Metric Learning. Face Recognition на LFW.

Я выбрал задачу face recognition. В ходе изучения материалов по metric learning я отметил пару моментов, которые, на мой взгляд, выделяют эту область от задач классификации. Это функции ошибок и техники семплирования объектов, именно с этими аспектами я и попытался поэкспериментировать в работе.

1 Contrastive Loss

Первыми техниками, которые я решил испытать, оказались сиамские сети и contrastive loss, так как они достаточно просты в реализации, и, как мне казалось, были неплохим началом для знакомства с новой для меня задачей. В качестве метрики было выбрано евклидово расстояние.

$$L(A,B,Y) = (Y) * ||f(A) - f(B)||^2 + (1-Y) * \{max(0, m^2 - ||f(A) - f(B)||^2)\}$$

Формула Contrastive Loss.

1.1 Siamese Network

Сначала я построил модель из свёрточных, линейных и пулинг слоёв. Модель обучалась 100 эпох и, как можно заметить по графику, довольно быстро вышла на плато с ошибкой в районе 1.

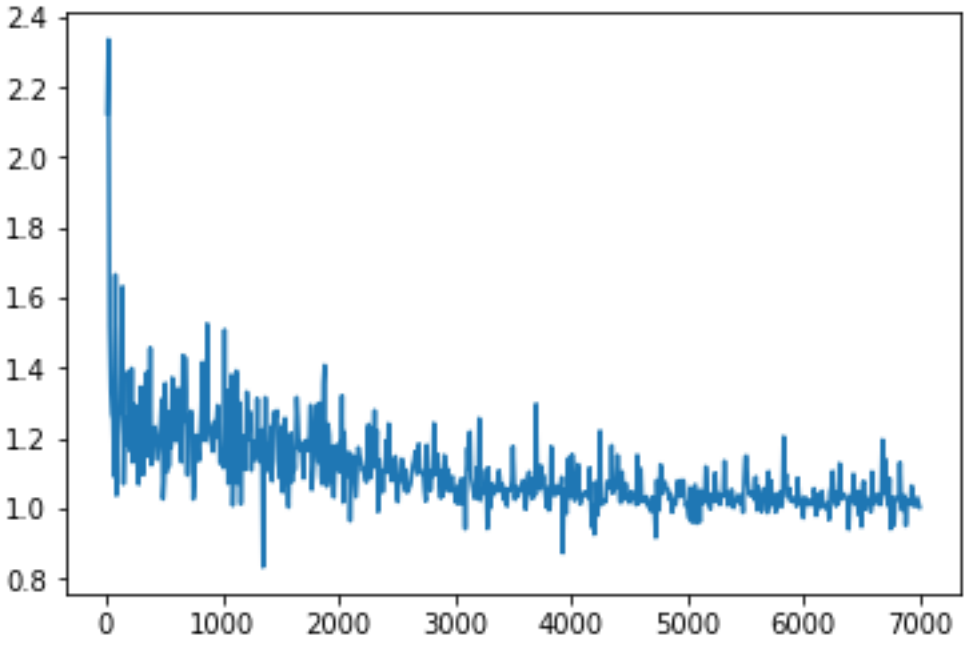

График 1

Во время валидации модель не справлялась с входными данными. Я предположил, что это связано с простотой архитектуры и перешёл к более серьёзной модели.

1.2 Resnet as backbone.

В качестве базовой архитектуры я выбрал Resnet-50, так как, я довольно часто встречал её в топах бенчмарков в других областях машинного обучения, кроме того, она использовалась в рассмотренных мной статьях. Я значительно увеличил размер выходного эмбеддинга, теперь он стал равен 128. Во всех последующих экспериментах будет также использоваться эта модель. Модель обучалась 100 эпох с оптимизатором Adam, а затем ещё 20 эпох с использованием scheduler'a. Ошибка стала меньше, перешла за порог 1, и лучшие значения ошибки доходили до 0.8.

Однако, в целом, судя по предсказаниям модели, результат получился неудовлетворительным.

2 Sampling matters

На тот момент мне было известно о проблеме семплирования тяжелых примеров в metric learning. Большое влияние этого аспекта на эффективное обучение модели хорошо продемонстрировано в статье Sampling Matters in Deep Embedding Learning.

Для проверки своей гипотезы, что для улучшения качества модели, можно воспользоваться более изощрёнными техниками майнинга, чем с помощью случайного набора, я воспользовался библиотекой pytorch-metric-learning.

Я испробовал две конфигурации:

- Contrastive Loss и Triplet Margin Miner (type of triplets = “semihard”)
- Margin Loss и Uniform Histogram Miner (как в статье Sampling Matters in Deep Embedding Learning)

В обоих случаях, на некоторых примерах модели исправно работали. Для оценки порогового значения близости, начиная с которого лицо на изображениях считать одинаковыми, я воспользовался F1 Score.

3 Arcface Loss

Насколько мне известно, Arcface Loss наравне с Cosface Loss на данный момент используются в state-of-the-art подходах, но их потенциал раскрывается с ростом числа классов, а для задач с меньшим числом классов лучше подходят контрастные функции ошибок, такие как Triplet Loss и Margin Loss.

Воспользовавшись формулой для нахождения гиперпараметров Arcface, представленной в статье AdaCos:

$\boldsymbol{s = \sqrt{2} * ln(C - 1)}$, где C – число классов в тренировочном датасете

В LFW 5749 классов в тренировочном наборе, значение параметра scale получилось равным примерно 12.24.

После тренировки модели я довольно поздно обнаружил факт переобучения и единственное что успел попробовать – оптимизатор AdamW с weight decay = 0.05. К

сожалению, это не помогло.

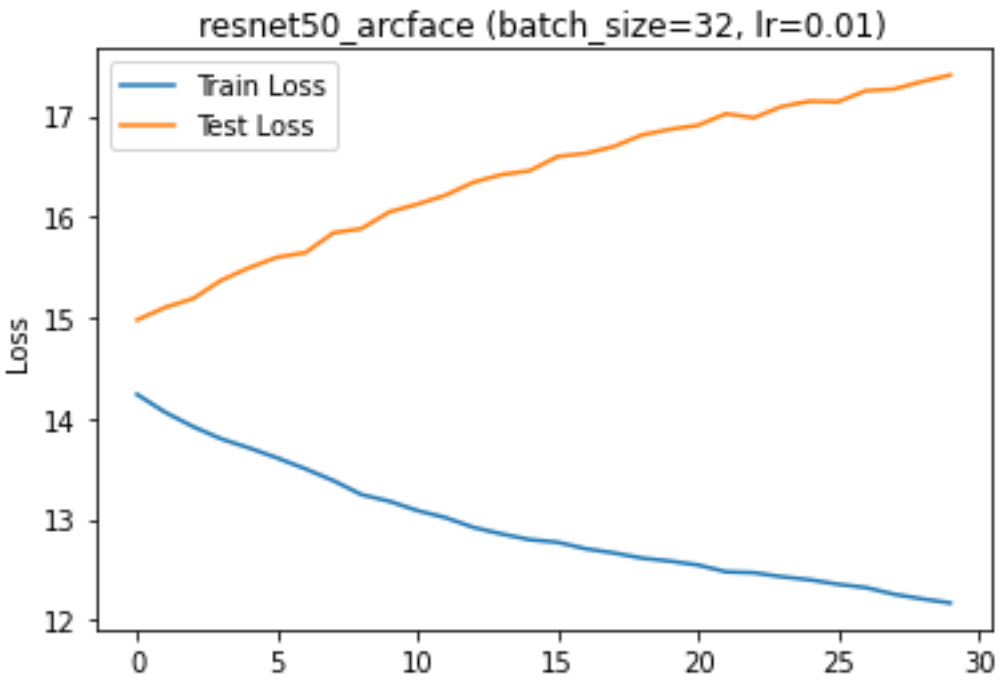

График 2

Расстояние между примерами которое выбирала модель, было близко к 1, как для фото одного и того же человека, так и для разных.

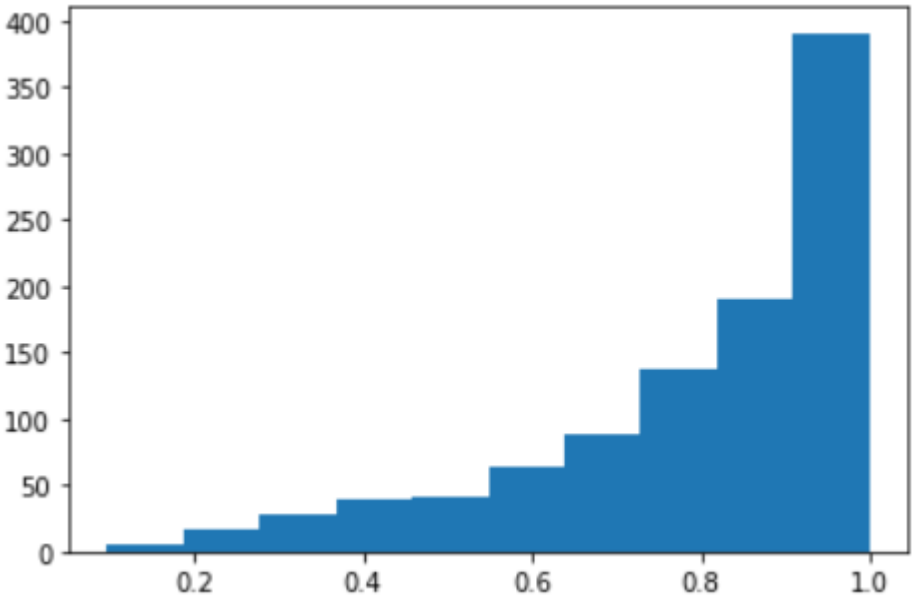

График 3

Возможные способы улучшения:

- Делать предобработку лиц. Во многих статьях, в том числе в статье Arcface, лица детектируются с помощью MTCNN и выравниваются для того, чтобы привести все данные к более общему виду.

- Обучать модель на датасетах большего размера. Размер LFW довольно мал по сравнению с другими наборами данных.

Datasets	#Identity	#Image/Video
CASIA [43]	10K	0.5M
VGGFace2 [6]	9.1K	3.3M
MS1MV2	85K	5.8M
MS1M-DeepGlint [2]	87K	3.9M
Asian-DeepGlint [2]	94 K	2.83M
LFW [13]	5,749	13,233
CFP-FP [30]	500	7,000
AgeDB-30 [22]	568	16,488
CPLFW [48]	5,749	11,652
CALFW [49]	5,749	12,174
YTF [40]	1,595	3,425
MegaFace [15]	530 (P)	1M (G)
IJB-B [39]	1,845	76.8K
IJB-C [21]	3,531	148.8K
Trillion-Pairs [2]	5,749 (P)	1.58M (G)
iQIYI-VID [20]	4,934	172,835

Table 1. Face datasets for training and testing. "(P)" and "(G)" refer to the probe and gallery set, respectively.

Большинство рассмотренных решений для тестирования используют уже предобученные модели. Например, в Sampling Matters in Deep Embedding Learning:

"For verification, we train our model on the largest publicly available face dataset, CASIA-WebFace, and evaluate on the standard LFW [16] dataset. … The CASIA-WebFace dataset contains 494,414 images of 10,575 people. The LFW dataset consists of 13,233 images of 5,749 people. "