Progress Report

BP

date

Introduction

We aim to investigate weather introducing narrow task specific pre-traning for the experts before more general MoE traning will increase the weak domain specialisation found by fan et al for sequence level routing.

If we can find ways to increase human like domain specialisation for MoE experts this would provide transparency into how exactly what recources the MoE draws from in order to adress a given prompt. This type of visibility would be interesting from an alignment perspective.

To do this we create a scaled down version of the experiments by fan et al showing this weak specailisation. We then pretrain the experts and comapre the outcome, with and without pretraining.

Method

The original experiment

The experiment orignally showing weak domain specialisation when trained on sequence level data was based on the base architecture of GPT2-mini, from the nanoGPT repository with a LoRA extentoin. We aim to reproduce a scaled down version of the result in figure2, here we observe slightly different activation patterns based on the subject, pointing towards weak specialisation (fig2). The figure was obtained by:

1. Training a GPT2-mini based transformer on openWebText a general dataset contining lots of information from teh internet.

2. Giving it subject specific tasks from MMLU - a test dataset continaning subject specific questions and answers.

3. Recording routing behaviour when the MoE was faced with these subject specific tasks. We have 4 experts in each of our 3 transformer layers. This is what we see in the figure. (fig2)

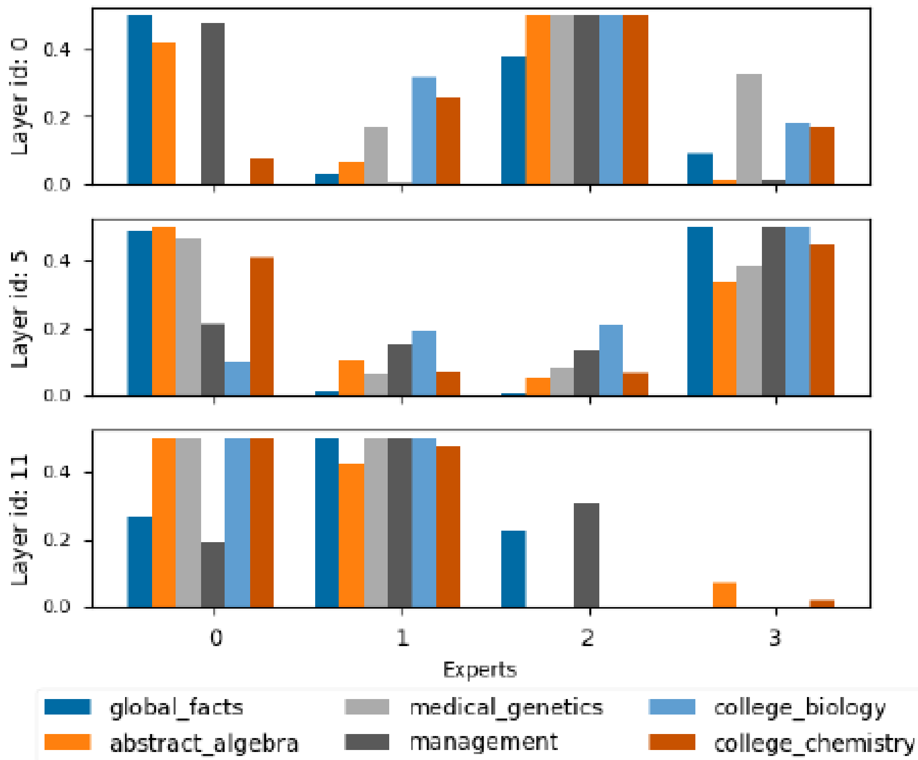

Figure 1:

4. For non specialisation we would see equal use of each expert based on subject. However here we see some different staple heights suggesting the MoE chooses to route differently based on subject. This is what hte authors describes as weak specialisatoin.

Fan et al trained on a on a single A100-SXM4-40GB GPU, with the following modifications to the nanoGPT architecture.

	Original Experiment
Base model	nanoGPT with LoRA extention
dropout	0.2
leaning rate	9.6e-4(min 9.6e-5)
weight decay	0.5
enabled biases	for 6k itterations
token pass thorugh/ itteration	1048576
gradient accumulation	128
batch size	8
sequence lenght	1024
total tokens seen by model	6B
N experts	4
tokens / parameter	20 (chincilla)
optioanl load balancing loss (experts)	weight $\lambda = 0.01$
tokensizer	openai gpt2

For our specific figure a top k=2 level routing was used, for the sequence level routing the softmax function was applied twise as below:

$$p_i(x) = \frac{e^{h_i(x)}}{\sum_j e^{h_j(x)}}, y = \sum_{i=\tau} \frac{e^{p_i(x)}}{\sum_{j\in\tau} e^{p_j(x)}} E_i(x)$$

Here:

- x - is the input token or sequence embedding
- τ is the set of top-k indices (K=2)
- $p_i(x)$ are the logits produced by the gating network as shown
- $E_i(x)$ is the output of the i-th expert
- y is the overall output, which is a weighted sum of the two selected exerpts.

Asymptotics of Routing Behaviour

Fan et al trained on a on a single A100-SXM4-40GB GPU. As we have a limit of 16GB vram this poses a harware constraint for our experiement, however note that we are not nesessarily looking to create a great lanauge model as evaluated on MMLU, but rather to observe routing behaviour when provided a subject specific task. Furhter experiments by fan et al shows that routing behaviour seems to stabilise quite early in the traning process. Especially for sequence level top 2 routing and top 3 routing when load balancing is applied. Fig 2 shows routing decitions during traning on open text web, in each figure there are 4 points one for each expert at any given traning itteration, if no specialisation was happening we would expect all experts to be used equally much. Or maybe some expert to be used a lot and some not at all. The task they are traning on is likely next token prediction, on open web text, for which we can imagine each expert learning some subtask taht is needed some given percent of the time, as seen on the y axis. When an expert learns a task that is used some given percent of the time we see the use on average converge to a line for that percentage, read y axis.

Based on this figure we see convergence quite quickly and hence we suggest that a smaller scale experiment with less traning itterations may still provide good insight into how routing behaviour differs with and without domain specific expert pretraning. The archietcture trained on, GPT2-small, uses 12 transformer blocks with the FFN layer in each block here changed out for a MoE instead with 4 experts. Hence we see 12 figures with 4 dots, one for each expert at any given traning itteration.

Scaling down Fan et al

For our experiment we propose a smaller version of fan et al to attempt reproducing weak topic specialisation for sequence level routing. Traning on a subset of openWebText to be able to run on a 16GB vram card.

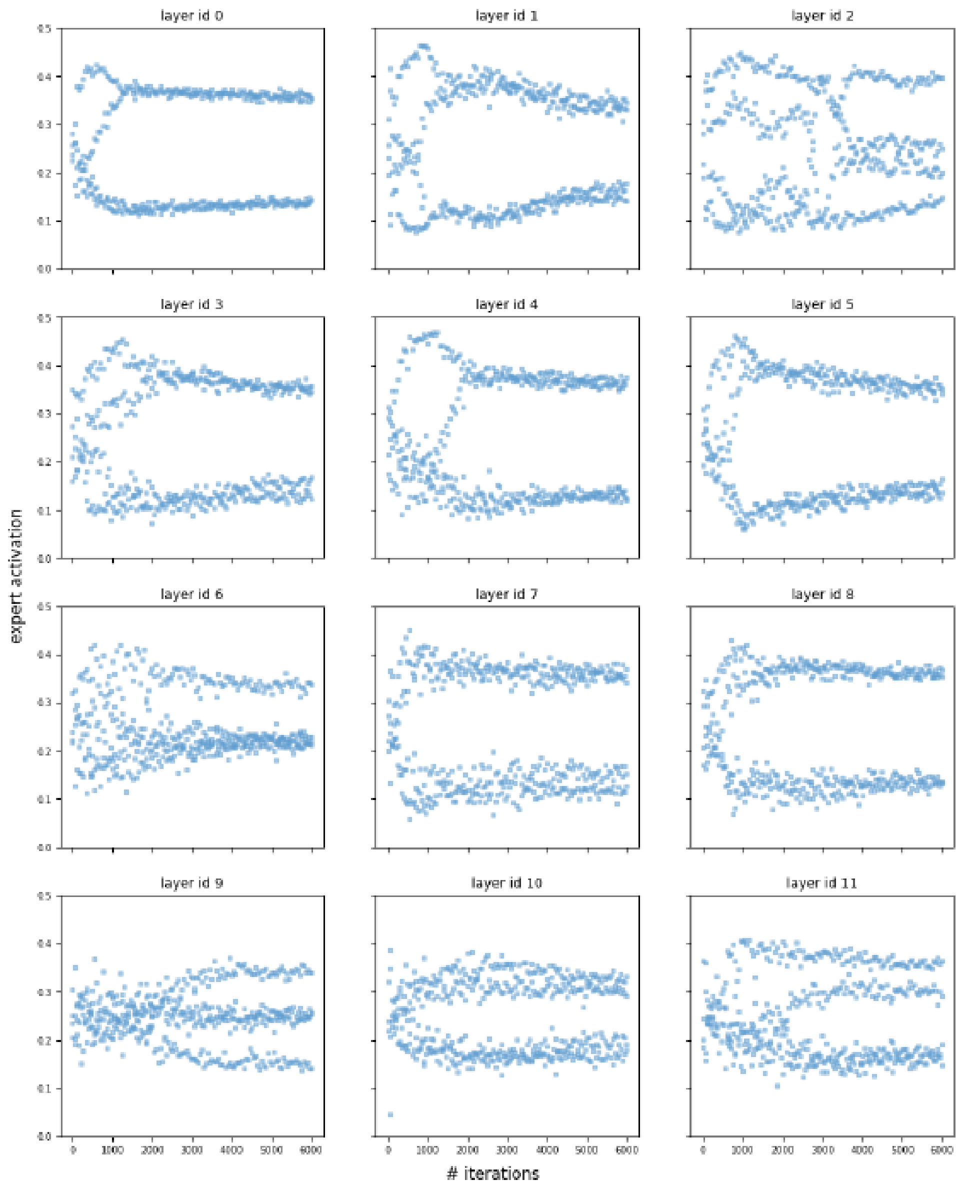

Figure 7: Expert activations from Layer-wise Sequence-level Top-2 routing when load balancing loss is applied.

Figure 2: