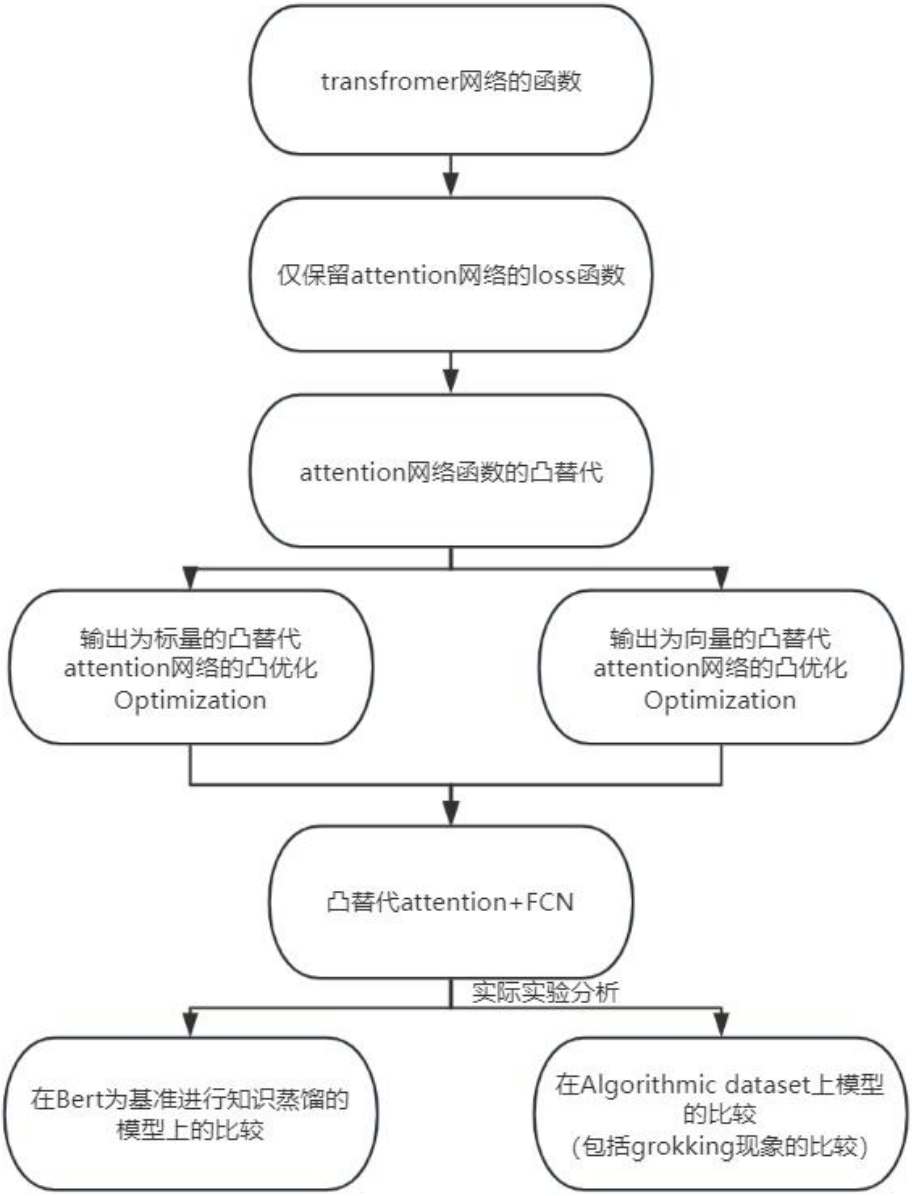
summary

文章：《Convexifying Transformers:
Improving optimization and understanding of transformer networks》
链接：https://arxiv.org/pdf/2211.11052.pdf

自 2017 年 Vaswani et al. (2017) 提出 transfromer 网络以来，其在许多领域如 NLP，CV 等都取得了巨大的成功。transfromer 网络的成功主要归功于 self-attention 机制，但对 transfromer 的本身的有力分析以及其所学到函数的解释的文章还比较少。该文研究了 transfromer 或 attention 的训练问题，并且引入一种新的凸分析方法以提高对 transfromer 网络的理解以及训练优化。

该文研究如何对 attention 函数进行凸替代以及如何对 attention 以及 attention+FCN 进行凸优化的 Optimization，该文指出对 attention 进行凸优化的 Optimization 具有更少的参数量以及更少的 FLOPs，并在某些任务上表现优于标准的 attention，且 grokking 现象更弱。
研究框架如下（下页）：



文章开始是引言部分，介绍 transfromer 的研究现状以及可能存在的一些问题，以及总结该文的成果。然后开始进入文章的正文，首先介绍了 transfromer 网络的结构，然后将其拆开，把 attention 网络单独拿出来，写出 attention 网络的 loss 函数，接着是文章的核心部分，即如何对 attention 网络函数进行凸替代，再接着分情况讨论，对输出为标量和向量两种情况分别讨论，分别对两者的 loss 函数进行凸优化的 optimization，再接着研究凸替代 attention 网络再加上 FCN 网络后的 loss 函数以及如何 optimization，文章最后将标准的非凸 attention 模块、凸替代 attention 网络函数的非凸

optimization 以及凸替代 attention 网络函数的凸 optimization 三个模型在以 bert 为基准的知识蒸馏模型进行性能比较，以及 transfromer 的凸优化 optimization 以及标准 optimization 在 Algorithmic datasets 上的性能比较，并额外比较了 grokking 现象在两者上的严重程度。

在以下公式中，大写字母代表矩阵，小写字母代表向量，小写斜体字幕代表标量。文章中引理以及定理的证明在此 Summary 中将省略。

1. Transfromer 结构

对一个输入样本 $\mathrm{X} \in \mathrm{R}^{\mathrm{h}\times \mathrm{d}}$，代表一个 sequence 包含 h 个 tokens，embdding 维度为 d，attention 首先对该样本乘三个矩阵得到 query，key，value：

$$\mathrm{Q} = \mathrm{X}\mathrm{W}_{\mathrm{q}},\ \mathrm{W}_{\mathrm{q}} \in R^{d\times d}$$

$$\mathrm{K} = \mathrm{X}\mathrm{W}_{\mathrm{k}},\ \mathrm{W}_{\mathrm{k}} \in R^{d\times d}$$

$$\mathrm{V} = \mathrm{X}\mathrm{W}_{\mathrm{v}},\ \mathrm{W}_{\mathrm{v}} \in R^{d\times d}$$

然后获得注意力值，再通过残差层，Normalization，以及一个前馈连接层得到结果：

$$\mathrm{A}_{\mathrm{s,j}} = softmax(\mathrm{Q}\mathrm{K}^{\mathrm{T}})\mathrm{V}$$

$$\mathrm{A}_{\mathrm{o}} = \mathrm{A}_{\mathrm{s}}\mathrm{W}_{\mathrm{o}}, \mathrm{W}_{\mathrm{o}} \in R^{d\times d}$$

$$\mathrm{X}_{\mathrm{A}} = LayerNorm(\mathrm{A}_{\mathrm{o}}) + \mathrm{X}'$$

$$\mathrm{X}_{\mathrm{B}} = \sigma(\mathrm{X}_{\mathrm{A}}\mathrm{X}_1)\mathrm{X}_2$$

其中$\sigma()$是 FCN 的激活函数，作者假设 transfromer 的成功主要归因于 attention，残差层，Normalization，以及前馈连接层可以忽略，在下面的研究中，只关注 attention，即输出为A_{o}。

2. attention 网络的损失函数

由上述推导，对 attention 网络，输出为$\hat{\mathrm{Y}} \in R^{n\times c}$，输入为 $\mathrm{X} \in R^{n\times \mathrm{d}}$，有：

$$\hat{\mathrm{Y}} = softmax\left(\mathrm{X}\mathrm{W}_{\mathrm{q}}\mathrm{W}_{\mathrm{k}}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}\right)\mathrm{X}\mathrm{W}_{\mathrm{v}}\mathrm{W}_{\mathrm{o}}$$

考虑 N 个样本，正则化损失函数为：

$$\min_{W_q,W_k,W_v,W_o} \sum_{i=1}^{N} L\left(softmax\left(X_i W_q W_k^T X_i^T\right) X_i W_v W_o,\ Y_i\right) + \frac{\beta}{2} \sum_{\# \in \{q,k,v,o\}} \|W_\#\|_F^2$$

其中 L 代表任意的凸损失函数，如平方损失或交叉熵等，β 是正则化系数，F 代表 Frobenius 范数。对以上损失函数的 Optimization 需要各种非凸优化启发法，接下来将对该损失函数进行凸替代。

由于 softmax 将其输入矩阵的行转换为概率分布，因此它可以被放宽为具有单位单纯形约束的线性运算：

$$\forall U \in R^{n\times n}, \exists W \in \Delta$$
$$s.t.\ \ softmax(U)X = WX$$

其中$\Delta \coloneqq \{W \in R^{n\times n} | w_i \geq 0,\ 1^T w_i = 1\}$，$w_i$为 W 的列。在此条件下，损失函数被重构为：

$$\min_{W_1\in\Delta,\ W_2\in R^{d\times d}, W_3\in R^{d\times c}} \sum_{i=1}^{N} L\left(W_1 X_i W_2 W_3,\ Y_i\right) + \frac{\beta}{2}\left(\|W_2\|_F^2 + \|W_3\|_F^2\right)$$

再引入多头机制，损失函数为：

$$\min_{W_{1j}\in\Delta,\ W_{2j}\in R^{d\times d}, W_{3j}\in R^{d\times c}} \sum_{i=1}^{N} L\left(\sum_{j=1}^{h} W_{1j} X_i W_{2j} W_{3j},\ Y_i\right) + \frac{\beta}{2}\left(\sum_{j=1}^{h} \|W_{2j}\|_F^2 + \|W_{3j}\|_F^2\right)$$

3. 对不同情况下损失函数的凸替代

接下来分别从输出为标量以及输出为向量的形式对以上函数进行凸优化。

首先是标量形式，此时$y_i \in R$, 损失函数为：

$$\min_{w_{1j}\in\Delta,\ w_{2j}\in R^d, w_{3j}\in R} \sum_{i=1}^{N} L\left(\sum_{j=1}^{h} w_{1j}^T X_i w_{2j} w_{3j},\ y_i\right) + \frac{\beta}{2} \sum_{j=1}^{h}\left(\|w_{2j}\|_2^2 + w_{3j}^2\right)$$

上述问题可以等价为以下 L1 正则化问题：

$$\min_{w_{1j}\in\Delta,\ \|w_{2j}\|_2\leq 1, w_{3j}\in R} \sum_{i=1}^{N} L\left(\sum_{j=1}^{h} w_{1j}^T X_i w_{2j} w_{3j},\ y_i\right) + \beta\|w_{3j}\|_1$$

基于以上等价，输出为标量的非凸优化问题可以等价于以下的凸优化问题：

$$\min_{\mathrm{Z}\in R^{n\times \mathrm{d}}}\frac{1}{2}\sum_{i=1}^{N}L\left(\mathrm{trace}(\mathrm{Z}^{\mathrm{T}}\mathrm{X}_{\mathrm{i}}),\ y_i\right)+\beta\sum_{k=1}^{n}\|\mathrm{z}_{\mathrm{k}}\|_2$$

其中 Z 是注意力矩阵，每一行都是相应 token 的注意力得分，此处转化为一个 group lasso 问题，可以将以组为单位的参数稀疏化。此时该损失函数可以解释为使用尽可能少的 token 来拟合输出y_i。

不同于等价前的$\mathrm{w}_{1\mathrm{j}}\in\Delta$是存在约束的，此处 Z 不需要任何约束，因此可以直接使用标准的 SGD 或者 Adam 算法训练该凸损失函数。且可以从该损失函数的最优解恢复到最开始函数的最优解，做如下变换即可：

$$\mathrm{w}_{1\mathrm{j}}^{*}=\mathrm{e}_{\mathrm{j}}$$

$$\mathrm{w}_{2\mathrm{j}}^{*}=\frac{\mathrm{z}_{\mathrm{j}}}{\sqrt{\|\mathrm{z}_{\mathrm{j}}\|_2}}$$

$$w_{3\mathrm{j}}^{*}=\sqrt{\|\mathrm{z}_{\mathrm{j}}\|_2}$$

其中e_{j}代表第 j 个值为 1 的基向量，$\mathrm{z}_{\mathrm{j}}\in R^{d}$是 Z 的第 j 行，假定稀疏化后 Z 有 h 行不为零向量。由此可以看出该凸优化问题与最开始的非凸优化问题的解存在一一映射的关系，只需要对凸替代后的损失函数进行求最优解就可以得到原非凸优化问题的最优解。

接着与输出为标量类似，当输出为向量，即$\mathrm{y}_{\mathrm{i}}\in R^{c}$时，损失函数为：

$$\min_{\mathrm{w}_{1\mathrm{j}}\in\Delta,\,\mathrm{w}_{2\mathrm{j}}\in R^{d},\mathrm{w}_{3\mathrm{j}}\in R^{c}}\sum_{i=1}^{N}L\left(\sum_{j=1}^{h}\mathrm{w}_{1\mathrm{j}}^{\mathrm{T}}\mathrm{X}_{\mathrm{i}}\mathrm{w}_{2\mathrm{j}}\mathrm{w}_{3\mathrm{j}},\,\mathrm{y}_{\mathrm{i}}\right)+\frac{\beta}{2}\sum_{j=1}^{h}\left(\|\mathrm{w}_{2\mathrm{j}}\|_2^2+\|\mathrm{w}_{3\mathrm{j}}\|_1^2\right)$$

此处$\mathrm{w}_{3\mathrm{j}}$使用 L1 范数是为了能够顺利进行接下来来的凸优化，作者认为这在实际中不会影响网络的性能。

接着同样，上述损失函数可以被凸替代为以下函数：

$$\min_{\mathrm{Z}_{\mathrm{l}}\in R^{n\times \mathrm{d}}}\sum_{i=1}^{N}\sum_{l=1}^{c}L\left(\mathrm{trace}(\mathrm{Z}_{\mathrm{l}}^{\mathrm{T}}\mathrm{X}_{\mathrm{i}}),\ y_{il}\right)+\beta\sum_{l=1}^{c}\sum_{k=1}^{n}\|\mathrm{z}_{\mathrm{lk}}\|_2$$

从以上函数可以看出其在 l 方向是可分离的，共有 c 个输出，每个Z_l矩阵对应与一个输出，同时也说明了 attention 网络的参数量可以由该凸替代损失函数间接由输出数量控制。

最后，将加上了 FCN 层的损失函数进行凸替代，原损失函数为：

$$\min_{w_{1j}\in\Delta,\, w_{2j},w_{3j}\in R^c} \sum_{i=1}^{N} L\left(\sigma\left(\sum_{j=1}^{h} w_{1j}^{T} X_i w_{2j}\right) w_{3j},\, y_i\right) + \frac{\beta}{2} \sum_{j=1}^{h} \left(\|w_{2j}\|_2^2 + \|w_{3j}\|_1^2\right)$$

其中σ代表 ReLu 函数，经过凸替代得到：

$$\min_{Z_{jl} \in R^{n\times d}} \sum_{i=1}^{N}\sum_{l=1}^{c} L\left(\sum_{j=1}^{h} I_{ij}\, \mathrm{trace}(Z_{jl}^{T} X_i),\, y_{il}\right) + \beta \sum_{l=1}^{c}\sum_{j=1}^{h}\sum_{k=1}^{n} \|z_{jlk}\|_2$$

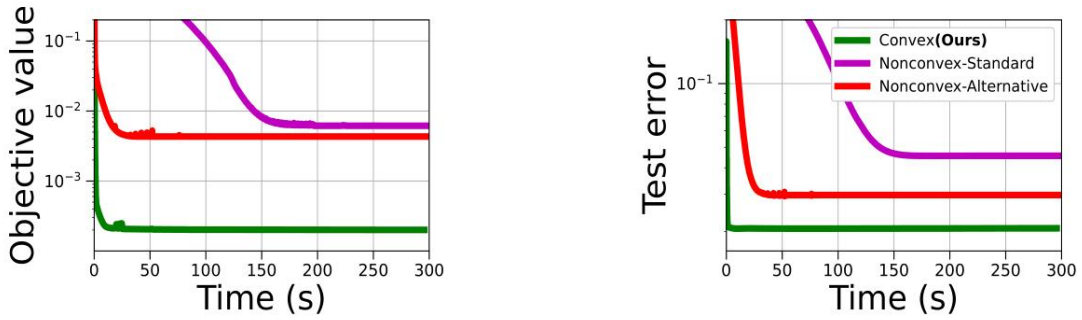
其中$I_{ij} = I\{u_{1j}^{T} X_i u_{2j} \geq 0\}$代表 ReLu 函数的示性函数替代，其中$u_{1j}$，$u_{2j}$是可以随机选择的固定向量，j=1,2,3,...,h。该式子表明引入激活函数进一步增加了网络的参数量（增加 h 倍）。

以上三个模型的参数量以及 FLOPs 数如下表所示：

	Nonconvex				Convex	
	Standard		Alternative (Ours)			
	# of params	FLOPs	# of params	FLOPs	# of params	FLOPs
Scalar output	$h(3d^2+d)$	$\mathcal{O}(n^2dh)$	$h(n+d+1)$	$\mathcal{O}(nd)$	nd	$\mathcal{O}(nd)$
Multi output	$h(3d^2+dc)$	$\mathcal{O}(n^2dh+ndhc)$	$h(n+d+c)$	$\mathcal{O}(nd+c)$	ndc	$\mathcal{O}(ndc)$
Multi output with FCN	$h(3d^2+dc)$	$\mathcal{O}(n^2dh+ndhc)$	$h(n+d+c)$	$\mathcal{O}(nd+c)$	$ndch$	$\mathcal{O}(ndch)$

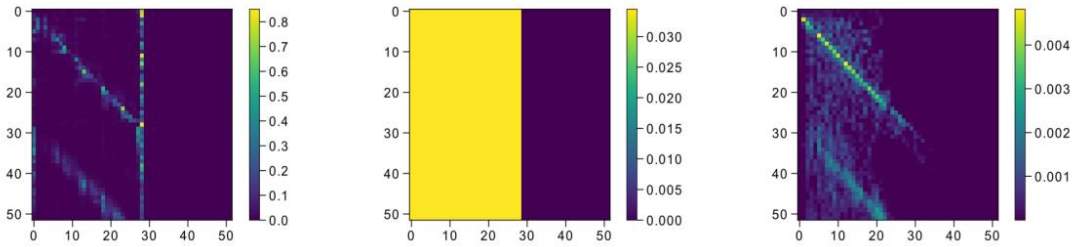
4. 实际性能比较

作者在两个数据集上使用不同优化方法进行比较，首先是以 Bert 为基准的蒸馏模型。以 glue 数据集的 mrpc 子集作为 Bert 的输入，再以预训练 Bert 的输入输出作为数据集分别比较标准 attention，本文中凸替代的 attention 但是非凸优化以及对凸替代的 attention 进行凸优化三个模型的表现，训练环境为 colab，单个 GPU，对三个模型的训练采用相同的正则化系数以及优化器 Adam，并通过对验证数据集执行网格搜索来调整学习率和正则化系数，结果如下图所示。



(a) Objective value (b) Test error

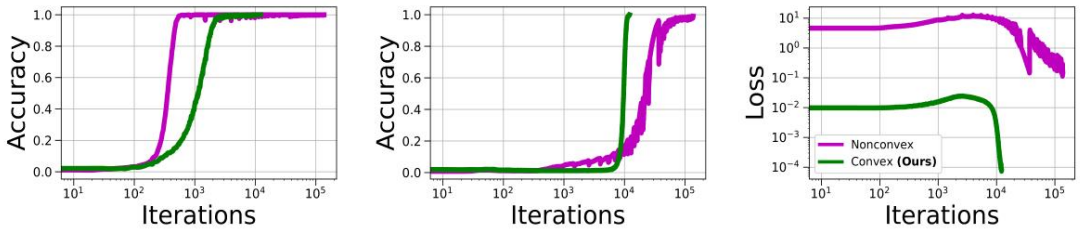
上面两张图分别代表训练损失以及测试损失，可以看出凸替代且凸优化的模型表现最优，标准模型其次，凸替代但不凸优化的模型最差。



(a) Ground truth (b) Nonconvex (c) Convex

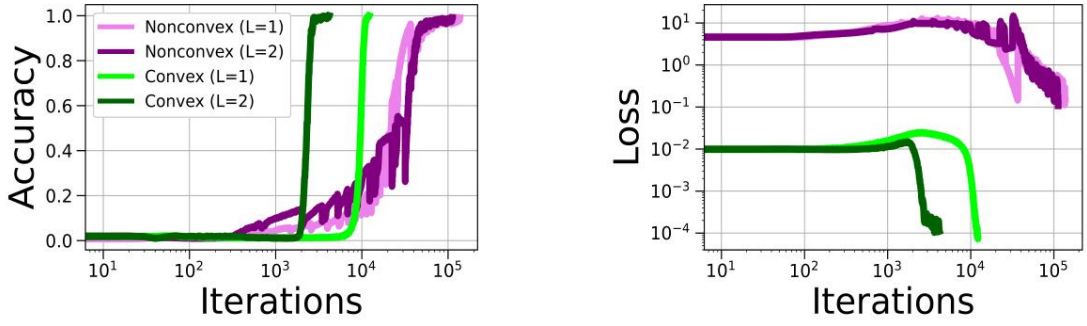
上面是三个模型的注意力图，可以看出非凸优化的模型并没有学习，图像为一个均匀分布的图像，而凸优化模型的注意力图跟 ground truth 的图像比较接近。

接下来作者在 Algorithmic 数据集上对标准模型与凸替代且凸优化的模型进行比较，并特别比较了 grokking 现象（当神经网络在小型算术生成数据集上训练时，即使模型在训练集上已经表现出过拟合的迹象，只要我们持续训练，模型有可能突破一个临界点，从而获得更高的泛化性能。），主要通过 mod97 与 mod15 来评估算法性能，并保证模型都能达到 99%的测试准确率。具体表现如下图所示



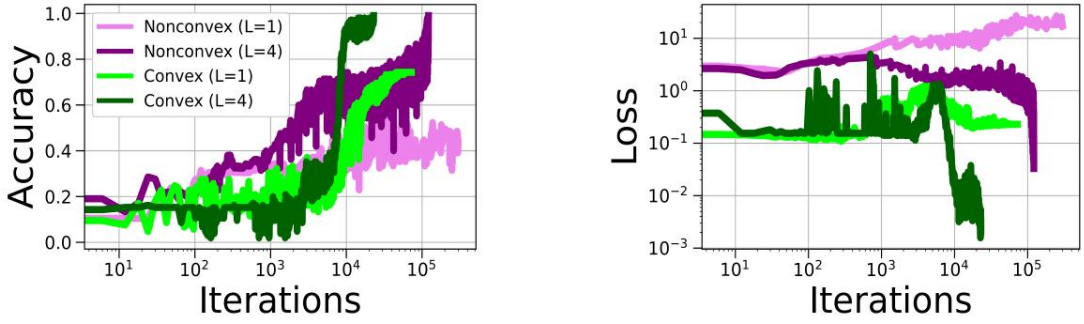
(a) Training accuracy (b) Test accuracy (c) Test loss

在上图中可以看到非凸优化模型在训练集上只需 10^3 数量级的迭代就达到 100%准确率，却在测试集上需要 10^5 数量级的迭代才达到接近 100%准确率，这说明确实出现了 grokking 现象，但使用凸优化模型后在测试集上得到了更快的收敛以及更低的损失函数值。



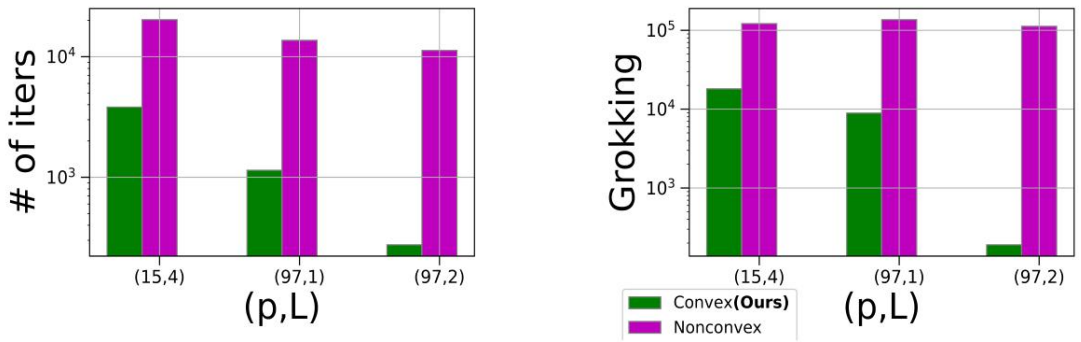
(a) Test accuracy (b) Test loss

上面两张图是在 mod97 任务上分别使用一层与两层 transfromer（attention+FCN）的比较，L 代表层数。从图片中可以看出使用凸优化的 transfromer 模型当使用两层堆叠时可以获得更快的测试收敛，但对非凸优化模型来说则没有明显影响。



(a) Training accuracy (b) Test loss

上面是在 mod15 任务上分别使用 1 层与 4 层 transfromer 的比较。在该任务上一层网络就无法达到 99%的测试准确率，在 4 层网络时可以看出凸优化的模型依然比非凸优化的模型收敛速度明显增快，且损失函数值更小。



(a) Training iterations (b) Grokking iterations

上面两张图示对前 4 张图的汇总，p 代表 modp 任务，L 代表网络层数，由于在 mod15 中 1 层网络达不到 99%测试精度所有没有放入图中，此处将迭代次数作为个肉 grokking 现象的量化，从图中可以看出凸优化的模型显著减轻了 grokking 现象。