CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP

Seminar in AI

Pascal Pilz — k12111234
k12111234@students.jku.at

February 28, 2024

1 Introduction

The 2022 paper "CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP" by Fürst et al. [1] presents an improvement on CLIP (Contrastive Language-Image Pre-training, Radford et al. [2]), which is considered to be foundational and has been used as a benchmark several times [1, 3].

Both methods use contrastive learning and aim to match text prompts with images in a zero-shot transfer learning setting, that is, performing predictions on classes that the model has not encountered during training. The main difference and improvement of CLOOB over CLIP is the use of modern Hopfield networks to overcome a problem of CLIP, namely the problem of "explaining away", which means that the model focuses on one object seen in the image while ignoring other important aspects [1, 4].

Fürst et al. [1] propose the integration of modern Hopfield networks into the model architecture of CLIP. They show that this, together with the use of a different objective function - InfoLOOB - leads to improved performance both in theory and in experimental setups.

2 Background

According to Radford et al. [2], there has been a revolution in the field of natural language processing in the past decade. The development of "text-to-text" as a standardized input-output format combined with significant increases in compute power, model size, and data, has enabled the development of task-agnostic models that perform well on zero-shot transfer learning tasks.

Radford et al. [2] go on to explain that these developments suggested to them "that the aggregate supervision accessible to modern pre-training methods within web-scale collections of text surpasses that of high-quality crowd-labeled NLP datasets", prompting

them to investigate whether scalable pre-trained methods learning directly from web text could achieve similar breakthroughs in computer vision.

3 Multimodal Setting

Both Fürst et al. [1] and Radford et al. [2] project text prompts and images into the same latent space. The objective is to have the projected embeddings of positive text-image pairs close to each other while having negative pairs further apart in terms of their cosine similarity. This makes both methods multi-modal and contrastive in nature.

In both settings we have a set of N iid drawn image-text pairs $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$, where $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are the image embeddings obtained via a method such as a ResNet or a vision transformer model, and $(\mathbf{y}_1, \ldots, \mathbf{y}_N)$ are the corresponding text encodings obtained via a continuous bag of words or a text transformer model [1, 2].

Both the image and text embeddings are normalized such that $\|\mathbf{x}_i\| = \|\mathbf{y}_j\| = 1$. Because of this, Fürst et al. [1] often talk of the uniformity of the sphere when discussing the distribution of the embeddings.

4 CLIP and CLOOB

4.1 CLIP

The basic architecture of CLIP can best be understood from Figure 1. During the training of CLIP, both an image encoder and a text encoder are trained in order to obtain the desired embedding properties. At inference time, the model predicts the most likely prompt from a given set of prompts that corresponds to a given image.

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I)  #[n, d_i]
T_f = text_encoder(T)   #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Figure 1: *Numpy-like pseudocode for the core of an implementation of CLIP. Adapted from Radford et al. [2]*

The training objective for CLIP is to minimize the InfoNCE loss function (1).

$$\mathrm{L}_{\mathrm{InfoNCE}} = -\frac{1}{N}\ln\sum_{i=1}^{N}\frac{\exp(\tau^{-1}\mathbf{x}_i^T\mathbf{y}_i)}{\sum_{j=1}^{N}\exp(\tau^{-1}\mathbf{x}_i^T\mathbf{y}_j)} - \frac{1}{N}\sum_{i=1}^{N}\ln\frac{\exp(\tau^{-1}\mathbf{x}_i^T\mathbf{y}_i)}{\sum_{j=1}^{N}\exp(\tau^{-1}\mathbf{x}_j^T\mathbf{y}_i)} \tag{1}$$

4.2 Modern Hopfield Networks

A modern Hopfield network is a neural network that can store and retrieve patterns, making it a form of associative memory. During training, patterns are fed into the network and the weights are adjusted until an equilibrium is reached, and during inference a partial or noisy pattern can be fed in to obtain the stored pattern with the closest resemblance. This essentially creates archetypes, allowing the network to associate similar patterns and filter out noise.

The basic concept is best illustrated with an example from Rumetshofer and Fürst [5], and the corresponding image in Figure 2: For demonstration purposes, only the network features corresponding to a teacup, a teapot, a piece of cake, and a fork are arranged so that, when activated, they visually form the corresponding objects. In the rightmost images, we can see the underlying concept of a tea house, which is understood as the co-occurrence of a teacup, a teapot, a piece of cake, and a fork. This is what the network

has stored in its weights. In the leftmost images, we have the objects observed in two different locations. This is the input to the network. We can see that all the objects corresponding to the stored concept of a tea house are present with varying levels of strength, but the modern Hopfield network is able to extract this information and, after performing a number of updates to network, the middle images can be observed.

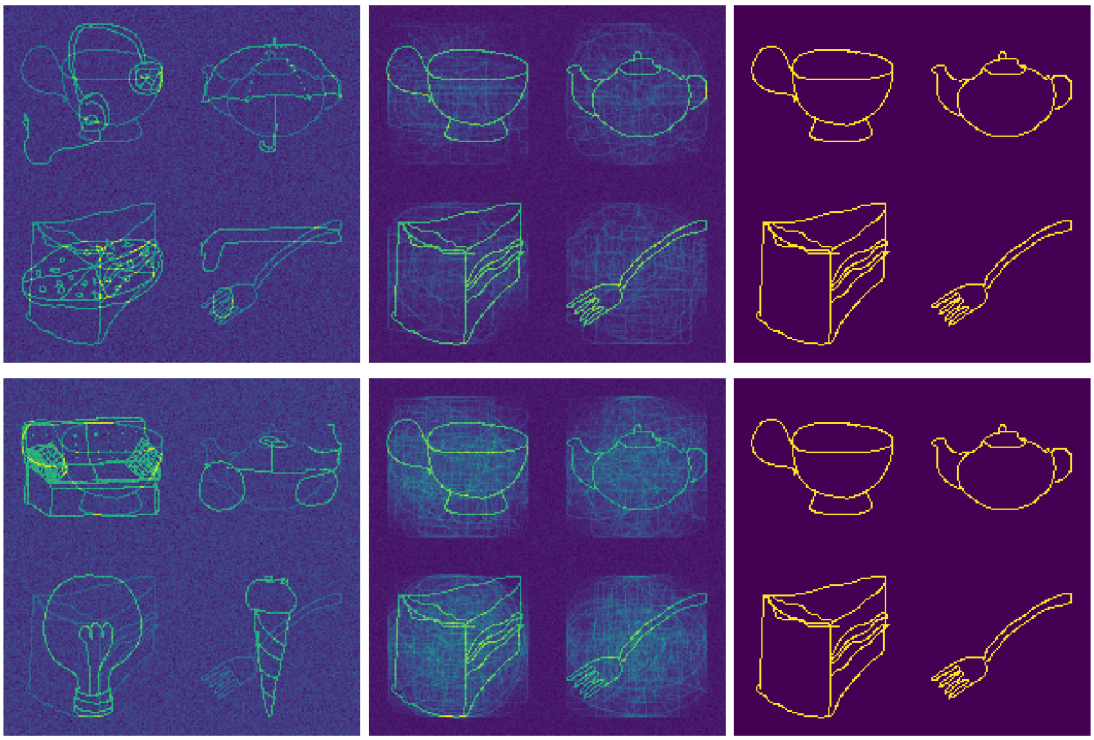

Figure 2: *Stylized representation of the input and output to a Hopfield network that has the concept of a tea house stored in it. Adapted from Rumetshofer and Fürst [5].*

Modern Hopfield networks can help with tackling the explaining away problem of CLIP, which is known in reasoning as the concept of accepting one cause of an event and dismissing all other causes [4]. According to Fürst et al. [1], this problem is caused by the model insufficiently extracting co-occurrences and covariance from the multimodal data. To combat this, Fürst et al. [1] suggest to use modern Hopfield networks to first store the calculated image and text embeddings and retrieve them immediately.

4.3 CLOOB

CLOOB's basic architecture is very similar to CLIP and can be seen in Figure 3. During training, CLIP trains both an image encoder and a text encoder in order to obtain the desired embedding properties. At inference time, the model predicts the most likely prompt from a given set of prompts that corresponds to a given image.

```
# image_encoder       - ResNet or Vision Transformer
# text_encoder        - CBOW or Text Transformer
# I [n , h , w , c ]  - minibatch of aligned images
# T [n , l ]          - minibatch of aligned texts
# W_i [ d_i , d_e ]   - learned proj of image to embed
# W_t [ d_t , d_e ]   - learned proj of text to embed
# b                   - inverse temperature Hopfield retrieval
# t                   - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I)  #[n, d_i]
T_f = text_encoder(T)   #[n, d_t]

# joint multimodal embedding
x = l2_normalize(I_f @ W_i)   #[n, d_e]
y = l2_normalize(T_f @ W_t)   #[n, d_e]

# Hopfield retrieval H with batch stored
# H(beta, A, B) = B.T @ softmax(beta * A @ B.T)
U_x = H(b, x, x).T #[n, d_e]
U_y = H(b, y, x).T #[n, d_e]
V_x = H(b, x, y).T #[n, d_e]
V_y = H(b, y, y).T #[n, d_e]

# normalize retrievals
U_x = l2_normalize(U_x)   #[n, d_e]
U_y = l2_normalize(U_y)   #[n, d_e]
V_x = l2_normalize(V_x)   #[n, d_e]
V_y = l2_normalize(V_y)   #[n, d_e]

# loss: info_loob(tau, anchors, samples)
loss_i = info_loob(t, U_x, U_y)
loss_t = info_loob(t, V_y, V_x)
loss = (loss_i + loss_t) * t
```

Figure 3: *Numpy-like pseudocode for the core of an implementation of CLOOB. Adapted from Fürst et al. [1]*

The training objective for CLOOB is to minimize the InfoLOOB loss function (2).

$$\mathrm{L}_{\mathrm{InfoLOOB}} = -\frac{1}{N}\ln\sum_{i=1}^{N}\frac{\exp(\tau^{-1}\mathbf{x}_i^T\mathbf{y}_i)}{\sum_{j\neq i}^{N}\exp(\tau^{-1}\mathbf{x}_i^T\mathbf{y}_j)} - \frac{1}{N}\sum_{i=1}^{N}\ln\frac{\exp(\tau^{-1}\mathbf{x}_i^T\mathbf{y}_i)}{\sum_{j\neq i}^{N}\exp(\tau^{-1}\mathbf{x}_j^T\mathbf{y}_i)} \tag{2}$$

4.4 InfoLOOB and the Saturation Effect

To explain the saturation effect, it is useful to consider only the second sum and the first sample of the InfoNCE (1) and InfoLOOB objective 2:

$$\mathrm{L}_{\mathrm{InfoNCE}}(\mathbf{y}_1) = -\ln \frac{\overbrace{\exp(\tau^{-1}\mathbf{x}_1^T\mathbf{y}_1)}^{a}}{\underbrace{\exp(\tau^{-1}\mathbf{x}_1^T\mathbf{y}_1)}_{a} + \underbrace{\sum_{j=2}^{N}\exp(\tau^{-1}\mathbf{x}_j^T\mathbf{y}_1)}_{b}} \tag{3}$$

$$\mathrm{L}_{\mathrm{InfoLOOB}}(\mathbf{y}_1) = -\ln \frac{\overbrace{\exp(\tau^{-1}\mathbf{x}_1^T\mathbf{y}_1)}^{a}}{\underbrace{\sum_{j=2}^{N}\exp(\tau^{-1}\mathbf{x}_j^T\mathbf{y}_1)}_{b}} \tag{4}$$

Inherently, minimizing the InfoNCE loss is equivalent to increasing a, the similarity of the positive pair, and decreasing b, the similarity of the negative pairs. The denominator includes both a and b, which means that if a is large relative to b, the value of the fraction will change little with any change in a or b.

Fürst et al. [1] call this the saturation effect, and according to them, it is a problem when using modern Hopfield networks, since the retrieved embeddings naturally have a higher similarity to each other. To combat the saturation effect, they suggest using the InfoLOOB objective, which excludes the positive pair a from the denominator.

5 Experiments

Fürst et al. [1] conducted a series of experiments to compare CLOOB with CLIP, pretraining once on the Conceptual Caption (CC) dataset and once on the Yahoo Flickr Creative Commons (YFCC) dataset. They also conducted ablation studies to determine the effects of two newly introduced changes, (1) modern Hopfield networks and (2) InfoLOOB.

For each of the three experiments, CLOOB was implemented based on OpenCLIP by Ilharco et al. [6]. Similarly, CLIP was reimplemented based on OpenCLIP. OpenCLIP achieves equivalent results to those reported by Radford et al. [2].

5.1 Zero-Shot Transfer Learning

5.1.1 Conceptual Caption

The Conceptual Captions (CC) dataset [7] consists of 2.9 million images with high-quality, detailed text descriptions. Since the original CLIP was not trained on this dataset, the authors of CLOOB used their own reimplementation for the comparison. For both models they used a ResNet-50 and a BERT architecture to encode the image-text pairs. The results of zero-shot transfer learning experiments on seven different datasets are shown in Table 1.

Dataset	CLIP RN-50	CLOOB RN-50	CLIP RN-50	CLOOB RN-50
Birdsnap	2.26 ± 0.20	$\mathbf{3.06 \pm 0.30}$	2.8 ± 0.16	$\mathbf{3.24 \pm 0.31}$
Country211	0.67 ± 0.11	0.67 ± 0.05	0.7 ± 0.04	0.73 ± 0.05
Flowers102	12.56 ± 0.38	13.45 ± 1.19	13.32 ± 0.43	14.36 ± 1.17
GTSRB	7.66 ± 1.07	6.38 ± 2.11	8.96 ± 1.70	7.03 ± 1.22
UCF101	20.98 ± 1.55	22.26 ± 0.72	21.63 ± 0.65	$\mathbf{23.03 \pm 0.85}$
Stanford Cars	0.91 ± 0.10	$\mathbf{1.23 \pm 0.10}$	0.99 ± 0.16	$\mathbf{1.41 \pm 0.32}$
ImageNet	20.33 ± 0.28	$\mathbf{23.97 \pm 0.15}$	21.3 ± 0.42	$\mathbf{25.67 \pm 0.22}$
ImageNet V2	20.24 ± 0.50	$\mathbf{23.59 \pm 0.15}$	21.24 ± 0.22	$\mathbf{25.49 \pm 0.11}$

Table 1: *Zero-shot results for models trained on CC with ResNet-50 vision encoders for two different checkpoints. Results are given as mean accuracy over 5 runs. Statistically significant results are shown in bold. CLIP and CLOOB were trained for 31 epochs while CLIP* and CLOOB* were trained for 128 epochs. Adapted from Fürst et al. [1].*

5.1.2 Yahoo Flickr Creative Commons

The Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset [8] consists of 99.2 million images, of which the authors chose a subset of 15 million images with less rich textual description then the CC dataset. Again, since the original CLIP was not trained on this dataset, the authors of CLOOB used their own reimplementation for the comparison. For both models they used a ResNet-50, a ResNet-101, and a ResNet-50x4, as well as a BERT architecture to encode the image-text pairs. The results of zero-shot transfer learning experiments on seven different datasets are shown in Table 2

	RN-50		RN-101		RN-50x4	
Dataset	CLIP	CLOOB	CLIP	CLOOB	CLIP	CLOOB
Birdsnap	21.8	**28.9**	22.6	**30.3**	20.8	**32.0**
Country211	6.9	**7.9**	7.8	**8.5**	8.1	**9.3**
Flowers102	48.0	**55.1**	48.0	**55.3**	50.1	**54.3**
GTSRB	7.9	**8.1**	7.4	**11.6**	9.4	**11.8**
UCF101	**27.2**	25.3	28.6	**28.8**	31.0	**31.9**
Stanford Cars	3.7	**4.1**	3.8	**5.5**	3.5	**6.1**
ImageNet	34.6	**35.7**	35.3	**37.1**	37.7	**39.0**
ImageNet V2	33.4	**34.6**	34.1	**35.6**	35.9	**37.3**

Table 2: *Zero-shot results for the CLIP reimplementation and CLOOB using different ResNet architectures trained on YFCC. Adapted from Fürst et al. [1].*

5.2 Ablation studies

Fürst et al. [1] found that the interplay of modern Hopfield networks with InfoLOOB is crucial, as either one alone shows little to no improvement (see Table 3).

Dataset	InfoNCE	InfoLOOB	InfoNCE Hopfield	nfoLOOB Hopfield	InfoNCE*	InfoLOOB*	InfoNCE Hopfield*	InfoLOOB Hopfield*
Birdsnap	**2.58**	2.37	1.67	2.53	2.15	1.89	2.15	**3.39**
Country211	0.53	0.63	0.54	**0.76**	0.62	0.62	0.66	**0.79**
Flowers102	13.16	13.03	11.53	**14.24**	11.79	11.57	10.86	**14.24**
GTSRB	4.47	4.39	5.76	**5.86**	**9.25**	6.93	6.24	8.67
UCF101	**23.68**	19.14	20.56	22.29	21.33	20.56	21.40	**24.05**
Stanford Cars	**1.38**	1.33	1.24	1.37	1.26	1.19	1.24	**1.62**
ImageNet	21.74	22.13	19.04	**24.21**	22.80	22.69	20.69	**25.59**
ImageNet V2	21.45	21.65	18.97	**23.80**	22.44	22.13	20.22	**25.50**

Table 3: *Influence of loss functions and Hopfield retrieval for models pre-trained on CC for 31 epochs (left) and 128 epochs (right, indicated by *). Adapted from Fürst et al. [1].*

They claim that this is due to the underfitting tendency of modern Hopfield networks counteracting the overfitting tendency of the InfoLOOB objective. To support this claim they show a comparison of the cosine similarity of matched and unmatched pairs. Figure 4 shows that modern Hopfield networks lead to higher similarity between unmatched and matched pairs, while InfoLOOB leads to higher similarity in matched pairs.

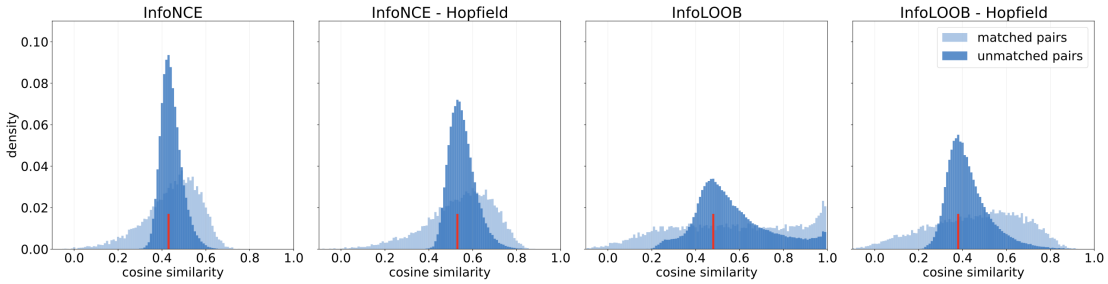

Figure 4: *Distribution of the cosine similarity of matched pairs and the cosine similarity of the 10 unmatched pairs that have the highest similarity score with the anchor. Adapted from Fürst et al. [1].*

In addition, Fürst et al. [1] discuss the ability of modern Hopfield networks to extract more covariance structure during learning. This can be seen in Figure 5, which shows the number of eigenvalues needed to reconstruct 99% of the covariance matrix of the corresponding embeddings at an early point in training compared to a later point in training. A higher number indicates that the covariance matrix is more complex.

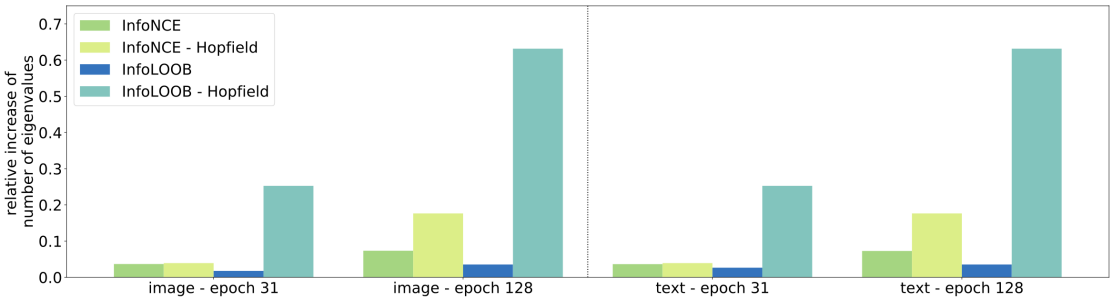

Figure 5: *Relative change in the number of the effective eigenvalues of the embedding covariance matrices, which were obtained from image and text encoders at two different training points. Adapted from Fürst et al. [1].*

6 Discussion and Critical Assessment

Unlike Table 1 (Table 1 in Fürst et al. [1]) and the corresponding experiment, Fürst et al. [1] do not explicitly explain the meaning of the bold-highlighted text in Table 2 (Table 3 in Fürst et al. [1]) and whether or not multiple tests were performed and the results averaged. Therefore, it is not clear to me how the YFCC experiment was performed and what the highlighting in Table 2 is supposed to show. The same problem exists with Table 3 (Table A1 in Fürst et al. [1]). Fürst et al. [1] do not explicitly discuss aspects such as data and energy efficiency when it comes to CLOOB. Nor do they discuss possible reasons for not using modern Hopfield networks.

The aspects of modern Hopfield networks concerning their covariance structure were difficult for me to understand. Nevertheless, it should be noted that Fürst et al. [1] take great care to explain concepts in an understandable and detailed way, making it easier to understand for readers with less experience in the field. Furthermore, they fulfil all relevant points of the "NeurIPS 2022 Paper Checklist Guidelines". In addition, they strive to include as many potentially interesting theoretical and experimental results as possible in the appendix.

References

[1] Andreas Fürst et al. *CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP*. 2022. arXiv: 2110.11316 [cs.LG].

[2] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[3] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *CoRR* abs/2108.07258 (2021). arXiv: 2108.07258. URL: https://arxiv.org/abs/2108.07258.

[4] M.P. Wellman and M. Henrion. "Explaining 'explaining away'". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.3 (1993), pp. 287–292. DOI: 10.1109/34.204911.

[5] Elisabeth Rumetshofer and Andreas Fürst. *CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP*. Accessed on Jone 19, 2023. 2023. URL: https://ml-jku.github.io/cloob/.

[6] Gabriel Ilharco et al. *Open Clip*. 2021.

[7] Piyush Sharma et al. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: Jan. 2018, pp. 2556–2565. DOI: 10.18653/v1/P18-1238.

[8] Bart Thomee et al. "YFCC100M: the new data in multimedia research". In: *Commun. ACM* 59.2 (Jan. 2016), pp. 64–73. ISSN: 0001-0782. DOI: 10.1145/2812802. URL: https://doi.org/10.1145/2812802.