

Homework 1 part 2 - Transformers

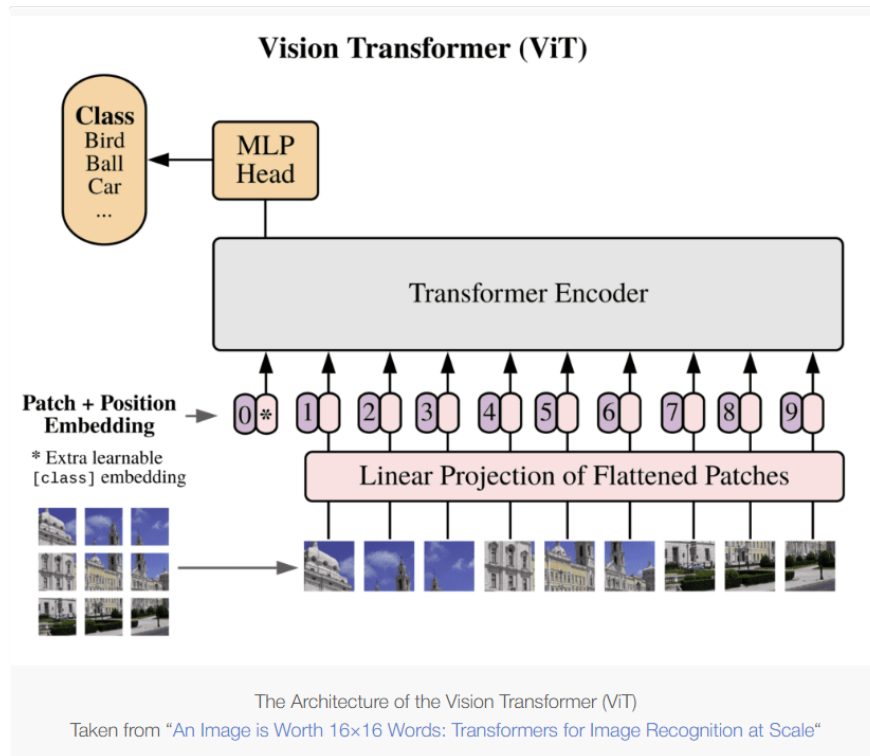


Figure 1: Vision Transformer

Still on the same dataset, we'll check out how to build a Vision transformer network to classify CIFAR10 images (The CIFAR10 dataset consists of 60,000 color images (32x32 pixels) divided into 10 classes of Airplane, Bird, Cat, Dog etc).

What to do

Understand and implement the vision transformers using pytorch (You can get the notebook for this homework from "<https://github.com/vita-epfl/DLAV-2025>" or on Moodle). There are several theory questions designed to evaluate your understanding. The main goal of this homework is to understand the architecture of vision transformers and compare it against the CNNs. Therefore, at the end of the notebook, you're asked to compare both models. For this task, try to discuss

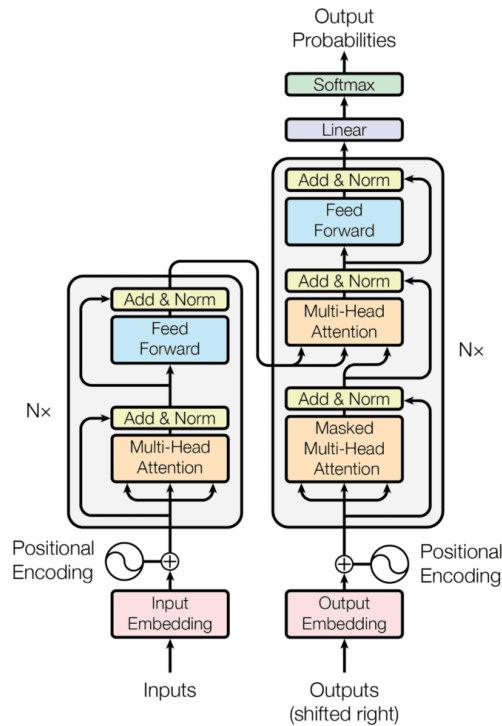


Figure 2: Transformer Architecture

the differences (if any) in terms of performance, training schemes as well as advantages and the disadvantages of each model.

1. Follow the notebook, fill in the empty classes to build a vision transformer.
2. Implement the training process and save your best model.
3. Compare the performance on the validation set with the CNN model you trained last week.

Deliverables

You need to submit the jupyter notebooks containing the code and answers to theory questions and the trained models into the moodle.

Grading

- Coding part accounts for 90% of the notebook assessment and remaining 10% will be for the theoretical questions.
- For the coding section, a student reaching 60% accuracy earns full credit (100 points). For each percentage point below 60%, 2 points are deducted from the total score (e.g., 40% accuracy results in 80 points). However, if a student scores below 30% accuracy, they receive no points.

- You will also get some bonus points for creative implementations/optimizations leading to better accuracy (Should be different from just increasing the number of epochs).

Helpful References

- Pytorch Documentation: <https://pytorch.org/docs/stable/>
- Pytorch Tutorial: [Official link](#), [Collection](#)
- Pytorch Transformer Layers: <https://pytorch.org/docs/stable/nn.html#transformer-layers>