Imagenet Finetuning

- We do not fine-tune the all tested (image-only) model variants, that have been pretrained, on imagenet
- Is expensive -> In terms of computation, time, and costs
- That is exactly why we use zero-shot validation

Cheap way to compare models, see their improvement over time and check how rich/good the created representations are -> Is the purpose of the model

- Will be even more expensive for the multimodal models
  - We need to fine-tune them on different modalities, each of which has potentially multiple datasets
    - This is especially the case later for the aligned multimodal model

Setup:

- First, we finetune the whole model, so no layer freezing
- For direct comparison with D2V, we follow the same setup:
  - As it is a classification task we add a linear layer on top, mapping from the embed dim of 768 to a dim of 1000, because imagenet contains 1k classes
  - We also add a LayerNorm layer before the linear projection head
  - the head has no actication function, instead, raw logits are returned, which are handled by the cross-entropy loss function in pytorch
  - of the 192 "timesteps" created by the patch embedding, we can only feed one into the linear layer, so we need to reduce the time step dimension to 1
  - in classical ViTs this is done by taking the CLS tokenm, also present in our model
  - surprisingly, the CLS token is not used in the d2v models, but the mean of all tokens except the CLS token
    - one the one hand reasonable, as the cls token has not learned to represent the image, because the pretraining process of d2v is to regress all time steps except the cls token to the image
    - on the other hand d2v2 adds a special cls loss, where the cls token regresses the mean of all time steps, except the cls token, of the teacher model
      - so there is an incentive for the cls token to represent the image, but it is not used in the linear projection head
    - also, this cls loss is scaled by a factor of 0.01, so it contributes 1% to the mse loss, which might sound low
    - however, other timesteps contribute with a factor of 1/192=0.0052 (192 total timesteps), so the cls token contributes twice as much as each other token
  - either way, we do the same as in the d2v models, and take the mean of all tokens except the cls token, which, as shown by the d2v authors, performs very well
    - we do the same to directly compare the models
- for images in the train set we use the same augmentation as in the pretraining phase
- add additional batch-wise mixup augmentation, with same hyperparamters as done in d2v finetuning
- we use AdamW as optimizer with cosine scheduler and warmup, we again take the same hyperparameters as in d2v finetuning
- both because finetuning is expensive in compute and therefore costs, and because we want to compare the models directly, and the hyperparameters have been tuned for d2v finetuning
- one significant difference to d2v is the number of steps that we use during finetuning

- the base model, to which we compare, uses 500k steps (batch size of 256), which is too long for us to run
- at the same time we also expect the model to converge faster, as it has close to 30 million parameters, which is much smaller than the close to 86 million parameters of the d2v base model
- we decide to train the model for 15 epochs,
  - a batch size of 256, with 1.281.167 images in the training set, results in 5.004 steps per epoch, which is 75.060 steps in total
- contrary to pretraining, now validation after each epoch
- Epoch training “now” possible
  - Is also possible in the still unimodal case, but we do not use epochs for consistency with the other experiments and models
  - there is an option to adjust the regularization parameters of the model, like dropout, but we do not do this, as the model is relatively small and imagenet-1k is a large dataset
  - overfitting is less of a concern
- First observation, train loss rather unstable, but decreases
- Try to use grad clipping again
- Take 2.0 again, as has worked good during pre-training, and tuning this value is not feasible, as too expensive
- D2V uses the same grad clipping value (4.0), for their big and huge models (not base), as used during pre-training, so we also adapt this decision, whether it is out of coincidence or something that is important
  - Did not mention why the same value was selected again
  - Maybe because it is the same dataset -> Pretrained on imagenet, now fine-tuning on imagenet
- Did not change anything in the results
- Interestingly: Learning rate is the same as used in d2v imagenet finetuning
- Is 1e-3, which is, even when considering warmup, so starting with lower learning rate, really high for finetuning
  - For comparison: When finetuning d2v audio on labeled audio dataset and d2v text on glue task learning rate is way lower (between 6e-5 - 1e-5)

Type	Hyperparameters	Distilled Unimodal Data2Vec2
Model	LayerNorm Weight Init	constant(1.0)
	LayerNorm Bias Init	constant(0.0)
	LayerNorm ε	1e-6
	Linear Head Weight Init	trunc_normal(std=0.02)
	Linear Head Bias Init	constant(0.0)
	Timestep aggregation	Mean without CLS
Training	Epochs	15
	Batch size	256
	AdamW ε	1e-6
	AdamW β	(0.9, 0.95)
	Peak learning rate	1e-3
	Learning rate schedule	Cosine
	Warmup steps	5k
	Weight decay	0.01
Mixup	mixup alpha	0.7
	cutmix alpha	1.0
	cutmix minmax	×
	prob	0.9
	switch prob	0.5
	mode	batch
	label smooting	0.1
	num_classes	1000

Table 1: Hyperparameters used for the imagenet finetuning of the Distilled Unimodal Data2Vec2. Data Augmentation on raw images is the same as used during pretraining.