Second Meeting

- 26th Sep / Thursday

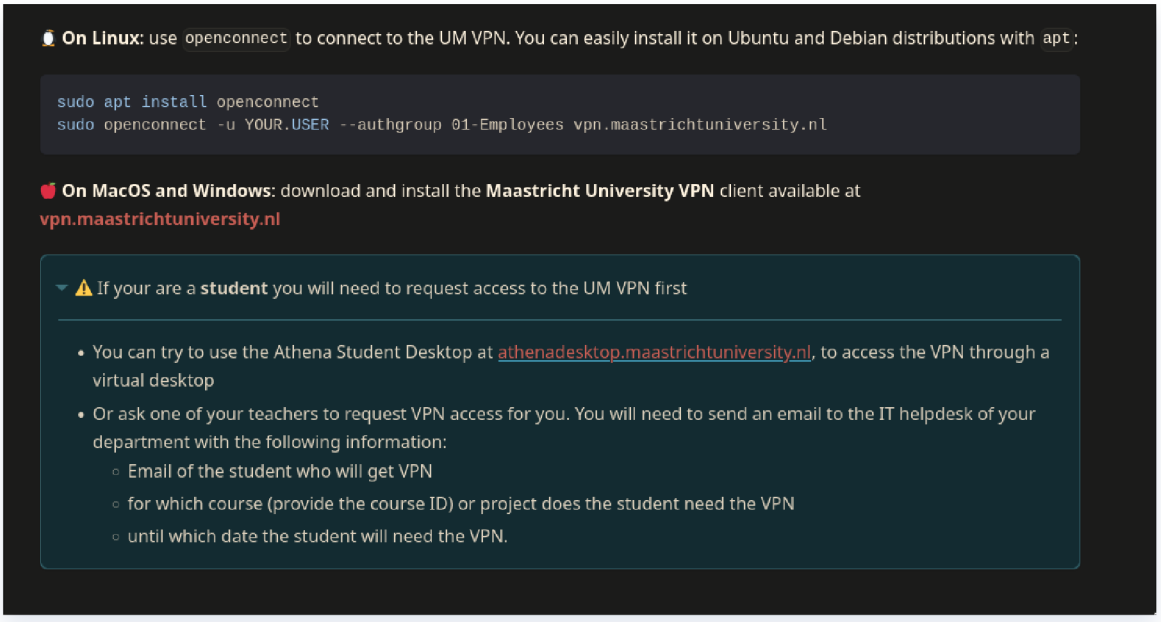
---

so we will go over :

```
* Last Week
* This Week
* Quick Demo from yesterday's checkpoint [very likely that it might be funny]
```

Administrative stuff

- vpn [still 🫠]



> ::

from previous meeting

1. who would use our model ? [grad students / radiologist / non medical]
    1. for multimodality : I said segmentation.. and you asked why ? [and i didnt have an answer]

2. Resarch gaps [besides using different kind of models]
3. Areas of improvement [efficieny or usage]

1. [Radiologist and patients][hope its not generic.. this is still very informal]

Radiologists typically know which bone or anatomical region they will be examining when they suggest the patient get a CT scan or an MRI

[so they dont need image segmentation]

However, the patient might.

[better in clinics which have long latency]

- After the scan, the patient can receive a copy of the scan and our model while they wair to see their doctor.
- During this time, they can use our model to educate themselves on these basic, broad labels. This will help them prepare better, more informed questions for their doctor.

Potential questions the patient might ask:

- What am I looking at? [would allow us to do : semantic segmentation]
- How does a healthy [xyz] look? [allows us to do: text to image but since they can look a bit cartoonish an even better option would be for the clinic to provide a healthy example of the [xyz] label for comparison, instead of them relying on googling image]
- What condition might I have? [allows us to do VQA but we will also have to release a guarded model so that it doesnt say horrible stuff while they wait for the doctor .. or more intensive allignment after pretraining]

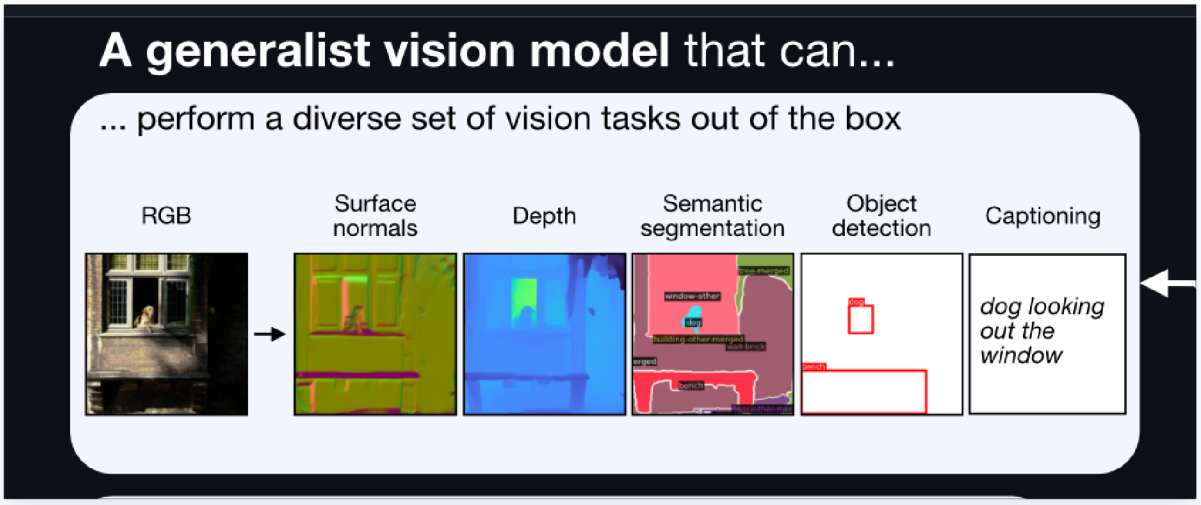
After consultation, we can have a human readable report generation:

- The scan image alongside its segmentation to ensure clarity in discussions.
- A summary of the questions the patient asked, providing them with a clear record of the conversation.
- and more... [this can probably be different for diffferent clinics]

But since 4m can do many modality generation.

- we only need to give them 1 model → so they wont need as much resource.

- and in future if they need support for other modality → they can finetune themselves solving their privacy issues
- ... more maybe



For improvement :

- masking [there is a lot of masking]
  - unified masking but not fused.
  - even we can improve by percentage this could save a lot. [since the generic models are pretrained on Millions of images]
  - 4m did on CC12M

Last Week

- Released a VQA Dataset
- we had a lot of captions

```
// U id is just patient id . each patient might have multiple images .
// but we are not doing that ...🥶

// captions per image
    {
        "Type": "CT",
        "U_id": "MPX1009",
```

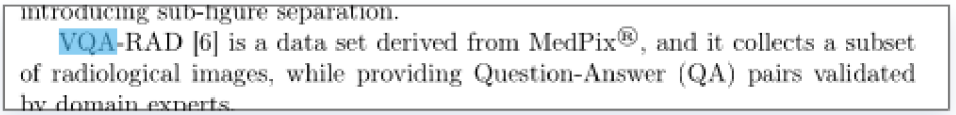
```
        "image": "MPX1009_synpic46283",
        "Description": {
            "ACR Codes": "8.-1",
            "Age": "73",
            "Caption": "The prostate is enlarged with several calcifications  noted
within.  No dominant prostate mass is evident.",
            "Figure Part": null,       <--------- This is almost always null
            "Modality": "CT - noncontrast",
            "Plane": "Coronal",
            "Sex": "male"
        },
        "Location": "Genitourinary",
        "Location Category": "Reproductive and Urinary System"
    },

// ----------------------------

// case captions [about the case the individual has][history][exams done][findings]
[literature]
    {
        "Case": {
            "Title": "Bladder Diverticulum",
            "History": "73-year-old male with hematuria and numerous white blood cells
found on UA",
            "Exam": "N/A",
            "Findings": "Bladder with thickened wall and diverticulum on the right.
Diverticulum is mostly likely secondary to chronic outflow obstruction.\n\nProstate
enlargement.",
            "Differential Diagnosis": "Bladder Diverticulum",
            "Case Diagnosis": "Bladder Diverticulum",
            "Diagnosis By": "N/A"
        },
        "Topic": {
            "Title": "Bladder Diverticulum",
            "Disease Discussion": "Bladder diverticula most often occur as a result of
outlet obstruction.  Occasionally, a congenital weakness in the bladder wall adjacent
to the ureteral orifice results in a diverticulum.  This is termed a \"Hutch\"
diverticulum.\nIn children, outlet obstruction causing a diverticulum is rare and can
be seen with urethral valves.  In men, diverticula are associated with outlet
obstruction from urethral stricture, prostatic hypertrophy, prostatic carcinoma etc.
acquired diverticula are rare in women.\nDiverticula usually occur on the lateral
bladder walls, rarely the dome.  They are often multiple.  Large diverticula often
displace the bladder and or ureters.  \ndiverticula can have wide or narrow necks.  The
wide necked variety empty urine readily.  The narrow neck type are slow to empty and
therefore are more likely to have urinary stasis.\nInfection, tumor and stone formation
can occur as a result of urine stasis within a diverticulum.  Tumor formation in a
diverticulum is more likely to spread beyond the bladder because the diverticulum wall
consists only of urothelium without muscle.\nBladder diverticula can be evaluated with
excretory urography, ultrasound, CT and cystoscopy.\n\nRef:\nDunnick, R., McCallum, R.,
Sandler, C., Textbook of Uroradiology.",
```

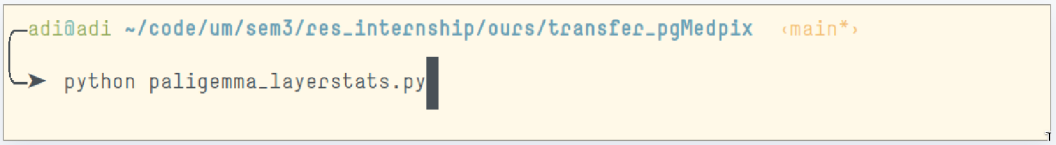
```
            "ACR Code": "8.9",
            "Category": "Diverticulum"
        }
    },
```


- Medpix also demos a vqa model [screenshot from the paper]


introducing sub-figure separation.

VQA-RAD [6] is a data set derived from MedPix®, and it collects a subset of radiological images, while providing Question-Answer (QA) pairs validated by domain experts.

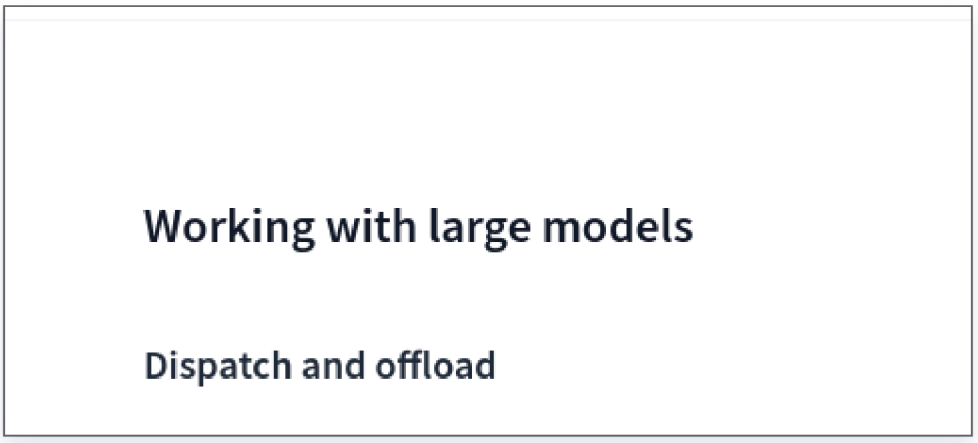
- I was struggling to load in the layers of the model
- Layerstats [base model]:



Total model size: 5.4454 GB

But this didnt even fit colab gpu [T4 Gpu 16 Gigs of HBM]

hugging face accelerate [mostly for inference]



allows certain layers to be :

- cpu offload
- disk offload

so i thought this is great i can train in full resolution in fp32. if i fit 1 layer at a time albeit it would have been slow

But i couldnt

> because we cant because we need 4 times the layer size to train the layer.
>
> - 1 layer
> - 1 gradients
> - 2 optimizer states (adam also holds previous gradients)

- LORA training

LoRA: Low-Rank Adaptation of Large Language Models

Edward Hu* **Yelong Shen*** **Phillip Wallis** **Zeyuan Allen-Zhu**
Yuanzhi Li **Shean Wang** **Lu Wang** **Weizhu Chen**
Microsoft Corporation
{edwardhu, yeshe, phwallis, zeyuana,
yuanzhil, swang, luw, wzchen}@microsoft.com
yuanzhil@andrew.cmu.edu
(Version 2)

We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. Compared to GPT-3 175B fine-tuned with Adam, LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times. LoRA performs on-par or better than fine-tuning in model quality on RoBERTa, DeBERTa, GPT-2, and GPT-3, despite having fewer trainable parameters, a higher training throughput, and, unlike adapters, no additional inference latency.

-- From Abstract

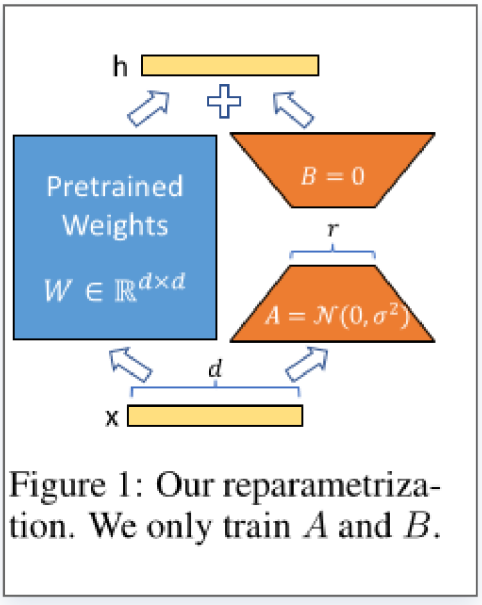


Figure 1: Our reparametrization. We only train A and B.

> so now we only have to store 4 times of dxr and rxd which is smaller than 4 times of dxd
>
> as r <<< d

And here is how it compares to finetuning :

Model & Method	# Trainable Parameters	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg.
RoB_{base} (FT)*	125.0M	**87.6**	94.8	90.2	**63.6**	92.8	**91.9**	78.7	91.2	86.4
RoB_{base} (BitFit)*	0.1M	84.7	93.7	**92.7**	62.0	91.8	84.0	81.5	90.8	85.2
RoB_{base} ($Adpt^D$)*	0.3M	$87.1_{\pm.0}$	$94.2_{\pm.1}$	$88.5_{\pm1.1}$	$60.8_{\pm.4}$	$93.1_{\pm.1}$	$90.2_{\pm.0}$	$71.5_{\pm2.7}$	$89.7_{\pm.3}$	84.4
RoB_{base} ($Adpt^D$)*	0.9M	$87.3_{\pm.1}$	$94.7_{\pm.3}$	$88.4_{\pm.1}$	$62.6_{\pm.9}$	$93.0_{\pm.2}$	$90.6_{\pm.0}$	$75.9_{\pm2.2}$	$90.3_{\pm.1}$	85.4
RoB_{base} (LoRA)	0.3M	$87.5_{\pm.3}$	**$95.1_{\pm.2}$**	$89.7_{\pm.7}$	$63.4_{\pm1.2}$	**$93.3_{\pm.3}$**	$90.8_{\pm.1}$	**$86.6_{\pm.7}$**	**$91.5_{\pm.2}$**	**87.2**
RoB_{large} (FT)*	355.0M	90.2	**96.4**	**90.9**	68.0	94.7	**92.2**	86.6	92.4	88.9
RoB_{large} (LoRA)	0.8M	**$90.6_{\pm.2}$**	$96.2_{\pm.5}$	**$90.9_{\pm1.2}$**	**$68.2_{\pm1.9}$**	**$94.9_{\pm.3}$**	$91.6_{\pm.1}$	**$87.4_{\pm2.5}$**	**$92.6_{\pm.2}$**	**89.0**
RoB_{large} ($Adpt^P$)†	3.0M	$90.2_{\pm.3}$	$96.1_{\pm.3}$	$90.2_{\pm.7}$	**$68.3_{\pm1.0}$**	**$94.8_{\pm.2}$**	**$91.9_{\pm.1}$**	$83.8_{\pm2.9}$	$92.1_{\pm.7}$	88.4
RoB_{large} ($Adpt^P$)†	0.8M	**$90.5_{\pm.3}$**	**$96.6_{\pm.2}$**	$89.7_{\pm1.2}$	$67.8_{\pm2.5}$	**$94.8_{\pm.3}$**	$91.7_{\pm.2}$	$80.1_{\pm2.9}$	$91.9_{\pm.4}$	87.9
RoB_{large} ($Adpt^H$)†	6.0M	$89.9_{\pm.5}$	$96.2_{\pm.3}$	$88.7_{\pm2.9}$	$66.5_{\pm4.4}$	$94.7_{\pm.2}$	$92.1_{\pm.1}$	$83.4_{\pm1.1}$	$91.0_{\pm1.7}$	87.8
RoB_{large} ($Adpt^H$)†	0.8M	$90.3_{\pm.3}$	$96.3_{\pm.5}$	$87.7_{\pm1.7}$	$66.3_{\pm2.0}$	$94.7_{\pm.2}$	$91.5_{\pm.1}$	$72.9_{\pm2.9}$	$91.5_{\pm.5}$	86.4
RoB_{large} (LoRA)†	0.8M	**$90.6_{\pm.2}$**	$96.2_{\pm.5}$	**$90.2_{\pm1.0}$**	$68.2_{\pm1.9}$	**$94.8_{\pm.3}$**	$91.6_{\pm.2}$	**$85.2_{\pm1.1}$**	**$92.3_{\pm.5}$**	**88.6**
DeB_{XXL} (FT)*	1500.0M	91.8	**97.2**	92.0	72.0	**96.0**	92.7	93.9	92.9	91.1
DeB_{XXL} (LoRA)	4.7M	**$91.9_{\pm.2}$**	$96.9_{\pm.2}$	**$92.6_{\pm.6}$**	**$72.4_{\pm1.1}$**	**$96.0_{\pm.1}$**	**$92.9_{\pm.1}$**	**$94.9_{\pm.4}$**	**$93.0_{\pm.2}$**	**91.3**

Table 2: $RoBERTa_{base}$, $RoBERTa_{large}$, and $DeBERTa_{XXL}$ with different adaptation methods on the GLUE benchmark. We report the overall (matched and mismatched) accuracy for MNLI, Matthew's correlation for CoLA, Pearson correlation for STS-B, and accuracy for other tasks. Higher is better for all metrics. * indicates numbers published in prior works. † indicates runs configured in a setup similar to Houlsby et al. (2019) for a fair comparison.

```
adi@adi ~/code/um/sem3/res_internship/ours/transfer_pgMedpix  ‹main*›
└➤ python lora_stats.py
```

Total quantized model size: 2.1095 GB

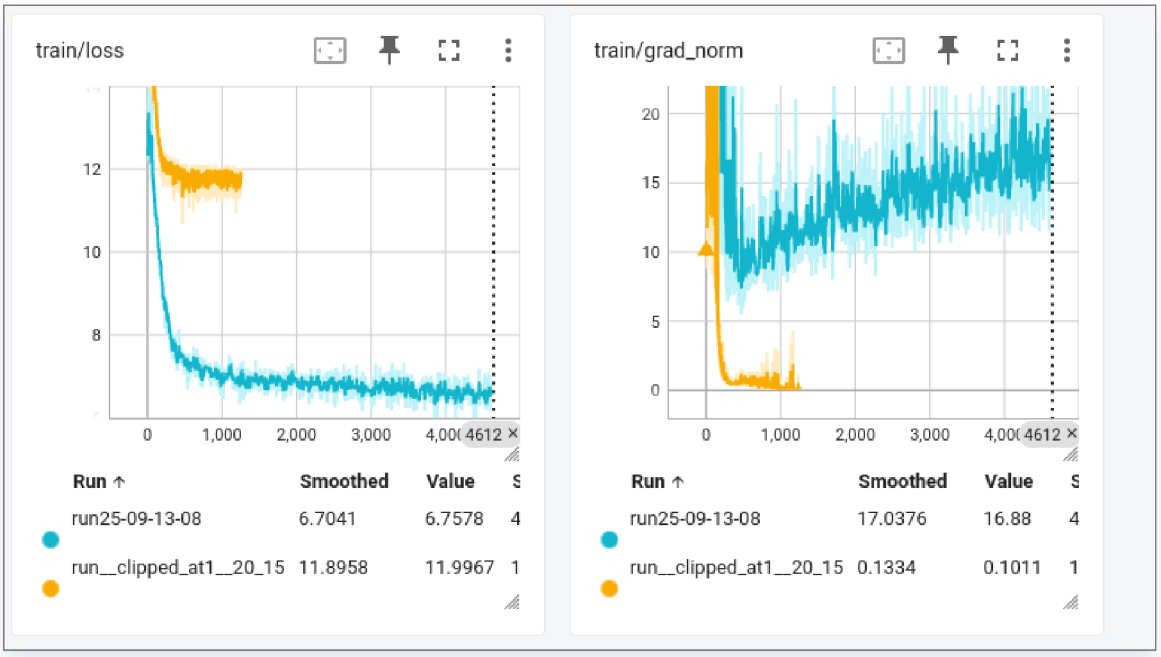
And i was able to fit this even on my gpu

This week

- 1 epoch pretrain [experimentation with unreleased hyperparameters]

> **A Generalization of Full Fine-tuning.** A more general form of fine-tuning allows the training of a subset of the pre-trained parameters. LoRA takes a step further and does not require the accumulated gradient update to weight matrices to have full-rank during adaptation. This means that when applying LoRA to all weight matrices and training all biases[2], we roughly recover the expressiveness of full fine-tuning by setting the LoRA rank r to the rank of the pre-trained weight matrices. In other words, as we increase the number of trainable parameters[3], training LoRA roughly converges to training the original model, while adapter-based methods converges to an MLP and prefix-based methods to a model that cannot take long input sequences.
>
> I missed a detail on gradient accumulation ?

- clipping [with no accumulation] at 1 killed all the gradients ...



Quick Demo

- i wrote a quick inference on how we will load in a checkpoint.
- hope cuda does not run out of memory.