

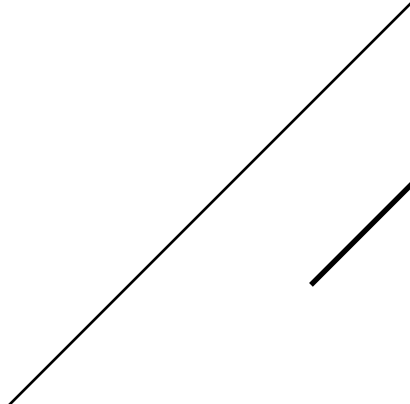
图神经网络入门节点分类 竞赛

举办方：百度大脑

2021.5

A vertical rectangular image on the left side of the slide, featuring a blue-tinted, low-angle shot of a modern building's glass and steel facade, with strong geometric lines and reflections.

CONTENT

- PART ONE 竞赛内容
 - PART TWO 图神经网络
 - PART THREE 算法实现
 - PART FOUR 模型对比
 - PART FIVE 提升方法
- 
- Two thin, dark gray diagonal lines in the bottom right corner of the slide, one longer and one shorter, intersecting near the bottom right edge.

The background is a collage of three European street scenes. The left image shows a narrow street with half-timbered houses and flower boxes. The middle image shows a street leading to a large stone church tower. The right image shows a street with a stone column and a flower bed in the foreground.

01. 竞赛内容

PART



任务」

赛题介绍

图神经网络 (Graph Neural Network) 是一种专门处理图结构数据的神经网络，目前被广泛应用于推荐系统、金融风控、生物计算等领域。图神经网络的经典问题主要有三类，分别为节点分类、连接预测和图分类。

本次任务的目标是预测未知论文的主题类别，如软件工程，人工智能，语言计算和操作系统等。比赛所选35个领域标签已得到论文作者和arXiv版主确认并标记。

本次比赛选用的数据集为arXiv论文引用网络——ogbn-arxiv数据集的子集。ogbn-arxiv数据集由大量的学术论文组成，论文之间的引用关系形成一张巨大的有向图，每一条有向边表示一篇论文引用另一篇论文，每一个节点提供100维简单的词向量作为节点特征。在论文引用网络中，我们已对训练集对应节点做了论文类别标注处理。本次任务希望参赛者通过已有的节点类别以及论文之间的引用关系，预测未知节点的论文类别。



竞赛内容





数据」

数据描述

本次赛题数据集由学术网络图构成，该图会给出每个节点的特征，以及节点与节点间关系（训练集节点的标注结果已给出）。

数据集简介：

1.学术网络图数据：

该图包含1647958条有向边，130644个节点，参赛者报名成功后即可通过比赛数据集页面提供edges.csv以及feat.npy下载并读取数据。图上的每个节点代表一篇论文，论文从0开始编号；图上的每一条边包含两个编号，例如3，4代表第3篇论文引用了第4篇论文。图构造可以参照AiStudio上提供的基线系统项目了解数据读取方法。

2.训练集与测试集：

训练集的标注数据有70235条，测试集的标注数据有37311条。训练数据给定了论文编号与类别，如3，15 代表编号为3的论文类别为15。测试集数据只提供论文编号，不提供论文类别，需要参赛者预测其类别。



具体数据介绍：

2.训练集： **train.csv**

字段	说明
nid	训练节点在图上的Id
label	训练节点的标签（类别编号从0开始，共35个类别）

3.测试集： **test.csv**

字段	说明
nid	测试节点在图上的Id



02 图神经网络

PART

图神经网络

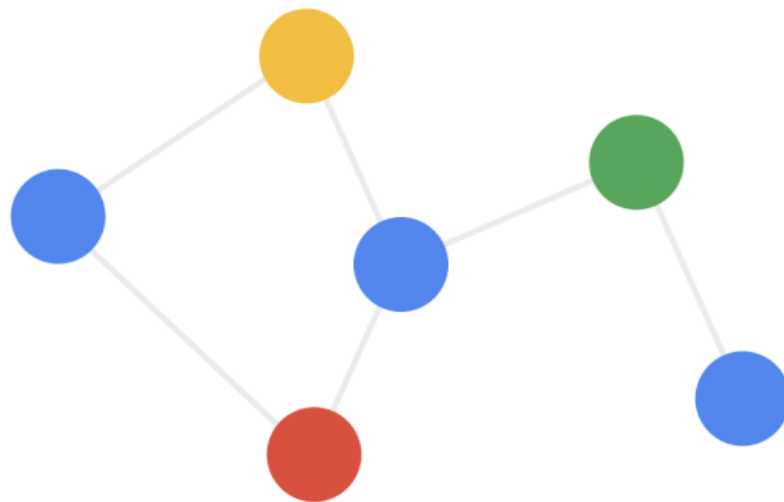
图神经网络就是将图数据和神经网络进行结合，在图数据上面进行端对端的计算。图神经网络的计算过程总结起来就是聚合邻居。如右面的动图所示，每个节点都在接收邻居的信息。为了更加全面的刻画每个节点，除了节点自身的属性信息，还需要更加全面的结构信息。所以要聚合邻居，邻居的邻居.....

单层的神经网络计算过程：

$$H = \sigma(XW)$$

相比较于神经网络最基本的网络结构全连接层（MLP），特征矩阵乘以权重矩阵，图神经网络多了一个邻接矩阵。计算形式很简单，三个矩阵相乘再加上一个非线性变换。

$$H = \sigma(A X W)$$



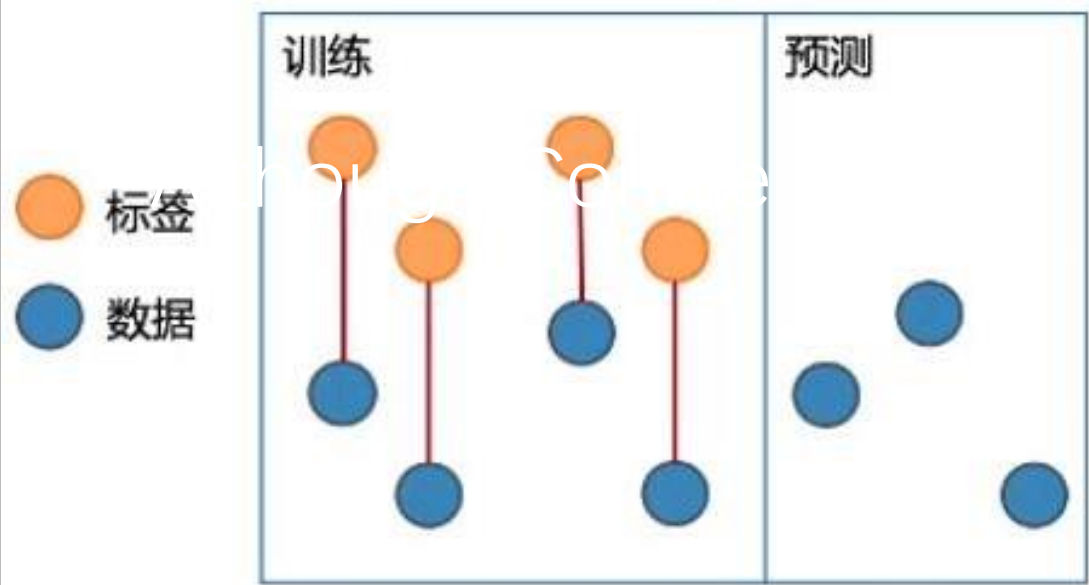
图网络特点

MOVIES

SHORT FILMS

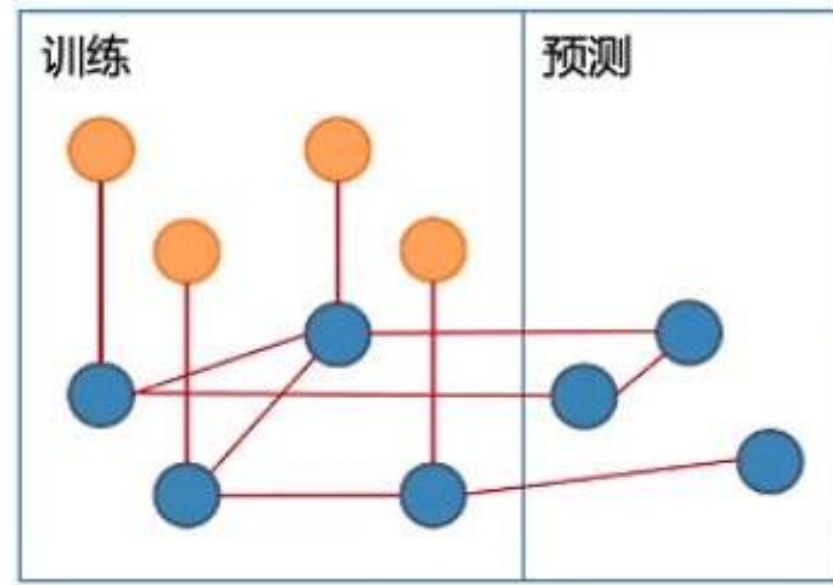
BEHIND THE SCENESS

一般机器学习场景



Niki Wu

图网络场景



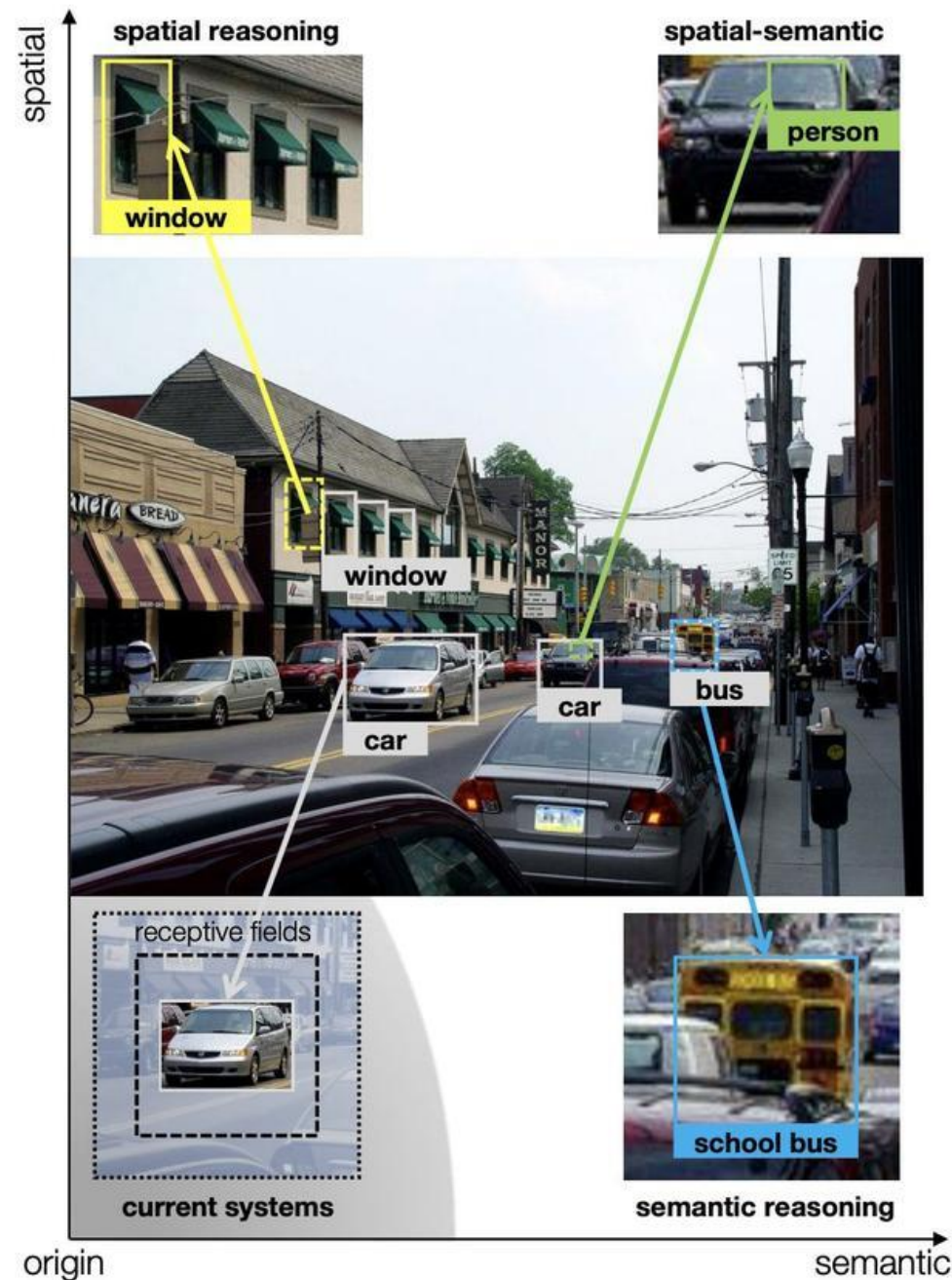
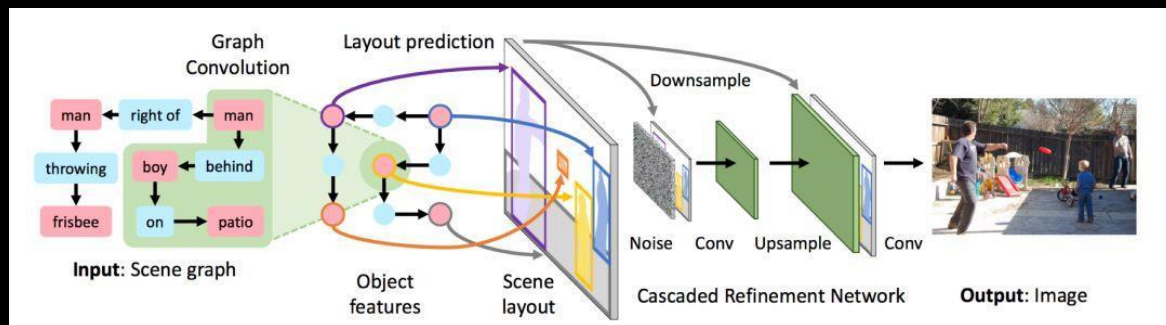
半监督节点分类



应用场景

万事万物皆有联系，节点+关系这样一种表示足以包罗万象。图数据可以说是一种最契合业务的数据表达形式。

计算机视觉、视觉推理





03 算法实现

PART



图网络配置



这里已经有很多强大的模型配置，可以尝试简单的改一下config的字段。例如，换成GAT的配置：

```
config = {  
    "model_name": "GAT",  
    "num_layers": 1,  
    "dropout": 0.5,  
    "learning_rate": 0.01,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.00,  
}
```


UniMP

```
In [5] from easydict import EasyDict as edict
```

```
config = {  
    "model_name": "UniMP",  
    "num_layers": 3,  
    "hidden_size": 16,  
    "heads": 2,  
    "learning_rate": 0.001,  
    "dropout": 0.3,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.3,  
    "use_label_e": True  
}
```

```
config = edict(config)
```



2020-09-20 14:22

百度又有“大动作”？9月18日，百度正式公布在图神经网络领域取得新突破，传递和图神经网络的统一模型UniMP（Unified Message Passing），在图神经网络单OGB（Open Graph Benchmark）取得多项榜首，引发业界关注。

Leaderboard for ogbn-products

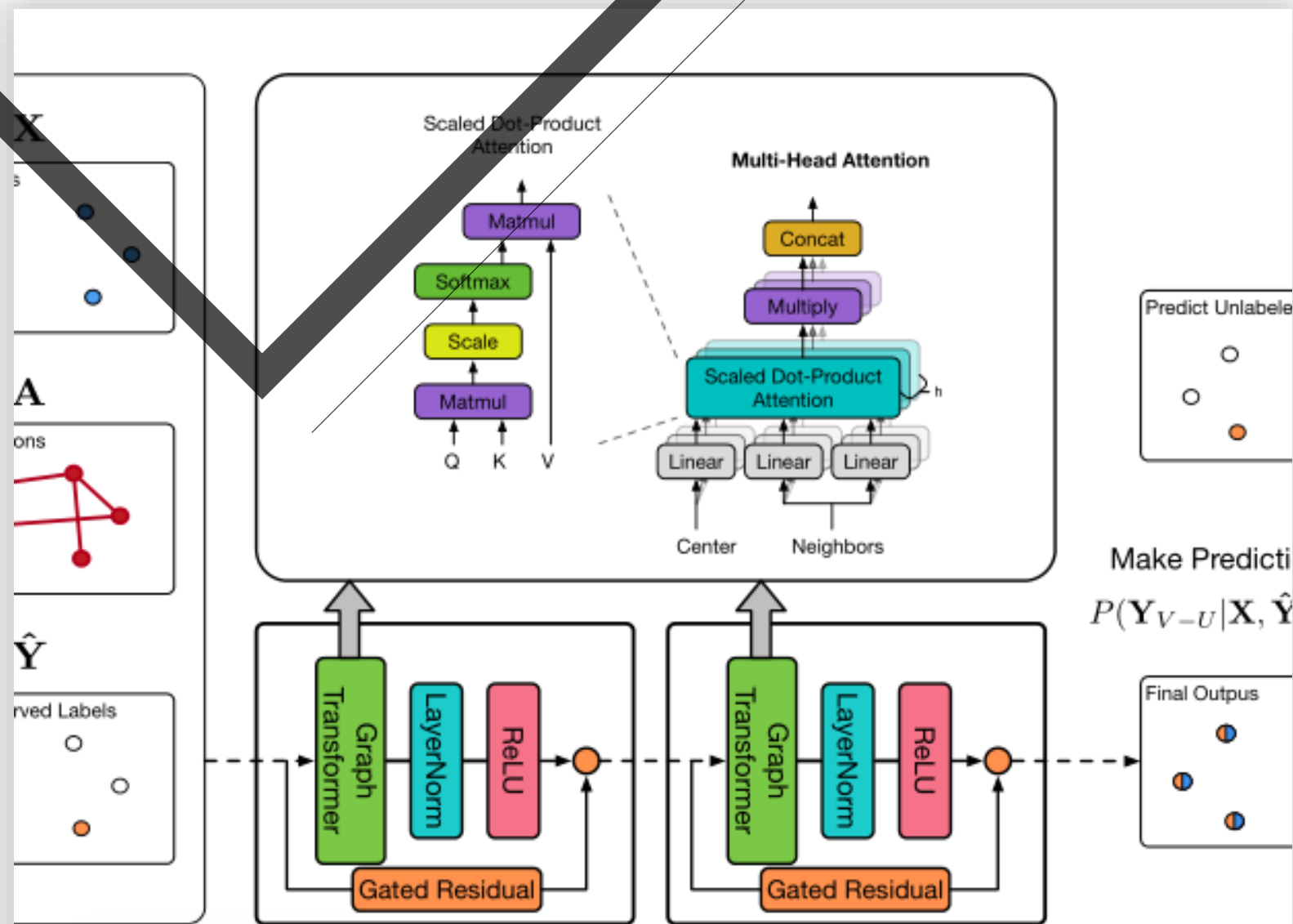
Leaderboard for ogbn-products

The classification accuracy on the test and validation sets. The higher, the better.

Package: >=1.1.1

	Test Accuracy	Validation Accuracy	Contact	References	#Params	Hardware
Niki Wu	0.8256 ±	0.9308 ± 0.0017	Yunsheng Shi (PGL team)	Paper, Code	1,475,605	Tesla V100

一般应用于半监督节点分类的算法分为图神经网络和标签传递算法两类，它们都是通过消息传递的方式进行节点标签的学习和预测。百度PGL团队提出的统一消息传递模型UniMP，将上述两种消息统一到框架中，同时实现了节点的特征与标签传递，显著提升了模型的泛化效果。



Frustrating it 3:00



Figure 1: The architecture of our UniMP.

训练结果



运用UniMP训练

次数

1000

Number

2000

团队成员用户名

acc

提交状态

提交时间

长夜

0.7426

完成

2021-05-31 22:05

长夜

0.74005

完成

2021-05-31 12:03

```
model_name": "ResGAT",  
num_layers": 3,  
hidden_size": 64,  
heads": 4,  
learning_rate": 0.05.
```

01

02

0.75455

```
UniMP(object):  
    __init__(self, config, num_class):  
        self.num_class = num_class  
        self.num_layers = config.get("num_layers", 2)  
        self.hidden_size = config.get("hidden_size", 64)  
        self.out_size = config.get("out_size", 40)  
        self.embed_size = config.get("embed_size", 100)  
        self.heads = config.get("heads", 8)  
        self.dropout = config.get("dropout", 0.3)  
        self.edge_dropout = config.get("edge_dropout", 0.3)  
        self.config = config.get("use_label_e", False)
```

03

04

```
config = {  
    "model_name": "UniMP",  
    "num_layers": 3,  
    "hidden_size": 128,  
    "heads": 2,  
    "learning_rate": 0.001,  
    "dropout": 0.1,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.3,  
    "use_label_e": False  
}
```


New Achievements

Break slide

Break slide

长夜的团队	0.75621	2021-06-03 23:38
-------	---------	------------------

团队成员用户名	acc	提交状态	提交时间
长夜	0.7426	完成	2021-05-31 22:05



04. 模型对比

PART



ResGCN &GCN

```
config = {  
    "model_name": "ResGCN",  
    "num_layers": 3,  
    "hidden_size": 128,  
    "learning_rate": 0.001,  
    "dropout": 0.1,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.3  
}
```

```
config = {  
    "model_name": "GCN",  
    "num_layers": 3,  
    "hidden_size": 128,  
    "learning_rate": 0.001,  
    "dropout": 0.1,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.3,  
}
```

ResGCN

团队成员用户名	acc	提交状态	提交时间
一叶幡过	0.72635	完成	2021-06-06 13:45

GCN

长夜	0.71204	完成	2021-06-21 20:22
----	---------	----	------------------

ResGAT &GAT

```
config = {  
    "model_name": "ResGAT",  
    "num_layers": 3,  
    "hidden_size": 128,  
    "num_heads": 2,  
    "learning_rate": 0.001,  
    "feat_drop": 0.6,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.3,  
    "attn_drop": 0.6  
}
```

```
config = {  
    "model_name": "GAT",  
    "num_layers": 3,  
    "hidden_size": 128,  
    "num_heads": 2,  
    "learning_rate": 0.001,  
    "feat_drop": 0.6,  
    "weight_decay": 0.0005,  
    "edge_dropout": 0.3,  
    "attn_drop": 0.6  
}
```

ResGAT

一叶幡过

0.72244

完成

2021-06-06 14:36

GAT

一叶幡过

0.66744

完成

2021-06-21 20:31





既然Res技巧能让GAT、GCN
提点，加入Unimp会不会有用？



最好结果

修改前

```
dropout_implementation='upscale_in_train')
for i in range(self.num_layers - 1):
    ngw = pgl.sample.edge_drop(graph_wrapper, edge_dropout)
    feature = self.get_gat_layer(i, ngw, feature,
                                hidden_size=self.hidden_size,
                                num_heads=self.heads,
```

长夜

0.75937

修改后

```
#改变输入特征维度是为了Res连接可以直接相加
feature = L.fc(feature, size=self.hidden_size * self.heads, name="init_feature")

for i in range(self.num_layers - 1):
    ngw = pgl.sample.edge_drop(graph_wrapper, edge_dropout)
```

完成

2021-06-15 00:19

```
concat=False, layer_norm=False, relu=False, gate=True,
pred = L.fc(
    feature, self.num_class, act=None, name="pred_output")
return pred
```

```
if dropout > 0:
    feature = L.dropout(feature, dropout_prob=dropout,
                        dropout_implementation='upscale_in_train')
```

```
# 下面这行便是Res连接了
feature = res_feature + feature
```

```
feature, attn, cks = graph_transformer(str(self.num_layers - 1), ngw, feature,
                                       hidden_size=self.out_size,
                                       num_heads=self.heads,
                                       concat=False, skip_feat=True,
                                       layer_norm=False, relu=False, gate=True)
```

```
feature = attn_appnp(ngw, feature, attn, alpha=0.2, k_hop=10)
```




05 提升方法

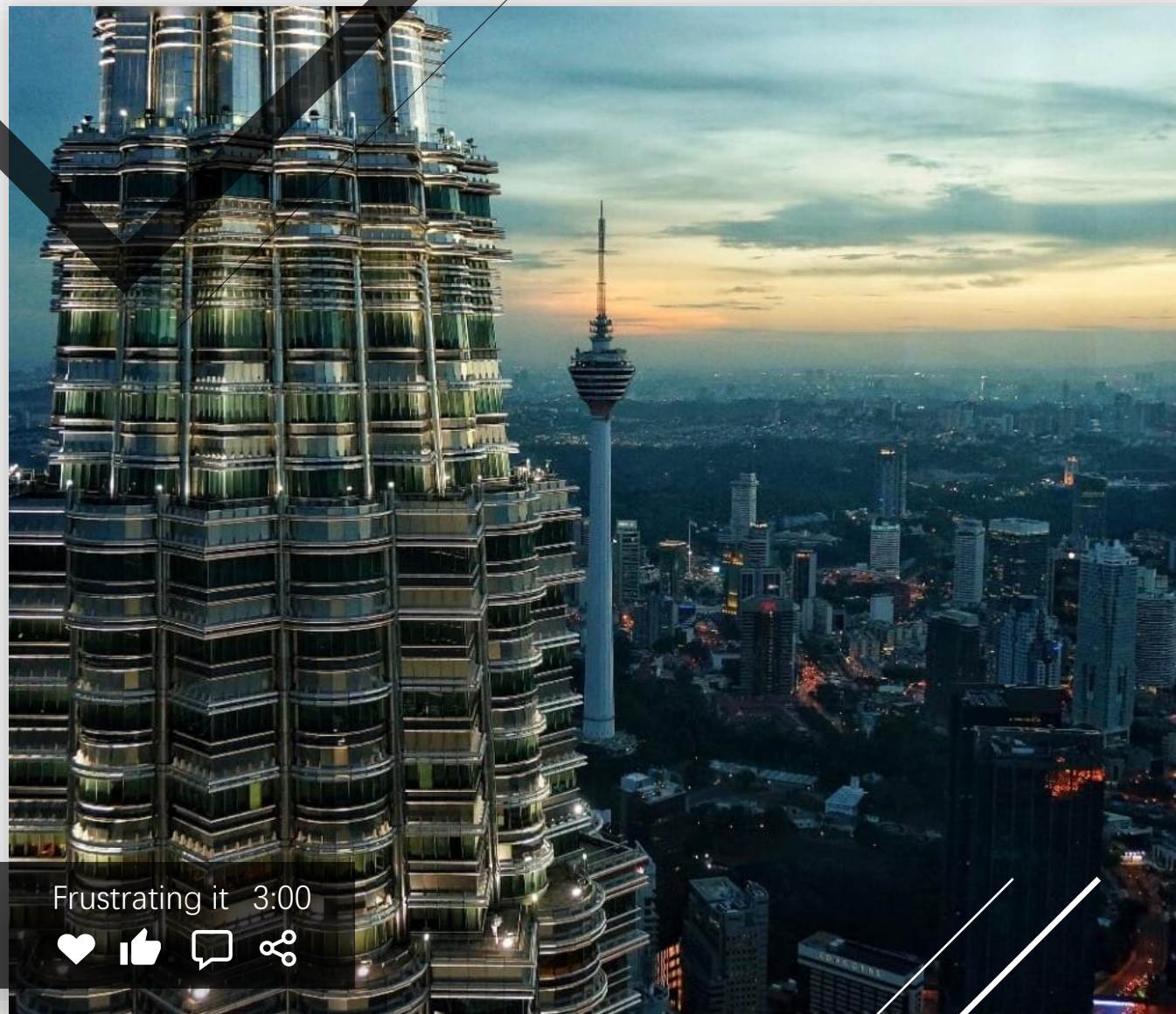
PART



绝对多数投票

绝对多数投票法很简单，分类器的投票数超过半数便认可预测结果，否则拒绝。

将所有提交文件的名称改为“测试精度.csv”，例如0.76087.csv；然后按照精度大小排序，首先使用绝对多数投票法进行投票，若某一投票不过半数，直接取精度最高csv的预测结果。



```
: import csv
#df10=pd.read_csv("submission10.csv")

nids=[]
labels=[]

for i in range(df4.shape[0]):
    label_zs=[]
    label_zs.append(df0.label[i])
    label_zs.append(df1.label[i])
    label_zs.append(df2.label[i])
    label_zs.append(df3.label[i])
    label_zs.append(df4.label[i])
    label_zs.append(df5.label[i])
    label_zs.append(df6.label[i])
    label_zs.append(df7.label[i])
    label_zs.append(df8.label[i])
    #label_zs.append(df9.label[i])
    #label_zs.append(df10.label[i])
    lab=publicnum(label_zs, d = 0)
    labels.append(lab)
    nids.append(df4.nid[i])

submission = pd.DataFrame(data={
    "nid": nids,
    "label": labels
})

submission.to_csv("submissiona.csv", index=False)
```


长夜

0.76235

完成

2021-06-14 16:59

202106

202105

202104

202103

202102

202101

202012

202011

排名

参赛团队

acc

提交时间

1

jsdbzcm的团队

0.76479

2021-06-09 18:12

2

飞雪の夏至的团队

0.7642

2021-06-08 11:39

3

孤木成林5的团队

0.7642

2021-06-11 11:45

4

长夜的团队

0.76235

2021-06-14 16:59



THANKS

—