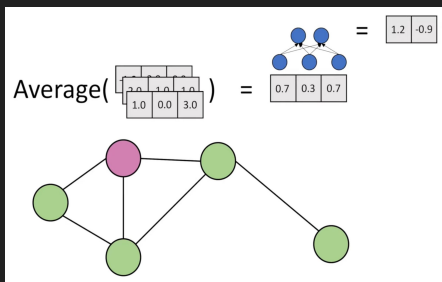
۱)



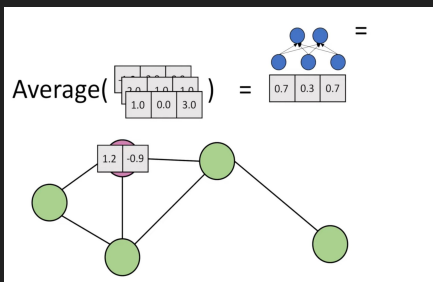
۲)



۳)



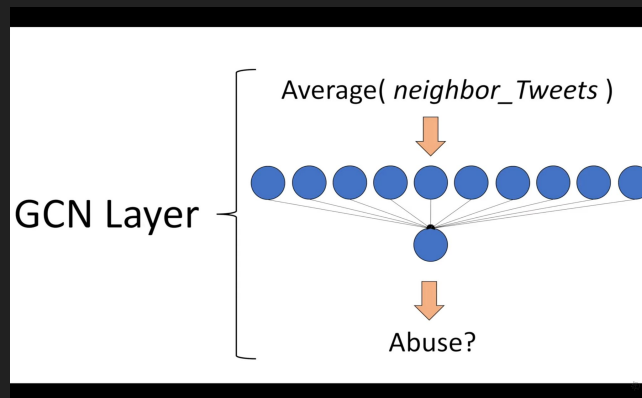
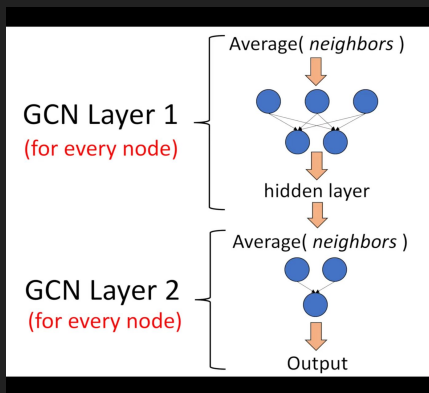
۴)



GCN (Graph Convolutional Networks) یا شبکه‌های پیچشی گرافی

تعداد نودون لایه آخر را می‌توان با توجه به وظیفه مورد بررسی تعیین کرد.

- تشخیص پیام نامناسب از پیام مناسب



Graph Attention Networks (GAT) یا شبکه‌های توجه گرافی

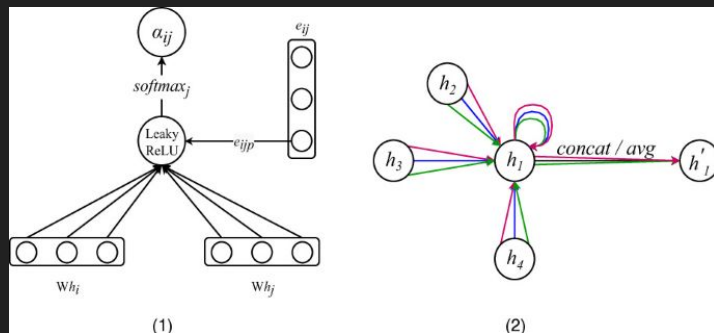
ساختار

$$e_{ij} = \text{LeakyReLU} \left(\vec{a}^T [W\vec{h}_i || W\vec{h}_j] \right)$$

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \text{softmax}(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_j]))$$

$$\vec{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W \vec{h}_j \right)$$

$$h'_i = \parallel_{k=1}^K \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j \right)$$



- ورودی

- مکانیسم توجه و محاسبه‌ی وزن همسایه‌ها

- استفاده از تابع به جای وزن دهی معمولی

- ضرب داخلی (h_j و h_i)

- اعمال شبکه عصبی

- توجه بیش‌تر به بعضی همسایه‌ها (حتی با درجه یکسان)

- اعمال Softmax برای نرمال‌سازی

- تجميع

- ادغام و خروجی

GATv2 (Graph Attention Networks version 2) یا شبکه‌های توجه گرافی نسخه ۲

ساختار

$$e_{ij} = \vec{a}^T \text{LeakyReLU} \left(W[\vec{h}_i || \vec{h}_j] \right)$$

- ورودی
- مکانیسم توجه و محاسبه‌ی وزن همسایه‌ها
- استفاده از تابع به جای وزن‌دهی معمولی
- اعمال Softmax برای نرمال‌سازی
- توجه بیش‌تر به بعضی همسایه‌ها (حتی با درجه یکسان)
- تجميع
- ادغام و خروجی

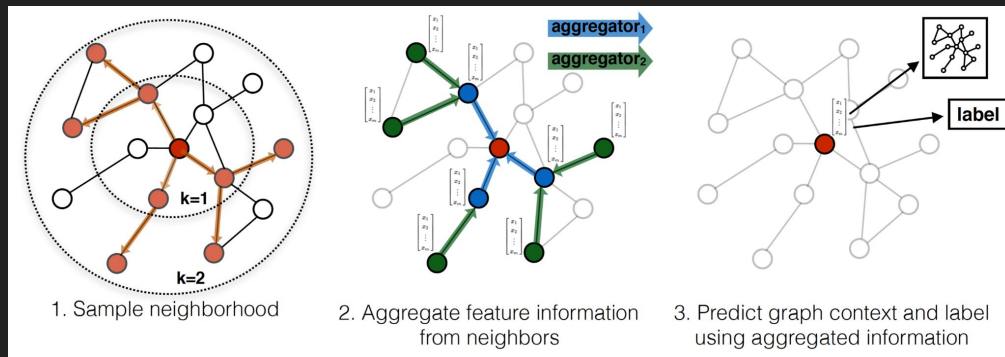
تفاوت در ترتیب عملیات‌ها

(Graph Sample and Aggregated Embeddings) GraphSAGE

تولید بازنمایی های برداری کم بعدی برای گره ها

یک چارچوب بر اساس مفهوم نمونه برداری و تجمع اطلاعات از همسایگی گره ها، برای ایجاد بازنمایی گره ها (به خصوص گره های دیده نشده) در گراف با استفاده از شبکه های عصبی معمولی

ساختار



- نمونه برداری K-Hop از همسایگان

- عمق K

- Hop

- تجمع

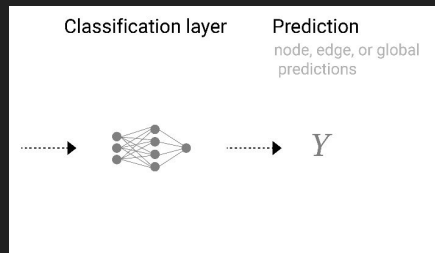
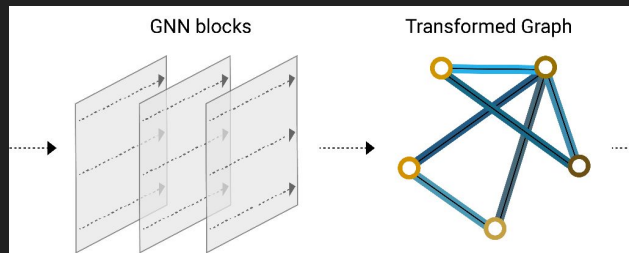
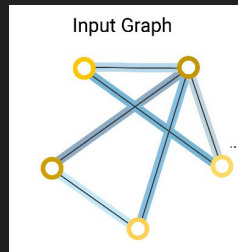
- Mean Aggregator, LSTM Aggregator,

Pooling Aggregator

- ادغام

پیاده‌سازی

لایه‌ها



- ورودی
- گراف (گره‌ها و مجاورت‌هایشان) همراه با ویژگی‌ها به عنوان ورودی مدل‌ها
- لایه ورودی با ابعاد تعداد ویژگی‌های ورودی و تعداد ویژگی‌های پنهان
- لایه‌های میانی
- لایه‌های میانی با ابعاد تعداد ویژگی‌های پنهان و تعداد ویژگی‌های پنهان
- لایه خروجی
- لایه‌ی خروجی با ابعاد تعداد ویژگی‌های پنهان و تعداد مقادیر خروجی (با توجه به وظیفه)
- در وظیفه ما (رگرسیون گره‌ها)، یک مقدار خروجی برای هر گره
- به کارگیری لایه‌ی تماماً متصل خطی‌ساز با ابعاد ویژگی‌های پنهان و یک خروجی

تابع رو به جلو (forward)

- استفاده از لایه‌ها
- لایه‌ها پشت سر هم
- عبور داده‌ها از شبکه عصبی گرافی و پردازش آن‌ها
- خروجی لایه‌ی قبل به عنوان ورودی لایه‌ی جدید
- توابع فعال‌ساز
- تابع فعال‌ساز مناسب وظیفه ReLU

۶. ارزیابی و نتیجه‌گیری

معیارهای ارزیابی

- یادگیری مدل
- ارزیابی عملکرد

توابع زیان

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- Mean Squared Error (MSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- Root Mean Squared Error (RMSE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- Mean Absolute Error (MAE)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%$$

- Mean Absolute Percentage Error (MAPE)

Squirrel Dataset

MSE	مدل/تابع زیان
0.0045	GAT
0.0046	GATv2
0.0069	GCN
0.0070	GraphSAGE

Crocodile Dataset

MSE	مدل/تابع زیان
0.0014	GAT
0.0016	GATv2
0.00175	GCN
0.00179	GraphSAGE

Chameleon Dataset

MSE	مدل/تابع زیان
0.0016	GAT
0.0015	GATv2
0.0019	GCN
0.0025	GraphSAGE

با توجه به مقادیر پیش‌بینی شده توسط مدل‌ها و نتایج ارزیابی‌ها از MSE استفاده می‌کنیم.

نتیجه و مقایسه

ساختار گراف، جنس دادگان، نرمال‌سازی و دیگر خصیصه‌های گراف در نتایج ارزیابی نقش دارند. اما مسئله و دادگان ما به طور کل می‌توان گفت:

۱- GAT

۲- GATv2

۳- GCN

۴- GraphSAGE

خلاصه فعالیت‌های عملی و پیاده‌سازی

دادگان

- محاسبه، شناسایی و حذف دادگان پرت از مقادیر هدف، لبه‌ها، ویژگی‌ها
- نرمال سازی مقادیر هدف
- اعمال رمزگذاری وان-هات بر ویژگی‌ها و ایجاد بردارهای ویژگی دودویی برای گره‌ها
- ایجاد گراف از دادگان پیش‌پردازش و آماده‌سازی شده
- ایجاد و افزودن ماسک‌های یادگیری، اعتبارسنجی و تست به گراف
- انجام پردازش‌ها بر دادگان نو

مدل‌ها

- وظیفه رگرسیون با سابقه فعالیت بسیار کم‌تر از بقیه وظایف
- ایجاد مدل شبکه عصبی گرافی با سه لایه - پیاده‌سازی و به‌کارگیری شبکه پیچشی گرافی (GCN) در لایه‌ها - پیاده‌سازی سر
- پیش‌بینی در لایه آخر - تنظیم ابعاد، ساختار و پارامترهای لایه‌های شبکه عصبی - پیاده‌سازی تابع رو به جلو (forward) برای مدل
- ایجاد مدل شبکه عصبی گرافی با سه لایه - به‌کارگیری شبکه توجه گرافی (GAT) در لایه‌ها - پیاده‌سازی سرپیش‌بینی در لایه آخر - تنظیم ابعاد، ساختار و پارامترهای لایه‌های شبکه عصبی - پیاده‌سازی تابع رو به جلو (forward) برای مدل

خلاصه فعالیت‌های عملی و پیاده‌سازی

مدل‌ها

- ایجاد مدل شبکه عصبی گرافی با سه لایه - به‌کارگیری شبکه توجه گرافی نسخه ۲ (GATv2) در لایه‌ها - پیاده‌سازی سرپیش‌بینی در لایه آخر - تنظیم ابعاد، ساختار و پارامترهای لایه‌های شبکه عصبی - پیاده‌سازی تابع روبه جلو (forward) برای مدل
- ایجاد و پیاده‌سازی مدل عصبی گرافی - به‌کارگیری GraphSAGE در لایه‌های مدل - پیاده‌سازی سرپیش‌بینی در لایه آخر - تنظیم ابعاد، ساختار و پارامترهای لایه‌های مدل - پیاده‌سازی تابع روبه جلو (forward) برای مدل
- پیاده‌سازی مدل‌ها با توابع فعال‌ساز مختلف و مقایسه و انتخاب تابع فعال‌ساز مناسب
- به‌کارگیری و تنظیم پارامترهای بهینه‌ساز
- پیاده‌سازی و به‌کارگیری توابع زیان متفاوت، مقایسه نتایج و انتخاب تابع زیان مناسب
- پیاده‌سازی و اجرای فرآیند یادگیری، پیش‌بینی، محاسبه زیان و بهبود پارامترهای مدل در تعداد دور (اپیک) مناسب
- پیاده‌سازی عملیات ارزیابی مدل با داده‌های آموزشی، اعتبارسنجی و آزمون

آمارگان پیاده‌سازی

منابع و مراجع

- [1] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. A comprehensive survey on graph neural networks. Ieee Transactions on Neural Networks and Learning Systems, .2021 ,24–32:4
- [2] Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In NIPS, 2003.
- [3] Fouss, F., Pirotte, A., Renders, J. M., and Saerens, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering, 19(11):1427–1443, .2007
- [4] Zhu, X. and Ghahramani, Z. Learning from labeled and unlabeled data with label propagation. 2002.
- [5] Rozemberczki, Benedek, Allen, Carl, and Sarkar, Rik. Multi-scale attributed node embedding, 2019.
- [6] Brody, Shaked, Alon, Uri, and Yahav, Eran. How attentive are graph attention networks?, Jan 2022.

منابع و مراجع

- [7] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lió, P., and Bengio, Y. Graph attention networks. 2017.
- [8] Brody, Shaked, Alon, Uri, and Yahav, Eran. How attentive are graph attention networks?, Jan 2022.
- [9] Kipf, Thomas N. and Welling, Max. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR '17, 2017.
- [10] Hamilton, William L., Ying, Rex, and Leskovec, Jure. Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems, 2017.

با تشکر از توجه شما!

سوال؟

راه ارتباطی: amzarrinnezhad@gmail.com