Convolutional Neural Networks and Metric Learning for Facial Verification

Islam Faisal
The American University in Cairo

Andrew Nguyen
UC San Diego

Prem Talwai
Cornell University

Surabhi Desai
University of St Andrews

Mentor: Dr. Shantanu Joshi
University of California Los Angeles

Motivation

Face verification and recognition is one of the central problems in computer vision. GumGum use image recognition to extract information related to people in images and video. This application of computer vision enables advertising to be targeted more effectively. Other uses of facial recognition include: biometric authentication, item tag suggestions for photos and videos.

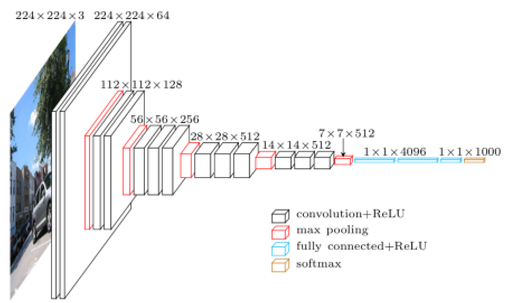
Training and Test Data

- **L**abeled **F**aces in the **W**ild (LFW) is a dataset of:
  - → **13233** face images from
  - → **5749** unique identities
- Training set of **2200** pairs is used to compute the threshold θ
  - → Find θ that results in an **E**qual Error **R**ate (EER), at which:

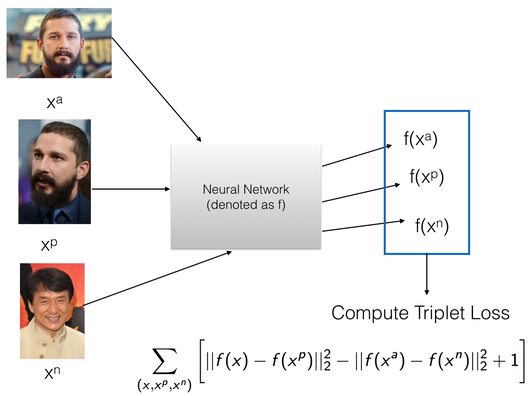
$$EER = \text{False Acceptance Rate} = \text{False Rejection Rate}$$

- Accuracy is tested against an unseen test set of **1000** pairs

Network architecture



Triplet loss function



$$\sum_{(x,x^p,x^n)} \left[||f(x) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + 1 \right]$$

Triplet Mining

For all matching pairs (x^a, x^p), fix m and choose an x^n where

$$\mathcal{L} = ||f(x^a) - f(x^p)||_2^2 - ||f(x^a) - f(x^n)||_2^2 + 1 > 0$$

but

$$\mathcal{L} < m$$

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

Triplet Loss with Metric Learning

$$\mathcal{L}(M) = \sum_{(f_a,f_p,f_n)\in\mathcal{T}} \left[d(f_a, f_p) - d(f_a, f_n) + 1 \right]_+$$

where $\mathcal{T}$ is the set of all feature triplets (f_a-anchor, f_p-positive, f_n-negative)

LDA Loss Function

$$\mathcal{L}(M) = \underbrace{\sum_{(f_a,f_p,f_n)\in\mathcal{T}} \left[d(f_a, f_p) - d(f_a, f_n) + 1 \right]_+}_{\text{triplet loss}} + \underbrace{\xi_1 ||M - LL^T||_2}_{\text{comparison to LDA}} + \underbrace{\xi_2 ||M||_*}_{\text{nuclear norm}}$$

where:

- M is the positive semi-definite matrix parameter to be learned
- L consists of the leading eigenvectors of $C_W^{-1} C_B$ (explains at least 80% of the variance)
- d is the distance function with $d(f_1, f_2) = (f_1 - f_2)^T M (f_1 - f_2)$ for $f_1, f_2 \in \mathscr{F}$ (feature space)
- Setting $M = LL^T$ reproduces LDA transform.

References

J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM Journal on Optimization, 20 (2010), pp. 1956–1982.

J. Chen, T. Yang, and S. Zhu, *Efficient low-rank stochastic gradient descent methods for solving semidefinite programs*, in Artificial Intelligence and Statistics, 2014, pp. 122–130.

F. Chollet et al., *Keras*. `https://github.com/fchollet/keras`, 2015.

R. Collobert, K. Kavukcuoglu, and C. Farabet, *Torch7: A matlab-like environment for machine learning*, in BigLearn, NIPS Workshop, 2011.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, *Online passive-aggressive algorithms*, Journal of Machine Learning Research, 7 (2006), pp. 551–585.

N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection*, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, IEEE, 2005, pp. 886–893.

Acknowledgements

- Mentors Shantanu Joshi, Cambron Carter and Divyaa Ravichandran
- RIPS and IPAM, especially Susana Serna and Dimi Mavalski
- AUC Travel Grant #1711739

SPGD for Metric Learning

Stochastic Proximal Gradient Descent is a new algorithm for metric learning, because the nuclear norm rebuffs established methods. Let $\mathcal{M}$ be all pos. semi def. matrices. $\mathcal{L}(M) = g + h$ where $h = \xi_2 ||M||_*$ and g has subgradient G_t. Let step size $\eta_t = c/\sqrt{t}$. The goal is to iteratively learn M. The algorithm:

- $M_{t+1} = \mathbf{prox}_h(M_t - \eta_t G_t) := \mathbf{argmin}_{M\in\mathscr{M}}\, g(M_t) + \eta_t \mathrm{Tr}((M - M_t)^T G_t) + \frac{1}{2}||M - M_t||_F^2 + \eta_t h(W)$
- Estimate G_t with a lower rank stochastic gradient per Chen et. al 2014.
- Per Cai et al 2010, $M_{t+1} = \mathbf{prox}_h(M_t - \eta_t \hat{G}_t) = U_1 D_{\eta_t \xi_2} U_2^T$ where U_1 and U_2 are reduced matrices of normalized eigenvectors of $M_t - \eta_t \hat{G}_t$, and $D_{\eta_t \xi_2}$ is a diagonal matrix with diagonal $\mathbf{d} = [\max(\lambda_1 - \eta_t \xi_2, 0), \max(\lambda_2 - \eta_t \xi_2, 0), \ldots, \max(\lambda_k - \eta_t \xi_2, 0)]$, where λ_i are absolute values of nonzero eigenvalues of $M_t - \eta_t \hat{G}_t$.

Convergence of SPGD

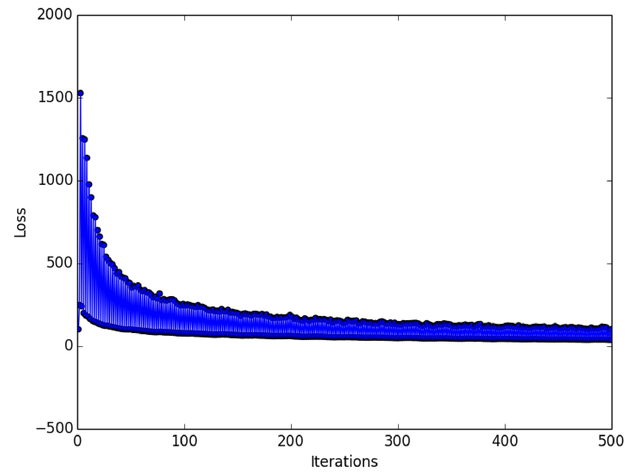


Fig. 1: Convergence of SPGD applied to LDA loss function

Final Results

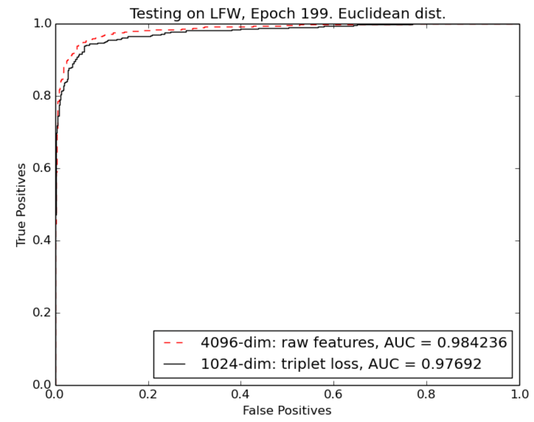


Fig. 2: Our accuracy on LFW: 0.937