Triplet Mining for Facial Recognition Convolutional Neural Networks

Team: Surabhi Desai, Islam Faisal, Andrew Nguyen, Prem Talwai

Industry Mentors: Cambron Carter and Divyaa Ravichandran

Academic Mentor: Shantanu Joshi

gumgum

ipam institute for pure & applied mathematics

Research in Industrial Projects for Students (RIPS)

- The Program provides an opportunity for undergraduate students to work in teams with an academic mentor on real-world research projects proposed by sponsors from industry or the public sector.
- Hosted and co-funded by the Institute for Pure and Applied Mathematics (NSF Funded Math Institute on UCLA campus)

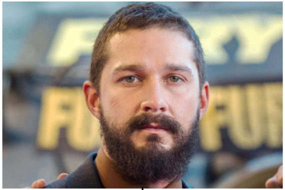
ipam
institute for pure & applied mathematics

GumGum

- Applied computer vision company and pioneers in in-image and in-screen advertising

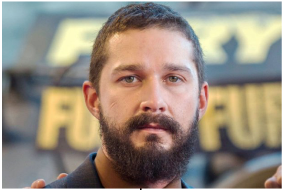
Example of in-image advertising

Facial Verification and Recognition



Recognition Engine (1 to N)

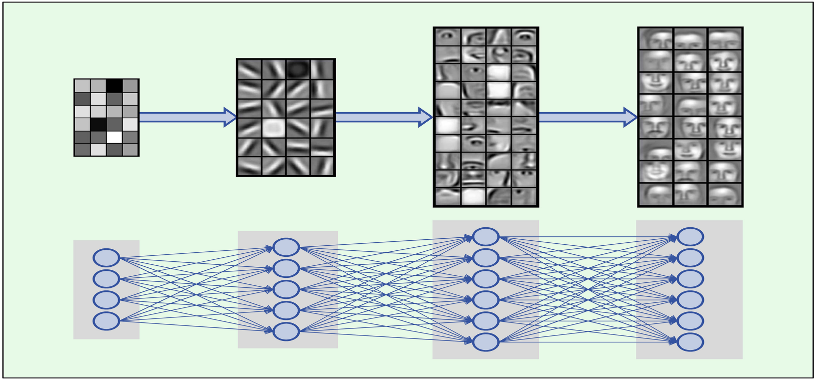
Shia LeBeouf: 84%
George Donner: 10%
Hannibal Lecter: 5%
...
Shantanu Joshi: 0.002%



Verification Engine (1 to 1)

Same identity: 90%

Neural Networks for Visual Tasks

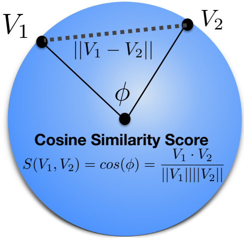


Deep Neural Networks Learning Hierarchical Representations
([Hochreiter-16] adopted from [Lee-2011])

Given the feature vectors of two faces, how can we tell if they represent the same person?

Similarity Between Feature Vectors

V_1



V_2

Target Image	Test Image	Similarity Score
		**0.73**
		**0.24**

Contrastive Loss (Hadsell-Chopra-LeCun-06)

- Consider a set $\mathcal{I}$ of high-dimensional training vectors X_i
- $\forall X_i \in \mathcal{I}$ there is a set S_{X_i} of training vectors that similar to X_i

Contrastive Loss (Hadsell-Chopra-LeCun-06)

- Consider a set $\mathcal{I}$ of high-dimensional training vectors X_i
- $\forall X_i \in \mathcal{I}$ there is a set S_{X_i} of training vectors that similar to X_i
- Let G_W denote the learned (neural network) embedding
- For $X_1, X_2 \in \mathcal{I}$, the parameterized euclidean distance function (written shortly as D_W) can be expressed as

$$D_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||_2$$

Contrastive Loss (Hadsell-Chopra-LeCun-06)

- Consider a set $\mathcal{I}$ of high-dimensional training vectors X_i
- $\forall X_i \in \mathcal{I}$ there is a set S_{X_i} of training vectors that similar to X_i
- Let G_W denote the learned (neural network) embedding
- For $X_1, X_2 \in \mathcal{I}$, the parameterized euclidean distance function (written shortly as D_W) can be expressed as

$$D_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||_2$$

- The piecewise loss function for a training pair is given by:

$$L(W, X_1, X_2) = \begin{cases} D_W^2 & \text{if similar} \\ max\{0, m - D_W\}^2 & \text{if dissimilar} \end{cases}$$

Dissimilar pairs contribute to the loss function only if their distance is within the margin m.

Contrastive Loss (Hadsell-Chopra-LeCun-06)

- Consider a set $\mathcal{I}$ of high-dimensional training vectors X_i
- $\forall X_i \in \mathcal{I}$ there is a set S_{X_i} of training vectors that similar to X_i
- Let G_W denote the learned (neural network) embedding
- For $X_1, X_2 \in \mathcal{I}$, the parameterized euclidean distance function (written shortly as D_W) can be expressed as

$$D_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||_2$$

- The piecewise loss function for a training pair is given by:

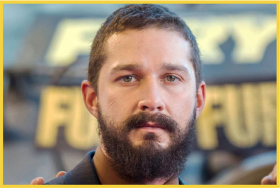
$$L(W, X_1, X_2) = \begin{cases} D_W^2 & \text{if similar} \\ max\{0, m - D_W\}^2 & \text{if dissimilar} \end{cases}$$

  Dissimilar pairs contribute to the loss function only if their distance is within the margin m.
- For P sampled training pairs and labels, the loss function is given by:

$$\mathcal{L}(W) = \sum^{P} L\left(W, (Y, X_1, X_2)^i\right)$$

Triplet Loss for Neural Network Training

- Consider now a triplet of images (x^a, x^p, x^n)

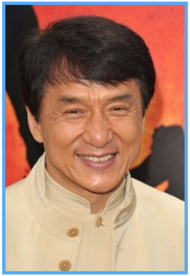


x^a

anchor

x^p

positive

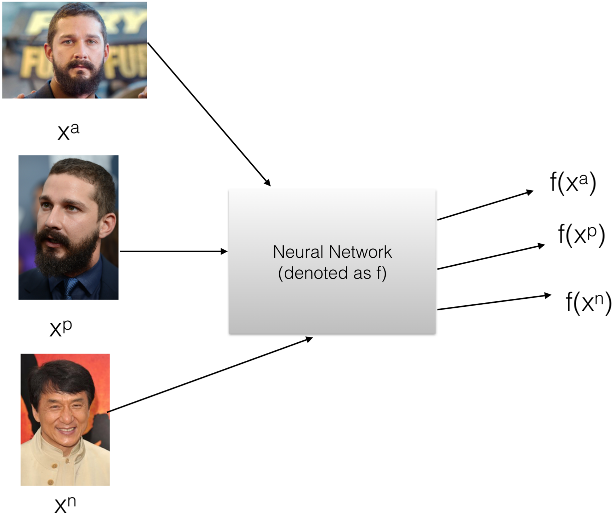


x^n

negative

Triplet Loss for Neural Network Training (cont)

- Pass them into the neural net, denoted f



Triplet Loss for Neural Network Training (cont)

- We wish to enforce the following condition for all triplets:

$$||f(x^a) - f(x^p)||_2^2 + m < ||f(x^a) - f(x^n)||_2^2$$

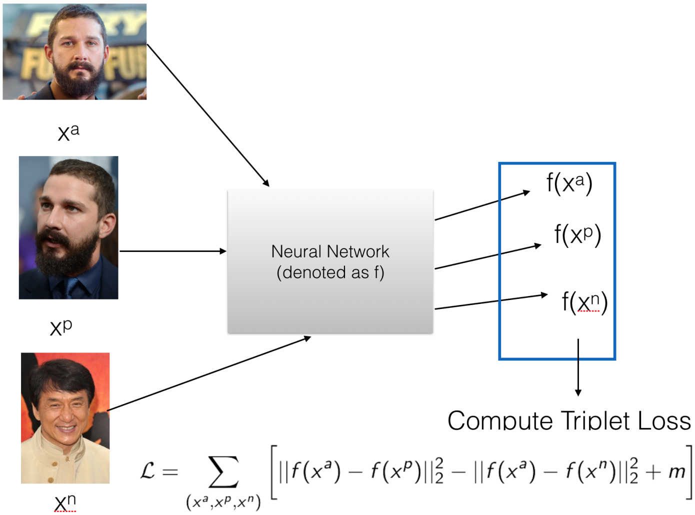
where m is some margin.

- Define the triplet loss:

$$\mathcal{L} = \sum_{(x^a, x^p, x^n)} \Big[||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + m \Big]_+$$

where $[\cdot]_+$ takes the positive part.

Triplet Loss for Neural Network Training (cont)



$$\mathcal{L} = \sum_{(x^a, x^p, x^n)} \left[||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + m \right]_+$$

Triplet Mining

- The triplet loss function cannot feasibly use all the data
- Simple combinatorics: 10000 images with 100 classes (relatively small) is about 50 *million* triplets!
- Must be able to **mine** for a subset of the triplets that will train best

Triplet Mining

- The triplet loss function cannot feasibly use all the data
- Simple combinatorics: 10000 images with 100 classes (relatively small) is about 50 *million* triplets!
- Must be able to **mine** for a subset of the triplets that will train best
- Select a smaller subset of “hard” triplets
- How to do so? Somewhat agnostic with respect to mining method—so we present a few ways

Triplet Mining (cont.)

- *Hard triplet mining:* For all matching pairs (x^a, x^p), choose the x^n that maximizes the triplet loss

$$\text{argmax}_{x^n} \Big[||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + m \Big]_+$$

  - Training is too "extreme", especially at first—bad local minima. Can implement this later in training

Triplet Mining (cont.)

- *Semi-hard mining:* For all matching pairs (x^a, x^p), fix m' and choose an x^n where

$$\mathcal{L} = ||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + m > 0$$

but

$$\mathcal{L} < m'$$

so $0 < \mathcal{L} < m'$

Triplet Mining (cont.)

- *Semi-hard mining:* For all matching pairs (x^a, x^p), fix m' and choose an x^n where

$$\mathcal{L} = ||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + m > 0$$

but

$$\mathcal{L} < m'$$

so $0 < \mathcal{L} < m'$

  - The triplets are not "as hard" because $\mathcal{L}$ is constrained
  - Usually $m' = m = 1$; no real principle to the choice

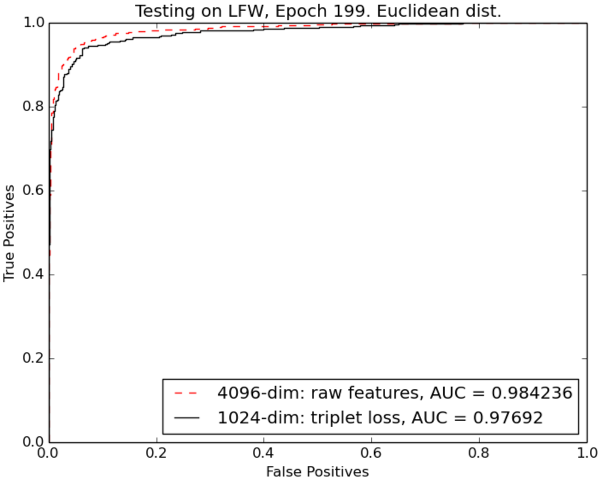
Triplet Mining (cont.)

- *Random-hard mining:* For all matching pairs (x^a, x^p), fix m'' and choose a random x^n where

$$\mathcal{L} = ||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + m > m''$$

  - Occupies a middle ground between hard and semi-hard and allows for quicker training

Results



Our accuracy on LFW: 0.937

Work summary

- Developed a Keras Implementation of triplet loss to fine-tune neural network feature embedding
- Developed a framework to combine the recognition and verification decision and a testing framework against a GumGum in-house dataset (***)
- Developed and implemented a novel metric learning method for face verification (***)

Acknowledgement

- Academic Mentor: Shantanu Joshi
- GumGum Computer Vision Team, specially: Cambron Carter and Divyaa Ravichandran
- RIPS Director: Susana Serna, Program Coordinator: Dimi Mavalski, IPAM Staff, IT
- GumGum, IPAM, and the NSF for funding
- AUC Undergraduate Research Office Travel Grant #1711739

Learn More...

- Poster Session: Friday January 12, 4:30 - 6:00pm - Exhibit Hall B2, Ground Level, San Diego Convention Center
- Stochastic Proximal Gradient Methods for Metric Learning: Saturday January 13, 2018 - Room 12, Mezzanine Level, San Diego Convention Center 8:45 am
- Get in touch: rips2017_gumgum@ipam.ucla.edu