Different triplet sampling techniques for lossless triplet loss on metric similarity learning

Gábor Kertész

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

John von Neumann Faculty of Informatics

SAMI 2021, January 21-23, 2021

Contents

Introduction

- Metric learning is a special type of supervised learning: learning is based on observation similarity
- Many applications in facial recognition:
  - FaceNet[1]
  - DeepFace[2]
  - OpenFace[3]

[1]Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823

[2]Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "Deepface: Closing the gap to human-level performance in face verification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1701–1708

[3]Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, et al. "Openface: A general-purpose face recognition library with mobile applications". In: *CMU School of Computer Science* 6 (2016), p. 2

Contrastive loss

The original loss function defined for Siamese Networks[4] is the contrastive loss, given as

$$\mathcal{L}_{\texttt{contrastive}} = (1 - Y)\texttt{max}(0, (m - d(x_1, x_2))) + Yd(x_1, x_2),$$

where Y represents the label, having $Y = 0$ for objects of different categories and $Y = 1$ for objects of the same category. The marginal distance of category representations is represented by m, and $d(a, b)$ function gives the distance between the two vectors, represented as x_1 and x_2.

[4] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems*. 1994, pp. 737–744

Contrastive loss

- The idea of not increasing the distance of different object embeddings after a marginal distance is reached is appropriate: after distinct clusters of vectors are formed, there is no need to further distinguish these sets.
- On the other hand however, as the loss for same-category objects will always be a positive value, later in the training a small positive value, these clusters will *implode*, if the model is trained for too long: overtraining could happen, and generalization will suffer.

Triplet loss

Triplet loss seems to deal with this issue: the embedding network is trained by using three inputs: an anchor, a positive pair from the same category, and a negative pair from a separate category.
Triplet loss can be formally defined as:

$$\mathcal{L}_{\mathtt{triplet}} = \mathtt{max}(0, m + d(x_a, x_p) - d(x_a, x_n)),$$

where x_a stands for the embedding vector of the anchor, x_p and x_n are the positive and negative pairs.

Triplet loss

$$\mathcal{L}_{\texttt{triplet}} = \texttt{max}(0, m + d(x_a, x_p) - d(x_a, x_n))$$

When interpreting this function, we can conclude, that a positive loss is produced in every case when $m + d(x_a, x_p) > d(x_a, x_n)$, in other cases the loss will be zero. As a result, a close negative pair and a distant positive pair is penalized during training; however, same category objects are not pushed together if their distance increased with the margin is below the measured negative distance.

Triplet loss

The problem with triplet loss is that this phenomena will occur more often during training, for most of the available triplets. As all available triplets can not be used during training due to the large numbers of possible variations, often random selection is applied. This however, might result in a large number of *easy* triplets, were the loss is zero[5]. To deal with this issue, different triplet mining techniques were proposed[6][7][8].

[5] Alexander Hermans, Lucas Beyer, and Bastian Leibe. "In defense of the triplet loss for person re-identification". In: *arXiv preprint arXiv:1703.07737* (2017)

[6] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. "Training region-based object detectors with online hard example mining". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 761–769

[7] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. "Sampling matters in deep embedding learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2840–2848

[8] Hong Xuan, Abby Stylianou, and Robert Pless. "Improved embeddings with easy positive triplet mining". In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 2474–2482

Contents

Methodology

In the following experiment, the triplet loss and a so-called lossless triplet loss function is compared on a simple dataset.

Lossless triplet loss

Marc-Olivier Arsenault described a different approach to deal with zero values in triplet loss[9]; this method is further referred to as lossless triplet loss. Lossless triplet loss is based on the logarithmic function, as

$$\begin{aligned}\mathcal{L}_{\texttt{lossless}} = &-\log\left(-\frac{d(x_a, x_p)^2}{N} + 1 + \epsilon\right)\\ &-\log\left(-\frac{N - d(x_a, x_n)^2}{N} + 1 + \epsilon\right),\end{aligned}$$

where N stands for the number of dimensions in embedding space, and ϵ is a small non-zero value, for example 10^{-8}. The operator log means the natural logarithm.

[9] Marc-Olivier Arsenault. *Lossless Triplet loss*. 2018. URL: https://coffeeanddata.ca/lossless-triplet-loss

Lossless triplet loss

$$\begin{aligned}\mathcal{L}_{\texttt{lossless}} = &-\log\left(-\frac{d(x_a, x_p)^2}{N} + 1 + \epsilon\right) \\ &-\log\left(-\frac{N - d(x_a, x_n)^2}{N} + 1 + \epsilon\right)\end{aligned}$$

When calculating the loss, the positive and negative distances are both transformed using this non-linear function, and are summarized.

Lossless triplet loss

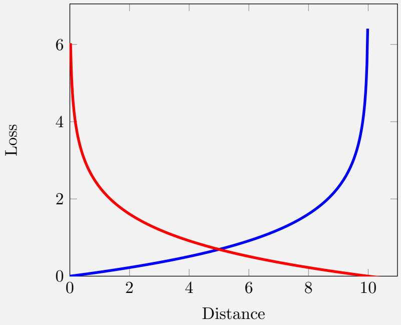


Figure: Plotting the positive and negative components of the lossless triplet loss. The blue curve visualizes the positive distance based $-\log\left(-\frac{x^2}{N}+1\right)$, where $N = 10$. The red curve shows $-\log\left(-\frac{N-x^2}{N}+1\right)$, representing the negative distance.

Negative selection

To optimize performance, the selection of negative pairs is important. Negative pair selection can be categorized as

- easy:
  random negative selection;
- semi-hard:
  the negative distance is below the marginal positive distance: $d(x_a, x_p) + m > d(x_a, x_n)$;
- hard:
  the selected negatives are closer to the anchor than the positive pair.

Negative selection

We define random negative selection as **easy mining**, as in this is the most efficient way to select negative samples. However, in case of random selection, it is possible to select triplets where the negative distance is significantly larger than the positive, resulting in zero calculated loss.
Semi-hard and hard negative mining yields those samples, where the negative distance is small: in case of hard mining the selected negatives are closer to the anchor than the positive pair. In case of semi-hard mining, a sample is satisfying if the negative distance is below the marginal positive distance:
$d(x_a, x_p) + m > d(x_a, x_n)$, resulting in a non-zero loss.

Contents

Data

- NIST SD19 dataset[10]: the original collection of handwritten digits and letters
- characters from 3669 hadwritten forms were collected and digitalized to create a set containing a total of 814,255 samples from a total of 62 classes
- 128×128 binary images, dark pixels on white background
- To highlight the details, the irrelevant parts were cropped out from the samples, while keeping the squared aspect ratio

[10] Patrick Grother and Kayee Hanaoka. "NIST special database 19 handprinted forms and characters 2nd Edition". In: *National Institute of Standards and Technology, Tech. Rep* (2016)

Data

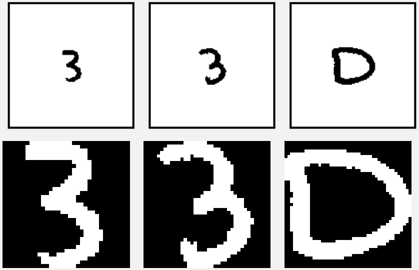


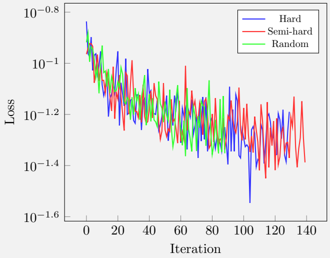
Figure: Samples of the NIST SD19 dataset before and after content highlighting. The top row shows the original samples, and the bottom row visualizes the samples after content highlighting.

Embedding network

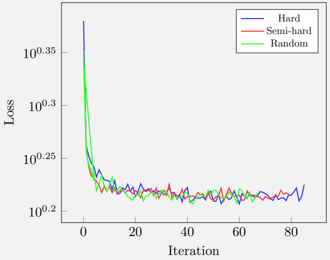
Table: The structure of the embedding neural network. The input samples are 96×96 matrices, which is followed by a number of convolutional and pooling layers, and finally fully connected *"dense"* layers are applied.

	Type	**Properties**
1	Input	96×96
2	Conv	5×5@64
3	MaxPool	2×2
4	Conv	3×3@128
5	MaxPool	2×2
6	Conv	3×3@256
7	Conv	3×3@256
8	Conv	3×3@256
9	MaxPool	2×2
10	Conv	3×3@512
11	Conv	3×3@512
12	MaxPool	2×2
13	Flatten	
14	Dense	512
15	Dense	256
16	Dense	64

Training loss



(a) Loss over training iterations using the triplet loss.



(b) Loss over training iterations using the lossless triplet loss.

Figure: Displaying the evolution of measured training loss after iterations when applying the original triplet loss (a), and the lossless triplet loss (b).

Training loss

- both show signs of converging to the minima;
- convergence staled, and training was stopped in less iterations for the lossless function;
- the original triplet loss is more volatile, while the lossless function results in a smoother curve;
- easy random triplet mining is suboptimal;
- semi-hard and hard negative mining seems promising for both cases;
- both methods have the discriminative ability to separate representations of the input samples.

Visualization

Model trained on the original triplet loss

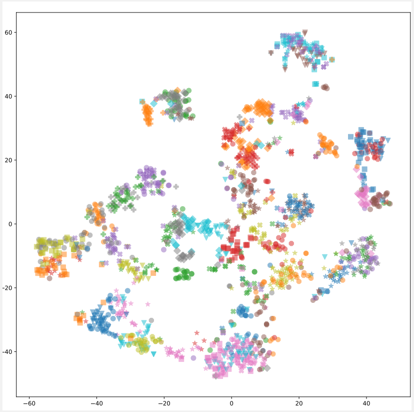


Figure: t-SNE based visualization of the validation samples of the NIST SD19 dataset using hard triplet mining technique.

Visualization

Model trained on the lossless triplet loss

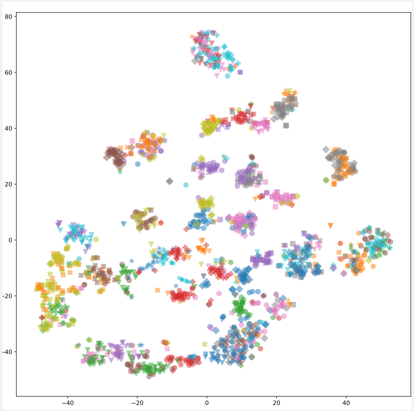


Figure: t-SNE based visualization of the validation samples of the NIST SD19 dataset using hard triplet mining technique.

Contents

Conclusions & future plans

- the lossless method seems less volatile, and shows promising behavior
- future plans include several other experiments, which are necesseraly performed before jumping to serious conclusions
- a larger and wider dataset, and the application of a more robust backbone architecture is interesting

Thank you for your attention!

Gábor KERTÉSZ
kertesz.gabor@nik.uni-obuda.hu

Stay safe!

The author is grateful to the members of the Óbuda University GPGPU Programming Research Group for constructive comments and suggestions.