
GNN Final Project

鲁东^①

2001213518

信息科学技术学院

刘墨杨

2001111388

信息科学技术学院

摘要

我们在 GAE(Graph Auto Encoder) 的基础上, 引入具有 Hessian Penalty 的 GAE。我们使用自动编码器模型 (VGAE), 它使用图卷积网络 (GCN) 编码器和简单的内积解码器。Hessian Penalty 是一个简单的正则化术语, 它鼓励将生成模型的 Hessian 的输入视为对角线。我们对链接预测任务进行了简单的实验。

代码: <https://github.com/mickelliu/GNNFinalProject>

1 介绍

深度学习在许多领域都表现出良好的性能, 并且被广泛用于各个领域的研究。但是, 这种基于深度学习的方法也存在很多问题, 其中无法解释训练模型的权重如何影响预测结果是一个典型的问题。最近有解耦潜在因子的研究, 让嵌入中的每个元素对应一个单独的影响因素。因此, 在本文中, 我们将参考先前发表的论文, 并将潜在因子的解耦应用于图自动编码器, 并将其用于链接预测任务。在第二节中, 我们将首先介绍本文相关研究, 包括图形自动编码器, 潜在因子的解耦研究。在第三节中, 我们将讨论模型和 Hessian Penalty, 在第四节中, 介绍进行实验所用到的数据集, 展示实验结果并对结果进行分析。

2 相关研究

图自动编码器 2016 年, Thomas 提出了基于自动编码器进行链接预测的 GAE 和 VGAE。这两个模型是通过解码-编码的结构去获取到图中节点的嵌入 (Z), 然后再去做具体的下游任务比如链接预测。编码部分直接使用 GCN 获得图中每个节点的嵌入, 解码部分直接采用内积作为编码器重构原始的图。ARGA 是一个自动编码器 (Auto-Encoder) 和生成式对抗网络 (GAN) 的融合模型, 整体模型包括 GAE 和判别器 (Discriminator), GAE 的编码器和解码器仍然分别使用 GCN 和内积, 用判别器使模型的嵌入 (Z) 拟合高斯分布。这两者模型都在链接预测任务获得了比较好的结果。

24 **潜在因子的解耦** 已经在与深度学习相关的许多领域中，正在研究潜在因素的解耦性。
 25 在图神经网络领域，ICML 上的一篇论文 Disentangled GCN 提出了 Disentangled Con-
 26 volutional Layer 和新的邻居路由机制获得了更好的解耦性。而且它在顶点分类任务上
 27 获得了比较好的结果，证明了解耦对分类结果的影响。自动编码器 (A-E) 领域上， β -
 28 VAE 在损失函数的 KL 上加一个超参数 β ，让 KL 更小。它使得潜在因素的分量独立，
 29 如果 β 取 1 的时候，它就是典型的 VAE。生成式对抗网络领域是潜在因子的解耦性研
 30 究中最活跃的领域之一，其中 Hessian Penalty 令 hessian 矩阵的非对角线元素小（最
 31 好为 0）的想法，提高了模型的解耦性。

32 3 主要方法

33 3.1 Variational Graph Auto-Encoders

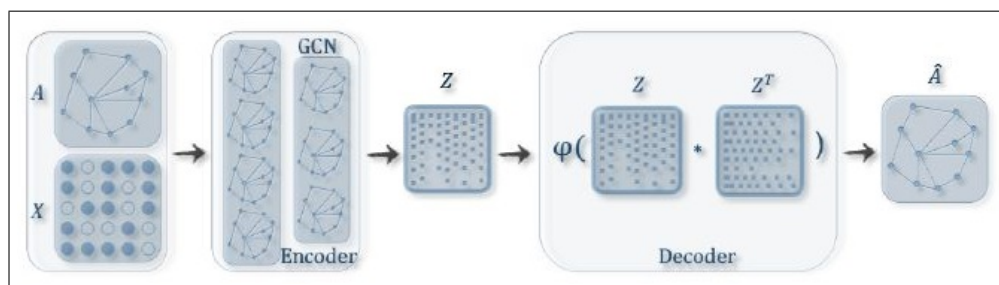


图 1: Variational Graph Auto-Encoders

34 本文中使用 VGAE 作为基本模型，它就是基于自动编码器 (A-E) 的一个图神经
 35 网络。它的编码部分从图信息获得图节点的嵌入 (Z)，解码部分从嵌入预测链接。使用
 36 GCN 作为模型的编码器，内积作为解码器。图 1 所示，输入图的邻接矩阵 A 和节点
 37 的特征矩阵 X ，编码器学习节点的特征向量生成嵌入，解码器生成图（预测链接）。解
 38 码部分一共使用三个图卷积层，第一层图卷积的输入为图的原节点特征向量 X 和连接
 39 矩阵 A ，第二层使用两个 GCN 来分别得到高斯分布的均值 μ 和方差 σ ，再将使其与
 40 随机生成的噪声相乘，相加，便得到高斯分布上采样到的一个嵌入 Z ，具体公式在 Eq1
 41 到 Eq3 所示。

$$X_1 = GCN(X, A) \quad (1)$$

$$\mu = GCN_{\mu}(X_1, A) \quad \sigma = GCN_{\sigma}(X_1, A) \quad (2)$$

$$Z = \mu + \epsilon \times \sigma \quad (3)$$

44 解码器由内积计算两个顶点之间存在链接的概率重构图，公式在 Eq4 所示。

$$A_{out} = sigmoid(Z^T, Z) \quad (4)$$

45 损失函数包括生成图和原始图之间的距离度量，以及节点表示向量分布和正态分布的
 46 散度两部分，损失函数公式在 Eq5 所示。

$$\mathcal{L}_{GAE} = E_{q(Z|X,A)} [\log(A|Z)] - KL [Q(Z|X,A)||p(Z)] \quad (5)$$

在 VGAE 的基础上，在损失函数的 KL 上加一个超参数 β ，让 KL 更小，这样可以使得潜在因素的分量独立。使用这种方法的模型叫做 β -VGAE，它的损失函数在 Eq6 所示

$$\mathcal{L}_{GAE} = E_{q(Z|X,A)} [\log(A|Z)] - \beta \times KL [Q(Z|X, A)||p(Z)] \quad (6)$$

当 β 大于 1 的时候可以获得解耦效果， β 等于 1 的时候，它就是一个 VGAE。

3.2 Hessian Penalty

Hessian Penalty 是本文中主要针对测试并使用的解耦方法，源于 W. Peebles 在 2020 年 ECCV 上发表的一篇同名工作。该方法主要用于 GAN 模型，设计初衷是通过计算 GAN 模型中生成器 (Generator) 所生成 Latent Representation 的 Hessian 损失函数，鼓励生成器生成解耦度更高的 Representation。这个方法简便易懂，应用方便，适用于大量设计 Encoding 的模型，故在这篇工作里我们应用该方法以检验它对 GCN 的优化效果。

通过计算 Hessian 矩阵优化解耦性的原理如 Eq 6 所示，Hessian 矩阵的 ij 元素代表的是潜在因子 z_i 与 z_j 之间的二阶导数。如果该导数为 0，那他们就是数学意义上的不相关，促成解耦。Eq 6 中 G 代表的是生成器函数或模型。

$$H_{ij} = \frac{\partial^2 G}{\partial z_i \partial z_j} = \frac{\partial}{\partial z_j} \left(\frac{\partial G}{\partial z_i} \right) \quad (7)$$

Hessian Loss 是 Hessian 矩阵非对角线元素元素之和 (Eq 8):

$$\mathcal{L}_H(G) = \sum_{i=1}^{|z|} \sum_{j \neq i}^{|z|} H_{ij}^2 \quad (8)$$

然而当 $|z|$ 也就是 Latent Representation 的维度增大时，涉及到大量的二阶导数计算导致计算效率的大幅度下降，因此作者提出了使用一个无偏随机估计量用于简化计算 (Eq 9, Eq 10):

$$\mathcal{L}_H(G) = Var_v(v^T H v) \quad (9)$$

其中:

$$v^T H v \approx \frac{1}{2} [G(z + \epsilon v) - 2G(z) + G(z - \epsilon v)] \quad (10)$$

ϵ 值为可调参数，参考值是 0.1。

由于图数据类型的特殊性，盲目在原始图数据上取梯度计算 Hessian Loss 并无意义，所以我们将原始数据中的特征数据 X 和图结构数据 A 进行一次编码，得到编码器输出 Z 也就是我们在本篇工作中计算 Hessian Penalty 时所用到的输入数据。训练整体 GAE 时所用到的损失函数使用 Eq 5 中的 \mathcal{L}_{GAE} 与 Hessian Loss 所组成:

$$\mathcal{L}_{Total} = \mathcal{L}_{GAE} + \mathcal{L}_H(G) \quad (11)$$

71 4 实验结果

72 我们进行三个实验, 在实验中均使用 Citeseer 数据集对该模型进行了链接预测。Hessian
73 Metrix、AUC (the area under a receiver operating characteristic curve) 和 AP (Average
74 precision) 作为实验的评价指标。

表 1: 用于实验的 Citeseer 数据集

Data Set	#Nodes	#Links	#Content Words	#Features
Citeseer	3,327	4,732	12,274,336	3,703

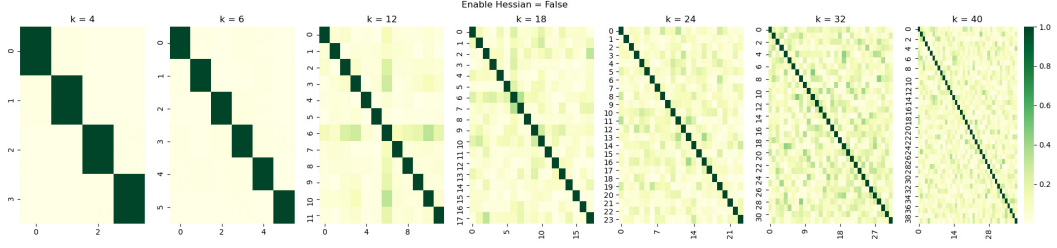
75 4.1 实验 1: 证实 Hessian Penalty 对于 Latent Representational Factors 的 76 聚合效果

77 在这个实验中, 我们主要验证的是 Hessian Penalty 对于 GCN 模型解耦性能的提升效
78 果。绝对相关图 (Absolute Correlation Plot) 是一种定性测量方法, 它能够直观的展现
79 出 \mathbf{z} 的解耦程度。如果任意图中的数值集中在对角线上, 便说明解耦程度高。反之如
80 果数值分布较为分散, 那么可以认为解耦程度较低。每两个潜在因子之间的相关度绝对
81 值越高, 数值越接近 1, 那么在图 2 中的颜色就越深 (绿色), 反之越接近 0 (无相关
82 性) 则越接近于浅黄色。在完美解耦的情况下, 潜在因子因独立对应于一个特征并且因
83 子之间相互独立, 那么 Latent Representation 的相关矩阵 (Correlation Matrix) 应当
84 为一个完全对角矩阵。注: 在本文中, $|z|$ = 潜在因子的数量 = k

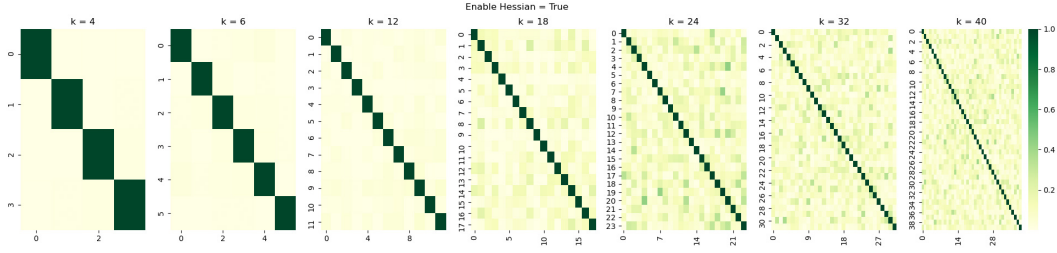
表 2: 三种模型在不同 k 下的平均测试精度, 单次试验数据

	$k = 4$	$k = 6$	$k = 12$	$k = 18$	$k = 24$	$k = 32$	$k = 40$
GAE	0.888	0.892	0.920	0.914	0.912	0.891	0.906
GAE + Hessian Penalty	0.883	0.886	0.919	0.910	0.895	0.909	0.905
β -GAE, $\beta = 1000$	0.830	0.785	0.817	0.812	0.823	0.818	0.836

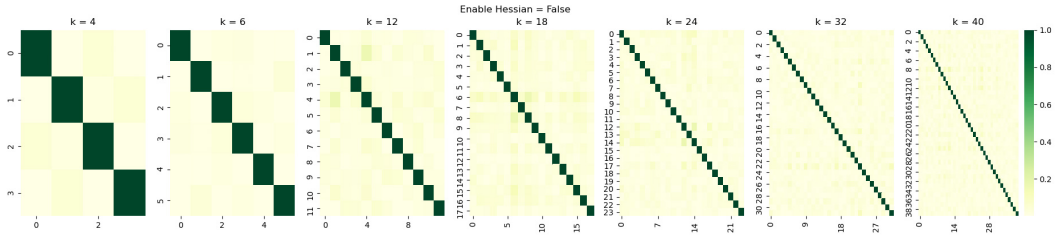
85 主观上看, 我们可以看出在 $k = 12$ 和 $k = 18$ 时 GAE+Hessian 的模型较于纯 GAE 相
86 比, 减小了数值分散, 然而在其他的情况下并无明显提升。原 GAE 在 k 小于等 6 时就
87 已经可以达成很大的解耦度, 而两者在 k 大于等于 18 时均表现出解耦都下降的区别。
88 精度方面, 单次数据结果表明 Hessian 对于预测效率有负面干扰, 精度平均下降了 1%
89 左右, 我们将在截下来的实验 2 中更准确的去认证两者之间的精度差。相比传统的在
90 GAE 中介入 β 参数从而达到解耦目的的方法, Hessian 虽然没有大幅度降低精度, 但
91 也未能综合情况下实现更明显的解耦效果。如 (c) 图中所示, 把 β 参数从 1 调到 1000
92 可以达到很好的解耦效果, 但是所承受的代价也是将近 10% 的精度下降。



(a) Pure GAE (No Hessian Penalty)



(b) GAE + Hessian Penalty



(c) β -GAE, $\beta = 1000$

图 2: Latent Factor Correlation Plot. GAE 模型在损失函数中加入 Hessian Penalty 前后的对比。两组模型除损失函数包不包含 Hessian Penalty 外的其他超参数均相同, 7 张图的横坐标为 Latent Representation \mathbf{z} 的 Latent Factor 总数 k , 分别为 $k = [4, 6, 12, 18, 24, 32, 40]$ 。

93 4.2 实验 2: Hessian Penalty 对于链接预测任务的影响

94 实验 2 的目的是证实 Hessian Penalty 是否能够真正的提升链接预测任务的效率。我们
 95 一共做了 3 组实验, 每个实验测试不同 k 参数下两个模型的预测精度的差距。为了保
 96 证数据的精确度 (Precision), 每个模型的结果均为 10 轮实验的平均, 并且我们提供了
 97 每个 AP 分数的标准误差值。

98 10 次实验数据平均结果表明, Hessian Penalty 对于链接预测精度是有负面影响的, 下
 99 降幅度约为 0.8% 至 1.1% 之间。本次实验中, Hessian Loss 在损失函数中权重为 $1E-5$,
 100 Hessian Loss 的大小与 \mathcal{L}_{GAE} 差距不大。

表 3: 两种模型在不同 k 下的平均测试精度, 10 次试验数据平均

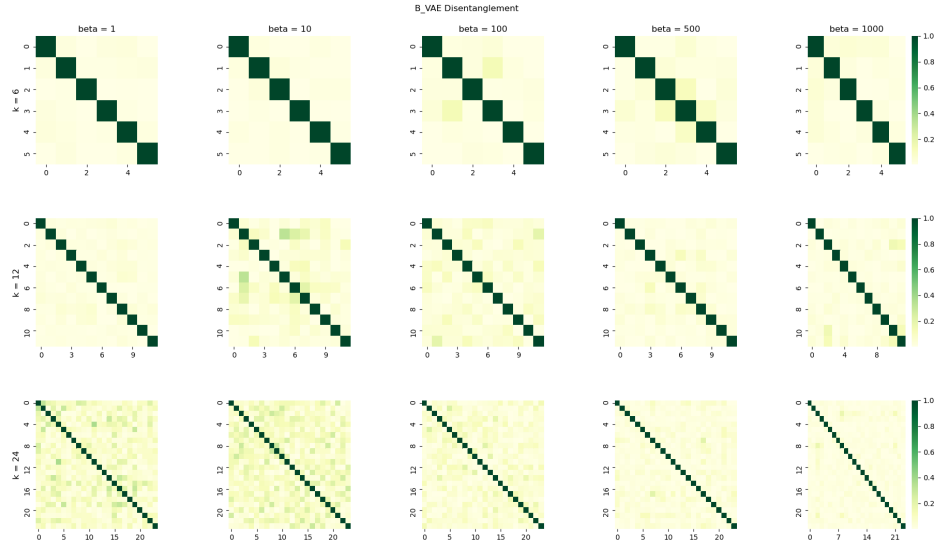
Models	$k = 6$	$k = 12$	$k = 18$
	AP Score	AP Score	AP Score
GAE	0.905 \pm 0.011	0.921 \pm 0.011	0.921 \pm 0.010
GAE + Hessian Penalty	0.896 \pm 0.010	0.913 \pm 0.009	0.910 \pm 0.009

101 4.3 实验 3: β 参数对于解耦性的影响以及其与 Hessian Penalty 的相互作用

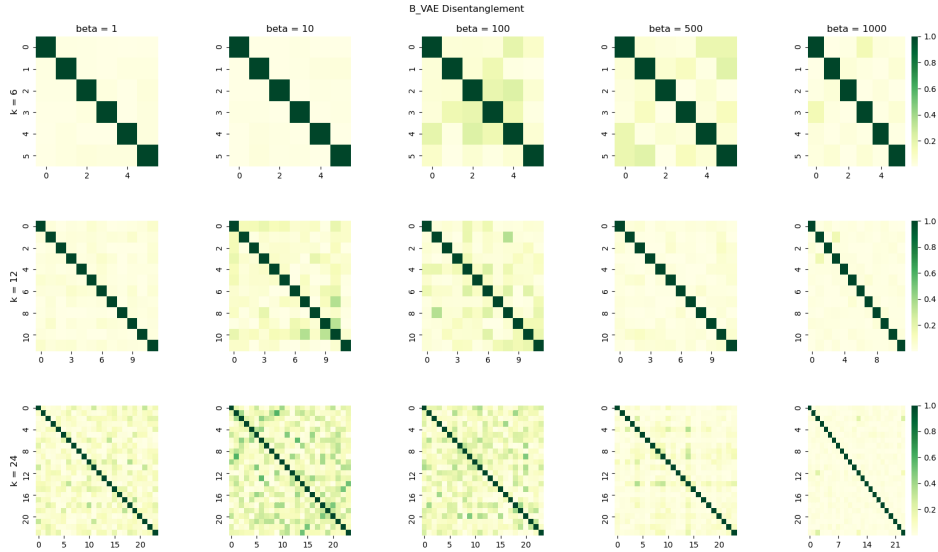
102 在以上两个实验使用的是纯 GAE 模型, β 参数预设为 1。 β 参数的应用原理与 Hessian
 103 Penalty 类似, 设计初衷都是提高模型解耦性能。在这个实验中, 我们将调整 β 参数
 104 的大小来分析它对解耦度的影响程度。另外, 在其中的一个实验中我们会将 β 参数与
 105 Hessian Penalty 结合使用, 观察两者的相互作用。

表 4: 改变 k 与 β 两个变量与添加 Hessian Penalty 所产生模型的平均测试精度, 单次试验数据

	$\beta = 1$	$\beta = 10$	$\beta = 100$	$\beta = 500$	$\beta = 1000$
β -GAE ($k=12$)	0.918	0.883	0.840	0.842	0.820
β -GAE + Hessian Penalty ($k=12$)	0.927	0.891	0.853	0.781	0.758
β -GAE ($k=24$)	0.917	0.889	0.845	0.863	0.796
β -GAE + Hessian Penalty ($k=24$)	0.920	0.904	0.835	0.785	0.747



(a) β -GAE (No Hessian Penalty)



(b) β -GAE + Hessian Penalty

图 3: Latent Factor Correlation Plot. 从左到右分别为 $\beta = [1, 10, 100, 500, 1000]$, 从上到下分别为 $k = [6, 12, 24]$ 。例如第二行第四列所对应模型的两个变量分别为 $(\beta = 500, k = 12)$ 。我们可从以上实验结果表明, 当 Hessian Penalty 和 β 参数一起生效时导致解耦因子分散, 主要表现在 (b) 图中 $(\beta = [10, 100], k = 24)$ 这两张图。另外, β 参数越大, 产生的解耦效果越好, 但是代价是预测精度的下降, 故无法确认其他工作中所提出的解耦度和预测精度的正相关关系。

106 5 总结

107 我们参考相关研究, 在 VGAE 的基础上引入 Disentanglement 的思路, 使用 Hes-
108 sian Penalty 和调整 β 参数进行了三种实验。通过这三种实验表明, Hessian Penalty 和。
109 但是加 Hessian Penalty 却导致之后模型的精度低于纯 GAE, Penalty 达到对 Model
110 Disentanglement 更加显著的优化效果, 但是随之的代价就是精度的大幅度下降 (10%)。
111 我们认为导致这种结果有几种原因。首先, 链接预测任务相对而言不是很复杂, 简易的
112 GAE 模型就能达到比较高的精度 (90%)。第二, β -VAE 和 Hessian Penalty 都是针对
113 生成式模型的方法, 其主要设计初衷是为了提升无监督的生成任务中模型的 Disentan-
114 glement 效果, 对于监督学习任务的精度提升并无保障, 然而, 由于我们的实验是基于
115 监督学习, 任务中需首要考虑模型的精度问题。通过我们的实验结果, 我们认为模型的
116 Disentanglement 和精度可能是一个 Trade-Off 的关系。根据 J. Ma 在 2019 年提出的
117 Disentangled GCN 工作中, DisenGCN 在节点分类上获得了好的 disentanglement 和
118 精度, GAE 的编码部分使用 Disentangled 图卷积会有效果, 因此作者引出了 Disentan-
119 glement 和预测精度的正相关关系。然而在我们的实验设定下, 数据无法证实出类似的
120 结论。Disentanglement 的提升是否能够真正提升监督学习任务的精度还有待考究。最
121 后, Factor VAE 提出了解耦性和精度两方面都获得比较好结果的方法, 以后的研究可
122 以在 VGAE 的基础上引入 Factor VAE 中的优化方法进行, 以便证明在监督学习任务
123 中两者之间是否存在正相关关系。

124 参考文献

- 125 [1] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders, 2016; arXiv:1611.07308.
- 126 [2] Kipf, Thomas N and Welling, Max Semi-supervised classification with graph convolutional
127 networks arXiv:1609.02907
- 128 [3] Pan, Shirui and Hu, Ruiqi and Long, Guodong and Jiang, Jing and Yao, Lina and
129 Zhang, Chengqi Adversarially regularized graph autoencoder for graph embedding, 2018;
130 arXiv:1802.04407.
- 131 [4] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros and Antonio Torralba. The Hessian
132 Penalty: A Weak Prior for Unsupervised Disentanglement, 2020; arXiv:2008.10599.
- 133 [5] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Des-
134 jardins and Alexander Lerchner. Understanding disentangling in -VAE, 2018; arXiv:1804.03599.
- 135 [6] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, Wenwu Zhu. Disentangled Graph Convolutional
136 Networks, 2019; Proceedings of the 36th International Conference on Machine Learning, PMLR
137 97:4212-4221.
- 138 [7] Kim, Hyunjik and Mnih, Andriy. Disentangling by factorising, 2018; arXiv:1802.05983.