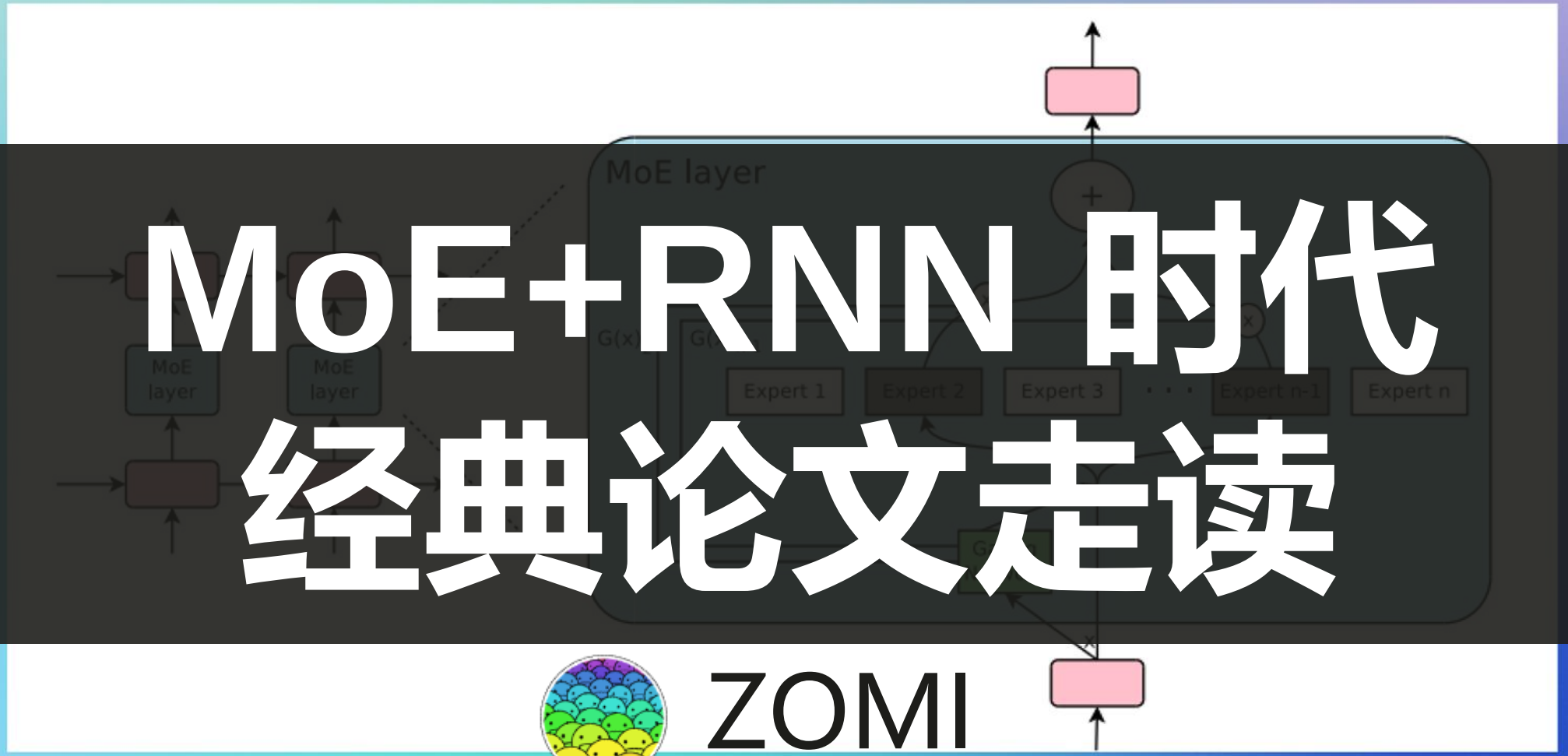
Mixture of Experts (MoE)



MoE+RNN 时代
经典论文走读

ZOMI

Contents

视频目录大纲

- MoE 混合专家
  - MoE 基本介绍 (4)
  - MoE 前世今生 (5)
  - MoE 核心工作
    - 奠基工作：Adaptive Mixtures of Local Experts
    - LSTM MoE：Outrageously Large Neural Networks ...
    - GShard: Scaling Giant Models with Conditional ...
    - Switch Transformers: Scaling to Trillion Parameter Models ...
    - GLaM: Efficient Scaling of Language Models with Mixture ...
    - ST-MoE: Designing Stable and Transferable Sparse Exp
  - MoE 架构原理 (3)
  - 大模型遇上 MoE (4)
  - 手撕 MoE 代码 (5)
  - MoE 的未来 (2)

02

Outrageously Large Neural Networks

基本介绍

- Google在2017年1月发布了《OUTRAGEOUSLY LARGE NEURAL NETWORKS: THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER》，把MoE应用到了LSTM上，训出了最大137B的LSTM模型。这样规模的模型哪怕放在7年后的今天，也是巨无霸的存在，需要解决很多工程问题。
- 相比1991年的工作，这里做到了真正的稀疏激活，从而可以在实际计算量较少的情况下，训练巨大的模型。

背景与挑战

- 虽然当时Transformer还没出来，大规模模型的竞赛也还不像今天这么激烈，但是在多个领域中（文本、图像、音频），已经有不少工作反复证实了一件事：模型容量越大，能训出来的效果越好，上限越高。但是模型越大，需要的训练数据也就越多，二者共同作用下，就造成了训练开销基本是随着模型增大，以平方关系在增长。
- 在这个背景下就出现一些conditional computation，条件计算的工作来解决这个问题。conditional computation就是根据输入，有选择地只激活部分网络模块。那么MoE其实就是一种条件计算的实现。由于不用激活全部参数，训练所需的计算量就大大减小，整体计算成本就不用以平方速度增长。

背景与挑战

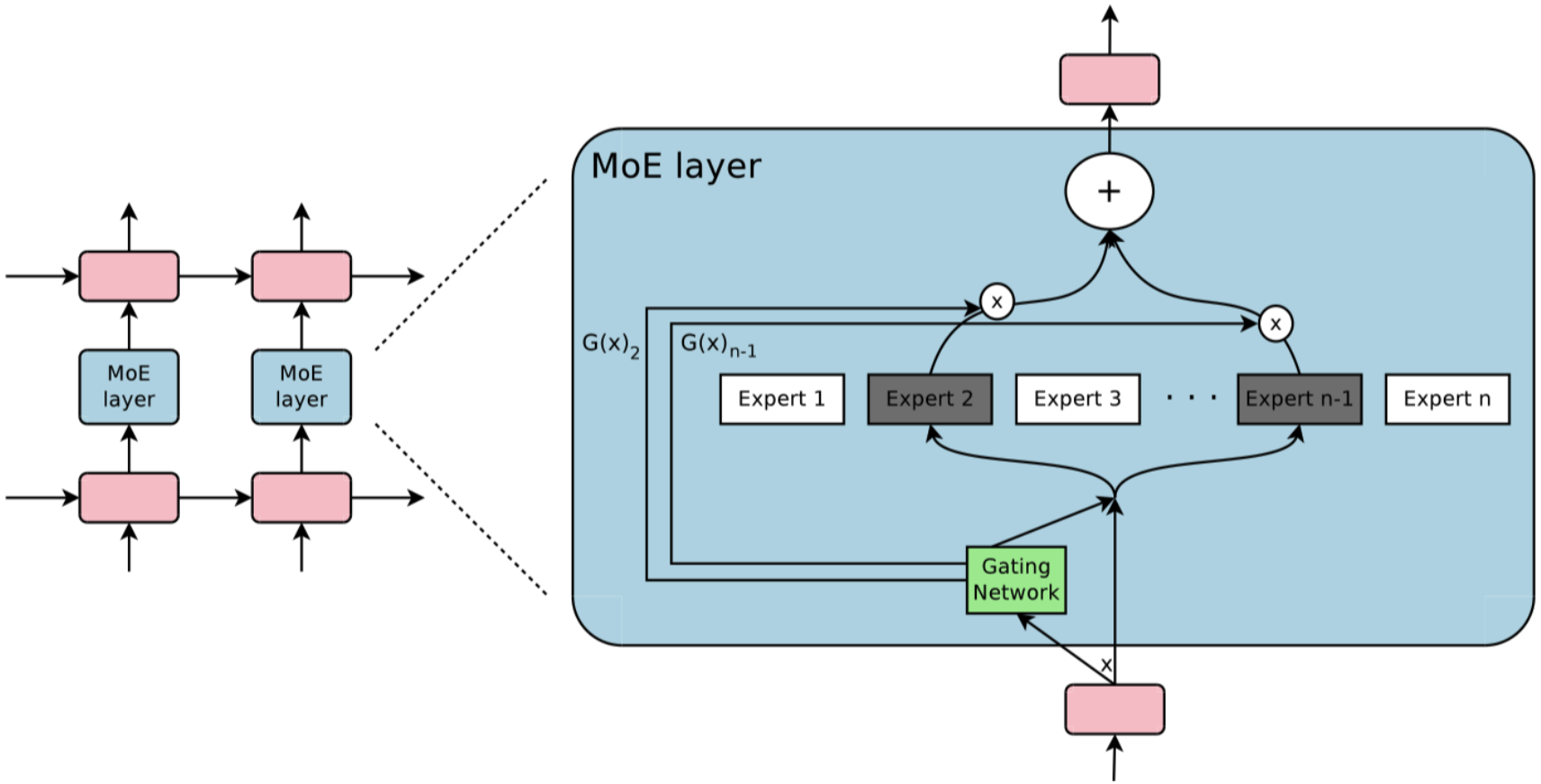
- 训练 Batch Size 较少：
  - 训练的时候，在MoE结构下，每个expert的batch size比整个模型的batch size小了。比如模型的batch size是32，一共有16个expert，那实际上一次迭代平均每个expert只能分到2个训练样本。而batch size对训练效率影响是很大的，大的batch size摊小了参数传输和更新的成本。如果直接增大模型的batch size，又会受显存和通讯效率的限制。
- 训练数据量不足：
  - 要训大模型就需要大量的数据，让模型参数充分学习。在当时的背景下，大规模的NLP数据是比较缺的。当然如今数据集多了很多，特别是预训练数据，这个问题现在来看没有那么突出了。

背景与挑战

- 损失函数的设计：
  - 如何使用合适的损失函数来训练模型，提升效果，并且使得模型的负载比较均衡，这是一个不容易解决的问题。
- 集群通讯问题：
  - 一个GPU集群的计算能力可能比设备间网络带宽的总和高出数千倍，因此设备间的通讯很可能成为训练效率的瓶颈。为了计算效率，就要使得设备内计算量和所需的通讯量的比值，达到相应的比例。
- GPU计算特点：
  - GPU做数学计算很快，但是并不擅长做branching（if/else），因此MoE的工作基本上都是用gating network来控制参数的激活。这个严格来说不算是新的挑战了，应该说是根据计算设备沿用下来的设计。

模型设计

- 论文里使用的是两个LSTM层，中间夹着一个MoE层，最上面和最下面分别还有一个embedding层和一个任务输出层，结构如下图所示。



模型设计

- 每个expert是一个简单的feed-forward neural network。一共有n个expert，gating network输出是一个稀疏的n维向量。
- 如果expert的数量特别多，可以用two-level hierarchical MoE，即使用两层gating network，第一层的gating network先选择一个包含一批expert的分支，每个分支又有一个单独的gating network来选择具体的expert。类似word2vec训练所用的hierarchical softmax。这样做可以节省一些计算。

gating network

- 如果对输入进行线性变换，再简单加上一个softmax，那得到的是一个非稀疏的gating function。
- 在这个基础上，使用一个topk函数，只保留最大的k个值，其他都设为 - ∞（softmax之后变成0），这样就能只选择部分expert，得到了稀疏性。
- 论文提到，虽然理论上这个形式的sparsity（topk）会造成gating function的不连续，不过在实操中暂时没有遇到相关问题。
- 在这个基础上，在输入再加上一个Gaussian noise，这个noise的大小由另外一个可学习的参数来控制。

负载均衡

- 在MoE模型训练的实验中观察到，如果不对gating network进行干预，任由模型自由学习，那么最终模型会倾向于收敛到“总是选那几个固定的expert”的状态，而其他expert几乎不会被使用。这就是负载不均衡的状态，如果这些专家分布在不同的计算设备上，结果就是有些设备输入排队特别长，而有些设备基本处于闲置状态，这明显不是我们想要的。
- 这种负载不均衡的状态有自我加强的属性，因为一旦开始出现部分专家被较多选中激活，这些专家就会得到更充分的训练，从而获得更好的效果，进而又提升被选中激活的概率。
- 针对这种情况，之前有一些工作使用hard constraint来缓解，比如当某个expert激活次数达到上限，就把它从候选集合中移除。hard constraint明显会对模型效果有影响。而这篇论文使用的是一种soft constraint。

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

GitHub https://github.com/chenzomi12/AIInfra

引用与参考

- https://mp.weixin.qq.com/s/6kzCMsJuavkZPG0YCKgeig
- https://www.zhihu.com/tardis/zm/art/677638939?source_id=1003
- https://huggingface.co/blog/zh/moe
- https://mp.weixin.qq.com/s/mOrAYo3qEACjSwcRPG7fWw
- https://mp.weixin.qq.com/s/x39hqf8xn1cUlnxEIM0_ww
- https://mp.weixin.qq.com/s/ZXjwnO103e-wXJGmmKi-Pw
- https://mp.weixin.qq.com/s/8Y281VYaLu5jHoAvQVvVJg
- https://blog.csdn.net/weixin_43013480/article/details/139301000
- https://developer.nvidia.com/zh-cn/blog/applying-mixture-of-experts-in-llm-architectures/
- https://www.zair.top/post/mixture-of-experts/
- https://my.oschina.net/IDP/blog/16513157

- PPT 开源：https://github.com/chenzomi12/AIInfra
- 夸克链接：https://pan.quark.cn/s/74fb24be8eff