

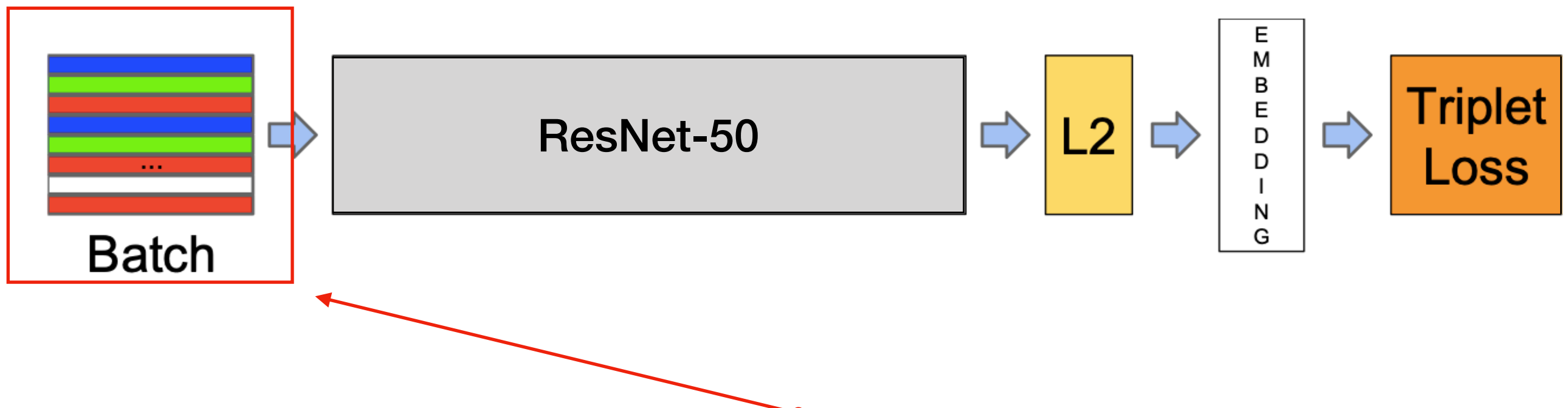
Компьютерное зрение

Практический курс
Савельева Юлия Олеговна
i.o.saveleva.kpfu@gmail.com
2-й семестр, 05.03.2022 г.



Image Retrieval

Составление батча



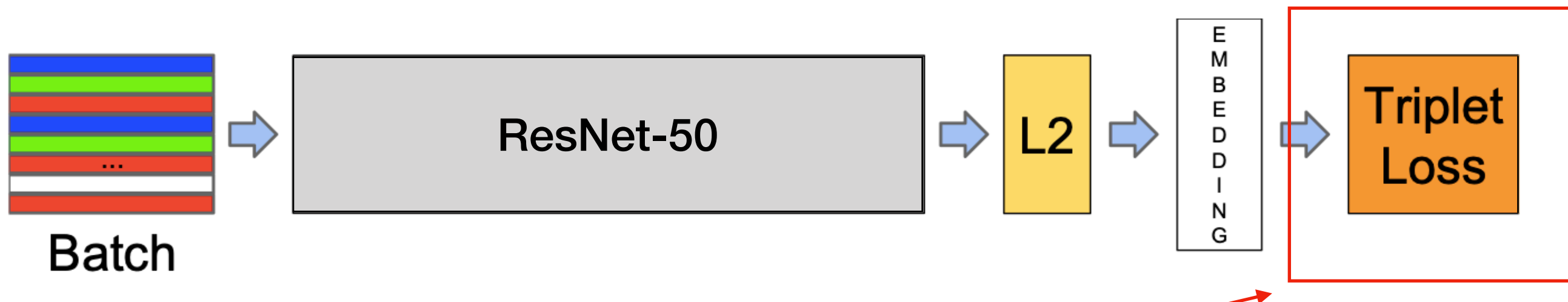
В батче $m \times n \times l$ должно быть n категорий по m товаров по l картинок (например $m=4$, $n=4$, $l=4$). Если l картинок набрать нельзя, этот товар не участвуют в обучении.

```
from torch.utils.data.sampler import BatchSampler
```

- на каждой итерации должен возвращать индексы элементов в датасете, предназначенные для текущего батча. Необходимо переопределить логику конструктора и метода `__iter__()` и `__len__()`. Объект класса можно подать вместе с `datasets.ImageFolder` в `torch.utils.data.DataLoader`.

Image Retrieval

Выбор триплетов

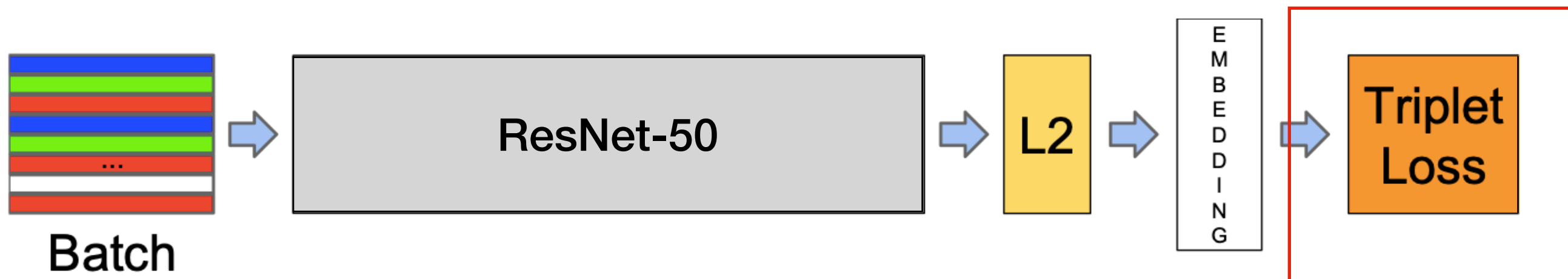


Для k -мерных векторов $\text{embeddings} = (x_1, x_2, \dots, x_n)$ необходимо векторными операциями рассчитать матрицу квадратов Евклидова расстояния всех со всеми:

$$A = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 & \dots & d_{1n}^2 \\ d_{21}^2 & 0 & d_{23}^2 & \dots & d_{2n}^2 \\ d_{31}^2 & d_{32}^2 & 0 & \dots & d_{3n}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \dots & 0 \end{bmatrix} \quad d_{ij}^2 = \|x_i - x_j\|^2$$

Image Retrieval

Выбор триплетов



Поиск триплетов:

for label in unique_batch_labels:

same_indices = where(labels == label)

negative_indices = where(labels != label)

anchors, positives = all_combinations(same_indices)

ap_dists = A[anchors, positives]

for a, p, ap_dist in (anchors, positives, ap_dists):

loss = ap_dist - A[a, negative_indices] + α

semi_hard = random_choice($0 < \text{loss} < \alpha$)

if semi_hard is not None:

triplets.append((a, p, semi_hard))

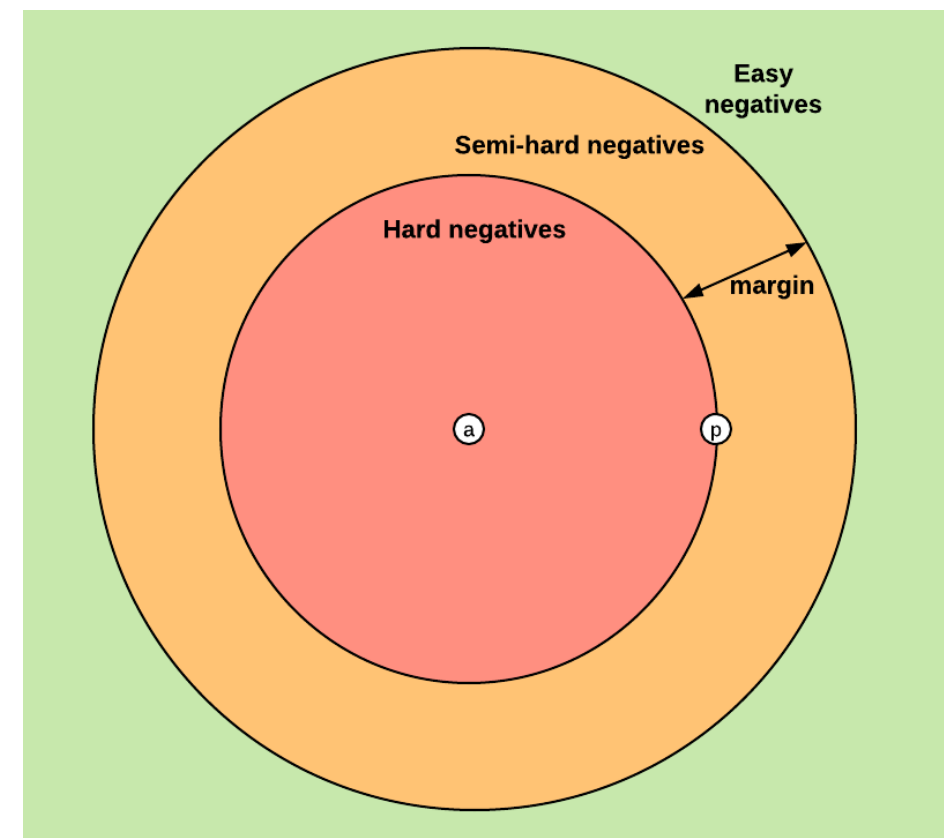
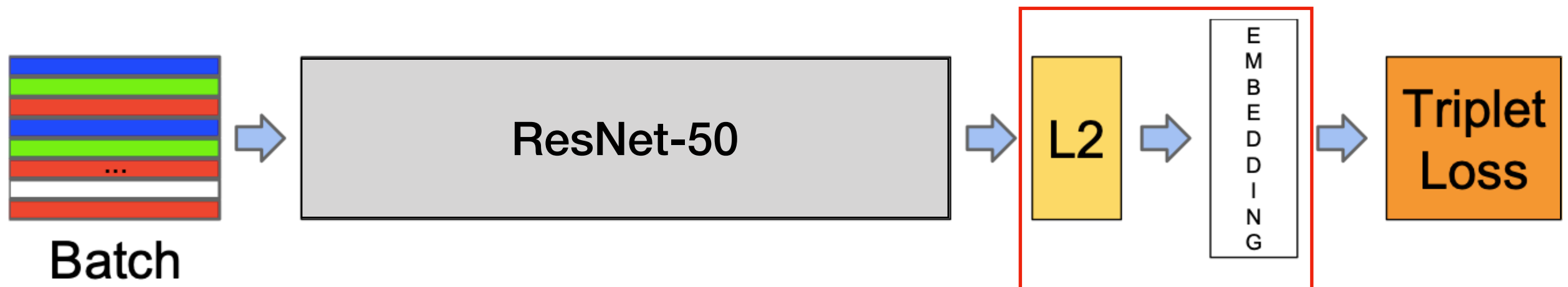


Image Retrieval

Нормализация и целевая функция



Так как целевая функция имеет следующий вид:

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+, \text{ где } [\dots]_+ - \max(\dots, 0)$$

α - гиперпараметр, ($\alpha > 0$, например, в оригинальной статье 0.2).

Image Retrieval

Точность

- Как работает HNSW (Hierarchical Navigable Small Worlds):
<https://www.pinecone.io/learn/hnsw/>
- Точность считать на `valid_dataset.pickle` и `test_dataset.pickle`, где лежат словари, имеющие следующую структуру и количество элементов:

```
>>> len(content['query'])
4566
>>> len(content['retrieval'])
19479
```
- HNSM в Python:
Предполагаем есть уже посчитанные `query_embeddings` и `retrieval_embeddings`.

```
index = nmslib.init(method='hnsw', space='l2')
index.addDataPointBatch(retrieval_embeddings)
index.createIndex(print_progress=True)
matches = np.zeros((len(query_labels)))
for i in range(len(query_labels)):
    query_label = query_labels[i]
    ids, distances = index.knnQuery(query_embeddings[i], k=5)
    local_retrieval_labels = retrieval_labels[ids]
    matches[i] = any(query_label == local_retrieval_labels[:5])
```

Image Retrieval

Дополнительно

У предобученной на задачу классификации модели нужно обучать на задачу Image Retrieval все слои нейронной сети (не только fully-connected).

Во время дообучения необходимо перевести Batch Normalization в режим test и не обучать его.

Качество embeddings необходимо оценить визуально с помощью T-SNE.

Подробнее о методе можно узнать в статье FaceNet: A Unified Embedding for Face Recognition and Clustering.

<https://arxiv.org/pdf/1503.03832.pdf>

Дедлайн 22.03.2022 23:59