Sampling matters

1. Introduction

1.1 About Task

Main Methods : transform images into rich, semantic representations with deep learning

- zero-shot learning
- visual search
- face recognition
- fine-grained retrieval

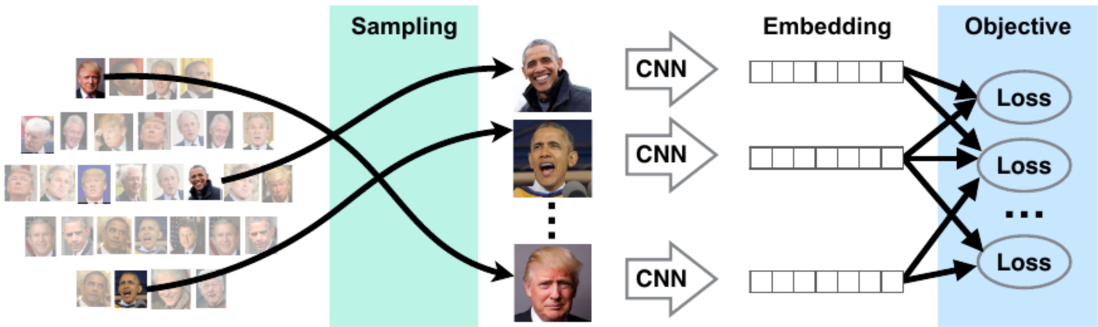


Figure 1: An overview of deep embedding learning: The first stage samples images and forms a batch. A deep network then transforms the images into embeddings. Finally, a loss function measures the quality of our embedding. Note that both the sampling and the loss function influence the overall training objective.

1.2 About Method - learn embedding

- **Simple insight :**

  pull similar images closer in embedding space and push dissimilar images apart.

- **Aplication in methods - loss function**
  - contrastive loss : positive->1, negative->0
  - pairwise losses : describe below
  - triplet loss : Not only **loss function** are changed, but also changes the way positive and negative example are selected **(sampling)**

- **From here**

  we know that there are two key point:

  - the loss
  - the sampling strategy

1.3 Conclusion

- sample selection $\geq$ loss.

- About **sample selection**:
    - **analyze** existing **sampling strategies**, and **show why they work and why not**
    - a new sampling strategy
        - propose
        - analyse :
            - **corrects the bias** induced by the geometry of embedding space
        - effect :
            - a lower variance of gradients $\rightarrow$ stabilizes training
- About **Loss functions**
    - Also matters
    - a new simple margin-based loss
        - It relaxes the loss, making it more robust
        - isotonic regression

2. Related Work

2.1 About loss function

1) show some loss function in recent research:

- triplet losses
    - [introduction]()
    - more constraint : PDDM , Histogram Loss
    - more examples : n-pair loss , Lifted Structure

      > defines constraints on all images in a batch
- other loss func:
    - Structural Clustering : optimizes for **clustering quality**
    - PDDM : proposes a new module to model **local feature structure**.
    - HDC : trains an ensemble to model examples of **different “hard levels”**

2) But

we show that **a simple pairwise loss** is **sufficient** if paired with **the right sampling strategy**.

2.2 About example selection (sampling)

- **common methods:**

select at all posible pairs at random

- **hard negative mining:**

  introduction

  - Creat a batch of negative samples
  - Train model on it
  - Use fasle positive (negative samples detected as postive samples) as negative sample
  - Creat new negative samples

- **semi-hard negative mining:**

  described in chapter 3

3. Preliminaries

3.1 Notations

- Data point : $x_i \in \mathbb{R}^N$
- Deep network : $f : \mathbb{R}^N \to \mathbb{R}^D$
- the distance between two datapoints : $D_{ij} := ||f(x_i) - f(x_j)||$
- Euclidean norm : $||\ ||$
- Positive/negative value : $y_{ij} = 1/0$

3.2 loss

1) contrastive loss

$$\ell^{\text{contrast}}(i, j) := y_{ij} D_{ij}^2 + (1 - y_{ij}) \left[\alpha - D_{ij}\right]_+^2$$

2) triplet loss

$$\ell^{\text{triplet}}(a, p, n) := \left[D_{ap}^2 - D_{an}^2 + \alpha\right]_+ .$$

3.3 Compution effiency

1) Risk minimization

Suppose have a n examples dataset :

- For constractive loss : $O(n^2)$ pairs
- For triplet loss : $O(n^3)$ pairs

Thus

- This is computationally infeasible

3.4 Convergences

Accelerate Convergences with sampling methods

- once the **network convergences**
    - most samples contribute in a minor way
    - very few of the negative margins are violated
- **lots of heuristics methods to accelerate convergence**
    - For the contrastive loss : hard negative mining
    - For the triplet loss : semi-hard negative mining

> - hard negative mining in triplet loss also lead to a collaspe model:**all images have the same embedding.**
> - About **semi-hard negative mining**:
>
> $$n_{ap}^{\star} := \underset{n:D(a,n)>D(a,p)}{\operatorname{argmin}} D_{an},$$
>
> n_{ap}^{*} : obtain a negative instance n within a batch

4. Distance Weighted Margin-Based Loss

4.1 About common sampling

1) Sampling strategy

sampling negative uniformly

2) Distance distribution

Take distance between a pair point as a random varibale:

the distribution of distance is :

$$q(d) \propto d^{n-2}\left[1-\frac{1}{4}d^2\right]^{\frac{n-3}{2}} .$$

> n : dimensions

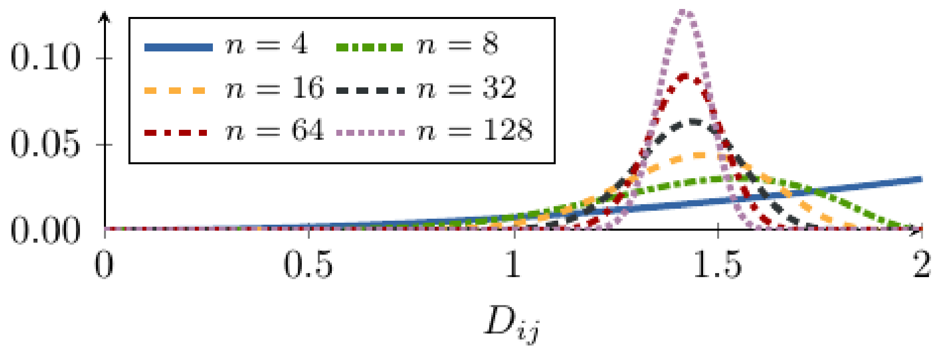


Figure 2: Density of datapoints on the D-dimensional unit sphere. Note the concentration of measure as the dimensionality increases — most points are almost equidistant.

4.2 About hard negative mining

1) differentiation of loss

Think about a triplet loss function on a triplet $t := (a, p, n)$

$$\ell^{\text{triplet}}(a, p, n) := \left[D_{ap}^2 - D_{an}^2 + \alpha\right]_+ .$$

The gradient with respect to the negative example $f(x_n)$ is in the form of:

$$\partial_{f(x_n)} \ell^{(\cdot)} = \frac{h_{an}}{\|h_{an}\|} w(t)$$

2) Noise:

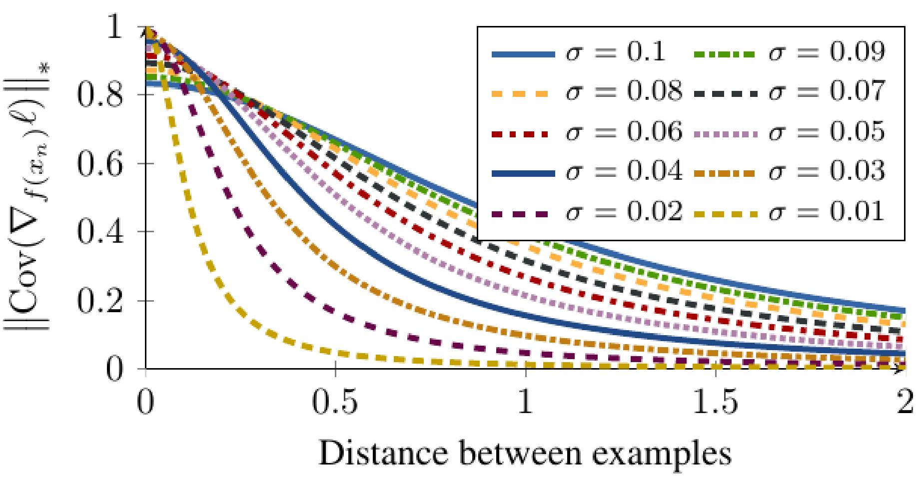
Z is noise in model , for example **dropout** , **L2** and **Data Augmentation**

$$\frac{h_{an}+z}{\|h_{an}+z\|}$$

if h_{an} is small, direction will be dominated by noise.

3) Norm of covariance matrix

- experiment result:



(a) Variance of gradient at different noise levels.

- Meaning of covariance matrix in optimization:

  High variance means the gradient is close to random, while low variance implies a deterministic gradient estimate.

> Why?

4.3 Distance weighted sampling

1) New sampling methods

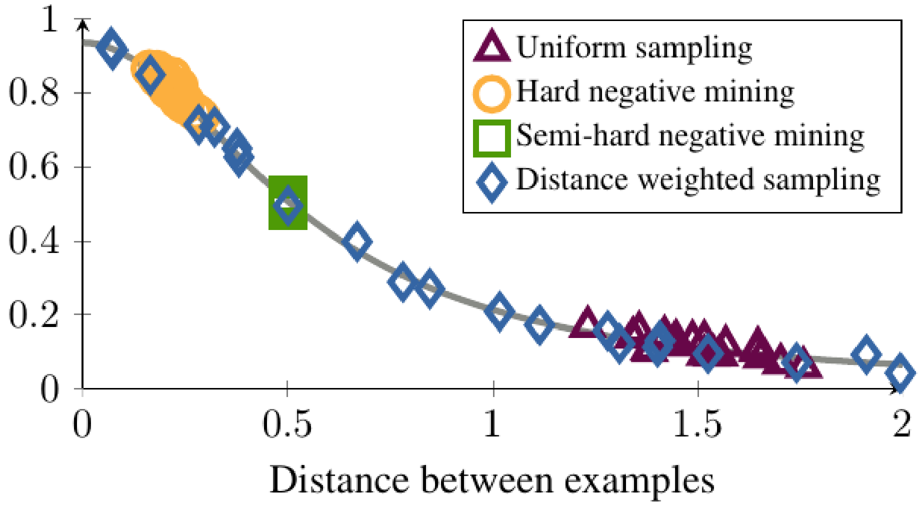
sample uniformly according to distance, sampling with weights $q(d)^{-1}$

$$\Pr\left(n^{\star}=n|a\right) \propto \min\left(\lambda, q^{-1}\left(D_{an}\right)\right).$$

2) Comparision to other sampling methods

2.1) The comparison standard

simulated examples drawn from different strategies along with their variance of gradients.



(b) Sample distribution for different strategies.

2.2) Analyse of graph

- Uniform sampling :
  - Property:

    Because of norm distribution of distance, most sampling distances are concentrate in 1.2-1.7, describe as below :

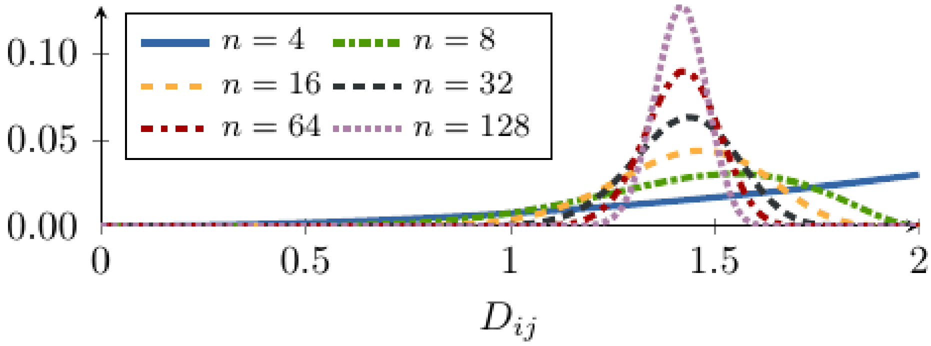


Figure 2: Density of datapoints on the D-dimensional unit sphere. Note the concentration of measure as the dimensionality increases — most points are almost equidistant.

- Result:

  Random sampling yields only easy examples that **induce no loss**

- Hard negative mining:
  - Property

    always use false positive (samples which suppose to have far distance but actually close).

  - Result:
    - This leads to **noisy gradients** that **cannot effectively push two examples** apart.
    - Lead to **a collapsed model**
- Semi-hard negative mining :
  - Property

    select the minimization distance in a mini-batch as negative sampling : **always have same distances between examples**

  - Result:

    It might **converge quickly at the beginning**, at some point no examples are **left within the band**

- Distance weigthed sampling
  - Property:

    We can't induce too high variance or too low variance or too steady variance. So balance is the best.

  - Result:

    steadily produce informative examples while controlling the variance

######

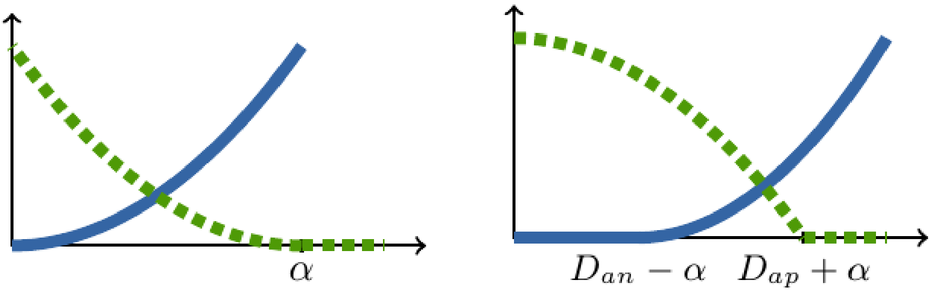
4.4 Margin based loss

1) New loss function

$$\ell^{\text{margin}}(i,j) := (\alpha + y_{ij}(D_{ij} - \beta))_+ \,.$$

2) Comparision to other loss functions

2.1) Why triplet loss better than constrastive loss



(a) Contrastive loss [11] (b) Triplet loss ℓ_2^2 [25]

> The solid blue : loss value for positive pairs
>
> the dotted green : loss value for negative pairs.

- The **triplet loss** does **not assume a predefined threshold** to separate similar and dissimilar images
- the triplet loss only **requires positive examples to be closer than negative examples**, while the contrastive loss **spends efforts on gathering all positive examples as close together as possible.**

  (flat part in blue line)

2.2) Why hard negative mining is not fitting with triplet loss?

- Concave shape in negative loss.

  Because **hard negative mining** always have **low loss for negative samples**.

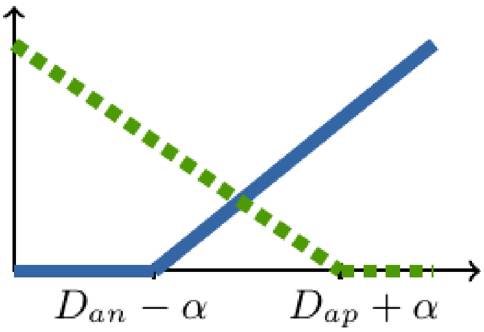
  the gradient with respective to negative example is **approaching zero**

2.3) Change squared norm to norm make it better for triplet loss.

A improvement of triplet loss:

$$\ell^{\text{triplet},\ell_2} := (D_{ap} - D_{an} + \alpha)_+ \,.$$

Why it works? **Turn concave to line**



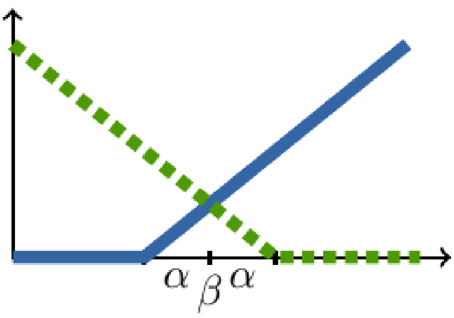
(c) Triplet loss ℓ_2

2.4) Margin based loss

- **Advantage**
  - Compared to **contrastive loss** : Enjoys the **flexibility** of the triplet loss.

    : have flat parts
  - Compared to **triplet loss** : Enjoys the **computational efficiency**

    : only $O(n^2)$



(d) Margin based loss

- How to determine value of β ?

  To enjoy the flexibility as a triplet loss, we need a more flexible boundary parameter β .

$$\beta(i) := \beta^{(0)} + \beta^{(\text{class})}_{c(i)} + \beta^{(\text{img})}_{i}$$