READ 网络复现

胡昕

摘要

布局是图形学与人机交互研究的重要课题之一。随着人工智能以及深度学习相关技术的发展，出现了大量使用神经网络发展对布局进行研究的工作。这些工作主要集中在布局的解析与生成上，借用神经网络理解布局的结构并使用生成模型生成。READ 是其中一片比较重要且基础的研究工作，其将 VAE 编解码结构应用于布局生成，做到了较好的效果。但是其没有提供代码，因此，本文研究尝试在论文的基础上对论文提出的网络结构进行复现测试。

关键词：READ；布局；VAE 结构；

1 引言

布局，作为图形设计的基本要素，在有效传达信息方面是十分重要的，良好的布局能够吸引目标的注意力并能有效传达信息。同时，布局的设计也在众多的文档中所应用并被给予重视——报纸、网页、程序界面等。

但是，在不同的应用场景下设计出多样且美观的各种布局设计是一件繁琐任务，需要满足诸多的格式风格等的要求，这些要求可以是局部元素之间的，也可以是全局相关的。

而近年来内容生成的研究工作更多的是把重心放到图像、音频以及 3D 内容的生成上，与之相对，在生成合理的布局的任务上，研究的内容则较少。

复现的目的主要在于使用程序生成合理的布局，而要生成这样的布局，主要需解决如下问题：如何合理地表示布局？如何生成符合前述表示方式的合理的新布局。

2 相关工作

2.1 传统的布局相关研究

这里传统的布局研究指不使用神经网络方法进行布局分析或生成的相关研究。针对布局分析，传统方法常采用启发式的方发进行研究，如《Structured Document Image Analysis》[1]、《The document spectrum for page layout analysis》[2]、《Document image analysis: A primer》[3]。针对布局生成，传统方法主要采用一系列的约束条件对输出 layout 进行约束，以达到理想效果，相关的文章有《Solving linear arithmetic constraints for user interface applications》[4]、《Hierarchical Layout Blending with Recursive Optimal Correspondence》[5]

2.2 基于神经网络的布局相关研究

基于神经网络在越来越多的领域被成功应用，与布局生成相关的神经网络模型也陆续被提出验证，早期有基于对抗生成网络（GAN）的 LayoutGAN[6]以及 LayoutGAN++[7]网络被应用于直接生成布局的图片，GAN 之后是本篇文章所复现的 READ[8]，旨在将布局中物件的长宽类别所在位置编码到 VAE 中的空间中，以此完成生成任务。最新的研究则更加着重于将 Transformer 网络结构应用于布局的研究中，如 BLT[9]，VTN[10]。

3 本文方法

3.1 VAE 结构

READ 主要使用 VAE 结构将已有布局进行编解码，VAE 的主要结构如图 1所示。

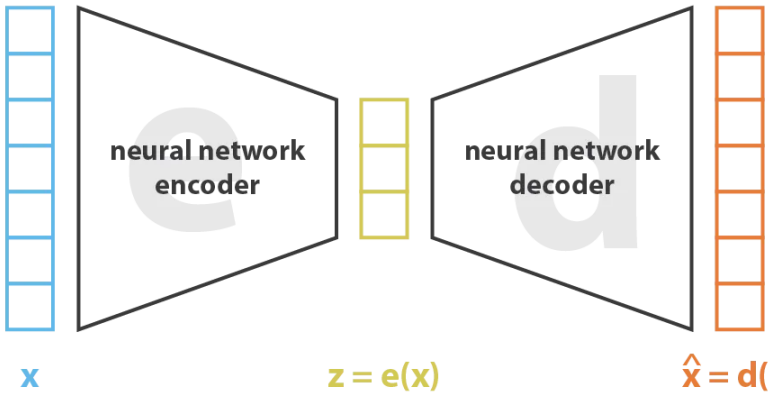

图 1: VAE 图示

其中 x 为输入特征，e 表示一个神经网络编码器，d 为神经网络解码器。e 以 x 为输入，经过神经网络的运算，将 x 转化到编码空间中的 z。d 以 z 为输入，根据编码空间中 z 的相关信息，尽可能地以 z 去解码复原出原来的 x，因此损失函数为复原的特征与原特征比较得出。以上即为一个基本的编码器-解码器结构。

与普通编解码器结构相比，VAE 在编解码的同时对 z 所在的空间进行了约束，使得这样的 z 的分布尽可能地满足正态分布。即在原损失函数的基础上引入了 KL 距离度量 z 分布于正态分布的距离，进行训练以是 z 分布接近正态分布。

3.2 布局结构表示

在使用神经网络进行编解码前，需要对布局的数据、层级结构进行定义。一个布局包括若干的边界盒（Bounding Box），每个边界盒包含（x，y，w，h，type）五项内容，x、y 为边界盒左上角点在整个布局中的位置，w、h 为边界盒的宽度与高度，type 表示当前边界盒的种类，其与数据集相关，可以有文字、图片、标题等类别。

更进一步地，为了突出边界盒之间的相互关系与层级结构，定义布局的树形结构如图 2所示。首先定义有两个边界盒之间的相对关系——Right、Bottom Right、Left、Buttom Left、Bottom、Wide Bottom。然后将布局中的边界盒从上往下，从左往右进行排序为序列 B，初始以第一个边界盒为目标盒。在构造树结构的每一步中，将 B 中未加入目标盒的第一个边界盒与目标盒比较得出相对关系，并按如图 2所示的方法将其加入到目标盒中得到新的目标盒，直到 B 中不存在未加入目标盒的边界盒。

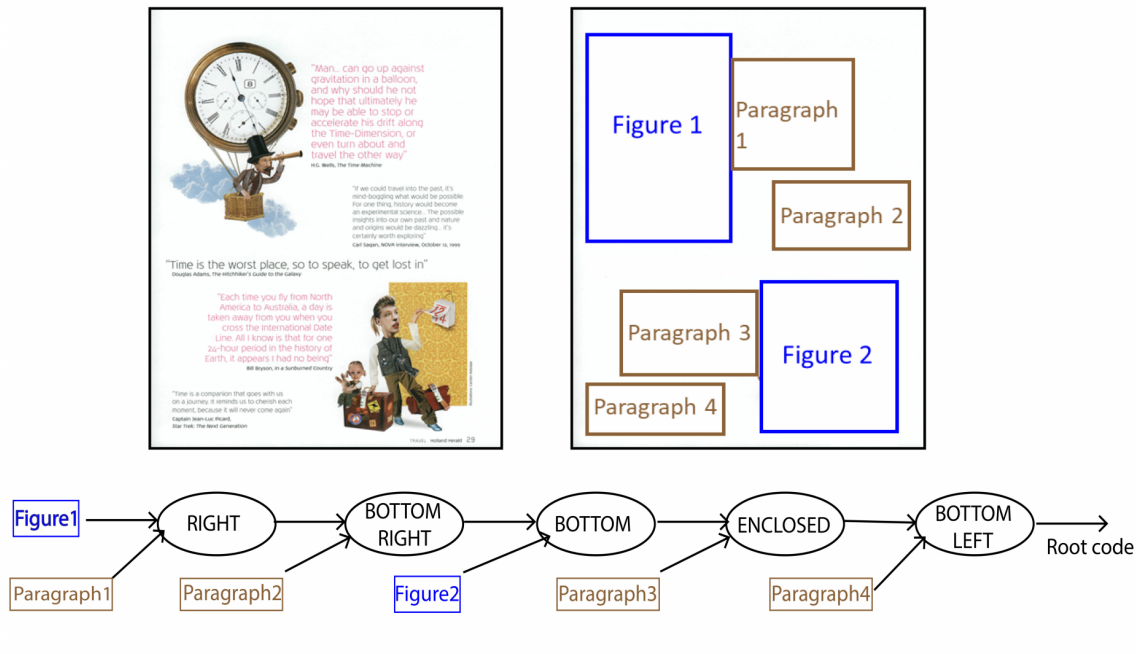

图 2: 布局的层级结构图示

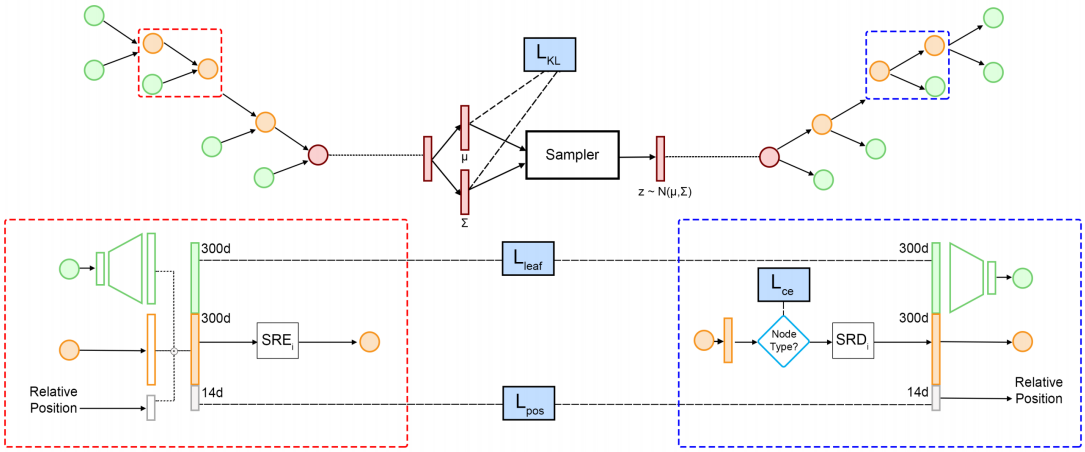

图 3: 布局的层级结构图示

3.3 神经网络结构与损失函数

READ 所构建的网络结构如图 3所示，上半部分为整体的结构，下半部分的红色蓝色虚线框对应整体结构中虚线框对应的结构。在整体结构的左边属性结构为由树结构递归调用编码结构的编码器，中间部分为 VAE 的采样器，右边部分为由递归调用解码器生成的文档树结构。其中绿色叶子节点表示的是在布局中每一个独立的边界盒的特征值，该特征值由边界和的五项属性经过同一个叶子节点特征提取网络得到。橙色节点对应上节提到的树形结构的中间特征，由当前已经经过编码的目标盒特征与新的边界盒特征以及相对位置编解码得到，针对不同的相对位置类型，分别使用不同的编解码器。红色节点则表示的是布局树经过编码得到的最终在 VAE 所定义空间中的特征。

根据以上结构引入损失函数 L_{KL}，L_{leaf}，L_{pos}，L_{ce}。L_{KL} 是 VAE 中使用的 KL 距离损失函数。L_{leaf} 是对于叶子节点还原程度的衡量，其表达式如下，其中，N 为布局中每个叶子节点的数量，b 为叶子节点的边界盒参数，b' 指经过解码器还原的对应叶子节点的边界盒参数。

$$L_{leaf} = \frac{1}{N}\sum_{i=1}^{N}(b' - b)^2$$

L_{pos} 是对于相对位置还原准确度的衡量，其表达式如下，r 代表的是节点之间相对位置的参数：

$$L_{pos} = \frac{1}{N-1}\sum_{i=1}^{N-1}(r^{'} - r)^2$$

L_{ce} 是在解码过程中，由于要使用不同的相对位置解码器解码不同的父节点，需要额外引入，因此引入对于每个处理节点的分类器以及对应的 L_{ce}，其为多分类问题中使用的标准交叉熵损失函数。

4 复现细节

4.1 与已有开源代码对比

本文章复现的 READ 在网上尚无任何代码，所有的代码皆为本人依据论文原文编写。

4.2 实验环境搭建

本文对于代码的复现使用 Python 平台以及 Pytorch 库编写，实验环境

CPU: AMD Ryzen R7 3700x

Memory: 32GBytes

GPU: NVIDIA RTX 3070

5 实验结果分析

如图 4所示为在 ICDAR 数据集上进行训练的网络参数，经过采样器随机采样生成得到的部分结果图。该结果主要可以说明了两方面的问题，第一就是，READ 网络模型可以从布局中学习到一些不同边界盒之间相对位置的关系，如上下对齐等，因此生成的部分结果与真实布局较为接近。另一方面 READ 模型虽然可以学习到相对位置，但是难以处理较为复杂的布局情况。

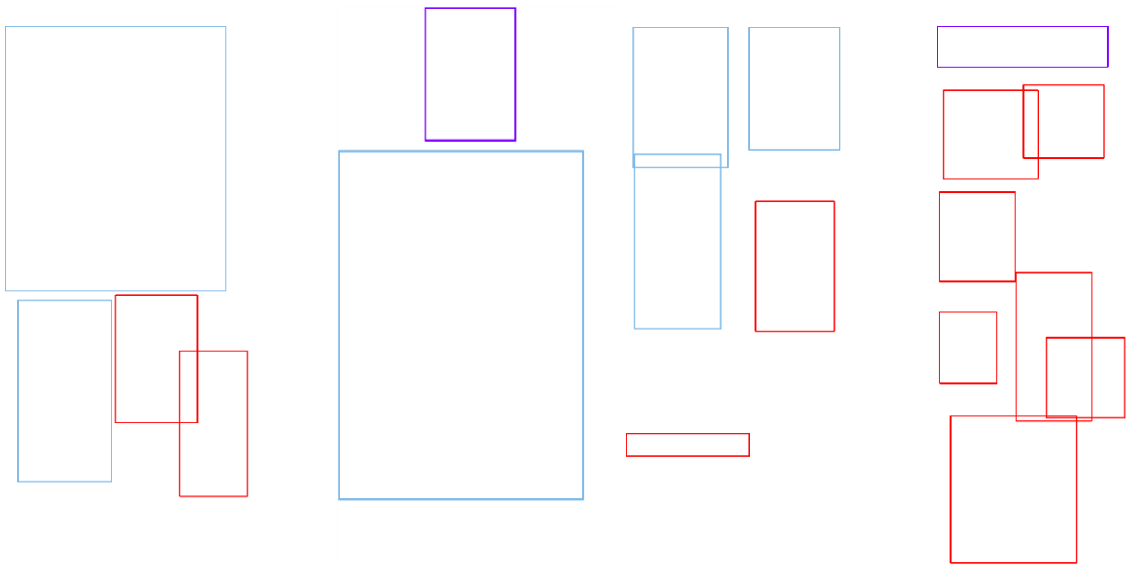

图 4: 生成结果示意

6 总结与展望

READ 模型在训练和结果上具有一定的缺点——使用的递归形式的神经网络，并行效率低导致训练过程的低效率，另一方面，布局最新的相关研究大部分都使用了 Transformer 网络结构，相比 READ 可以达到更好的效果。未来的研究一方面可以考虑更好的网络结构对布局问题进行优化，另一方面可以再更复杂的布局上进行研究。

参考文献

[1] HENRY S. BAIRD K Y, Horst Bunke. Structured Document Image Analysis[J]. Springer Berlin Heidelberg, 1992.

[2] O’GORMAN L. The document spectrum for page layout analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993, 15(11): 1162-1173.

[3] RANGACHAR KASTURI L O, GOVINDARAJU V. Document image analysis: A primer.[J]. Sadhana, 2002, 27(1): 3-22.

[4] BORNING A, MARRIOTT K, STUCKEY P, et al. Solving linear arithmetic constraints for user interface applications[C]//Proceedings of the 10th annual acm symposium on user interface software and technology. 1997: 87-96.

[5] XU P, LI Y, YANG Z, et al. Hierarchical Layout Blending with Recursive Optimal Correspondence[J]. ACM Transactions on Graphics (TOG), 2022, 41(6): 1-15.

[6] LI J, YANG J, HERTZMANN A, et al. Layoutgan: Generating graphic layouts with wireframe discriminators[J]. arXiv preprint arXiv:1901.06767, 2019.

[7] KIKUCHI K, SIMO-SERRA E, OTANI M, et al. Constrained Graphic Layout Generation via Latent Optimization[C]//MM ’21: ACM International Conference on Multimedia. 2021: 88-96. DOI: 10.1145/3474085.3475497.

[8] PATIL A G, BEN-ELIEZER O, PEREL O, et al. Read: Recursive autoencoders for document layout generation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 544-545.

[9] KONG X, JIANG L, CHANG H, et al. BLT: bidirectional layout transformer for controllable layout generation[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII. 2022: 474-490.

[10] ARROYO D M, POSTELS J, TOMBARI F. Variational transformer networks for layout generation [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13642-13652.