No Fuss Distance Metric Learning using Proxies

Radek Bartyzal

GLAMI AI

14. 2. 2020

Goal

We have:

- dataset of products with multiple images per product

We want:

- create embeddings of each product image
- embeddings of the same product images are closer to each other than to the embedding of images of the other products

Distance Metric Learning

- learning a distance consistent with a notion of semantic similarity
- an anchor point x is similar to a set of positive points Y, and dissimilar to a set of negative points Z
- a loss defined over these distances is minimized

Classic approaches

Triplet Ranking Loss

- sample 1 anchor, 1 positive, 1 negative point = a triplet
- optimize the 3 embeddings to:
  - enlarge the distance between the anchor and negative point
  - shrink the distance between the anchor and positive point

Contrastive Loss (Pairwise Ranking Loss)

- sample 1 anchor and 1 positive OR 1 negative point = a pair
- optimize the 2 embeddings with the same goal as the triplet

Classic approaches

Triplet Ranking Loss:

$$L(x, y, z) = max(0, m + d(x, y) - d(x, z))$$

[4]

Contrastive Loss (Pairwise Ranking Loss):

$$L(x, y) = \begin{cases} d(x, y) & if \quad PositivePair \\ max(0, m - d(x, y)) & if \quad NegativePair \end{cases}$$

[4]
m = margin used to ignore triplets that have good enough embeddings

Problems

Problem:

- There are too many triplets to go through.

Solution:

- we need to select **informative triplets** that will guide the optimization
- informative triplets = not too easy, not too hard => **Semi-Hard negative mining** = select the right triplets from the mini-batch [5]
- downside: requires large mini-batches (1800 images)
- other approaches incorporating information outside the single triplet improve the convergence at the cost additional computation

Approach with proxy embeddings

- create a proxy embedding for each product = class
- optimize distance to the proxy embeddings = proxies
- optimize the proxies end-to-end with all the embeddings [1]

Approach with proxy embeddings

Algorithm 1 Proxy-NCA Training.

Randomly init all values in θ including proxy vectors.
for $i = 1 \dots T$ **do**
  Sample triplet (x, y, Z) from D
  Formulate proxy triplet $(x, p(y), p(Z))$
  $l = -\log\left(\frac{\exp(-d(x,p(y)))}{\sum_{p(z)\in p(Z)} \exp(-d(x,p(z)))}\right)$
  $\theta \leftarrow \theta - \lambda \partial_\theta l$
end for

Figure: Still have to sample the triplets, which is what we want to avoid.

One proxy embedding per class

Algorithm 1 Proxy-NCA Training.

Randomly init all values in θ including proxy vectors.
for $i = 1 \ldots T$ **do**
  Sample x from D
  Formulate proxy triplet $(x$, p(C(x)), p(other classes) Z
  $l = -\log\left(\frac{\exp(-d(x,p(y)))}{\sum_{p(z)\in p(Z)} \exp(-d(x,p(z)))}\right)$
  $\theta \leftarrow \theta - \lambda \partial_\theta l$
end for

Figure: Optimization: sample the negative classes. [2]

Sources

1. Movshovitz-Attias, Yair, et al. "No fuss distance metric learning using proxies." Proceedings of the IEEE International Conference on Computer Vision. 2017. https://arxiv.org/abs/1703.07464

2. Zhai, Andrew, and Hao-Yu Wu. "Classification is a Strong Baseline for Deep Metric Learning." arXiv preprint arXiv:1811.12649 (2018). https://arxiv.org/abs/1811.12649

3. Zhai, Andrew, et al. "Learning a Unified Embedding for Visual Search at Pinterest." Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019. https://arxiv.org/abs/1908.01707

4. Blog: Understanding Ranking Loss, Contrastive Loss, Margin Loss, Triplet Loss, Hinge Loss and all those confusing names https://gombru.github.io/2019/04/03/ranking_loss/

Sources

5. Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Schroff_FaceNet_A_Unified_2015_CVPR_paper.pdf