

GGNN: Gated Graph Neural Network (门控图神经网络)

基本概念

GGNN 是一种基于 GRU 的经典的空域 message passing 的模型

问题描述

一个图 $G = (V, E)$, 节点 $v \in V$ 中存储 D 维向量, 边 $e \in E$ 中存储 $D \times D$ 维矩阵, 目的是构建网络 GGNN. 实现每一次参数更新时, 每个节点既接受相邻节点的信息, 又向相邻节点发送信息

主要贡献

基于 GRU 提出了 GGNN, 利用 RNN 类似原理实现了信息在 graph 中的传递

传播模型

$$h_v^{(1)} = [x_v^T, 0]^T \quad (1)$$

$h_v^{(1)}$ 是节点 v 的初态, 是 D 维向量,

当节点输入特征 x_v 维度小于 D 时, 在后面补 0

$$a_v^{(t)} = A_v^T : [h_u^{(t-1)T} \dots h_v^{(t-1)T}]^T + b \quad (2)$$

$a_v^{(t)}$ 是 $2D$ 维向量, 表示节点和相邻节点间通过 edges 的相互作用的结果, A_v 是矩阵 A 中选出对应节点 v 的两列

update gate $\leftarrow \boxed{z_v^t} = \sigma(W^z a_v^{(t)} + U^z h_v^{(t-1)})$ (3)

reset gate $\leftarrow \boxed{r_v^t} = \sigma(W^r a_v^{(t)} + U^r h_v^{(t-1)})$ (4)

产生的新信息 $\leftarrow \boxed{\tilde{h}_v^{(t)}} = \tanh(W a_v^{(t)} + U(\boxed{r_v^t} \odot h_v^{(t-1)}))$ (5)

最终更新的节点状态 $\leftarrow \boxed{h_v^{(t)}} = \underbrace{(1 - z_v^t)}_{\text{遗忘}} \odot h_v^{(t-1)} + z_v^t \odot \tilde{h}_v^{(t)}$ (6)

(3)-(6) 类似于 GRU 的计算过程, z_v^t 控制遗忘信息,

r_v^t 控制新产生信息, \rightarrow 选择遗忘哪些信息, \rightarrow 决定从哪些过去的信息中产生新信息.

输出模型

有两种输出:

- ① 每个结点分别输出值
- ② 整张图输出一个值

①: $O_v = g(h_v^{(T)}, x_v) \Rightarrow$ 利用每个 nodes 的最终状态和初始输入分别求输出

② $h_g = \tanh(\sum_{v \in V} \sigma(i(h_v^{(T)}, x_v)) \odot \tanh(j(h_v^{(T)}, x_v)))$

i, j 表示神经网络, 输入: $\text{concat}(h_v^{(T)}, x_v)$

输出: 实值向量

\downarrow
- 一种 attention 机制

用于选出哪些节点和整个 graph 的输出最相关