COMP0197-Applied Deep Learning

Packages Used

- torch==2.2
- torchvision==0.17
- torchmetrics
- datasets
- matplotlib

Downloading Model Checkpoints

Please use this link to download the `models` folder containing all our trained model checkpoints:

https://liveuclac-my.sharepoint.com/:f:/g/personal/ucabs55_ucl_ac_uk/EsZ7FI-MzORPgHm_M0AHgaEB_bhD33sP7qiPo-DQtfvdpw?e=7OQIil (https://liveuclac-my.sharepoint.com/:f:/g/personal/ucabs55_ucl_ac_uk/EsZ7FI-MzORPgHm_M0AHgaEB_bhD33sP7qiPo-DQtfvdpw?e=7OQIil)

After finish downloading, please unzip and place the `model` folder within the project root directory

Install your project in editable mode

```
pip install -e .
```

Login to HuggingFace

To login to huggingface, use the following command and enter your hugging face token:

```
huggingface-cli login
```

If you don't have a valid token, use the provided login script:

```
python hg_login.py
```

Comparison Summary

To view the summary of all experiments, please check the `summary.ipynb`

Pre-training MAE

Normal Pre-training

```
python .\src\mae_code\run_mae.py
```

View Results from pre-trained MAE

To visualize the reconstructions from MAE You can use the following command:

```
python .\src\mae_code\plot_trained_mae.py [--model <the path to MAE that you trained, defaults to our trained MAE>]
```

Or through our interactive Jupyter Notebook `view_mae_examples.ipynb`.

Finetuning MAE

Finetuning Pipeline

You may wish to use our already finetuned model, to do so, use:

```
python .\src\finetune\main.py
```

This will test the finetuned model and output example semantic segmentations.

Re-finetune a model

To finetune our pre-trained MAE, use the following command

```
python .\src\finetune\main.py --refinetune [--finetune_percentage <percentage of dataset to finetune on>] [--n_epoch] ...
```

This command will finetune a model, test the model and show example output of semantic segmentations.

Interactive Notebook

You can also use our interactive Jupyter Notebook `view_mae_examples.ipynb`, which visualizes the metrics monitored during training as well

Fully Supervised SegNet

Training

To train our SegNet model, you can use the following command:

```
python src/segnet_bm/train.py [--epochs <number of epochs, default 100>] [--dataset_proportion <proportion 0-1, default 1>]
```

This will train a standard SegNet model and a SegNet with Depthwise Separable Convolution

Testing

To test our SegNet model, use the following command:

```
python .\src\segnet_bm\test.py
```

(OEQ) MAE Skip Connection

We have added skip connections to the decoder and the detection heads to make predictions.

Testing

Use the following command:

```
python .\src\oeq_mae_decoder_conv_layer\main.py
```

Re-finetuning

If you don't want to use our finetuned mode and wish to re-finetune, use the following command:

```
python .\src\oeq_mae_decoder_conv_layer\main.py --refinetune
```

Interactive Notebook

Please see `oeq.ipynb` which visualizes test results, example outputs and the metrics monitored during training

(OEQ) ImageNet Pruning for efficient pretraining

The current code offers both Label Mapping based and Feature Mapping based Dataset Pruning as proposed in https://arxiv.org/abs/2310.08782 (https://arxiv.org/abs/2310.08782). The code is adopted from the paper's original repo https://github.com/OPTML-Group/DP4TL (https://github.com/OPTML-Group/DP4TL)

Running Pruning

Running Label Mapping Based Dataset Pruning

Using the following command:

```
python .\src\dataset_pruning\run_pruning.py [--lmdp] [--reprune] [--retain_class_nums <Number of classes you wish to retain>]
```

Running Feature Mapping Based Dataset Pruning

Using the following command:

```
python .\src\dataset_pruning\run_pruning.py [--fmdp] [--reprune] [--retain_class_nums <Number of classes you wish to retain>]
```

Additional Note

If you have already pruned previously, then you can ignore the `--reprune` tag. The pruned class labels would be saved within the src/dataset_pruning/save folder.

Pre-training with Dataset Pruning

Use the following command:

```
python .\src\mae_code\run_mae.py --train_mode pruned_pretrain
```

Then you can use the existing pipeline mentioned before to finetune this model, test and view results.