面向语言模型的神经元连接自动学习方法

姜雨帆,李 北,林 野,李垠桥,肖 桐*,朱靖波
（东北大学计算机学院 自然语言处理实验室， 辽宁 沈阳 110819）

摘要：在自然语言处理中，神经网络的结构需要人工设计，这导致复杂的神经网络结构中存在大量冗余。为了减少冗余人们常剪枝等模型压缩方法，但是这类方法通过对一些与训练过程无关的指标直接对模型进行裁剪，往往造成性能损失。因此探索了神经网络中神经元连接的自动学习方法，通过在训练中对神经元连接进行动态生长和删除的方法，可以更好的对网络连接进行动态操作，从而得到更紧凑、高效的模型结构。使用该方法在神经语言模型上进行自动生长和消去，在保证模型性能不变的前提下，模型规模可缩小 49%。

关键词：语言模型；神经元连接；剪枝

中图分类号：TP 391 **文献标志码：**A

基于神经网络的模型是目前自然语言处理广泛使用的框架，并且在很多任务中取得了令人瞩目的成绩。与传统神经网络一样，这类模型的最基本单元是神经元，神经元之间连接组成层，层和层之间连接构成网络。神经网络的结构有很多种，大部分神经网络都是根据先验知识进行人工设计的网络结构（如前馈神经网络、循环神经网络），通过训练网络中的参数来优化整个网络，这些网络结构在很多任务中都取得很好的效果，如语言模型[1]、机器翻译[2]、语音识别[3]、图像分类[4]等。这些强大的网络都得益于规模庞大的多层神经元结构，层与层之间采用全连接，模型参数量大，模型性能优越。但是这样的网络结构也存在大量的冗余[5]，而且人工设计的语言结构不容易捕捉神经元之间的关系，没有考虑到在模型训练过程中神经元连接动态变化的过程，网络结构的表示并不高效。

为了得到更加紧凑、有效的神经网络，研究人员提出了剪枝等模型压缩方法[6]。基本思想是裁剪掉一些不合理、低权重的连接。通过模型结构剪枝，将全连接裁剪为较稀疏结构，可以得到紧凑的网络结构。但是，这类方法与网络的训练无关，不是通过学习得到的，而是通过一些外部指标直接对模型进行裁剪。此外，这些方法仅仅考虑去除而没有考虑网络结构的增加，所以这种机械的剪枝方法无法得到一个最优的结果。

一种更为有效的方法是让神经元连接可以动态地生长和删除[7]。基于这个思路，本研究提出了一种动态网络连接的自动学习方法。动态的去学习网络的连接，这种连接生长和剪枝的方式完全和训练相关，参考了训练过程中的梯度，层和层之间信息流传递的有效性，可以更好的对网络连接进行动态操作，从而得到更紧凑、高效的模型结构。本文将使用动态网络连接方法对神经语言模型的连接进行自动生长和剪枝，使得在保证压缩后的模型性能不变的前提下，模型规模可进一步缩小为原本的 51%，同时当只保留 10%的连接数量的时候，仍能保证模型具有 176 的困惑度而对全连接网络保留相同连接数的困惑度为 230。

收稿日期：2018-11-16
基金项目：国家自然科学基金(61432013，61732005，61876035)；中央高校基本科研业务费；辽宁省高等学校创新人才支持计划
***通信作者：**xiaotong@mail.neu.edu.cn

1　神经语言模型

语言建模任务是通过某种方式对语言建立数学模型的过程。在神经网络出现之前，一般使用统计的方法来设计语言模型。比较常见的为 n-gram 模型，它对文本中若干词语出现的频率进行统计，并使用平滑算法对未见词语搭配进行修正，最终得到该语言中不同词语连续出现的概率值。神经语言模型相对传统基于统计的模型而言，能够在学习词语搭配的同时学习到词汇之间的相似性，相对平滑算法而言有效提高了对已知单词的未见搭配的预测效果，获得了更好的性能。

神经语言模型最早由 Bengio 等人[1]系统化提出并进行了深入研究，其整体结构和普通的前馈神经网络类似，由输入层、隐藏层和输出层组成，层和层之间存在连接，每一层将本层接收到的向量映射到另一维空间上作为该层的输出。在神经语言模型中，输入层将离散表示的词汇转化成连续空间上的词向量，不同词汇之间的向量距离能够反映词语之间的相似性。隐藏层将输入层传递来的词向量进行更深层次的表述。输出层结合隐藏层所传递来的信息对可能出现在下一个位置的词进行预测，输出当前状态下词汇表中每个词的预测概率。在神经网络的训练过程中比较常用的方法为随机梯度下降（stochastic gradient descent），通过反向传播计算得到损失函数 L 对参数 W 的导数$\frac{\partial L}{\partial W}$，之后我们根据式 1 对参数 W 进行更新，其中η是学习率。

$$W_{k+1} = W_k - \eta * \frac{\partial L}{\partial W_k}. \qquad (1)$$

2　生长裁剪模型

生长裁剪模型主要包括 2 个过程、4 个状态，从起始的种子结构，依次通过生长阶段和裁剪阶段达到最终的模型结构，转换过程如图 1 所示。这里我们借鉴 D. Xiaoliang 等工作[7]。

从种子结构开始，该结构由全连接的网络随机削减而成。第一个阶段是生长阶段，这个过程通过构建新的连接来使得网络结构达到所期待的性能（逼近全连接的性能）。第二阶段是裁剪，在对性能不影响或影响较小的情况下，从网络中删减掉部分权重绝对值较小的连接，使网络结构更加紧凑，同时降低模型在存储压力和计算消耗。通过上述的两个过程最后得到规模较小的模型结构，达到加速以及模型压缩的目的。整个过程的算法流程如算法 1 所示。

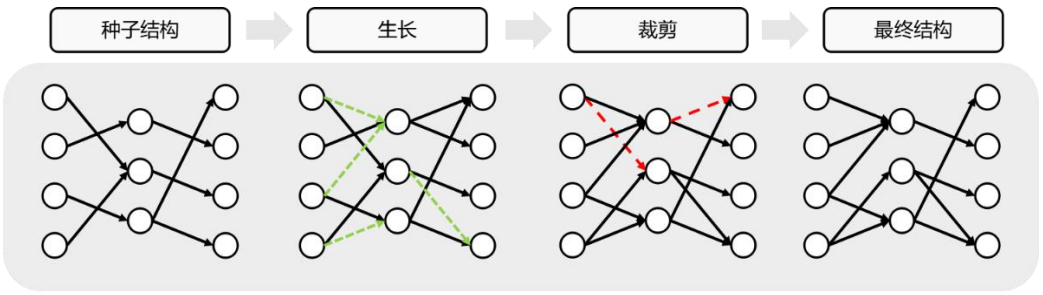

图 1 生长裁剪流程

Figure 1 Growth pruning process

算法 1 使用先生长后裁剪策略对网络结构进行学习	
输入：A - desired accuracy,Net-neural network	
1	*a* = test(Net),train(Net)
2	**If *a* < *A* then**
3	Repeat
4	Grow connections
5	Net <- train(Net)
6	*a* = test(Net)
7	Until *a* > *A*
8	**End if**
9	Repeat
10	Prune connections
12	*a* = test(Net)
13	Until *a* < *A*
14	Return Net

2.1 生长过程

种子结构并非全连接网络，因此本研究采用 mask 矩阵来对原本的连接状况进行遮盖，如图 2 所示。mask 为 0-1 矩阵，其中值为 1 代表两个神经元之间存在连接（激活状态），值为 0 代表其间不存在连接（未激活状态）。具体来说，第 l 层的种子结构的权重矩阵$\boldsymbol{W^l}$为该层中的 mask 矩阵$\boldsymbol{M^l}$和全连接权重$\boldsymbol{W^{l'}}$点乘后的结果，如式 2 所示：

$$W^l = M^l \odot W^{l'}. \tag{2}$$

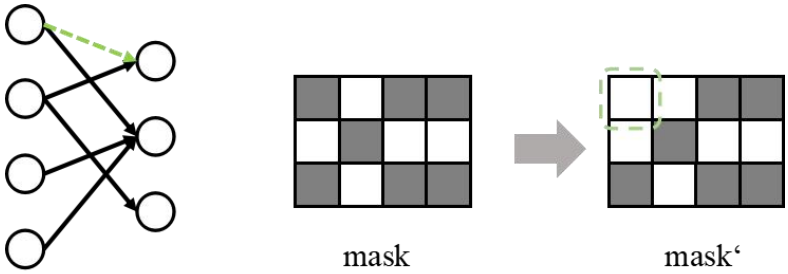

图 2 mask 矩阵操作流程

Figure 2 mask matrix operation

而在连接的生长过程中，该方法每次均将激活部分未激活的连接，将优先选择为优先能够使得模型损失值更快下降的未激活的连接进行激活。具体来说将对当前状态下全部的连接进行梯度的计算（包括被 mask 矩阵遮盖的连接，其值视为 0），然后根据梯度值的大小对连接重要性进行排序，对于其中重要且未激活的连接执行激活操作，使得网络能够在稀疏网络结构的基础上不断增加新的连接，促进整个模型快速收敛。该过程 mask 矩阵具体的计算公式如式 3 所示：

$$M_{t+1}^l[i][j] = \begin{cases} M_t^l[i][j], & \left|\frac{\partial L}{\partial W_t^l[i][j]}\right| \notin TopK\left(\left|\frac{\partial L}{\partial W_t^l}\right|\right), \\ 1, & \left|\frac{\partial L}{\partial W_t^l[i][j]}\right| \in TopK\left(\left|\frac{\partial L}{\partial W_t^l}\right|\right). \end{cases} \tag{3}$$

其中$M_{t+1}^{l}[i][j]$代表第 l 层 mask 矩阵在 t+1 时刻下第 i 行、第 j 列元素的值，W_t^l为第 l 层中权重矩阵 W 在 t 时刻下的值，L 代表网络的损失函数,。根据上述公式，模型即可根据权重矩阵在每一时刻的梯度，在未连接的节点之间构建新的连接，该过程即为连接的生长过程。

2.2 裁剪过程

虽然使用了连接生长的方式以尽可能减少不必要连接的产生，但由于初始化种子结构以及贪心策略下的连接增长均有可能给最终的网络结构带来部分非重要的连接，所以想要获得更紧凑、精简的网络结构仍需要使用传统的剪枝的方法对网络结构中非重要连接进行删减。具体来说，这里我们采取的方式同样是对 mask 矩阵进行操作，将权重矩阵中绝对值较小的位置使用 mask 矩阵进行置 0 遮盖，达到剪枝的目的。其中 mask 矩阵的计算如式 4 所示：

$$M_{t+1}^{l}[i][j]=\begin{cases}M_t^l[i][j], & W_t^l[i][j]\notin(0,thres),\\ 0, & W_t^l[i][j]\in(0,thres).\end{cases} \tag{4}$$

其中$M_{t+1}^{l}[i][j]$代表第 l 层 mask 矩阵在 t+1 时刻下第 i 行、第 j 列元素的值，W_t^l为第 l 层中权重矩阵 W 在 t 时刻下的值,$thres$ 代表权重的阈值，对权重小于这个阈值但不等于 0 的连接进行裁剪。据裁剪和生长操作执行的次序可以将生长裁剪方式中的裁剪方法分为先生长后裁剪以及边生长边裁剪两种方法。

2.2.1 先生长后裁剪

先生长后裁剪方法的核心思想在于将模型结构的生长和裁剪两个过程完全切分成两个阶段来操作。当网络在种子结构的基础上，通过若干次的迭代训练后生长出了一个性能稳定模型结构之后，裁剪阶段将根据权重矩阵中数值的大小对连接的重要性进行排名分析，最终对非重要连接进行删减，通过这种裁剪和重训练的相结合的方式达到模型性能和参数规模相互平衡的目的。这种裁剪方式的优势在于实现方式简单，但由于裁剪操作的执行对象已经是一个相对稳定的网络结构，即使使用了重训练的方式对裁剪后的结构进行补救也容易在模型性能上造成比较严重的损伤。

2.2.2 边生长边裁剪

针对先生长后裁剪方法存在的问题，将裁剪的操作次序提前，在生长过程中就伴随着裁剪的执行从一定程度上对原有问题进行了缓解。在原本的连接生长过程中，根据权重矩阵中梯度的大小在稀疏的网络结构中构建新的连接，而边生长边裁剪的方式将在这个过程中加入连接删减的操作，即根据当前状态下模型中连接的重要程度进行筛选，剔除掉非重要的网络连接。这样做的好处在于可以更早地对结构中的连接进行去留选择，一方面使得在训练过程中模型的规模相对较小，网络结构更易训练；另一方面将裁剪阶段提前可以有效使得削减后的网络有更加充分的时间进行重新训练，更易达到较好的模型效果。

3 实验系统

本研究相关实验使用自主研发的开源工具包，在 NVIDIA TITAN X(Pascal)设备上对前馈神经网络语言模型的网络结构进行学习。根据神经网络语言模型训练过程中的梯度指导网络结构中的连接生长，同时采用裁剪的方式对种子结构以及生长过程中产生的冗余连接进行剔除。将得到结果与面向全连接网络的普通裁剪操作进行性能以及模型规模上的对比分析。

本研究实验数据采用 Brown 数据集，（1 161 169 词，57 341 句），从中抽取 40 000 句作为训练集，统计 16 400 词作为词表。在模型参数方面，前馈神经网络的输入层、隐藏层节点个数分别为 100、256（128），每次处理数据批大小为 128。前馈神经网络训练 5-gram 语言模型。实验中，仅对隐藏层和输出层的权重进行连接生长。语言模型的评价采用困惑度（perplexity）作为标准（困惑度越低，模型性能越强）。主要对隐藏层和输出层中的连接数进行统计。全连接状态下隐藏层的总连接数为 256*400，输出层的总连接数为 256*16400。

主实验中，隐藏层分别设置为 256 和 128。首先在神经网络语言模型的隐藏层和输出层权重中随机初始化 20%的连接，作为连接生长的初始种子结构。每隔 10 个 step 新建一次连接，每次选梯度绝对值前 200 的连接。将它们中的没有连接在一起的神经元连接在一起。同时，训练参数设置相同的全连接的神经网络语言模型，并将隐藏层和输出层的权重按比例进行剪枝，与后面实验进行对比。

3.1 生长连接效果分析

从表 2 和图 3 的实验结果可以看出，根据梯度生长连接得到的网络结构在增长到全连接模型规模 50%的时候就可以达到与其相近的性能。然后对连接生长得到的最终结构与全连接的结构进行对比。同时对两种结构进行裁剪操作，这里我们的裁剪方法是裁剪掉权重绝对值较低的连接。例如当我们剪掉全连接模型 90%的连接数，语言模型的困惑度为 230.37。由于普通的剪枝操作从全连接的结构开始对参数进行削减，而连接生长的结构在开始削减的时候连接数仅为全连接的 50%左右，结果发现，全连接的结构在连接数减少 40%左右时会出现性能下降的情况，而通过生长连接得到的结构相比于全连接的网络结构，保留相同的连接数比例，性能要更加稳定，从某种程度说明了通过该方法得到的网络结构更加紧凑。这是由于生长中新建立的连接都是对网络拟合训练目标做出贡献的连接。在全连接的网络中，训练实际上是调整网络中的参数，让网络预测的结果与目标更加接近。但是，不是所有的连接都能在训练当中得到有效利用，从而对预测目标做出贡献。即常说的网络冗余，这些冗余绝大部分都是一些数值非常小的参数。如果在训练过程中只选择梯度下降大的两个神经元进行连接，这就使得网络中得到激活的均为所谓的有效连接，大幅度提升模型的紧凑程度。从图 3 中和表 2 中也可以证明这一点。

表 2 隐藏层分别为 256 和 128 裁剪比例

Table 2 hidden layers are 256 and 128

隐藏层数	剪枝比例/%	困惑度		
		全连接	先生长后裁剪	生长同时裁剪
隐藏层数 256	0	159.14	158.48	158.23
	10	159.09	158.48	158.23
	20	159.17	158.48	158.23
	30	159.30	158.48	158.23
	40	160.23	158.48	158.23
	50	161.23	158.48	158.23
	60	161.39	158.38	158.34
	70	166.87	159.08	158.53
	80	187.70	163.31	160.99
	90	230.37	177.94	176.76
隐藏层数 128	0	154.96	159.05	159.75
	10	155.01	159.05	159.75
	20	155.25	159.05	159.75
	30	155.51	159.05	159.75
	40	156.13	159.05	159.75
	50	157.10	159.06	159.76
	60	158.94	158.97	159.63
	70	164.46	159.83	161.20
	80	181.43	162.85	166.02
	90	227.55	179.92	186.10

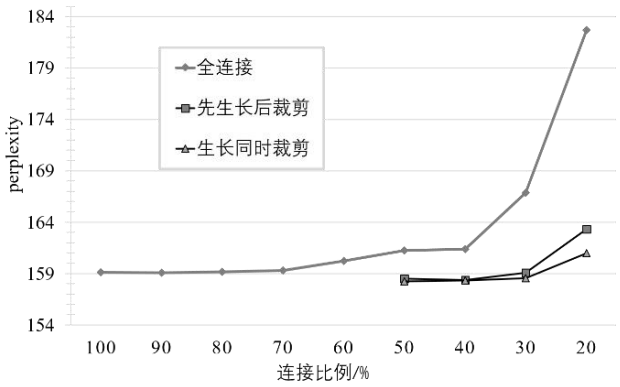

图 3 隐藏层为 256 时，三种裁剪方式裁剪比例的变化对困惑度的影响

Figure 3 When the hidden layer is 256, the effect of the cropping ratio of the three cropping methods on the confusion

3.2 生长连接同时裁剪效果分析

相比先生长后剪枝的方法，在生长的同时，对权重进行剪枝将会在性能和规模上得到更好效果。表 3 可以看出，无论是隐藏层还是输出层，在生长过程中增加剪枝的过程均会使得网络中的连接数变得更少。从表 3 的实验结果可以看出，在生长的过程中进行剪枝可以在不影响性能的情况下进一步减小网络的最终结构。边生长边裁剪的操作的时序提前，使得剪枝后的网络结构能够得到更充分的训练。所以我们要在训练的过程中引入裁剪操作，在训练的时候削减权重较小的连接，这样可以使网络的结构进一步精简。

表 3 不同生长时间间隔连接比例

Table 3 Connection ratio at different growth intervals

	隐藏层/%		输出层/%	
	先生长后裁剪	生长同时裁剪	先生长后裁剪	生长同时裁剪
隐藏层数 256	52.28%	45.54%	55.71%	51.66%
隐藏层数 128	61.69%	51.81%	55.28%	50.11%

3.3 其他参数的影响

隐藏层大小对连接生长的影响 从实验结果可以发现，当隐藏层大小为 256 时，仅依靠连接生长就可以达到全连接的效果。而当隐藏层大小为 128 时，采用连接生长的方法就无法达到全连接的效果。这是因为，隐藏层为 128 时，网络结构相比隐藏层大小为 256 模型要更加紧凑，自身存在的冗余也就更少，从图 3 中也可以看出，剪枝对全连接网络的性能影响更加明显。但是我们发现经过生长得到的结构和全连接结构相比依旧更加稳定。

种子结构的预先训练 实验中，发现，如果让种子结构预先得到有效的训练，那么在生长剪枝过程之后就很难达到最好的性能。如表 4 所示。当先训练 5epoch 和 10epoch 再增加连接时，语言模型的性能没有一开始就增加连接的效果好。这是因为，网络中的参数通过一段时间的训练后将达到一个平衡状态，当我们加入新的连接时，新的连接很难打破之前的平衡状态，导致新增加的连接不能有效参与到训练当中。

表 4 初始化不同种子结构连接比例

Table 4 Initialize the connection ratio of different seed structures

增加连接	连接比例/%		困惑度
	隐藏层	输出层	
直接开始 增加连接	52.28	55.71	158.48
先训练 5epoch	44.18	52.68	171.82
先训练 10epoch	41.59	51.62	179.87

增加连接的间隔 在实验中，增加连接的时间间隔和每次增加的连接数也很重要。将增加连接的时间间隔分为每 10 个、100 个、1000 个 step。为了保证增加总的连接次数相同。设置间隔越大，每次增加的连接数量越多。实验结果如表 5 所示。发现这三种时间间隔生长的模型性能接近，但是模型最终连接数不同，间隔越短，增加的连接数越少，网络结构就越紧凑。这是因为每次增加的连接数过多，可能会将模型中冗余的连接加入进来。

表 5 对种子结构进行预训练连接比例

Table 5 Pre-training connection ratio for seed structure

更新间隔	连接比例/%		困惑度
	隐藏层	输出层	
更新间隔 10steps	52.28	55.71	158.48
更新间隔 100steps	74.37	56.47	161.52
更新间隔 1000steps	95.80	65.45	159.87

种子结构的初始化 种子结构的设置初始化 10%，30%，50%的连接。结果如表 6、图 6 所示，当初始化 10%的连接时效果没有其他初始化效果好，其可能的原因是因为网络太过于稀疏，导致网络中大量的神经元节点被孤立，梯度无法有效传回，导致网络性能偏低。随着初始化的连接数越多，网络中的冗余也就越多。从图中可以看出，当网络中的连接数减少时，初始化 10%的网络表现的最为稳定，也就是说初始化 10%的网络结构最为紧凑。

表 6 不同初始化种子结构

Table 6 Different initialization seed structure

初始化	连接比例/%		困惑度
	隐藏层	输出层	
初始化 10%	38.62	48.02	160.84
初始化 30%	61.28	62.26	158.57
初始化 50%	77.28	74.09	157.68

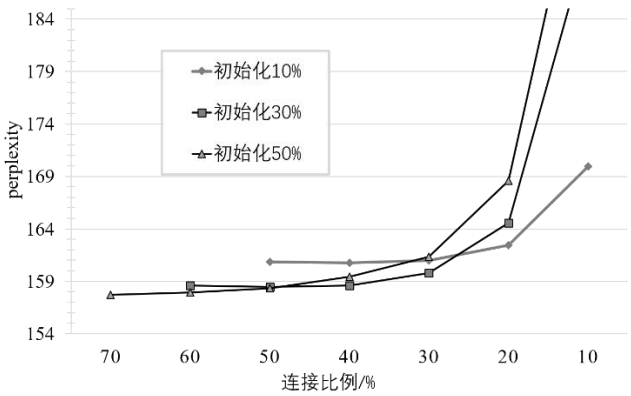

图 6 不同初始化种子结构连接比例

Figure 6 Different initialization seed structure

耗时分析 由于在训练的过程中加入了连接增长和裁剪的操作，这会导致训练的时间变长，以下时间是在隐藏层设置为 128，随机初始化 20%的连接，作为连接生长的初始种子结，每隔 10 个 step 新建一次连接，每次选梯度绝对值前 200 的连接。全连接、先生长后裁剪、生长同时裁剪的实验收敛时间分别为 2335.3247 和 4891s。时间的增长主要来源于对所有网络连接上的权值进行排序所需的时间。虽然耗时增加，但是整个网络结构的空间表示更加高效，每一个连接都是对模型有贡献的。下一步可以通过优化算法来减少时间开销。

为了证明进一步验证该方法的效果，在 1Billionword 数据集上取出 100 w 句进行实验。实验的结果如表 7 所示。

表 7 1Billionword 数据集上对比实验

Table 7 Experiment on 1Billionword

初始化	连接比例/%		困惑度
	隐藏层	输出层	
全连接	100	100	134.48
先生长后裁剪	73.30	82.09	137.89
生长同时裁剪	62.23	71.62	135.92

实验的隐藏层设置为 128，随机初始化 20%的连接，作为连接生长的初始种子结构。我们每隔 10 个 step 新建一次连接，每次选梯度绝对值前 200 的连接。由于训练的数据增加，需要模型具有较强的抽取信息的能力，所以网络的连接数也会相应的增加。

4 相关工作

一般来说，增加网络结构的复杂度可以有效提升模型本身的表达能力，从而提升神经网络在各种任务中的性能。但许多研究人员也提出，实际上过于复杂的网络结构并不能对模型性能产生实质上的提升，反而会出现训练过程缓慢、计算存储等资源消耗大等情况。针对该问题，Thodberg 等最初提出了使用剪枝的方法对网络中部分非重要连接进行删减的策略[8]，其核心思想在于通过网络连接重要性判别、剪枝、重训练 3 个步骤的结合，使得在对网络参数进行训练的同时对其中的冗余连接进行合理剔除，在不损伤性能的前提下减少网络结构大小，之后 S.Han[9]、Nabhan [10]、Kadetotad[11]等人又从剪枝模式角度对该方法进行了改进，在原有的剪枝策略基础上融合了哈夫曼编码以及量化的模型压缩策略，最终在 AlexNet 网络结构上达到了 35 倍的压缩效果。但由于剪枝方法的操作对象为一个经过训练的网络结构，并没有对其中连接存在的合理性进行探索，所以 M. Mézard 等反其道行之，考虑从模型生长的角度使用更小的结构对问题进行建模进行求解[12]，该方法采用一种增量式的方法对结构改进，从小模型开始，通过不断尝试，逐渐向其中增加能使模型性能提升的新的节点以及层的结构，最终获得更加紧凑的网络结构[13]。近些年研究人员发现，模型生长和剪枝方式的结合相较单一使用其中任意一种方法对于网络结构的学习都更有助益。2017 年，Dai.Xiaoliang [7]等人提出将生长和剪枝的方式进行结合，同时利用网络训练过程中的梯度信息对其进行指导，使得其所获得的模型参数量更小，并在图像识别任务中相对基线系统取得了更为优秀的成绩[7]。

本文所做的工作与上述研究的差异点在于有效地将生长剪枝的结构学习模式应用到了自然语言处理任务中来，针对语言建模任务对该方法的有效性以及实现策略进行探索。最终在前馈神经语言模型任务中分析了不同的生长剪枝策略对网络结构压缩效果的影响，在保证模型性能的前提下大幅度缩减模型的规模，有效减少了神经语言模型结构中不必要的计算。

5 结论

本文主要针对如何让神经网络语言模型能够自动生成更有效更紧凑的网络结构进行实验，利用训练过程中的梯度指导神经网络中的连接生长，并与传统的剪枝方法进行对比。

实验发现在训练过程中，利用梯度指导一个稀疏的种子神经网络结构生长，可以得到更紧凑、有效的模型结构。在保留相同数量的连接数时，模型性能比全连接网络结构更加稳定。针对在训练过程中，网络产生的冗余连接问题，分别采用先生长后裁剪和生长同时裁剪的裁剪方法，结果证明两种方法都可以获得更精简的模型结构。以上实验证明，一个更好的网络结构是可以通过网络自身学习得到的，下一步课题组将继续基于神经网络语言模型结构学习方法，尝试在连接生长的基础上对神经元节点生长方法进行探究，进一步提升神经语言模型的性能。

参考文献

[1] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.

[2] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. 2017.

[3] Graves A, Mohamed A R, Hinton G. Speech recognition with deep recurrent neural networks[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013:6645-6649.

[4] Haraclick R M. Texture Features for Image Classification[J]. IEEE Trans Smc, 1973, 3(6):610-621.

[5] Dean J, Corrado G S, Monga R, et al. Large scale distributed deep networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1223-1231.

[6] Cun Y L, Denker J S, Solla S A. Optimal brain damage[C]// International Conference on Neural Information Processing Systems. MIT Press, 1989:598-605.

[7] Dai X, Yin H, Jha N K. NeST: A Neural Network Synthesis Tool Based on a Grow-and-Prune Paradigm[J]. 2017.

[8] Thodberg H H. Improving generalization of neural networks through pruning[J]. International Journal of Neural Systems, 1991, 1(04): 317-326.

[9] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint arXiv:1510.00149, 2015.

[10] Nabhan T M, Zomaya A Y. Toward generating neural network structures for function approximation[J]. Neural Networks, 1994, 7(1): 89-99.

[11] Kadetotad D, Arunachalam S, Chakrabarti C, et al. Efficient memory compression in deep neural networks using coarse-grain sparsification for speech applications[C]//Proceedings of the 35th International Conference on Computer-Aided Design. ACM, 2016: 78.

[12] Mézard M, Nadal J P. Learning in feedforward layered networks: The tiling algorithm[J]. Journal of Physics A: Mathematical and General, 1989, 22(12): 2191.

[13] Ash T. Dynamic node creation in backpropagation networks[J]. Connection science, 1989, 1(4): 365-375.

[14] Fahlman S E, Lebiere C. The cascade-correlation learning architecture[C]//Advances in neural information processing systems. 1990: 524-532.

[15] Gloger W, Häusler G. Neural nets with reduced connectivity for the processing of large pictures[J]. Int J Opt Comp, 1993, 2: 425.

[16] Aran O, Yildiz O T, Alpaydin E. An incremental framework based on cross-validation for estimating the architecture of a multilayer perceptron[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2009, 23(02): 159-190.

Learning Neuron Connections for Language Models

JIANG Yufan , LI Bei , LIN Ye, LI Yinqiao, XIAO Tong* ,ZHU Jingbo

(Natural Language Processing Laboratory, Northeastern University, Shenyang, Liaoning, 110819, China)

Abstract : In the field of natural language processing, the structure of the neural network requires manual design, which leads to a large amount of redundancy in the complex neural network structure. For the purpose of reducing the redundant model parameters, people often adopt model compression methods such as pruning, but these methods directly compress the model by taking some indicators that are not related to the training process, which often results in performance loss. This paper explores the automatic learning method of neural connection in neural network. This method can dynamically grow and delete the neuron connection during training, which can better operate the network connection dynamically, thus achieving more compact and efficient model structure. Using this method, automatic growth and elimination are performed on the neural language model, the model scale can be further reduced to 51% while sticking to the original model performance.

Keywords: language model; neuron connection; pruning