A Metric Learning Reality Check

Охрименко Дмитрий, 172

Metric learning

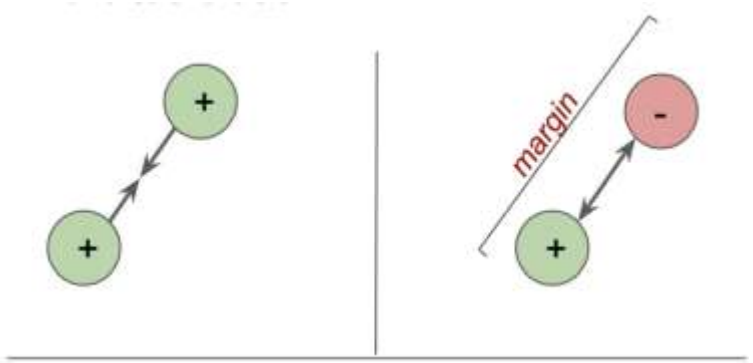
- Задача – определить похожесть двух объектов
- Пытаемся сопоставить данные с пространством эмбеддингов, где похожие данные находятся близко друг к другу, а разные данные далеко друг от друга.
- Можно использовать embedding losses и classificational losses

Classificational losses

- Основаны на использовании весовой матрицы, где каждый столбец соответствует определенному классу
- Обучение состоит из матричного умножения весов на векторы эмбеддингов для получения логитов и последующего применения функции потерь к логитам.
- Есть множество вариаций (normalized softmax loss, ProxyNCA, SphereFace, CosFace, ArcFace, SoftTriple losses)

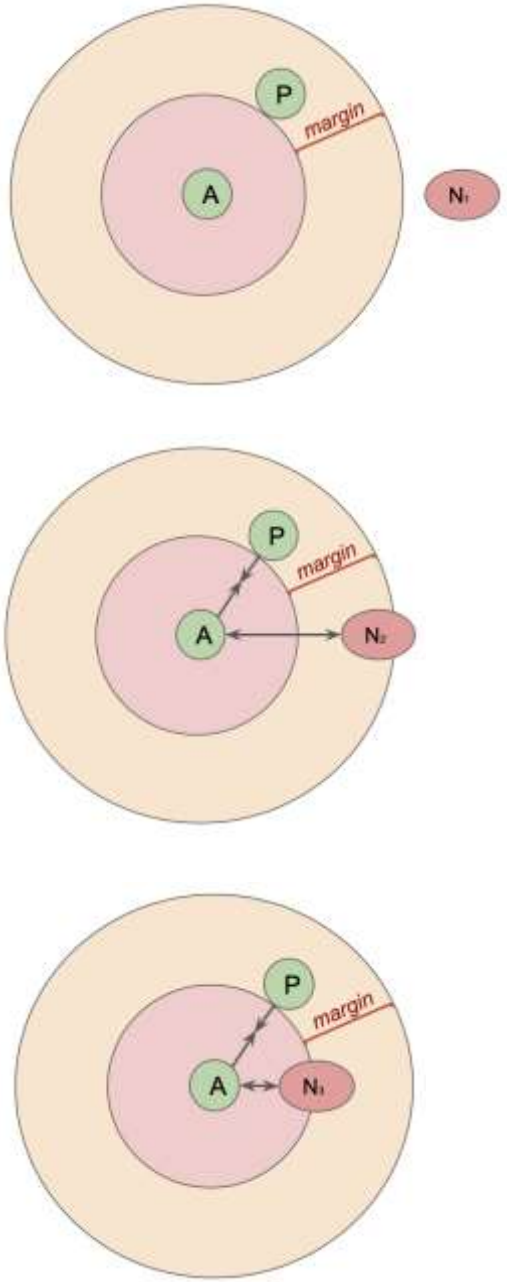
Embedding losses

- Contrastive loss – одинаковые сдвигаем, только если они дальше m_{pos}, разные отталкиваем, только если они ближе m_{neg}
- $L_{contrastive} = [d_p - m_{pos}]_+ + [m_{neg} - d_n]_+$
- m_{pos} может быть равен 0 (как на картинке)



Embedding losses

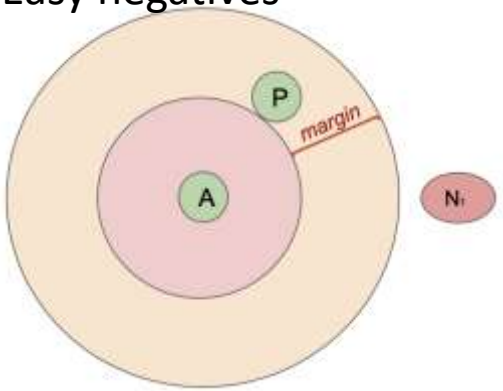
- Triplet loss – A (anchor) похож на Positive больше, чем на Negative. Цель – сделать расстояние AP меньше чем AN на выбранный margin.
- $L_{triplet} = [d_{ap} - d_{an} + m]_+$



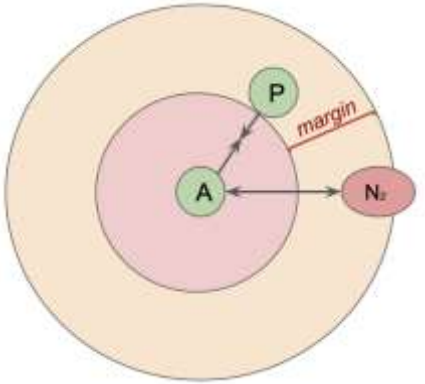
Pair and triplet mining

- Offline mining – выбираем тройки до конструирования батчей
- Online mining – находим тройки в каждом рандомном батче
- Easy negatives – не влияют на обучение
- Hard negatives – вызывают обучение на плохих данных
- Semi-hard negatives – остается только этот вариант

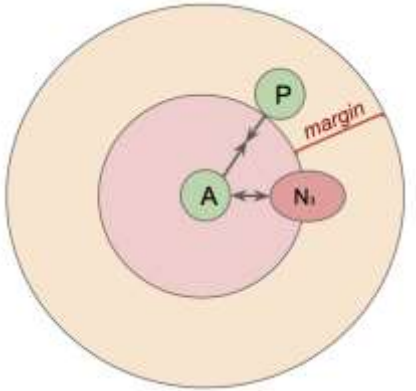
Easy negatives



Semi-hard negatives



Hard negatives



Проблемы в существующих статьях

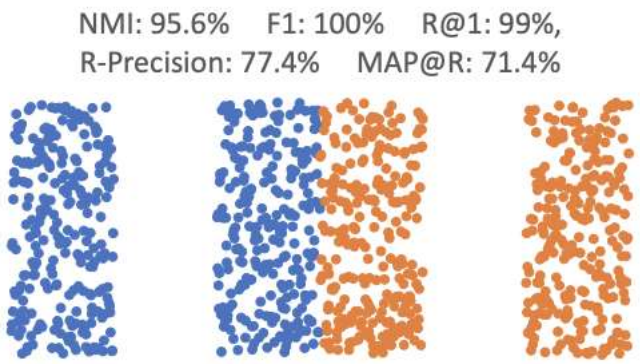
- Нечестные сравнения
- Неточность используемых метрик
- Переобучение на тестовой выборке

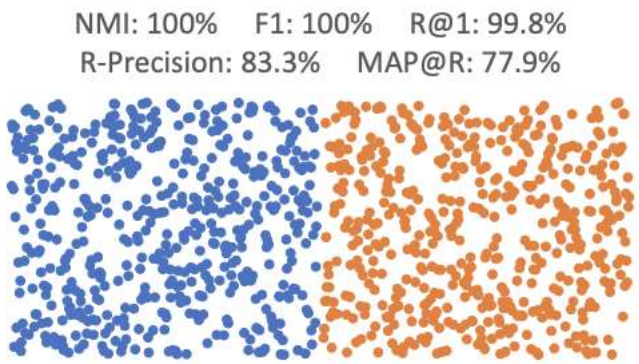
Нечестные сравнения

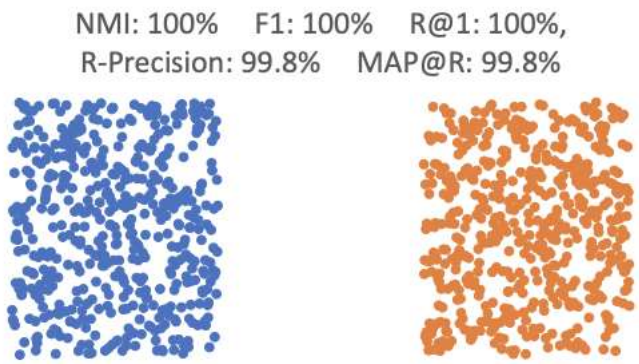
- Чтобы заявить, что новый алгоритм превосходит существующие методы, важно сохранять как можно больше параметров постоянными, однако в большинстве существующих работ этого не происходит
- Простые способы повысить точность - обновить сетевую архитектуру или использовать более сложные способы увеличения изображения

Неточность используемых метрик

- Обычно используются Recall@K, Normalized Mutual Information (NMI) и F1 score
- Три картинки снизу демонстрируют различные разделения на классы, однако показания метрик очень схожи







Переобучение на тестовой выборке

- В большинстве статей данные делятся на тестовые и обучающие 50/50 и гиперпараметры настраиваются в зависимости от поведения на тестовых данных
- Это может привести к переобучению на тестовых данных, что является большой проблемой

Предлагаемые решения

- Честные сравнения и воспроизводимость
- Информативные метрики
- Подбор гиперпараметров через кросс-валидацию

Честные сравнения и воспроизводимость

- Использовалась предоубченная на ImageNet BN-Inception сеть с размером выходных эмбеддднигов = 128
- Размер батча = 32
- Батчи создаются случайной выборкой C классов, а затем случайным разделением M картинок на C классов
- Для embedding losses C=8, M=4; для classification losses C=32, M=1
- Во время обучения изображения увеличиваются с использованием стратегии обрезки со случайным изменением размера.
- Все параметры сети оптимизированы с помощью RMSprop с lr = 1e-6
- Эмбеддинги L2 нормализуются перед подсчетом лосса и во время оценки

Информативная метрика

- R-precision = $\frac{r}{R}$, R – ближайших к запросу элементов, r – относящиеся к тому же классу
- R-precision не учитывает ранжирование правильных выборок, поэтому используем MAP@R – Mean Average Precision at R:

$$MAP@R = \frac{1}{R}\sum_{i=1}^{R} P(i)$$

$$P(i) = \begin{cases} precision_i, \text{если i–е извлечение верно} \\ 0, \text{иначе} \end{cases}$$

Информативная метрика

- MAP@R информативнее R@1
- Может быть вычислен прямо из пространства эмбеддингов
- Вознаграждает хорошо сгруппированные пространства эмбеддингов
- Легкая в понимании
- Стабильнее, чем R@1

$$MAP@R = \frac{1}{R}\sum_{i=1}^{R} P(i)$$

$$P(i) = \begin{cases} precision_i, \text{если i–е извлечение верно} \\ 0, \text{иначе} \end{cases}$$

Информативная метрика

- MAP@R информативнее R@1
- Может быть вычислен прямо из пространства эмбеддингов
- Вознаграждает хорошо сгруппированные пространства эмбеддингов
- Легкая в понимании
- Стабильнее, чем R@1

Retrieval results	**Recall@1**	**R-Precision**	**MAP@R**
10 results, of which only the 1st is correct	100	10	10
10 results, of which the 1st and 10th are correct	100	20	12
10 results, of which the 1st and 2nd are correct	100	20	20
10 results, of which all 10 are correct	100	100	100

Подбор гиперпараметров через кросс-валидацию

- Запускается 50 итераций байесовской проверки, каждая состит из 4-х кратной перекрестной проверки
- В результате наборы для обучения и проверки всегда не пересекаются с классами, поэтому оптимизация производительности набора проверки должна быть хорошим показателем точности для задач с открытым набором
- Гиперпараметры оптимизированы для максимизации средней точности проверки. Для получения лучших гиперпараметров загружается контрольная точка наивысшей точности для каждого раздела обучающего набора, вычисляются его эмбеддинги для тестового набора и нормализуются L2 нормой. Точность вычисляется объединенным и раздельным способами
- Результаты слабо подвержены начальному шуму, так как проводится 10 тренировочных прогонов с использованием лучших гиперпараметров и выбирается среднее

Эксперименты

- Провели эксперименты с 13 лоссами на трех датасетах: CUB200, Cars196 и Stanford Online Products (SOP)

Method	Year	Loss type
Contrastive [16]	2006	Embedding
Triplet [63]	2006	Embedding
NT-Xent [50,38,6]	2016	Embedding
ProxyNCA [35]	2017	Classification
Margin [65]	2017	Embedding
Margin / class [65]	2017	Embedding
Normalized Softmax (N. Softmax) [58,31,72]	2017	Classification
CosFace [57,59]	2018	Classification
ArcFace [11]	2019	Classification
FastAP [3]	2019	Embedding
Signal to Noise Ratio Contrastive (SNR) [70]	2019	Embedding
MultiSimilarity (MS) [62]	2019	Embedding
MS+Miner [62]	2019	Embedding
SoftTriple [41]	2019	Classification

Эксперименты

Table 4. Accuracy on CUB200

	Concatenated (512-dim)			Separated (128-dim)		
	P@1	**RP**	**MAP@R**	**P@1**	**RP**	**MAP@R**
Pretrained	51.05	24.85	14.21	50.54	25.12	14.53
Contrastive	**68.13 ± 0.31**	37.24 ± 0.28	26.53 ± 0.29	59.73 ± 0.40	31.98 ± 0.29	21.18 ± 0.28
Triplet	64.24 ± 0.26	34.55 ± 0.24	23.69 ± 0.23	55.76 ± 0.27	29.55 ± 0.16	18.75 ± 0.15
NT-Xent	66.61 ± 0.29	35.96 ± 0.21	25.09 ± 0.22	58.12 ± 0.23	30.81 ± 0.17	19.87 ± 0.16
ProxyNCA	65.69 ± 0.43	35.14 ± 0.26	24.21 ± 0.27	57.88 ± 0.30	30.16 ± 0.22	19.32 ± 0.21
Margin	63.60 ± 0.48	33.94 ± 0.27	23.09 ± 0.27	54.78 ± 0.30	28.86 ± 0.18	18.11 ± 0.17
Margin/class	64.37 ± 0.18	34.59 ± 0.16	23.71 ± 0.16	55.56 ± 0.16	29.32 ± 0.15	18.51 ± 0.13
N. Softmax	65.65 ± 0.30	35.99 ± 0.15	25.25 ± 0.13	58.75 ± 0.19	31.75 ± 0.12	20.96 ± 0.11
CosFace	67.32 ± 0.32	**37.49 ± 0.21**	**26.70 ± 0.23**	59.63 ± 0.36	31.99 ± 0.22	21.21 ± 0.22
ArcFace	67.50 ± 0.25	37.31 ± 0.21	26.45 ± 0.20	**60.17 ± 0.32**	**32.37 ± 0.17**	**21.49 ± 0.16**
FastAP	63.17 ± 0.34	34.20 ± 0.20	23.53 ± 0.20	55.58 ± 0.31	29.72 ± 0.16	19.09 ± 0.16
SNR	66.44 ± 0.56	36.56 ± 0.34	25.75 ± 0.36	58.06 ± 0.39	31.21 ± 0.28	20.43 ± 0.28
MS	65.04 ± 0.28	35.40 ± 0.12	24.70 ± 0.13	57.60 ± 0.24	30.84 ± 0.13	20.15 ± 0.14
MS+Miner	67.73 ± 0.18	37.37 ± 0.19	26.52 ± 0.18	59.41 ± 0.30	31.93 ± 0.15	21.01 ± 0.14
SoftTriple	67.27 ± 0.39	37.34 ± 0.19	26.51 ± 0.20	59.94 ± 0.33	32.12 ± 0.14	21.31 ± 0.14

Эксперименты

Table 5. Accuracy on Cars196

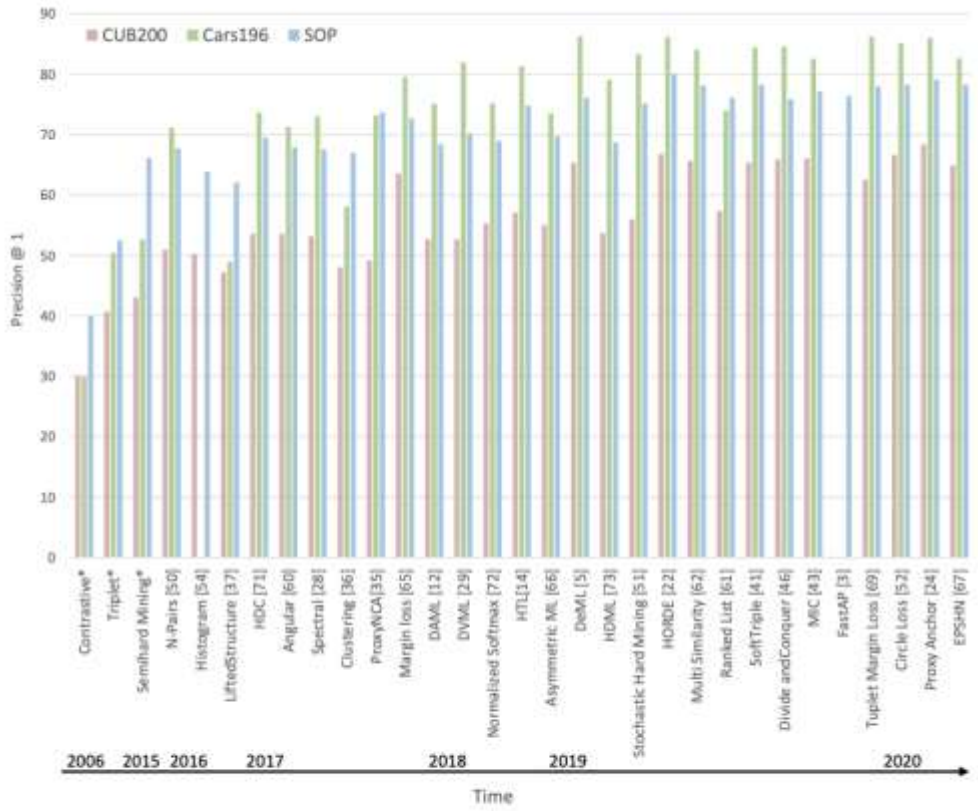
	Concatenated (512-dim)			Separated (128-dim)		
	P@1	**RP**	**MAP@R**	**P@1**	**RP**	**MAP@R**
Pretrained	46.89	13.77	5.91	43.27	13.37	5.64
Contrastive	81.78 ± 0.43	35.11 ± 0.45	24.89 ± 0.50	69.80 ± 0.38	27.78 ± 0.34	17.24 ± 0.35
Triplet	79.13 ± 0.42	33.71 ± 0.45	23.02 ± 0.51	65.68 ± 0.58	26.67 ± 0.36	15.82 ± 0.36
NT-Xent	80.99 ± 0.54	34.96 ± 0.38	24.40 ± 0.41	68.16 ± 0.36	27.66 ± 0.23	16.78 ± 0.24
ProxyNCA	83.56 ± 0.27	35.62 ± 0.28	25.38 ± 0.31	73.46 ± 0.23	28.90 ± 0.22	18.29 ± 0.22
Margin	81.16 ± 0.50	34.82 ± 0.31	24.21 ± 0.34	68.24 ± 0.35	27.25 ± 0.19	16.40 ± 0.20
Margin / class	80.04 ± 0.61	33.78 ± 0.51	23.11 ± 0.55	67.54 ± 0.60	26.68 ± 0.40	15.88 ± 0.39
N. Softmax	83.16 ± 0.25	36.20 ± 0.26	26.00 ± 0.30	72.55 ± 0.18	29.35 ± 0.20	18.73 ± 0.20
CosFace	$\mathbf{85.52 \pm 0.24}$	37.32 ± 0.28	27.57 ± 0.30	$\mathbf{74.67 \pm 0.20}$	29.01 ± 0.11	18.80 ± 0.12
ArcFace	85.44 ± 0.28	37.02 ± 0.29	27.22 ± 0.30	72.10 ± 0.37	27.29 ± 0.17	17.11 ± 0.18
FastAP	78.45 ± 0.52	33.61 ± 0.54	23.14 ± 0.56	65.08 ± 0.36	26.59 ± 0.36	15.94 ± 0.34
SNR	82.02 ± 0.48	35.22 ± 0.43	25.03 ± 0.48	69.69 ± 0.46	27.55 ± 0.25	17.13 ± 0.26
MS	85.14 ± 0.29	$\mathbf{38.09 \pm 0.19}$	$\mathbf{28.07 \pm 0.22}$	73.77 ± 0.19	$\mathbf{29.92 \pm 0.16}$	$\mathbf{19.32 \pm 0.18}$
MS+Miner	83.67 ± 0.34	37.08 ± 0.31	27.01 ± 0.35	71.80 ± 0.22	29.44 ± 0.21	18.86 ± 0.20
SoftTriple	84.49 ± 0.26	37.03 ± 0.21	27.08 ± 0.21	73.69 ± 0.21	29.29 ± 0.16	18.89 ± 0.16

Эксперименты

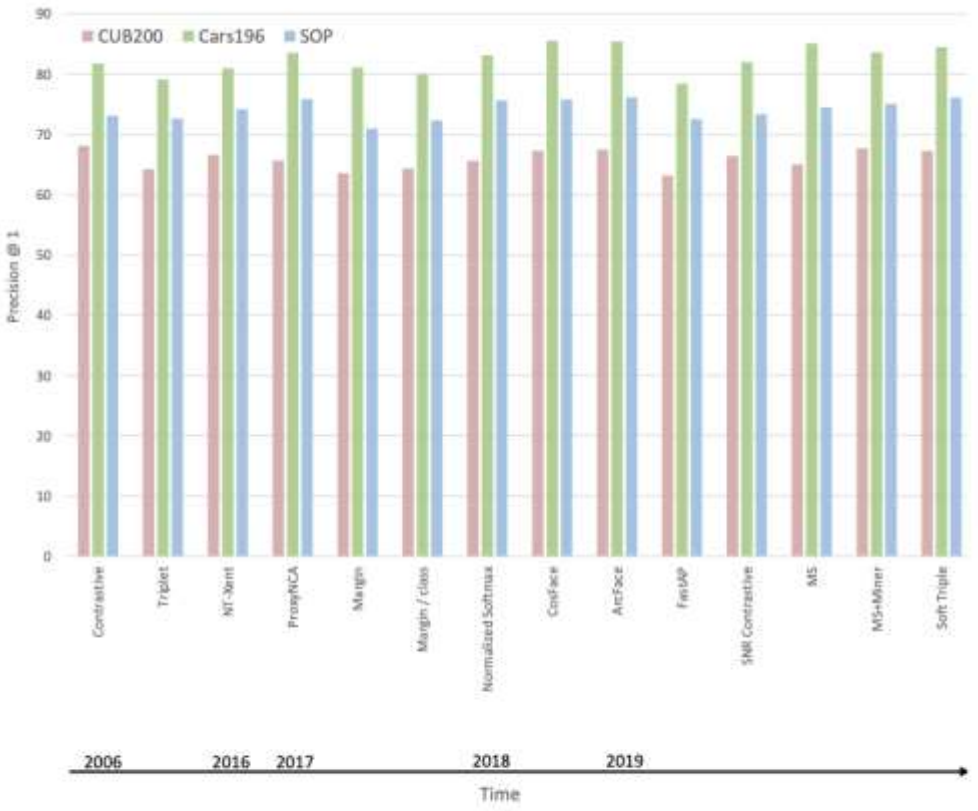
Table 6. Accuracy on SOP

	Concatenated (512-dim)			Separated (128-dim)		
	P@1	**RP**	**MAP@R**	**P@1**	**RP**	**MAP@R**
Pretrained	50.71	25.97	23.44	47.25	23.84	21.36
Contrastive	73.12 ± 0.20	47.29 ± 0.24	44.39 ± 0.24	69.34 ± 0.26	43.41 ± 0.28	40.37 ± 0.28
Triplet	72.65 ± 0.28	46.46 ± 0.38	43.37 ± 0.37	67.33 ± 0.34	40.94 ± 0.39	37.70 ± 0.38
NT-Xent	74.22 ± 0.22	48.35 ± 0.26	45.31 ± 0.25	69.88 ± 0.19	43.51 ± 0.21	40.31 ± 0.20
ProxyNCA	75.89 ± 0.17	50.10 ± 0.22	47.22 ± 0.21	71.30 ± 0.20	44.71 ± 0.21	41.74 ± 0.21
Margin	70.99 ± 0.36	44.94 ± 0.43	41.82 ± 0.43	65.78 ± 0.34	39.71 ± 0.40	36.47 ± 0.39
Margin / class	72.36 ± 0.30	46.41 ± 0.40	43.32 ± 0.41	67.56 ± 0.42	41.37 ± 0.48	38.15 ± 0.49
N. Softmax	75.67 ± 0.17	50.01 ± 0.22	47.13 ± 0.22	$\mathbf{71.65 \pm 0.14}$	$\mathbf{45.32 \pm 0.17}$	$\mathbf{42.35 \pm 0.16}$
CosFace	75.79 ± 0.14	49.77 ± 0.19	46.92 ± 0.19	70.71 ± 0.19	43.56 ± 0.21	40.69 ± 0.21
ArcFace	$\mathbf{76.20 \pm 0.27}$	$\mathbf{50.27 \pm 0.38}$	$\mathbf{47.41 \pm 0.40}$	70.88 ± 1.51	44.00 ± 1.26	41.11 ± 1.22
FastAP	72.59 ± 0.26	46.60 ± 0.29	43.57 ± 0.28	68.13 ± 0.25	42.06 ± 0.25	38.88 ± 0.25
SNR	73.40 ± 0.09	47.43 ± 0.13	44.54 ± 0.13	69.45 ± 0.10	43.34 ± 0.12	40.31 ± 0.12
MS	74.50 ± 0.24	48.77 ± 0.32	45.79 ± 0.32	70.43 ± 0.33	44.25 ± 0.38	41.15 ± 0.38
MS+Miner	75.09 ± 0.17	49.51 ± 0.20	46.55 ± 0.20	71.25 ± 0.15	45.19 ± 0.16	42.10 ± 0.16
SoftTriple	76.12 ± 0.17	50.21 ± 0.18	47.35 ± 0.19	70.88 ± 0.20	43.83 ± 0.20	40.92 ± 0.20

Paper vs Reality

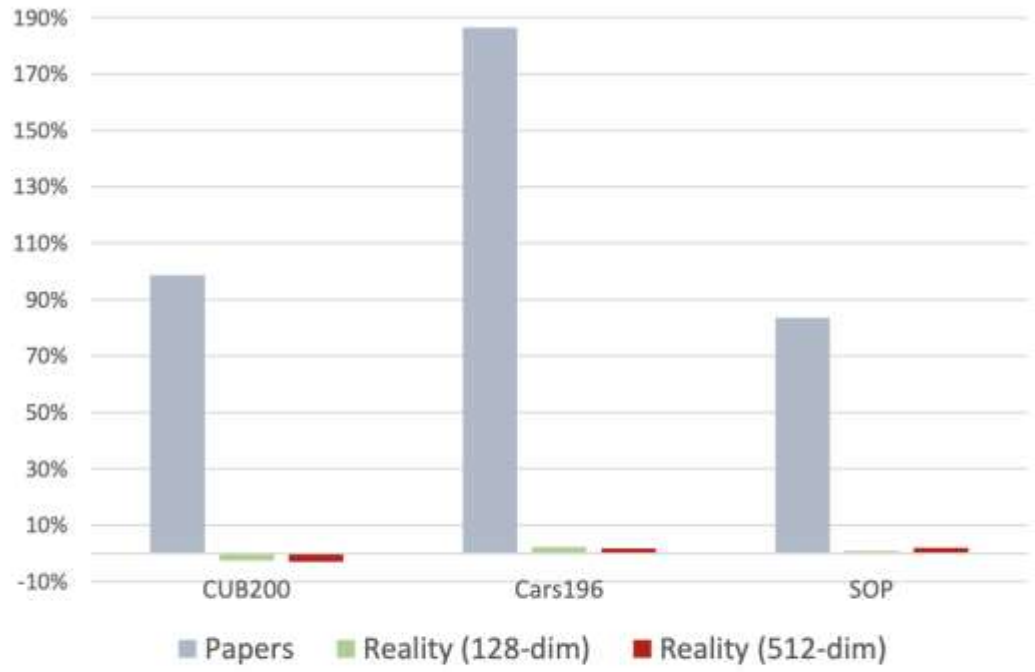


(a) The trend according to papers

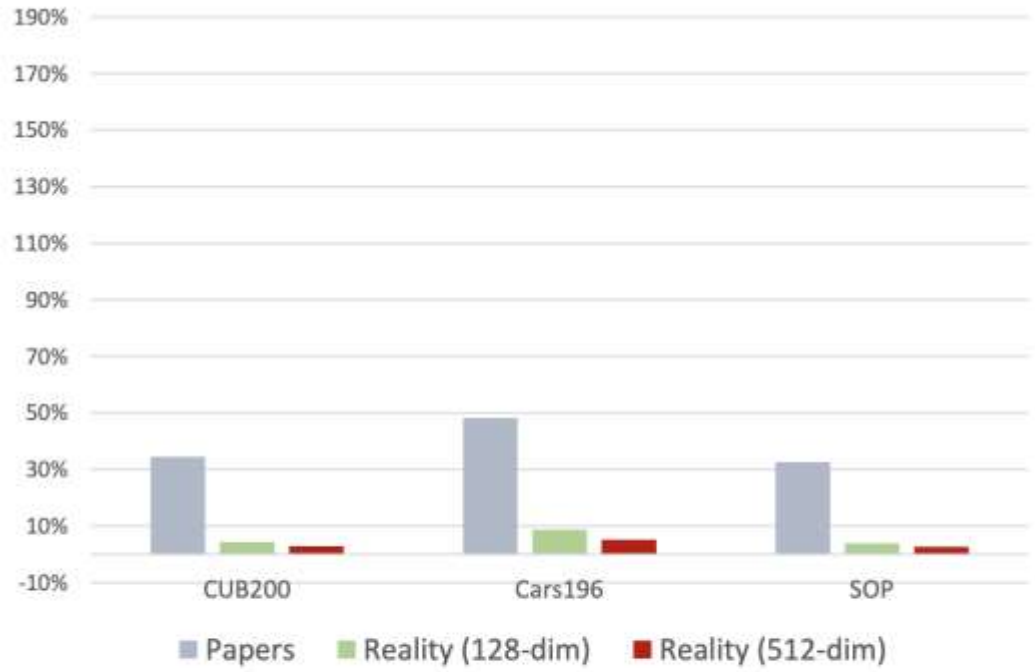


(b) The trend according to reality

Paper vs Reality



(a) Relative improvement over the contrastive loss



(b) Relative improvement over the triplet loss

Paper vs Reality

- Обнаружилось, что в статьях резко преувеличены улучшения по сравнению с contrastive и triplet лоссами. Это происходит из-за чрезвычайно низкой точности, связанной с этими потерями.
- При правильной реализации мы получим, что методы 2006 и 2019 работают +- одинаково. Другими словами, алгоритмы метрического обучения не добились впечатляющего прогресса, которого, как они утверждают, добились.

Заключение

- Обнаружили некоторые недостатки в современной литературе по Metric Learning:
- Несправедливые сравнения, вызванные изменениями в сетевой архитектуре, размерах встраиваемых файлов, способах увеличения изображений и оптимизаторах.
- Использование метрик, которые либо вводят в заблуждение, либо не дают полной картины пространства эмбеддингов.
- Обучение без валидационной выборки, то есть с обратной связью только с тестовым набором.

Заключение

- Затем провели эксперименты с исправлением этих проблем и обнаружили, что современные функции потерь работают немного лучше, а иногда и наравне с классическими методами. Это резко контрастирует с утверждениями, сделанными в статьях.
- В будущем можно будет изучить взаимосвязь между оптимальными гиперпараметрами и комбинациями наборов данных / архитектуры, а также причины, по которым разные потери работают одинаково.

Вопросы

- 1. В чем идея triplet loss и в чем он превосходит более простые версии, например, contrastive loss?
- 2. В чем проблема использования обычных метрик в задачах metric learning? Какую метрику предлагают авторы статьи и чем она лучше? (С формулой)
- 3. Как авторы предлагают подбирать гиперпараметры для модели?