Video Streaming and Tracking 113 Autumn Homework 4 - Vision Transformer Explainability

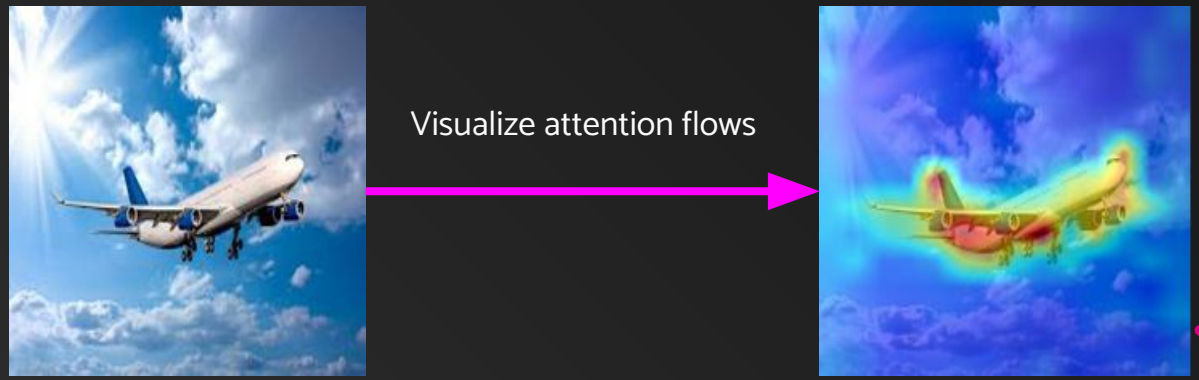
Prof. 蔡文錦
TA 林浩君, 周彥昀, 許承壹

Objective

This assignment focuses on exploring which parts of an image the attention layer focuses on, thereby **visualizing** and explaining the model's behavior.

You are restricted to use VIT classification model deit_tiny_patch16_224 from facebookresearch

1. Learn how to record **attention maps** and **gradients** during inference
2. Learn to implement **Attention Rollout** algorithm, and add some tricks to convert it to **Gradient Rollout**.



Part 1. Attention Rollout(80%)

Attention Rollout helps to aggregate attention maps across multiple layers, providing a cumulative view of which input tokens (e.g., image patches or words) the model focuses on when making predictions.

Algorithm of Attention Rollout

Given a transformer with L layers, for each layer l:

1. Attention map fusion:
   - Since the model we use have **3 heads** in each attention layer, We need to fuse the 3 attention maps into 1 representative map by using **mean/min/max filter**
2. Attention Matrix with skip connection:

the attention map $\tilde{A}^l$ of layer l includes a skip connection, which is represented by adding the identity matrix:

$$\tilde{A}^l = A^l + I$$

where A^l is the fused result of 3 attention heads

3. Cumulative Attention Rollout

We recursively multiply the current attention map with previous result:

$$R_l = \tilde{A}^l \times R_{l-1}$$

where R_{l-1} represents attention rollout of $l-1$ layer, and R_1 is an identity matrix I

Requirements for Part 1:

1. Please **follow templates**
2. You need to implement Attention Rollout algorithm
3. You are only allowed to use pretrained model **deit_tiny_patch16_224**
4. Do not fine-tune the model

Hint:

- Since the model we use has **3 heads** for each attention layers, it is recommended to fuse the 3 attention maps into 1 attention map by **mean/min/max filter** before doing attention rollout calculation.

(Bonus) Part2. Gradient Rollout(20%)

Gradient Rollout is an method that aggregates attention maps with respect to **specific category** across multiple attention layers.

This time you need to also take gradients of specific category into consideration.

Algorithm of Gradient Rollout

Given a transformer with L layers, for each layer l:

1. Attention map fusion:
   - Since the model we use have **3 heads** in each attention layer, We need to fuse the 3 attention maps into 1 representative map by using **mean/min/max filter**
2. Attention Matrix with skip connection:

the attention map $\tilde{A}^l$ of layer l includes a skip connection, which is represented by adding the identity matrix, and this time you need to take gradient of specific category into consideration:

$$\tilde{A}^l = g_l \cdot A^l + I$$

where A^l is the fused result of 3 attention heads, and g_l is the gradients of layer l of specific category

3. Cumulative Attention Rollout

We recursively multiply the current attention map with previous result:

$$R_l = \tilde{A}^l \times R_{l-1}$$

where R_{l-1} represents attention rollout of $l-1$ layer, and R_1 is an identity matrix I

How to Get Gradients?

Typically, during inference, we don't calculate gradients, since we are not training the model.

Hints:

1. You need to perform **back propagation** manually during inference.
2. You need to define a simple **loss function** that can gather information about specific category.

Requirements for Part 2:

1. Please **follow templates**
2. You need to implement Gradient Rollout algorithm
3. You need to perform backpropagation during inference manually.
4. You are only allowed to use pretrained model **deit_tiny_patch16_224**
5. Do not fine-tune the model

Hint:

- Since the model we use has **3 heads** for each attention layers, it is recommended to fuse the 3 attention maps into 1 attention map by **mean/min/max filter** before doing gradient rollout calculation.

Grading Policy

- Attention Rollout(80%): 10 points per image
    - 6 public images, 2 private images(TAs will test for you)
- (Bonus)Gradient Rollout(20%): 5 points per image
    - 2 public images, 2 private images

  –image 1: 87_dogbird.png, category=87 (parrot)

  –image 2: 258_samoyed.png, category=258 (samoyed)
- Report(20%): You just need to answer the following two questions:

  1. Which filters(mean/min/max) have the best performance for part 1.? Please provide a reasonable explanation.
  2. Paste all your results of each image.

Grading Policy(2)

The evaluation is quite simple, you just need to have red region spotted in the right place. If some region focuses on background, it is allowable.

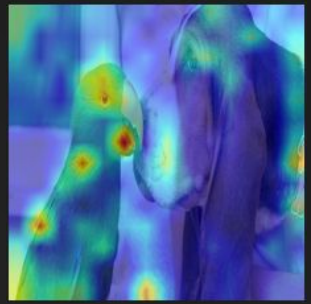
- Attention Rollout Example:

Acceptable:

or

- Gradient Rollout Example(category id=87)

Acceptable:



or

Submission and Rule

```
python attention_rollout.py --image ./images/both.png
```

```
python gradient_rollout.py --image ./images/dogbird.png --category 87
```

Submission:

Please upload your homework in such format:

- HW4_{studentID}.zip (e.g. HW4_312551062.zip)
  - attention_rollout.py
  - gradient_rollout.py
  - report_{studentID}.pdf (e.g. report_312551062.pdf)

Rule:

- Your code should be able to execute with this command format:
  python attention_rollout.py --image <path/to/image>
  python gradient_rollout.py --image <path/to/image> --category <int number>
- No **plagiarism**
- Incorrect filename / file format will get -10% point.
- Delayed submission will get -20% point per day.

Reference

- Register_forward_hook
- 如何取得模型中特定 layer的輸出
- Attention module source code
- attention rollout paper