What should the targets be for each modality, and what should we take as feature representation for zero-shot?

Why ask that question?

* In is not really apparent how a pretrained D2V model provides semantic information
* Yes, obviously through all time steps, but for training we need to store the targets efficiently, and for zero shot we need just one time step or an average
  + Regarding zero shot: We need to somehow reduce the time steps to get a final embedding/representation
* Each modality seems to be handled differently though
* Audio has no special token, and during D2V pre-training all masked time steps of the teacher are simply regressed
* For text, a <bos>/<cls> token is appended, but its presence is ignored during pre-training and the targets (masked time steps) are regressed the same as audio, that is, all time steps (only during fine-tuning on GLUE task the <bos> token is extracted by the classification head
* For images, we have a <CLS> token, when regressing the masked time steps this is removed, and all masked time steps are regressed, at the same time, the loss between the <CLS> token and the mean of all (including unmasked) time steps of the teacher is regressed, contributing to the loss by a factor of 0.01. This leads to an increase in imagnet accuracy by 0.2%.
* For each mode it is different!!! So what approach (averaging time steps or special token) holds the most information?
  + This is important, as the distillation process should be aligned or at least similar to the way in which D2V was pretrained, as we want to achieve the best possible result and we reuse some parts of the models (the modality encoders) which are trained with the respective strategies described above.
  + Also, we want to maximize the amount of information contained in the targets, while minimizing their size/memory footprint (which is done by the time step reduction)
* Simple test: We have validation datasets for zero-shot -> Use a pretrained d2v and apply this to the val datasets using KNN. As d2v returns time steps we have to somehow reduce them -> Try with: CLS token, Mean over all time steps, Mean over all except CLS token
* See which d2v-modality model performs best with each aggregation strategy on its respective task
  + Image -> Cifar10, Cifar100
  + Audio -> SpeechCommands
  + Text -> Imdb (is binary, but distribution 50/50, so accuracy metric sufficient)
* For each modality, take the aggregation strategy that yielded the best accuracy -> embedding used contains the most (important) information
* For image CLS token performs best, so the addition CLS loss seems the have a unexpected big impact
* For audio, less surprising, the mean performs the best, as there has never been a special token at the beginning used.
* For text, CLS performs best, surprising as the token is not really taken into account during pretraining
* In all cases there are only minor differences between both mean approaches, which makes sense, as we only take one time step less into account when computing the mean.
* Also interesting, if the CLS token performs well alone, there is barely a difference between taking it into account during the mean and not.

Question: Means have been taken over all time steps, also potential padded ones -> Try non-padded

* Surprisingly does not get better, but a little worse
* Keep aggregation also on padded time steps for audio
* Also a little faster, since mean can be vectorized instead of for-looped (ca. 1.5x faster) -> Padding mask is different for each time step!

For normal d2v and just speechcommands alreads 12gb of vram gpu memory used -> Already a bad sign? 4090 with 24 gb was used