0.1 Self-Supervised Learning

0.1.1 Motivation

Supervised models, while powerful, are not inherently scalable. Although their architecture can be extended to create larger models that achieve better performance, these larger models require more data for training. In the context of supervised learning, this data must be labeled, which presents a significant challenge. Labeled data is scarce and expensive to obtain, as it requires human annotation, thereby limiting the scalability of supervised models.

The primary objective of self-supervised learning is to learn representations of data without relying on human-annotated labels. However, self-supervised learning is not unsupervised learning. Unsupervised learning operates without any form of supervision, meaning that no labels are required at all, as seen in clustering methods like K-means. In contrast, self-supervised learning requires, as supervised learning, labeled data, but in contrast to supervised learning labels are generated directly from the data itself.

A prominent example of self-supervised learning is Masked Language Modeling (MLM) in Natural Language Processing (NLP), which is used in the popular NLP model BERT, the latter being one of the first models trained using self-supervised methods to achieve state-of-the-art performance in NLP [1]. In BERT, certain tokens, or words, are masked, i.e., removed, from a sentence, and the model is tasked with predicting the masked tokens. Since the labels are derived from the data itself — the words to predict are part of the original data — no human annotation is needed [1]. This allows for the utilization of large amounts of unlabeled data, as any text data can be used.

What makes self-supervised learning particularly powerful is its applicability to any type of data with a hierarchical structure, such as text, images, audio, or video. In these cases, part of the data can be masked, and the model must predict the masked part based on the context provided by the remaining data. An intuitive example, presented by Yann LeCun and Ishan Misra of Meta, illustrates why this approach is effective. Consider the sentence “The lions chase the wildebeests in the savanna.” If “lions” and “wildebeests” are masked, the input becomes “The [MASK] chases the [MASK] in the savanna.”. To successfully predict the masked words, the model must understand the real-world concepts expressed by the sentence. While “The cat chases the mouse in the savanna” might be a valid prediction in the context of “chase,” the word “savanna” provides additional context, as it is not a typical habitat for cats and mice, but rather for lions and wildebeests. Thus, the model must understand that lions and wildebeests are animals that inhabit savannas, in order to make a correct prediction. Through this process of predicting masked words, the model learns about the concepts of the world we live in [2].

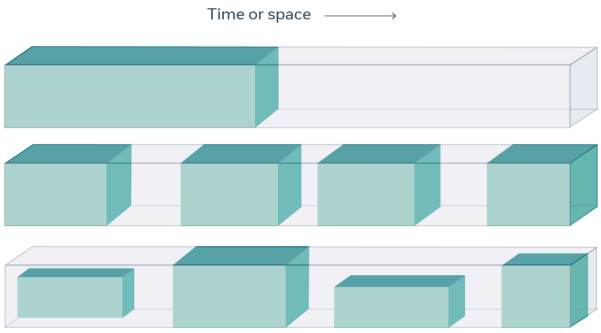

Figure 1: In self-supervised learning parts of the data are masked (grey), and the task of a model is to predict the masked parts using the visible data (green) [2].

0.1.2 Masked Data Modeling

0.1.2.1 Language

As introduced in Section 0.1, the self-supervised training of language models is mostly done through masked language modeling (MLM). While there are also approaches with Self-Distillation, which will be introduced in (TODO: cite Self-Distillation), MLM is by far the most popular approach.

MLM operates by masking tokens in a sentence and predicting the masked tokens based on the context provided by the remaining tokens. For each masked token, the model predicts a probability distribution over the vocabulary, representing all possible tokens. The token with the highest probability is considered the predicted token. Fundamentally, MLM is a classification task where the model classifies each masked token into one of the tokens in the vocabulary.

Given a text model f and a text representation input $\boldsymbol{H}_{w,0}$, where a fraction $p \in (0,1)$ of tokens $\mathcal{M}$ (not to be confused with M, denoting the number of text tokens) are masked, typically between 15% and 50% of tokens as per [1] and [3]. The set of masked tokens is defined as:

$$\mathcal{M} = \{i \mid i \sim \text{Uniform}(\{1, ..., M\})\} \in \mathbb{N}^{\lfloor M*p \rfloor} \tag{1}$$

The process begins by passing the text representation through the model f:

$$\boldsymbol{H}_{w,L} = f\big(\boldsymbol{H}_{w,0}\big) \tag{2}$$

The representations for the masked tokens are then gathered:

$$\boldsymbol{H'}_{w,L} = \big\{\boldsymbol{h}_{w,l,j} \mid j \in \mathcal{M}\big\} \tag{3}$$

These representations are passed through the classification head g, which returns logits for each token in the vocabulary:

$$\widehat{\boldsymbol{H}}_{w,L} = g\big(\boldsymbol{H'}_{w,L}\big) = \big\{g\big(\boldsymbol{h'}_{w,L,j}\big) \mid j \in \mathcal{M}\big\} \tag{4}$$

The model is trained to minimize the cross-entropy loss of the correct token i given the output representation of the corresponding masked token $\hat{\boldsymbol{h}}_{w,L,j}$. The MLM loss for an input text is the sum of the cross-entropy losses for the masked tokens:

$$\mathcal{L}_{\text{MLM}} = \sum_{j \in \mathcal{M}} -\log \frac{\exp\Big(\hat{\boldsymbol{h}}_{(w,L,j)_i}\Big)}{\sum_{k=1}^{V} \exp\Big(\hat{\boldsymbol{h}}_{(w,L,j)_k}\Big)} \tag{5}$$

Here, i denotes the index of the correct token in the vocabulary, and V the size of the vocabulary. j loops over all tokens in the vocabulary. The fraction, of which the logarithm is taken, essentially represents the probability of the correct token i at time step j, given the model's output $\hat{\boldsymbol{h}}_{w,L,j}$ for the masked token at time step j.

0.1.3 Vision

- not as straightforward as in language
- text is discrete, images are continuous
- for text, there are fixed borders -> words or subwords, smallest possible level is single characters.
- and there is a fixed number of possible tokens that can occur at a time step
- image has fixed number of pixels, but the number of possible values for each pixel is infinite
- also, single pixel has no meaning, way too low level, not like a subword or word
- in vision Transformers, the smallest meaningful unit is a patch of pixels, usually 16x16 or 14x14 pixels in size [4]

- even if a patch of pixels is used, the possible content of the patch is still infinite -> we can't do a probability distribution

over all possible patches/pixels, as done with (sub-)words in text

- solution is training a visual tokenizer as a preprocessing step, before the actual self-supervised training
- converts patches of pixels into discrete tokens -> represents the content of the patch
- patches can then be masked, and the model can predict the visual token of the masked patch

-> number of possible visual tokens is finite, and the model can predict a probability distribution over these tokens

- set of all possible visual tokens is called codebook, can be thought of as the same as the vocabulary in language models
- as content of a patch is still infinite, the visual token of the patch is the visual token in the codebook with the highest

cosine similarity to the content of the patch

- most popular approach introduced by BEiT [5] and BEiT v2 [6]

-> train a visual tokenizer with a codebook of 8,192 visual tokens -> each image patch is classified into one of 8,192 visual tokens, or rather semantic concepts

- for details on how a visual tokenizer is trained and how it works, please refer to papers BEiT [5] and BEiT v2 [6]
- training process is then the same as in language models

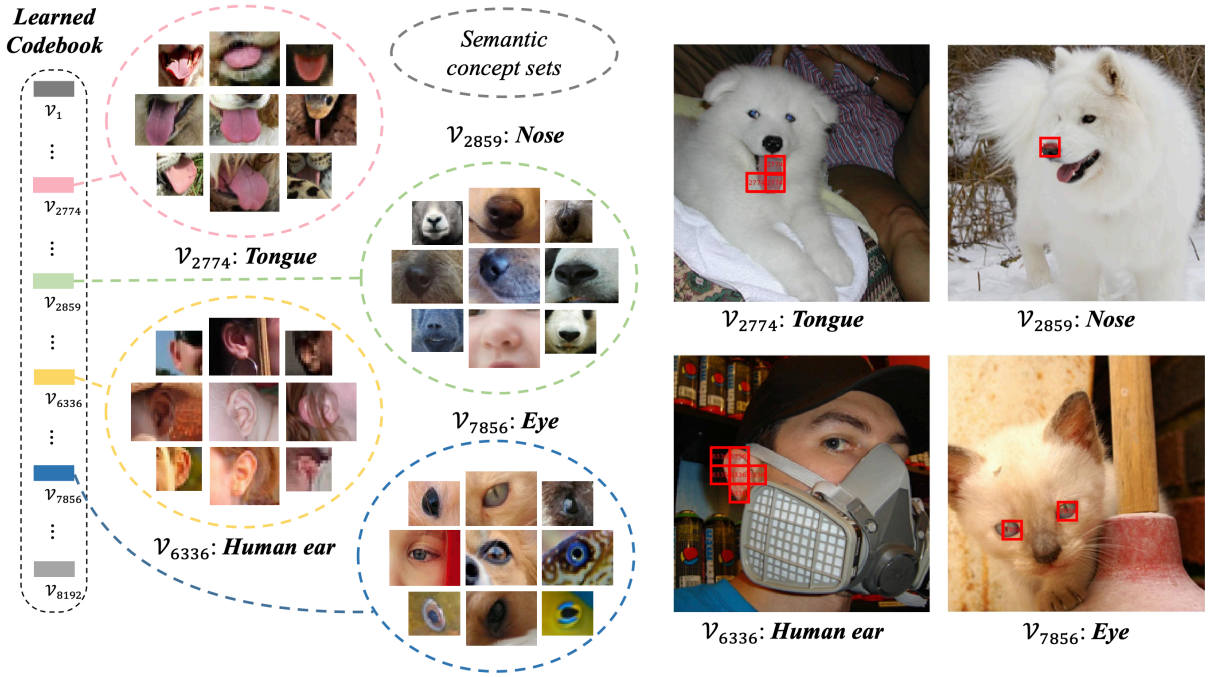

Figure 2: [6].

Bibliography

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds.,

Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[2] Y. LeCun and I. Misra, "Self-supervised learning: The dark matter of intelligence." [Online]. Available: https://ai.meta.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

[3] W. Wang *et al.*, "Image as a Foreign Language: BEIT Pretraining for Vision and Vision-Language Tasks," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 19175–19186. doi: 10.1109/CVPR52729.2023.01838.

[4] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR*, 2021.

[5] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT Pre-Training of Image Transformers," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=p-BhZSz59o4

[6] Z. Peng, L. Dong, H. Bao, Q. Ye, and F. Wei, "BEiT v2: Masked Image Modeling with Vector-Quantized Visual Tokenizers," 2022.