Exponential triplet loss

Evalds Urtans
Riga Technical University
Riga, Latvia
+37126401317
evalds.urtans@rtu.lv

Agris Nitkitenko
Riga Technical University
Riga, Latvia
agris.nikitenko@rtu.lv

Valters Vecins
Riga Technical University
Riga, Latvia
valters.vecins@rtu.lv

ABSTRACT

This paper introduces a novel variant of the Triplet Loss function that converges faster and gives better results. This function can separate class instances homogeneously through the whole embedding space. With Exponential Triplet Loss function we also introduce a novel type of embedding space regularization Unit-Range and Unit-Bounce that utilizes euclidean space more efficiently and resembles features of the cosine distance. We also examined factors for choosing the best embedding vector size for specific embedding spaces. Finally, we also demonstrate how new function can train models for one-shot learning and re-identification tasks.

CCS CONCEPTS

• **Theory of computation** → **Design and analysis of algorithms**;
• **Applied computing**;

KEYWORDS

Triplet loss, Feature embedding, Sample mining, One-shot learning, Identification, Re-identification

1 INTRODUCTION

Models that are capable of creating embedding representation that is somewhat disentangled and can be interpreted using distance metrics empowers many kinds of deep learning fields tasks starting with representation learning [1] [2], one-shot learning [3] [4], auto-encoders, generative models and reinforcement learning.

Nowadays, it is often not feasible to store all raw data from sensors and method for extracting and compressing only valuable data in the form of embedding is needed. Also, edge cases in data can be found using embedding queries thereby improving the quality of training data. Using the same queries these models can perform classification tasks on novel classes that have not been seen during training. Contrary to standard classification models, the precision of these models can be adjusted using cluster radiuses and other proximity metrics after they have been trained. These models also learn to generalize well and fewer samples to identify novel classes whereas classic classification models need a lot more data samples [5].

ICCDA 2020, March 9–12, 2020, Silicon Valley, CA, USA

ACM ISBN 978-1-4503-7644-0/20/03…$15.00
https://doi.org/10.1145/3388142.3388163

Sample-based methods like triplet loss [6], contrastive loss [7] [8] and N-tuple loss [8] has a goal to reduce the distance in embedding space between same class samples and increase the distance between different class samples. These methods require extensive sample mining methods and sample selection constraints to converge class representations into clusters. In this paper, we propose a new type of function to replace triplet loss that does not need sample selection constraints. This new function has an exponential shape to help converge faster samples that are further away from the desired state.

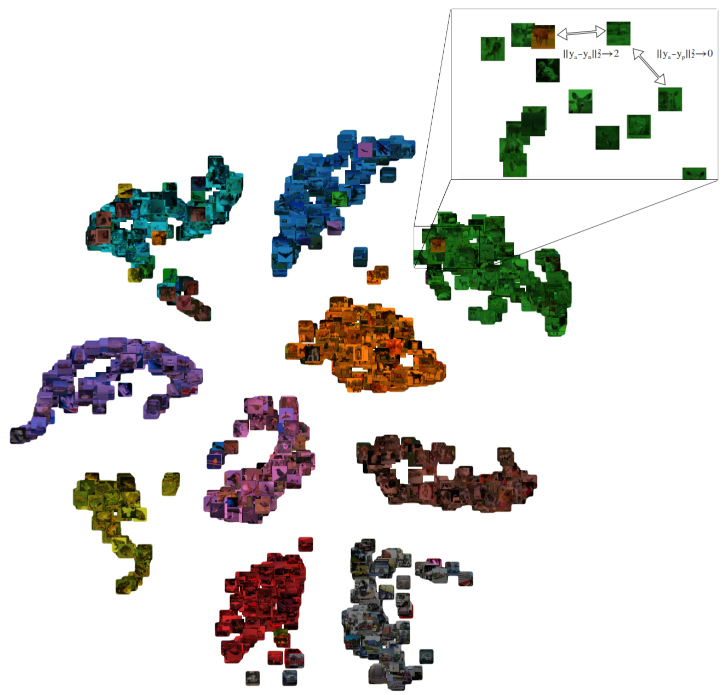

Figure 1. t-SNE embeddings of CIFAR10 trained with $\mathcal{L}_{exp}$. Colors denote different classes of samples. Exponential Triplet Loss decreases the distance between anchor y_a and positive same class sample y_p and increases the distance between different class sample y_n. Zoomed area shows sample from horse class that looks very similar to deer class samples.

2 RELATED WORK

Important research papers in Triplet Loss come from applications in face identification and re-identification tasks [6] [9].

Many variations of Triplet Loss have been explored by scientific community, notable mentions include Lifted Structured Loss [2], Histogram Loss [10], PDDM [11], n-pair loss [12], Triplet Ranking Loss [13], Additive Angular Margin Loss [14], Lossless Triplet Loss [15], N-Tuple Loss [8]. Similarly to our research, Margin Ranking Loss [16] worked on loss function shaping and described several shortcomings in standard Triplet Loss. One of the recent advances in Triplet Loss is Margin loss [17] that has some similar properties to our proposed loss function.

There has been a number of recent advances also in sample mining methods. Distance weighted sampling improves sample distributions in mini-batches [17], but is more useful in tandem with Margin Loss function. Doppelganger Mining [18] that takes into account most similar samples but with different classes. Hard example mining with auxiliary embeddings focuses on additional class features that can be used for better sampling strategy [19].

3 METHODOLOGY

Standard Triplet loss function (1) works with distances between embedding vectors y_p of same class as anchor embedding vector y_a and embedding vectors y_n of a different class than anchor embedding vector. Within the function, α is used as a margin between classes to not push them too far away and not to have them too close to each other.

$$\mathcal{L}_{std} = |||y_a - y_p||_2^2 - ||y_a - y_n||_2^2 + \alpha|_+ \quad (1)$$

In Figure 2 of function $\mathcal{L}_{std}$ it is possible to see that many pairs in lower bound of loss function converge inconsistently.

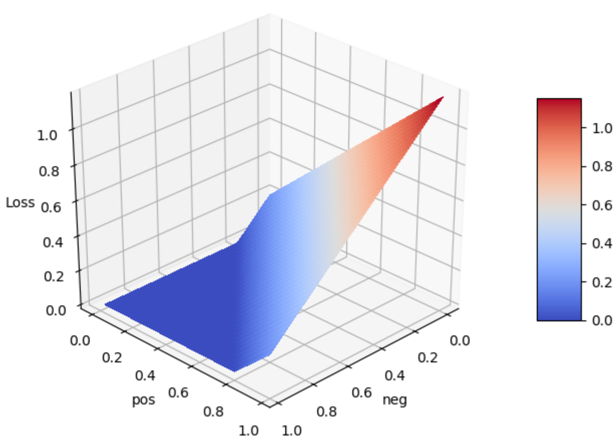

Figure 2. $\mathcal{L}_{std}$ function depending on positive pair $||y_a - y_p||_2^2$ (pos) and negative pair $||y_a - y_n||_2^2$ (neg) of embedding vectors

To have stable convergence of $\mathcal{L}_{std}$ so that negative embedding vector pairs are further away and positive are closer to each other at least one of two constraints must be enforced [9] [6] [20]:

(1) "Hard constraint" $||y_a - y_p||_2^2 + \alpha < ||y_a - y_n||_2^2$
(2) "Semi-hard constraint" $||y_a - y_n||_2^2 < ||y_a - y_p||_2^2$

After applying constraints in Figure 3 of function $\mathcal{L}_{std}$ it is possible to observe that area of sample combinations that guides model to converge is very small. To have convergence using $\mathcal{L}_{std}$ function it is necessary to have sample mining procedure before each mini-batch. Common approach is to use "batch hard" or "batch all" sample mining [21].

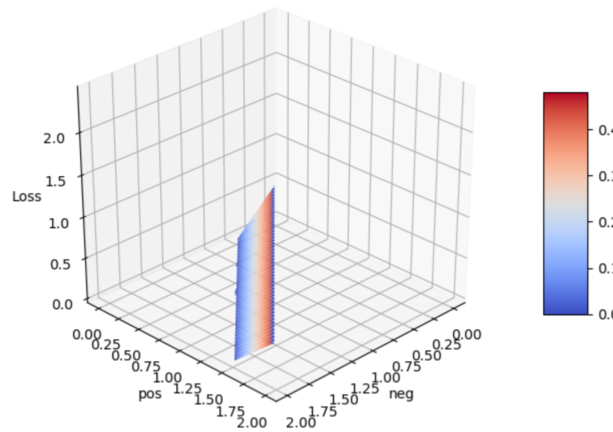

Figure 3. $\mathcal{L}_{std}$ function with constraints depending on positive pair $||y_a - y_p||_2^2$ (pos) and negative pair $||y_a - y_n||_2^2$ (neg) of embedding vectors

Our proposed Exponential Triplet Loss function $\mathcal{L}_{exp}$ has higher area where it can converge as seen in Figure 4. It also exploits exponential shape of function so that samples further away from desirable locations have much higher loss value. $\mathcal{L}_{exp}$ has asymmetric plateau where it does not draw closer pairs that are within half of embedding space maximum distance $\max(f_{emb}(x)$. As this loss function have a lot larger area of convergence it is less dependent on sample mining. Hyper-parameters C_{pos} and C_{neg} can be usually set to 1.0.

$$\begin{aligned}\mathcal{L}_{exp} = &-C_{pos} \cdot log(1.0 - \frac{|emb_p - c_n|_+}{1 - c_n} + \epsilon) \\ &-C_{neg} \cdot log(1.0 - \frac{|0.5 - emb_n|_+}{0.5} + \epsilon)\end{aligned}$$

$$emb_p = \frac{y_p}{\max(f_{emb}(x))} \qquad emb_n = \frac{y_n}{\max(f_{emb}(x))} \quad (2)$$

It is not beneficial to push embedding vectors too far away from each other when the model would be used for the one-shot learning task. Sample of an unseen class that has not been introduced during training would "jump" between modalities of training data-set distribution. For one-shot learning task, homogeneous distribution of classes withing embedding space would be desirable. In order to enforce this, we propose overlap coefficient c_o that describes how much instances of different classes in data-set should overlap with each other. Good value of c_o for clean data-sets is 1.5 that produces partial overlap in between all classes.

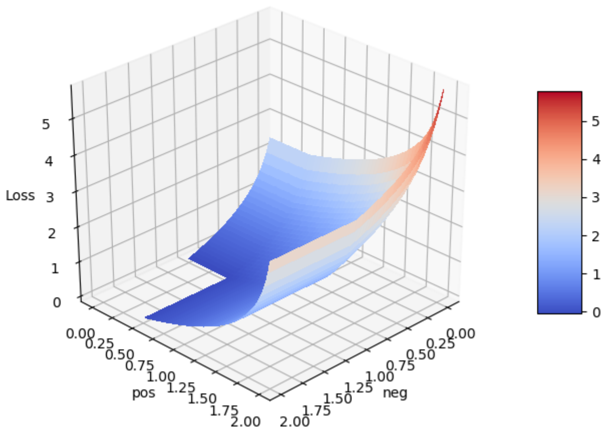

Figure 4. $\mathcal{L}_{exp}$ function depending on positive pair $\|y_a - y_p\|_2^2$ (pos) and negative pair $\|y_a - y_n\|_2^2$ (neg) of embedding vectors

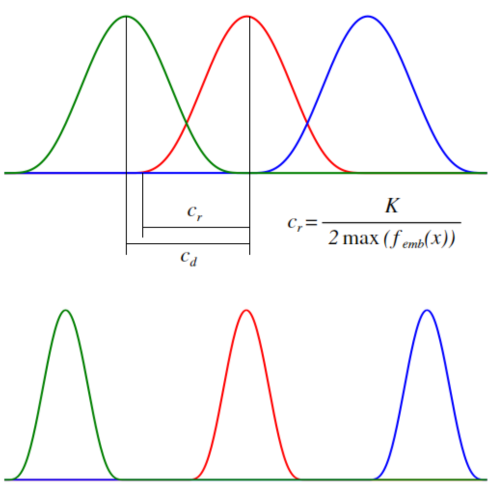

Figure 5. On the top example of overlapping class clusters in 1D with homogeneous embedding space. On the bottom example of non-homogeneous embedding space where new class samples will "jump" between clusters.

Function (3) evenly splits space into c_d embedding distances for each class in data-set. The number of classes in training data-set is K. When the model is used in the inference stage then the same K value is used as in the training stage even though the number of classes in inference might differ from training. $\mathcal{L}_{exp}$ can be used with cosine distance that has upper bound of $f_{emb} = 2$. It enables calculation of overlap distance that is calculated dividing this upper bound of distance by the number of classes and multiplied by overlap coefficient.

$$c_d = \frac{c_o \cdot K}{\max(f_{emb}(x))} \tag{3}$$

To use euclidean distances with $\mathcal{L}_{exp}$ it is necessary to have the upper bound of distance. It can be solved either by applying L2 normalization in which case it will always be $f_{emb} = 2 \cdot c_s$ depending on scale the maximum distance. But it constrains embedding vector positions within spherical space. We propose to use hybrid space that behaves as euclidean space when it is closer to the centre and when it reaches the radius of L2 then it behaves like spherical space. We call this function Unit-Range (4). Looking at embedding vectors using simple PCA it is possible to observe that Unit-Range space is more homogeneous than spherical space enforced by L2 normalization.

$$f_{emb}(x) = \begin{cases} c_s \frac{x}{|x|^2}, & \text{if } |x|^2 \geq 1 \\ x, & \text{otherwise} \end{cases} \tag{4}$$

Another function to normalize embedding space that is introduced by this paper is called Unit-Bounce (5). It resembles features of cosine distance as when embedding vector reaches the edge of a sphere it bounces back towards the centre of embedding space. When embedding reaches the opposite side of the sphere with a radius of c_s it bounces back towards centre again, similarly how the maximum angle between two vectors is always 180 degrees when calculating cosine distance.

$$f'_{emb}(x) = \begin{cases} f_{bounce}(x), & \text{if } |x|^2 \geq 1 \\ x, & \text{otherwise} \end{cases} \tag{5}$$

$$f_{bounce}(x) = \begin{cases} |x|^2 - \left\lfloor \frac{|x|^2}{c_s} \right\rfloor - c_s \frac{x}{|x|^2}, & \text{if } \left\lfloor \frac{|x|}{c_s} \right\rfloor \bmod 2 = 0 \\ c_s \frac{x}{|x|^2} - |x|^2 - \left\lfloor \frac{|x|^2}{c_s} \right\rfloor, & \text{otherwise} \end{cases} \tag{6}$$

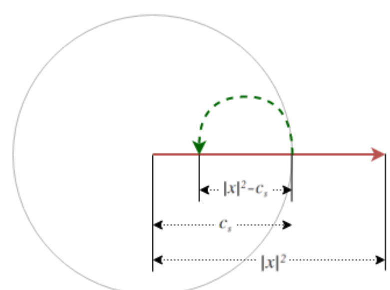

Figure 6. Illustration of Unit-Bounce embedding normalization function within L2 spherical space.

4 EXPERIMENTS

Proposed exponential triplet loss function $\mathcal{L}_{exp}$ has been tested on several image data-sets: MNIST, Fassion-MNIST, EMNIST, CIFAR10, CIFAR100 and VGGFace2 [22]. VGGFace2 dataset is typically used for face re-identification and verification tasks as well as one-shot learning evaluation. Implementation in PyTorch is available as an open-source code repository: https://github.com/evaldsurtans/exp-triplet-loss.

As an architecture of models we used pre-trained DenseNet-121 on ImageNet [23]. At the end of the model, we added the global average pooling function and max-out function with 16 linear units.

For all $\mathcal{L}_{exp}$ and $\mathcal{L}_{std}$ experiments grid search of best hyper-parameters were done to compare best results for each method. Experiments were repeated 5 times with the same set of hyper-parameters because in some cases convergence is sensitive to initialization.

For measuring accuracy two metrics were used:

- Closest accuracy (nearest neighbour) - after each epoch class cluster centres were calculated and each sample was assigned to the class closest to its embedding vector [24].
- Range accuracy - during each epoch class cluster centres were calculated and also their maximum distances within a class. Each sample added value of 1 to the class one-hot encoded vector so that embedding vector is in class cluster range. Afterwards, a one-hot encoded vector was L1 normalized to have probabilistic representation.

4.1 The dimensionality of a embedding vector

From empirical tests depending on an initialization method and the embedding space, we concluded that there is a dimension size limit for each combination these two parameters at which it does not improve performance of embedding models. It matches a simple experiment done by initializing different size vectors with a chosen pair of parameters shown in Figure 7. For example in L2 normalized spherical embedding space using cosine distances there are very small differences between vector size of 256 and 1024. For euclidean distances when L2 or Unit-Range even smaller embedding vector could be used at the size of in between 32 and 128. Experiments with training models with different embedding sizes confirmed this finding as shown in Figure 8. Intuitive explanation to this is that once distances between all samples at the beginning of training are same by random initialization then it is a good starting point for converging samples into homogeneous clusters covering the whole embedding space. At this point, it does not improve results to increase the dimension count of embedding vector. The same behaviour applies to cosine space as well as Euclidean space.

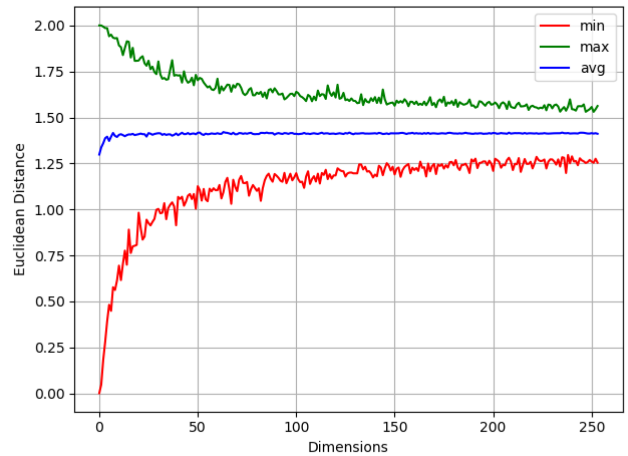

Figure 7. Cosine distances between 1000 sampled vectors that have been initialized by uniform distribution depending on dimension size.

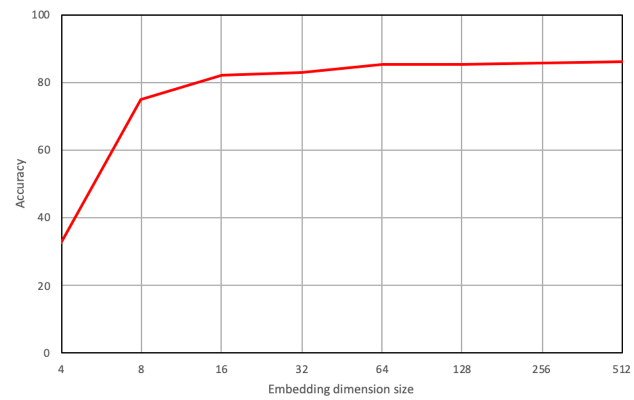

Figure 8. Accuracy of EMINST test data-set depending on dimension size of embedding vector. Unit-range and euclidean distances has been used in training.

4.2 Initialization of embedding vector

From empirical tests, we noticed that c_d had a higher effect on training if linear units of the embedding vector have been initialized using a uniform distribution. Such initialization yielded more uniform distribution of class clusters in the embedding space than more uni-modal initialization distributions. In Figure 9 distances between a sample and closest class cluster centre has been displayed. On left, there are distances before training, middle during training and on right after training. On top embedding, vectors are initialized using

Xavier initialization, but on the bottom using uniform initialization. Measurements have been taken from CIFAR10 data-set.

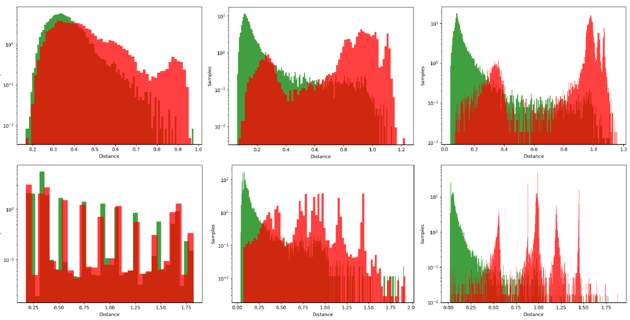

Figure 9. Comparison of xavier initialization on top and uniform initialization on bottom of embedding vectors. On left before training, in middle mid-part of training, on right after training.

4.3 Composite loss function

In order to improve performance of embedding models composite loss function were used. For all comparisons between $\mathcal{L}_{exp}$ and $\mathcal{L}_{std}$ composite loss functions were used. We introduced L2-constrained Softmax [25] with cross entropy $\mathcal{L}_{class}$ and center loss $\mathcal{L}_{center}$. We extended center loss (8) adding margin or radius of desired cluster to maintain (9). The idea is to discourage the collapse of embedding into one point in the embedding space. Within $\mathcal{L}_{class}$ input in Softmax function $f(x)$ is L2 normalized and scaled by s. Within $\mathcal{L}_{class}$ during training class instances are accumulated and c_{y_i} is calculated centre of cluster.

$$\mathcal{L}_{class} = -\sum_{i=1}^{M} y_i log \frac{e^{W_i^T s |f(x_i)|_2^2 + b_i}}{\sum_{j=1}^{C} e^{W_j^T s |f(x_i)|_2^2 + b_j}} \tag{7}$$

$$\mathcal{L}_{center'} = \sum_{i=1}^{M} ||x_i - c_{y_i}||_2^2 \tag{8}$$

$$\mathcal{L}_{center} = \sum_{i=1}^{M} |||x_i - c_{y_i}||_2^2 - \frac{c_d}{2}|_+ \tag{9}$$

$$\mathcal{L}_{\rfloor\wr\Updownarrow_{\surd}} = \mathcal{L}_{exp} + C_{center}\mathcal{L}_{center} + C_{class}\mathcal{L}_{class} \tag{10}$$

4.4 One-shot learning

A model trained for one-shot learning task or re-identification task will work with novel classes in the test phase that has not been seen in the training phase. As $\mathcal{L}_{exp}$ models shown in Figure 10 and Figure 11 tend to have homogeneous embedding space they are able to cluster unseen class samples in between classes that have been seen during the training phase. Uniform initialization mentioned in the previous section is also beneficial for one-shot learning task where novel samples would be more evenly distributed in the embedding space.

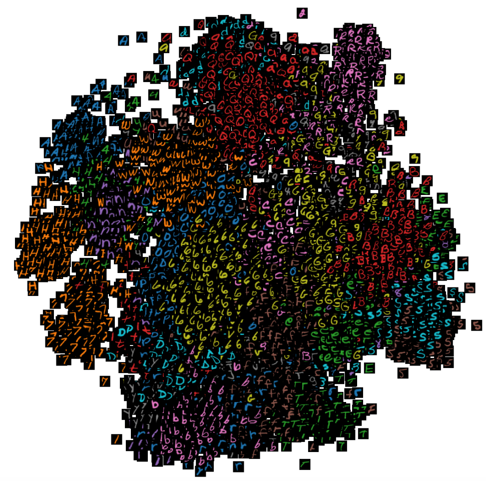

Figure 10. PCA of 2D of embeddings of class clusters seen in training. Model trained using $\mathcal{L}_{exp}$ and EMINST.

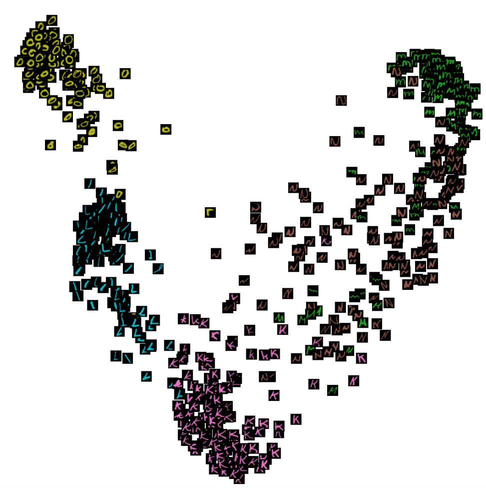

Figure 11. PCA of 2D of embeddings of class clusters not-seen in training. Model trained using $\mathcal{L}_{exp}$ and EMINST.

4.5 Results

Comparisons were made using the identification task wherein the inference phase there were the same classes as in the training phase. For each set of methods, hyper-parameters were tuned and multiple repeat training runs were made to compare only the best-performing model. All models were trained using state-of-the-art optimizer algorithm RAdam [26]. For some models, accuracy is close of those achieved by state of the art classification models. After analysis of mislabeled samples in the embedding space, it was possible to find many indistinguishable and wrong samples within training data-sets. All results of VGGFace2 have been obtained from re-identification task and one-shot learning task where test data-set classes were not included in train data-set. For VGGFace2 we used only 1000 samples of each class in training and testing data-sets with lower resolution of 128x128 pixels.

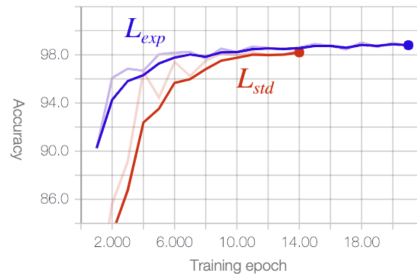

Figure 12. Comparison of convergence speed between $\mathcal{L}_{exp}$ and $\mathcal{L}_{std}$ on Fassion-MINST data-set.

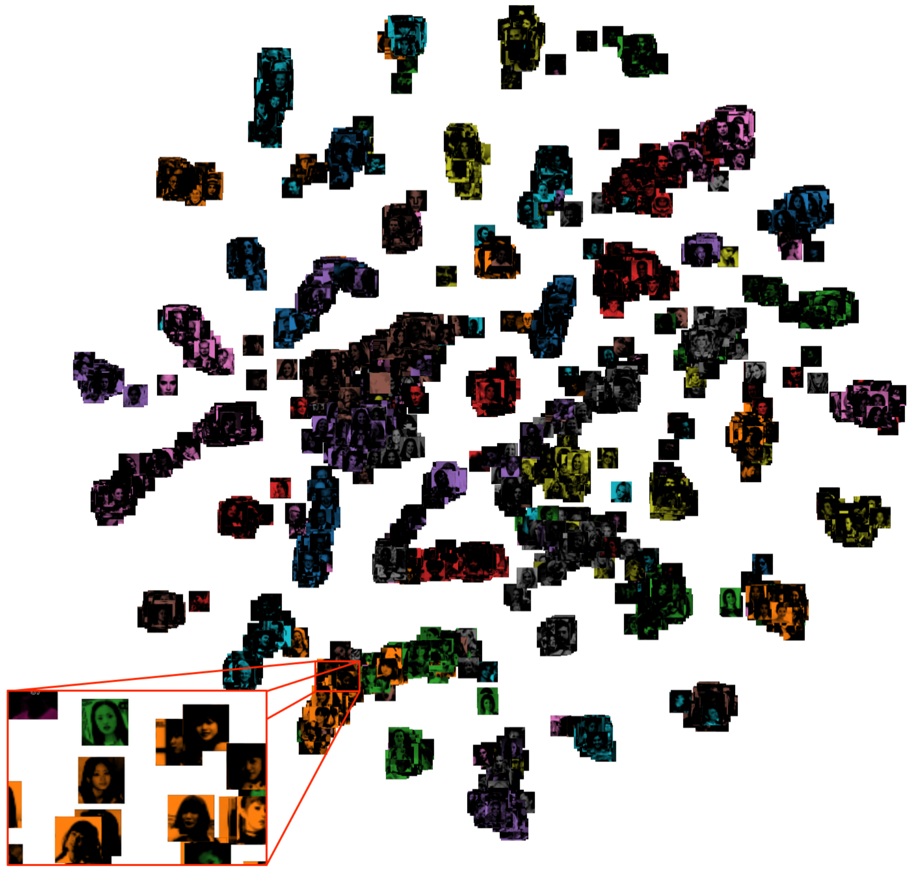

Figure 13. t-SNE of 50 color-coded samples of VGGFace2 test data-set trained by $\mathcal{L}_{exp}$. In lower-left corner sample given where model clusters together close samples of similar Asian woman face with dark hair.

Table 1. Comparison of accuracy between models trained with different types of loss functions. Accuracy calculated by closest cluster centre to a sample. All models use best fitted hyper-parameters and superior Unit-Range type of embedding normalization.

Loss func. / Accuracy %	MNIST	Fassion-MNIST	EMINST	CIFAR10	VGGFace2
L_{std}	**99.6**	91.4	82.0	56.2	77.4
$L_{std} + L_{class}$	**99.6**	92.1	85.0	79.8	76.3
$L_{std} + L_{center}$	97.5	71.5	61.7	52.1	76.4
$L_{std} + L_{center} + L_{class}$	97.7	82.0	70.9	62.8	78.6
L_{exp}	**99.6**	92.7	82.7	85.7	85.0
$L_{exp} + L_{class}$	**99.6**	**93.1**	85.2	87.2	84.1
$L_{exp} + L_{center}$	**99.6**	**93.1**	85.7	85.3	84.0
$L_{exp} + L_{center} + L_{class}$	**99.6**	**93.1**	**86.0**	**87.3**	**85.7**

5 CONCLUSIONS

Proposed Exponential Triplet Loss function provides an easier way to train embedding models. With this function, models converge faster and have higher accuracy and class separation. They produce embeddings in better utilized and more homogeneous Unit-Range and Unit-Bounce embedding spaces than in L2 spherical embedding space. The embedding normalization function Unit-Bounce resembled the same properties as cosine distances but using euclidean distances. Also, the training relies on less sample mining as the convergence space covers more of the sample space. The embedding models in this paper generalizes well also in one-shot learning task where novel class samples grouped in clusters even though they were not seen during the training phase.

6 ACKNOWLEDGMENTS

Research has been completed with support from High-Performance Computing Center of Riga Technical University that provided 12 nVidia K40 GPUs and 8 nVidia V100 GPUs. Special thanks to Lauris Cikovskis.

REFERENCES

[1] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, , T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 1199–1208. http://openaccess.thecvf.com/content_cvpr_2018/html/Sung_Learning_to_Compare_CVPR_2018_paper.html

[2] H. O. Song, Y. Xiang, S. Jegelka, , S. Savarese, "Deep Metric Learning via Lifted Structured Feature Embedding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 4004–4012. https://doi.org/10.1109/CVPR.2016.434

[3] M. Bucher, S. Herbin, , F. Jurie, "Improving Semantic Embedding Consistency by Metric Learning for Zero-Shot Classiffication," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pp. 730–746. https://doi.org/10.1007/978-3-319-46454-1_44

[4] Y. Yuan, K. Yang, , C. Zhang, "Hard-Aware Deeply Cascaded Embedding," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 814–823. https://doi.org/10.1109/ICCV.2017.94

[5] C. Wang, X. Lan, , X. Zhang, "How to train triplet networks with 100K identities?" pp. 1907–1915.

[6] F. Schroff, D. Kalenichenko, , J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 815–823. https://doi.org/10.1109/CVPR.2015.7298682

[7] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, , R. Shah, "Signature Verification Using A "Siamese" Time Delay Neural Network," vol. 7, no. 4, pp. 669–688. https://doi.org/10.1142/S0218001493000339

[8] S. Chopra, R. Hadsell, , Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pp. 539–546. https://doi.org/10.1109/CVPR.2005.202

[9] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, , Y. Wu, "Learning Fine-Grained Image Similarity with Deep Ranking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 1386–1393. https://doi.org/10.1109/CVPR.2014.180

[10] E. Ustinova , V. S. Lempitsky, "Learning Deep Embeddings with Histogram Loss," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4170–4178. http://papers.nips.cc/paper/6464-learning-deep-embeddings-with-histogram-loss

[11] C. Huang, C. C. Loy, , X. Tang, "Local Similarity-Aware Deep Feature Embedding," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1262–1270. http://papers.nips.cc/paper/6368-local-similarity-aware-deep-feature-embedding

[12] B. Amos, B. Ludwiczuk, , M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications," in *CMU.*

[13] M. Ye , Y. Guo, "Deep Triplet Ranking Networks for One-Shot Recognition," vol. abs/1804.07275. http://arxiv.org/abs/1804.07275

[14] J. Deng, J. Guo, , S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," vol. abs/1801.07698. http://arxiv.org/abs/1801.07698

[15] Marc-Olivier Arsenault. Lossless Triplet loss. https://towardsdatascience.com/lossless-triplet-loss-7e932f990b24

[16] D. P. Vassileios Balntas, Edgar Riba , K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson , W. A. P. Smith, Eds. BMVA Press, pp. 119.1–119.11. https://dx.doi.org/10.5244/C.30.119

[17] R. Manmatha, C.-Y. Wu, A. J. Smola, , P. Krähenbühl, "Sampling Matters in Deep Embedding Learning," pp. 2859–2867.

[18] E. Smirnov, A. Melnikov, S. Novoselov, E. Luckyanets, , G. Lavrentyeva, "Doppelganger Mining for Face Representation Learning," in *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, pp. 1916–1923. https://doi.org/10.1109/ICCVW.2017.226

[19] E. Smirnov, A. Melnikov, A. Oleinik, E. Ivanova, I. Kalinovskiy, , E. Luckyanets, "Hard Example Mining With Auxiliary Embeddings," in *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 37–46. http://openaccess.thecvf.com/content_cvpr_2018_workshops/w1/html/Smirnov_Hard_Example_Mining_CVPR_2018_paper.html

[20] O. M. Parkhi, A. Vedaldi, , A. Zisserman, "Deep Face Recognition," in *BMVC.*

[21] A. Hermans, L. Beyer, , B. Leibe, "In defense of the triplet loss for person re-identification," vol. abs/1703.07737.

[22] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, , A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," pp. 67–74.

[23] G. Huang, Z. Liu, , K. Q. Weinberger, "Densely Connected Convolutional Networks," pp. 2261–2269.

[24] K. Q. Weinberger , L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," vol. 10, pp. 207–244. https://dl.acm.org/citation.cfm?id=1577078

[25] R. Ranjan, C. D. Castillo, , R. Chellappa, "L2-constrained Softmax Loss for Discriminative Face Verification," vol. abs/1703.09507. http://arxiv.org/abs/1703.09507

[26] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, , J. Han, "On the variance of the adaptive learning rate and beyond," vol. abs/1908.03265.

Columns on Last Page Should Be Made As Close As Possible to Equal Length

Authors' Research Background

*The form itself will not be published.

*Title can be chosen from: master student, Ph.D. candidate, assistant professor, lecturer, associate professor, full professor

Your Name	Title*	Research Field	Personal website
Evalds Urtans	PhD. cand.	Deep learning, loss functions	http://yellowrobot.xyz
Agris Nitkitenko	Professor	Robotics	https://ortus.rtu.lv/science/en/experts/1522
Valters Vecins	MSc.	Deep learning, UNets	N/A