Dataset Cartography

Background/Paper Info

Training efficiency: knowledge distillation makes the model smaller

Dataset cartography: make dataset smaller
https://github.com/allenai/cartography

We have 3 metrics:
Confidence
Variability
Correctness

Confidence is the classifier's confidence
Average the confidence over several epochs. For each datapoint calculate confidence.
Also calculate std dev of confidence for that datapoint.

Potential Directions

1. **Verify that the conclusions of this paper apply to knowledge distillation. Also other compression techniques: pruning, different architecture sizes.**
    a. **Do compressed models**

Train model A on full dataset. This also constructs a datamap.
Construct compressed model B. (compressed using any technique: pruning, octave convolution, simply smaller, quantization)
Train model B only using ambiguous datapoints from datamap.
Questions:
    Training time improvement?
    Accuracy improvement?
    Is there a tradeoff, and is it worth it?
Common approach: Start with big trained model, make it smaller while retaining accuracy.
How can we make this common approach slightly better (better accuracy and quicker training)?
Change dataset: Images/audio/synthetic
Focus on compression
Focus on improving training efficiency. Reducing training FLOPs for achieving same accuracy.

Idea1:

How to pick subset using datamap?

Model A (big, teacher). When training model A, we got a data map.
Model B(small i.e.: compressed using {pruning, quantization,octave convolution, simply smaller}, student): trained by:
1) using a subset of data chosen using datamap,
    a) train using knowledge distillation
    b) normal training.

Baseline: (small model i.e.: compressed),
1) train using full dataset

Assessments:
1) *Assess on ID dataset*
2) *Assess on OOD dataset*

Do data-mapped teachers make better knowledge distillation teachers?
Better in terms of accuracy? Better in terms of efficiency?

Idea2:
Federated Learning w/ Training Dynamics:

- How to construct data-map when federated learning? Each user makes their own datamap
- How to use data-map to improve accuracy? Pick same ambiguous datapoints
- Does using data-map affect convergence?

In federated learning, not all devices and not all datapoints participate in a training round. Hypothesis: We can potentially optimise the impact of a training round for a user by only training using ambiguous datapoints. By optimise impact we mean achieve higher global model accuracy.

2. **Make a data map with a small model, then use it to train a bigger model. This greatly accelerates training.**
    a. **Falls under curriculum learning**
    b. **We can experiment with pre-quantization methods**
    c. **We can experiment with same/similar architecture, just smaller size (e.g. RoBERTa large vs RoBERTa small, ResNet 50 vs ResNet 10)**
    d. **Break down into smaller ideas**

Research Questions:

RQ1) Can we use smaller architectures to construct transferable DMs to reduce training cost on larger architectures? If so, how is performance affected.
RQ2) What strategies can we deploy to optimally select data points from smaller architectures DMs when training larger architectures.
RQ3) Do DM properties (selecting subset of ambiguous points for higher OOD distribution) also apply to smaller models produced with compression techniques (pruning, knowledge distillation, smaller architecture)

Relevant and/or Conflicting Papers

- https://aclanthology.org/2023.findings-acl.674/
    - It seems they only work on datamap transfer from one model to another…and show that it works
        - No mention of compression/smaller architectures
        - Only on NLI (Natural Language Inference) tasks…no CV, no tabular
- https://arxiv.org/abs/2310.06588
    - Use of compression models…pretty much what we wanted to do
    - Might be an area for improving, increasing robustness, ofc extending domains…
    - *Criticism*
        - Still only use NLP
        - Did not look into compression techniques…just smaller architectures (e.g. DeBERTa_small vs DeBERTa_large)
        -

TODO:

Topics to Learn:

- Read up on active learning and curriculum learning.
- Read up on using ambiguous data points for training

Questions for Filip:

- GPU Usage?
    - Are we expected to use Colab and get our receipts billed by the department?
    - Are we allowed departmental GPU usage?
- NLP + CV + tabular domains
    - Good enough for two people?
- Ensemble of different models
    - Would this be too much?
    - Do we have enough already? What would we need to get a high distinction

---

Plan:

Dec 13, 2023

Goals for next meeting:

a) Timo:
   - Install Repo on dev-gpu

b) Evan:
   - Figure out how to run on dev gpu and HPC using PL and WANDB

c) Both of us:
   - First pass of Model A and Model B

Dec 15, 2023

Catch-Up:

a) What was/wasn't accomplished
b) Any hurdles?
c) Any new ideas?

Goals for next meeting:

d) Timo:
   -

e) Evan:
   -

f) Both of us:
   - Install repo on HPC
   - Run on dev-gpu
   - Run on HPC

Dec 26, 2023

Catch-Up/Minutes:

a) KD and LSP Loss
   i) How is LSP loss implemented in the student paper?
   ii) https://www.cl.cam.ac.uk/teaching/2324/L46/examples/project5.pdf
   iii)

b) What more code needs to be done?
   i) Training dynamics subset testing; see comment.
   ii) EfficientNet implementation + testing
   iii) Knowledge Distillation Finish implementation + testing
   iv) Vision Transformer implementation + testing
   v) Find out what the default parameters are for our large models on dummy datasets
      1) Resnets
         (a) 18

(b) 34
(c) 50

2) Efficientnet
(a) 0
(b) 2
(c) 4

3) ViT

c) Experiment Setup Questions:
i) (no experiment needed) **EQ1**: What are the best hyperparameters for our models and datasets?
ii) (no experiment needed) **EQ2**: Can we map training dynamics faithfully to image models and datasets?
iii) **EQ2.5**: What are metrics of our large models? (Tag5)
iv) **EQ3**: Which subset of data works best for knowledge distillation? (only on Cifar100) (Tag1)
v) **EQ4**: From (EQ3), can we further probe the BEST regions and take different percentages? (Tag 2)
vi) **EQ5**: (Possibly wont be needed) Can we further grab data subsets from TWO different regions to improve performance? (Tag3)
vii) **EQ6**: Do the results from EQ4 hold across different datasets? (Tag4)
viii) (no experiment needed) EQ7: Do data points from different regions correlate with human judgement?

Tag5: Simply train the teacher models until convergence

Dataset	Model
Cifar10	Large Resnet
	Large EfficientNet
	Large ViT
Cifar100	Large Resnet
	Large EfficientNet
	Large ViT
MNIST	Large Resnet
	Large EfficientNet
	Large ViT

Datamap selection params

% of train selected	**Combination ratios (variability vs hard vs easy)**	
# Tag 1		
100% train (baseline)		
33% random		
33% hard to learn		
33% ambiguous		
33% easy to learn		
33% low correctness		
33% high correctness		
33% low confidence		
33% high confidence		
33% low variability		
33% high variability		
33% low forgetfulness		
33% high forgetfulness		
# Tag 2		
1% ambiguous		
5% ambiguous		
10% ambiguous		
15% ambiguous		
20% ambiguous		
33% ambiguous		
50% ambiguous		
# Tag3		
17% ambiguous + 10% easytolearn		

15.3% ambiguous + 1.7% easytolearn		
17% ambiguous + 20% easytolearn		
17% ambiguous + 25% easytolearn		
17% ambiguous + 33% easytolearn		
17% ambiguous + 50% easytolearn		

 # Tag4 Hyperparameters to explore

Dataset	Teacher architectures	Student architecture
CIFAR10	**Large Resnet**	**Small Resnet**
		Small EfficientNet
		ViT Small
	ViT Large	**Small Resnet**
		Small EfficientNet
		ViT Small
CIFAR100	**Large Resnet**	**Small Resnet**
		Small EfficientNet
		ViT Small
	ViT Large	**Small Resnet**
		Small EfficientNet
		ViT Small
MNIST	**Large Resnet**	**Small Resnet**
		Small EfficientNet
		ViT Small
	ViT Large	**Small Resnet**
		Small EfficientNet
		ViT Small

TODO:

g) Timo:
- Knowledge Distillation (27th 15:00) done
- Training dynamics subset testing; see comment. (27th 15:00) done
- Subset behaviour (27th 15:00) in progress
- PR to main
- YAML files:
  - EQ3 (28th 16:00)
  - EQ4 (28th 16:00)
  - EQ5 (28th 16:00)

- EQ6 (28th 16:00)

```
(batch_idx=0) x[0] in training_dynamics_callback.py w ith val_split_seed=123
tensor([[[ 0.9843,  0.9686,  0.8745,  ...,  0.2863,  0.7647,  0.9686],
         [ 1.0000,  0.9686,  0.8588,  ...,  0.2392,  0.8196,  1.0000],
         [ 0.9922,  0.9765,  0.9216,  ...,  0.2078,  0.8196,  1.0000],
         ...,
         [-0.5294, -0.3804, -0.2941,  ...,  0.8118,  0.5686,  0.3882],
         [-0.5765, -0.6000, -0.5922,  ...,  0.1373, -0.0196, -0.1294],
         [-0.6941, -0.7490, -0.7412,  ..., -0.2235, -0.3333, -0.3333]],

        [[ 0.9922,  0.9608,  0.8196,  ...,  0.3333,  0.7725,  0.9765],
         [ 1.0000,  0.9686,  0.8118,  ...,  0.3569,  0.8588,  1.0000],
         [ 0.9843,  0.9765,  0.9137,  ...,  0.3333,  0.8667,  1.0000],
         ...,
         [-0.5059, -0.3647, -0.2784,  ...,  0.9137,  0.7412,  0.6000],
         [-0.4824, -0.4824, -0.4824,  ...,  0.3882,  0.2314,  0.0902],
         [-0.4824, -0.4980, -0.4745,  ..., -0.0431, -0.1922, -0.2157]],

        [[ 0.9686,  0.9451,  0.5686,  ...,  0.4353,  0.8431,  0.9765],
         [ 1.0000,  0.9137,  0.5922,  ...,  0.4667,  0.9059,  1.0000],
         [ 0.9765,  0.9686,  0.8667,  ...,  0.4431,  0.8980,  1.0000],
         ...,
         [-0.5529, -0.4196, -0.3412,  ...,  0.9451,  0.7804,  0.6471],
         [-0.5137, -0.5137, -0.5137,  ...,  0.4353,  0.2314,  0.0588],
         [-0.4824, -0.4902, -0.4745,  ..., -0.0980, -0.2784, -0.3098]]])

(batch_idx=1) x[0] in training_dynamics_callback.py w ith val_split_seed=123
tensor([[[ 0.0275, -0.0275, -0.0902,  ...,  0.4431,  0.4510,  0.4431],
         [-0.0039, -0.0275, -0.0902,  ...,  0.3882,  0.4118,  0.4588],
         [-0.0353, -0.0510, -0.1137,  ...,  0.4039,  0.4667,  0.5216],
         ...,
         [-0.2235, -0.3490, -0.5765,  ...,  0.0824,  0.2157,  0.2941],
         [-0.2392, -0.3961, -0.6627,  ...,  0.0980,  0.2078,  0.2784],
         [-0.2471, -0.4431, -0.7255,  ...,  0.0980,  0.2235,  0.2863]],

        [[-0.3647, -0.4039, -0.4510,  ...,  0.1922,  0.2000,  0.1922],
         [-0.4039, -0.4118, -0.4588,  ...,  0.1137,  0.1451,  0.1922],
         [-0.4275, -0.4275, -0.4745,  ...,  0.1137,  0.1765,  0.2392],
         ...,
         [-0.6078, -0.6706, -0.7882,  ..., -0.1922, -0.1137, -0.0745],
         [-0.6078, -0.6941, -0.8353,  ..., -0.1765, -0.1216, -0.0902],
         [-0.6157, -0.7176, -0.8745,  ..., -0.1843, -0.1059, -0.0745]],

        [[-0.7490, -0.7882, -0.8431,  ..., -0.1059, -0.0824, -0.0588],
         [-0.7882, -0.7961, -0.8510,  ..., -0.1686, -0.1373, -0.0902],
         [-0.8196, -0.8118, -0.8667,  ..., -0.1686, -0.1059, -0.0745],
         ...,
         [-0.9216, -0.9608, -0.9608,  ..., -0.4667, -0.4353, -0.4196],
         [-0.9216, -0.9608, -0.9686,  ..., -0.4588, -0.4431, -0.4431],
         [-0.9216, -0.9686, -0.9765,  ..., -0.4588, -0.4275, -0.4275]]])
```

h) Evan:
   - Training dynamics subset testing (26th 23:59)
   - Vision Transformer implementation + testing (27th 15:00)

- Double check default params (27th 15:00)
- PR to main
- Run EQ3 (28th 20:00)

i) Both of us:
   i) YAML config files
      1) EQ3 (28th 16:00)
      2) Run EQ3 (28th 20:00)
      3) EQ4 (28th 16:00)
      4) EQ5 (28th 16:00)
      5) EQ6 (28th 16:00)
   ii) Report file up on WANDB
   iii)

Dec 29, 2023

Catch-Up:

Minutes:

Goals:

j) Timo:
   -

k) Evan:
   -

l) Both of us:

TODO 14.1.24

Background:
Definitions for ambiguity, confidence, correctness etc.

Methodology:

- ☑ ~~Table with param count for each architecture.~~
- ☑ ~~Numbered list explaining the general procedure. The training flow.~~
- ☑ ~~And then do the numbered list for the experimental flow.~~
- ☐ Hyperparameter search

Regarding tables:

- ☑ ~~Replace p_forgetfulness=0.33, selector_forgetfulness to high-forgetfulness etc.~~

- ☑ ~~Make tables 3 column - one for each student architecture~~

- ☑ ~~Change p_random for cifar10 tables to be the correct p_random from Table 4~~

- ☑ ~~Change Tables to be Test f1 instead of val-acc~~
- ☑ ~~Remove lines from combination tables that contain experiments which were not run.~~

☑~~Make 1 big table~~