

All of the steps were done on google colab pro on L4 GPU, the code can be modified to run on other platforms.

In order to run the code please follow the following steps:

- 1.) First we get conda downloaded and installed from the dissertation.py file

```
! wget https://repo.anaconda.com/miniconda/Miniconda3-py37_4.12.0-Linux-x86_64.sh
! chmod +x Miniconda3-py37_4.12.0-Linux-x86_64.sh
! bash ./Miniconda3-py37_4.12.0-Linux-x86_64.sh -b -f -p /usr/local/
```

- 2.) Then conda environment was created

```
!conda create -n prune_llm python=3.9
```

- 3.) After this we git clone wanda repository

```
!git clone https://github.com/locuslab/wanda.git
```

```
Cloning into 'wanda'...
remote: Enumerating objects: 150, done.
remote: Total 150 (delta 0), reused 0 (delta 0), pack-reused 150 (from 1)
Receiving objects: 100% (150/150), 122.69 KiB | 252.00 KiB/s, done.
Resolving deltas: 100% (70/70), done.
```

Note: Every step after this was done through terminal

- 4.) Now in the bash terminal conda environment was initiated

```
conda init --all
```

- 5.) After initialising the following code was used to bring bash into base mode

```
source ~/.bashrc
```

- 6.) After this conda environment was activated

```
conda activate prune_llm
```

- 7.) Now we changed the directory to wanda

```
cd wanda
```

- 8.) Now the following libraries were installed using pip install in the environment

```
pip install transformers datasets wandb sentencepiece accelerate scikit-learn
```

- 9.) Before pruning certain changes were to be done in wanda prune.py code where under prune_wanda function get_loaders was changed from c4 to wikitext2 for calibration through wikitext2

```
def prune_wanda(args, model, tokenizer, device=torch.device('cuda:0')):
    use_cache = model.config.use_cache
    model.config.use_cache = False

    print("loading calibration data")
    dataloader, _ = get_loaders("wikitext2", nsamples=args.nsamples)
    print("dataset loading complete")
    with torch.no_grad():
        inps, outs, attention_mask, position_ids = dataloader.get_batch(
```



- 10.) After this we proceed to pruning where the models were pruned and saved in sparsity_type unstructured and sparsity_ratio 0.2, 0.4, 0.6, and 0.8, and lastly sparsity_type 2:4 under sparsity_ratio 0.5. This was done using the following command format

```
python main.py --model microsoft/phi-2 --sparsity_type
unstructured --sparsity_ratio 0.2 --prune_method wanda --save
/content/results/ --save_model /content/pruned_model/
```

- 11.) After all models were pruned and saved the code in test.py was used to evaluate zero-shot evaluation/ validation testing on the models. To use this code we need to change the path in the code for the datasets(if being used on a new computer) and model path (for each pruned and baseline model).

```
1 import pandas as pd
2 import torch
3 import json
4 from transformers import AutoModelForCausalLM, AutoTokenizer
5 from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
6
7 # Define a consistent model path
8 model_path = '/content/pruned_model/Phi-2/0.4' # Replace with your actual model path
9
10 # Load the tokenizer and model
11 tokenizer = AutoTokenizer.from_pretrained(model_path)
12 model = AutoModelForCausalLM.from_pretrained(model_path).to('cuda')
13
14 # Ensure padding token is set
15 if tokenizer.pad_token is None:
16 | | tokenizer.add_special_tokens({'pad_token': tokenizer.eos_token})
17 model.config.pad_token_id = tokenizer.pad_token_id
18
19 print(f"Model loaded successfully from {model_path}.")
20
```

NOTE: The validation files for each of the datasets are kept in this format

-  validation_commonsenseqa.parquet
-  validation_cosmosqa.csv
-  validation_ekar.json
-  validation_logiqa.txt
-  validation_reclor.json

Example of result on the next page

Example of result on distilgpt2 :

```
.Cache;
CommonsenseQA Evaluation Metrics:
Validation Accuracy: 49.63%
Validation Precision: 49.76%
Validation Recall: 49.63%
Validation F1-Score: 49.64%
Validation set size: 2985
CosmosQA Evaluation Metrics:
Validation Accuracy: 34.41%
Validation Precision: 34.38%
Validation Recall: 29.04%
Validation F1-Score: 29.04%
Loaded 651 examples from LogiQA dataset.
LogiQA Evaluation Metrics:
Accuracy: 19.66%
Precision: 25.47%
Recall: 19.66%
F1-Score: 19.45%
Dataset size: 118
{'id': '982f17-en', 'question': 'plant:coal', 'choice': 'D', 'text': ['white wine:aged vinegar', 'starch: cabbage:cabbage']}, {'answerKey': 'C', 'explanation': 'of "coal".', 'both "white wine" and "aged vinegar" e of "corn", and the order of words is inconsistent made from "milk".', '"pickled cabbage" is made of s inconsistent with the query.'}, {'relation': [['p te wine', 'aged vinegar', 'R2.4']], [['corn', 'star t', 'R3.7']], [['cabbage', 'pickled cabbage', 'R3.7 e-KAR Evaluation Metrics:
Accuracy: 32.20%
Precision: 32.96%
Recall: 32.20%
F1-Score: 32.30%
Validation set size: 500
{'context': "In a business whose owners and employe e employees can be paid exceptionally low wages. He are much lower than they would be for other busine her. So a family business is a family' s surest roa uestion': 'The reasoning in the argument is flawed ': ["ignores the fact that in a family business, pa ay itself reduce the family's prosperity", "presume ion, that family members are willing to work for lo cause they believe that doing so promotes the famil act that businesses that achieve high levels of cus ofitable even if they pay high wages', 'presumes, w that only businesses with low general operating ex 0, 'id_string': 'val_0'}
ReClor Evaluation Metrics:
Validation Accuracy: 19.00%
Validation Precision: 19.23%
Validation Recall: 19.00%
Validation F1-Score: 19.07%
```