Lecture Notes On Triple Loss Function (Part III)

Harry Li [‡], Ph.D.
Computer Engineering Department
College of Engineering, San Jose State University
San Jose, CA 95192, USA

Research and Development Center for Artificial Intelligence and Automation [‡]
CTI Plus Corporation, 3679 Enochs Street
Santa Clara, CA 95051, USA

Email[†]: hua.li@sjsu.edu

***Abstract*—This note is part III from my lecture on convolutional neural networks. In particular, the tripple loss functions in FaceNet for facial recognitions.**

I. Objective Function in Facical Recognition

Now, we will discuss the objective function in FaceNet. First, let's define an image x as an input layer, and the FaceNet CNN output for the image feature extraction as function f. Obviously, the above objecitve function is also a loss function.

Denote all output neurons errors from i equal to 1, 2, ..., n_o as

$$D = \sum_{\mu=1}^{M} ||(\zeta^{\mu} - f^{\mu}(h(l)))||_2^2 \quad (1)$$

$$= \sum_{\mu=1}^{M} ||(\zeta^{\mu} - f^{\mu}(l))||_2^2 \quad (2)$$

where ζ_i^{μ} for i = 1,2, ..., n_o, for the errors at each individual dimenson, is replaced by the summation of all the errors at the experiment μ, e.g., we denote

$$\zeta^{\mu} = \sum_{i=1}^{n_o} ||(\zeta_i{}^{\mu} - f^{\mu}(h(l))||_2^2 \quad (3)$$

In order to make comparison with Google team's mathematical notation, let input image layer l

$$l_i(u,v) = x_i \quad (4)$$

for images of i=1,2, ..., n. And $x_i \in R^3$, the outpout of CNN of the faceNet defines feature vectors, e.g., embeddgings of the input image. The outpout is denoted as

$$f : R^3 \rightarrow R^N \quad (5)$$

for N=68 facial land marks, see google facenet reference and Sandburg github repo (detector_dlib.py). Using this notation to rewrite the objective function above, after some mathematic manipulations as in Note II, let $\zeta(x_i) = f_{true}(x_i)$, we have

$$D = \sum_{i=1}^{M} ||(f_{true}(x_i) - f(x_i))||_2^2 \quad (6)$$

Lemma 1. ***Classes from the image data set.*** Suppose there are P classes of faces (persons), where each class of the images counts for $M_1, M_2, ..., M_i, ...M_P$, so the total number of images involved in the training for P classes is

$$M = \sum_{i=1}^{P} M_i \quad (7)$$

The recognition of a person from class i can be formulated as a training process of class i against the rest of all other classes. We call these rest of all other classes as the negative class and it is

$$M_n = \sum_{j=1, j\neq i}^{P} M_j \quad (8)$$

then, we have

Definition 1. ***P-classes objective function for training.*** P classes objective function takes the form of

$$\begin{aligned} D \quad &= \alpha_1 \sum_{j=1}^{M_1} ||(f_{true}(x_j) - f(x_j))||_2^2 + \\ &\alpha_2 \sum_{j=M_1+1}^{M_2} ||(f_{true}(x_j) - f(x_j))||_2^2 + ... \\ &... + \alpha_p \sum_{j=M_{P-1}+1}^{M_P} ||(f_{true}(x_j) - f(x_j))||_2^2. \end{aligned} \quad (10)$$

where

$$\sum_{i=1}^{p} \alpha_i = \alpha_1 + \alpha_2 + ... + \alpha_p = 1. \quad (11)$$

If the recognition tasks for the above P classes treat the correct detection rate for each class is equally likely, then we have

$$\alpha_1 = \alpha_2 = ... = \alpha_p = 1/p. \quad (12)$$

which is just a scaling factor and can be removed from the equation. However, not all loss functions carries the same significance, nor are the corrected dection rate are euqally likely demanded. For higher significant class(es), whose loss functions can be treated with bigger cofficient α (weighting factors). For example if we would like to more recognition accuracy for class i and class j than the reset of the classes, we can set

$$\alpha_i, \alpha_j > \alpha_k \tag{13}$$

for $k = 1, 2, ..., P$ and $k \neq i$ and $k \neq j$.

Lemma 2. *P classes recognition problem can be decomposed as two classes detection at a time, by grouping P-1 classes into one combined negative class.* This Lemma basically saying, one can re-group P classes by the following

$$\{P_i\}, \{P_1, P_2, ..., P_{i-1}, P_{i+1}, ..., P_p\} \tag{14}$$

where the negative group consists of P-1 groups.

For training purpose of P classes, let's assume we break the training tasks as $\{P_i\}$ group, e.g., positive group vs. $\{P_1, P_2, ..., P_{i-1}, P_{i+1}, ..., P_p\}$ group, e.g., negative group. Adopting the notation commoly used, denote the positive and negative group with super script as $\{P^p\}$ and $\{P^n\}$, then in the frame work of the FaceNet, each of these groups is defined by the embeddings as the output of the CNN output (see FaceNet paper), so we have

$$\{f(x_i^p)\}, \{f(x_j^n)\} \tag{15}$$

where $i \in \{1, 2, ..., i-1, i+1, ...M_i\}$ and j for all images in the rest of all classes.

We now have reached to the following triple loss functions which is given in the FaceNet paper.

Property 1. *Triple loss functions for training based on the equally likely assumptions.* For P classes recognition, if the detection task is to recognize an image x_i, based on Lemma 2, then its loss function can be defind as follows with the triplet

$$D = \sum_{i=1}^{M} [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha] \tag{16}$$

Proof From Definition 1, let P=2, so we have

$$\begin{aligned} D = \alpha_1 \sum_{j=1}^{M_1} ||(f_{true}(x_j) - f(x_j))||_2^2 + \\ \alpha_2 \sum_{j=M_1+1}^{M_2} ||(f_{true}(x_j) - f(x_j))||_2^2 \end{aligned} \tag{17}$$

where

$$\alpha_1 = \alpha_2 = 1/2. \tag{18}$$

the above equation holds good for equally likely case, with the following fine tuning: (1) the plus sign of the second term become negative to minimize the distance of the same positive class images, and to maximize the distance of negative classes images, in the sense of norm L2 per FaceNet paper contribution; (2) add α in the objective function to even further minimizing the distance of the positive class.

Therefore, we have

$$\begin{aligned} D &= \sum_{j=1}^{M_1} ||(f_{true}(x_j) - f(x_j))||_2^2 + \alpha \\ &- \sum_{j=M_1+1}^{M_2} ||(f_{true}(x_j) - f(x_j))||_2^2 \end{aligned} \tag{19}$$

In the data set, make $M_1 = M_2$, and let $M = M_1$, and let $f_{true}(x_j)$ be denoted as in the FaceNet common notation, $f(x_j^a)$, where the superscript a stands for anchor image, we have

$$\begin{aligned} D &= \sum_{j=1}^{M} [||(f(x_j^a) - f(x_j^p))||_2^2 + \alpha \\ & \quad -||(f(x_j^a) - f(x_j^n))||_2^2] \end{aligned} \tag{20}$$

QED.

II. Coding Aspects

The FaceNet github by Sandburg provides a base line implementation of the code. However, the code is using pre-trained model. So, you will have to find the module for training. To be explained in the following.

In addition, anchor images are selected from the positive class, but with "higher" and "unique" quality and characteristic requirements. (To be further addressed in the separate notes)

III. Future Directions

(1) For the negative classes, further divide the them into multiple classes, so triplet loss function can be modified to include 4th, 5th, or even higher split of the loss functions for the purpose of better, faster training, and higher accuracy.

(2) Look into training vs. deployment while using statistical based approach to sign α weighting coefficients;

(3) Develop statistical ranking approach for the recognition tasks during the deployment.

Acknowledgment

CTI One team and San Jose State University graduate, undergraduate students have assisted me in many aspects of stimulating work and conversations.

References

[1] *B.K.P. Horn, Robot Vision.* MIT Press, 1982.