Learning a Similarity Metric Discriminatively, with Application to Face Verification

Sumit Chopra Raia Hadsell Yann LeCun

Courant Institute of Mathematical Sciences
New York University
New York, NY, USA
{sumit, raia, yann}@cs.nyu.edu

Abstract

We present a method for training a similarity metric from data. The method can be used for recognition or verification applications where the number of categories is very large and not known during training, and where the number of training samples for a single category is very small. The idea is to learn a function that maps input patterns into a target space such that the L_1 norm in the target space approximates the "semantic" distance in the input space. The method is applied to a face verification task. The learning process minimizes a discriminative loss function that drives the similarity metric to be small for pairs of faces from the same person, and large for pairs from different persons. The mapping from raw to the target space is a convolutional network whose architecture is designed for robustness to geometric distortions. The system is tested on the Purdue/AR face database which has a very high degree of variability in the pose, lighting, expression, position, and artificial occlusions such as dark glasses and obscuring scarves.

1. Introduction

Traditional approaches to classification using discriminative methods, such as neural networks or support vector machines, generally require that all the categories be known in advance. They also require that training examples be available for all the categories. Furthermore, these methods are intrinsically limited to a fairly small number of categories (on the order of 100). Those methods are unsuitable for applications where the number of categories is very large, where the number of samples per category is small, and where only a subset of the categories is known at the time of training. Such applications include face recognition and face verification: the number of categories can be in the hundreds or thousands, with only a few examples per category. A common approach to this kind of problem is distance-based methods, which consist in computing a similarity metric between the pattern to be classified or verified and a library of stored prototypes. Another common approach is to use non-discriminative (generative) probabilistic methods in a reduced-dimension space, where the model for one category can be trained without using examples from other categories. To apply discriminative learning techniques to this kind of application, we must devise a method that can extract information about the problem from the available data, without requiring specific information about the categories.

The solution presented in this paper is to *learn a similarity metric from data*. This similarity metric can later be used to compare or match new samples from previously-unseen categories (e.g. faces from people not seen during training). We present a new type of discriminative training method that is used to train the similarity metric. The method can be applied to classification problems where the number of categories is very large and/or where examples from all categories are not available at the time of training.

The main idea is to find a function that maps input patterns into a target space such that a simple distance in the target space (say the Euclidean distance) approximates the "semantic" distance in the input space. More precisely, given a family of functions $G_W(X)$ parameterized by W, we seek to find a value of the parameter W such that the similarity metric $E_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||$ is small if X_1 and X_2 belong to the same category, and large if they belong to different categories. The system is trained on pairs of patterns taken from a training set. The loss function minimized by training minimizes $E_W(X_1, X_2)$ when X_1 and X_2 are from the same category, and maximizes $E_W(X_1, X_2)$ when they belong to different categories. No assumption is made about the nature of $G_W(X)$ other than differentiability with respect to W. Because the same function G with the same parameter W is used to process both inputs, the similarity metric is symmetric. This is called a *siamese architecture* [4].

To build a face verification system with this method, we first train the model to produce output vectors that are nearby for pairs of images from the same person, and far away for pairs of images from different persons. The model can then be used as a similarity metric between face images of new persons that were not seen during training.

An important aspect of the proposed method is that we have complete freedom in the choice of $G_W(X)$. In particular, we will use architectures which are designed to extract representations that are robust to geometric distortions of the input, such as convolutional networks [8]. The resulting similarity metric will be robust to small differences of the pose between the pairs of images.

Since the dimension of the target space is low and the natural distance in that space is invariant to irrelevant distortions of the input, we can easily estimate probabilistic models of each new category from a very small number of samples.

1.1. Previous Work

The idea of mapping face images to low dimensional target spaces before comparison has a long history, starting with the PCA-based Eigenface method [16] in which $G(X)$ is a linear projection trained non-discriminatively to maximize the variance. The LDA-based Fisherface method [3] is also linear, but trained discriminatively so as to maximize the ratio of inter-class and intra-class variances. Nonlinear extensions based on Kernel-PCA and Kernel-LDA have been discussed [5]. See [14] for a review of subspace methods for face recognition. One major shortcoming of all those approaches is that they are very sensitive to geometric transformations of the input images (shift, scaling, rotation) and to other variabilities (changes in facial expression, glasses, and obscuring scarves). Some authors have described similarity metrics that are locally invariant to a set of known transformations. One example is the Tangent Distance method [19]. Another example, which has been applied to face recognition, is elastic matching [6]. Others have advocated warping-based normalization algorithms to maximally reduce the variations of appearance due to pose [10]. The invariance properties of all these models are hand-designed in advance. In the method described in this paper, the invariance properties do not come from prior knowledge about the task, but they are learned from data. When used with a convolutional network as the mapping function, the proposed method can learn a wide range of invariances present in the data.

Our approach is somewhat similar to that of [4], which uses a siamese architecture for signature verification. The main difference between their method and ours is the nature of the loss function minimized by the training process. Our loss function is derived from the discriminative learning framework for energy-based models (EBM).

Our method is very different from other dimensionality reduction techniques such as Multi-Dimensional Scaling (MDS) [13] and Local Linear Embedding (LLE) [15] MDS computes a target vector from each input object in the training set based on known pairwise dissimilarities, without constructing a mapping. By contrast, our method produces a non-linear mapping that can map any input vector to its corresponding low-dimensional version.

2. The General Framework

While probabilistic models assign a normalized probability to every possible configuration of the variables being modeled, energy-based models (EBM) assign an unnormalized energy to those configurations [18, 9]. Prediction in such systems is performed by searching for configurations of the variables that minimize the energy. EBMs are used in situations where the energies for various configurations must be compared in order to make a decision (classification, verification, etc). A trainable similarity metric can be seen as associating an energy $E_W(X_1, X_2)$ to pairs of input patterns. In the simplest face verification setting, we simply set X_2 to all the available images of the claimed identity and compare the minimum $E_W(X_1, X_2)$ to a pre-determined threshold.

The advantage of EBMs over traditional probabilistic models, particularly generative models, is that there is no need for estimating normalized probability distributions over the input space. The absence of normalization saves us from computing partition functions that may be intractable. It also gives us considerably more freedom in the choice of architectures for the model [9].

Learning is performed by finding the W that minimizes a suitably designed *loss function*, evaluated over a training set. At first glance, we might think that simply minimizing $E_W(X_1, X_2)$ averaged over a set of pairs of inputs from the same category would be sufficient. But this generally leads to a catastrophic collapse: The energy and the loss can be made zero by simply making $G_W(X_1)$ a constant function. Therefore our loss function needs a *contrastive term* to ensure not only that the energy for a pair of inputs from the same category is low, but also that the energy for a pair from different categories is large. This problem does not occur with properly normalized probabilistic models because making the probability of a particular pair high automatically makes the probability of other pairs low.

2.1. Face Verification with Learned Similarity Metrics

The task of face verification [12], is to accept or reject the claimed identity of a subject in an image. Performance is assessed using two measures: percentage of *false accepts* and the percentage of *false rejects*. A good system should minimize both measures simultaneously.

Our approach is to build a trainable system that non-linearly maps the raw images of faces to points in a low dimensional space so that the distance between these points is small if the images belong to the same person and large otherwise. Learning the similarity metric is realized by training a network that consists of two identical convolutional networks that share the same set of weights - a *Siamese Architecture* [4] (see figure 1).

2.2. The energy function of the EBM

The architecture of our learning machine is given in figure 1. The details of the architecture of $G_W(X)$ are given in section 3.2.

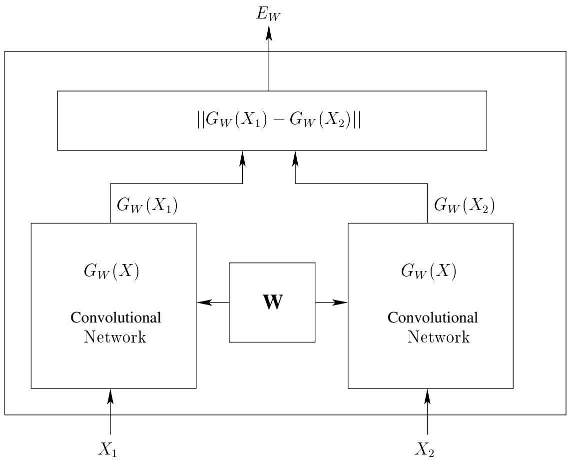

Figure 1. Siamese Architecture.

Let X_1 and X_2 be a pair of images shown to our learning machine. Let Y be a binary label of the pair, $Y = 0$ if the images X_1 and X_2 belong to the same person (a "genuine pair") and $Y = 1$ otherwise (an "impostor pair"). Let W be the shared parameter vector that is subject to learning, and let $G_W(X_1)$ and $G_W(X_2)$ be the two points in the low-dimensional space that are generated by mapping X_1 and X_2. Then our system can be viewed as a scalar "energy function" $E_W(X_1, X_2)$ that measures the compatibility between X_1, X_2. It is defined as

$$E_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)|| \tag{1}$$

Given a genuine pair from the training set (X_1, X_2), and an impostor pair from the training set (X_1, X_2'), the machine behaves in a desirable manner if the following condition holds:

Condition 1 $\exists m > 0$, *such that* $E_W(X_1, X_2) + m < E_W(X_1, X_2')$,

The positive number m can be interpreted as a margin.

To simplify notations $E_W(X_1, X_2)$ is written as E_W^G and $E_W(X_1, X_2')$ as E_W^I for the remainder of the paper.

2.3. Contrastive Loss Function used for Training

We assume that the loss function depends on the input and the parameters only indirectly through the energy. Our loss function is of the form:

$$\mathcal{L}(W) = \sum_{i=1}^{P} L(W, (Y, X_1, X_2)^i)$$

$$L(W, (Y, X_1, X_2)^i) = (1 - Y) L_G \left(E_W(X_1, X_2)^i \right) + Y L_I \left(E_W(X_1, X_2)^i \right)$$

where $(Y, X_1, X_2)^i$ is the i-th sample, which is composed of a pair of images and a label (genuine or impostor), L_G is the partial loss function for a genuine pair, L_I the partial loss function for an impostor pair, and P the number of training samples. L_G and L_I should be designed in such a way that the minimization of L will decrease the energy of genuine pairs and increase the energy of impostor pairs. A simple way to achieve that is to make L_G monotonically increasing, and L_I monotonically decreasing. However, there is a more general set of conditions under which minimizing L will make the machine approach condition 1. Our arguments are similar to those given by LeCun et al in [9]. We will consider a training set composed of one genuine pair (X_1, X_2) with energy E_W^G and one impostor pair (X_1, X_2') with energy E_W^I. Let us define:

$$H(E_W^G, E_W^I) = L_G(E_W^G) + L_I(E_W^I) \tag{2}$$

as the total loss function for the two pairs. We will assume that H is convex in its two arguments (note: we do not assume convexity with respect to W). We also assume that there exists a W for single training sample such that condition 1 is satisfied. The following conditions must hold for the loss function H for all values of E_W^G and E_W^I.

Condition 2 *The minima of* $H(E_W^G, E_W^I)$ *should be inside the half plane* $E_W^G + m < E_W^I$.

This condition clearly guarantees that when we minimize H with respect to W, the machine is driven to a region where the solution satisfies condition 1.

For H whose minima lie at infinity (see figure 2), the following condition is sufficient:

Condition 3 *The negative of the gradient of* $H(E_W^G, E_W^I)$ *on the margin line* $E_W^G + m = E_W^I$ *has a positive dot product with the direction* [-1, 1].

To prove this, we state and prove the following theorem.

Theorem 1 *Let $H(E_W^G, E_W^I)$ be convex in E_W^G and E_W^I and have a minimum at infinity. Assume that there exists a W for a sample point such that condition 1 is satisfied. If condition 3 holds, then minimizing H with respect to W would lead to finding a W that satisfies condition 1.*

Proof. Consider the positive quadrant of the plane formed by E_W^G and E_W^I (see figure 3). Let the two half planes $E_W^G + m < E_W^I$ and $E_W^G + m \geq E_W^I$ be denoted by HP_1 and HP_2 respectively. We minimize H over E_W^G and E_W^I for all the values of W in its domain. Let R be the region inside the plane formed by E_W^G and E_W^I which correspond to all the values in the domain of W. In the most general setting R could be non-convex and could lie anywhere in the plane. However by our assumption that there exists at least one W such that condition 1 is satisfied, we can conclude that a part of R intersects the half plane HP_1. In order to prove the theorem in the light of condition 3, we need to show that there exists at least one point in the intersection of R and HP_1, such that the loss H at this point is less than the loss at all the points in the intersection of R and HP_2.

Let E_G^* be the point on the margin line $E_W^G + m = E_W^I$, for which H is minimum. That is,

$$E_G^* = \text{argmin}\{H(E_W^G, E_W^G + m)\} \tag{3}$$

Since the negative of the gradient of H at all the points on the margin line is in the direction which is inside the half plane HP_1 (condition 3), by convexity of H we can conclude that

$$H(E_G^*, E_G^* + m) \leq H(E_W^G, E_W^I) \tag{4}$$

when $E_W^G + m > E_W^I$.

Now consider a point at a distance ϵ away from $(E_G^*, E_G^* + m)$, and inside the half plane HP_1. That is the point

$$H(E_G^* - \epsilon, E_G^* + m + \epsilon). \tag{5}$$

Using a first-order Taylor expansion, we can write the above as::

$$\begin{aligned} & H(E_G^* - \epsilon, E_G^* + m + \epsilon) \\ & = H(E_G^*, E_G^* + m) - \epsilon \frac{\partial H}{\partial E_W^G} + \epsilon \frac{\partial H}{\partial E_W^I} + O(\epsilon^2) \\ & = H(E_G^*, E_G^* + m) + \epsilon \left[\frac{\partial H}{\partial E_W^G} \frac{\partial H}{\partial E_W^I} \right] \left[\begin{array}{r} -1 \\ 1 \end{array} \right] + O(\epsilon^2) \end{aligned} \tag{6}$$

By condition 3, the second term on the right hand side of equation 6 is negative. Thus for sufficiently small ϵ,

$$H(E_G^* - \epsilon, E_G^* + m + \epsilon) \leq H(E_G^*, E_G^* + m) \tag{7}$$

Thus there exists a point in the intersection of the region R and the half plane HP_1 at which the loss function is less than at any point in the intersection of R and HP_2. Hence the claim follows. □

Note that condition 3 clearly holds for any H whenever L_0 is a monotonically increasing function, and L_1 is a monotonically decreasing function.

The exact loss function that we use for a single sample is

$$\begin{aligned} L(W, Y, X_1, X_2) &= (1 - Y) L_G(E_W) + Y L_I(E_W) \quad (8) \\ &= (1 - Y) \frac{2}{Q} (E_W)^2 + (Y) 2Q \, e^{-\frac{2.77}{Q} E_W} \quad (9) \end{aligned}$$

where $E_W = \|G_W(X_1) - G_W(X_2)\|$. In our architecture, the components of G_W are bounded, hence E_W is also bounded. The constant Q is set to the upper bound of E_W.

One can clearly see that the above loss function is monotonically increasing in E_W^G and monotonically decreasing in E_W^I, and it is convex with respect to both E_W^G and E_W^I. Hence by the above arguments we conclude that minimizing this loss function would lead the machine to a W where it would behave in the desired manner.

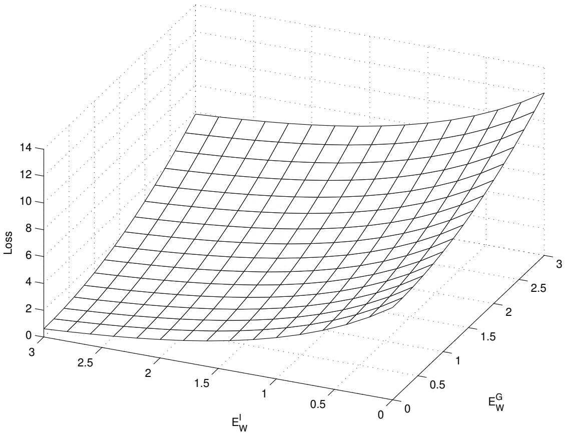

Figure 2. Graph of the loss function H against E_W^G and E_W^I in 3D.

We make two further remarks concerning our loss functions. First, the constants in the loss function are explained. The optimization algorithm that we use to minimize our loss function is based on the gradient. These constants are chosen so as to make sure that the direction of the negative of the gradient of the loss function on the margin line always point inside the region R. This is required to avoid the situation where our algorithm is stuck at a point on the boundary of the R with the gradient pointing outside R. In such a case a gradient based algorithm may identify that point as a local minimum of the loss function and terminate.

Second, we must emphasize that using the square norm instead of the $L1$ norm for the energy would not be appropriate. Indeed, if the energy were the square norm of the difference between the output vectors of the two patterns, the

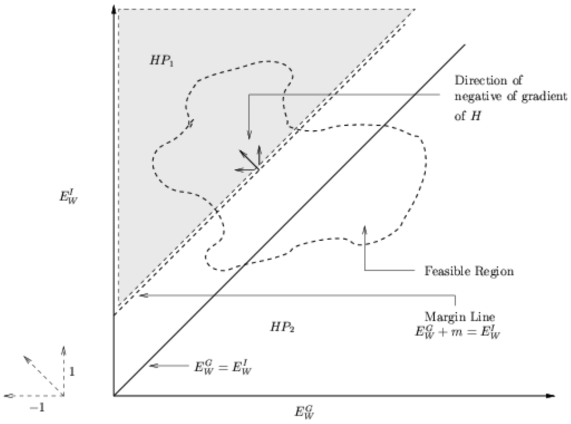

Figure 3. Plot showing the two half planes HP_1 and HP_2 and the feasible region.

gradient of the energy with respect to the parameter would vanish as the energy approached zero. This would create a dangerous plateau in the loss function. This could lead to failure of the machine to learn in cases where the two images are impostors and the corresponding energy is near zero.

2.4. Convolutional Networks

In order to map the raw images to points in a low dimensional space and hence to realize a learned similarity metric, we use two identical convolutional networks [8] with a common parameter vector (see figure 1). Convolutional networks are trainable, multi-layer, non-linear systems that can operate at the pixel level and learn low-level features and high-level representations in an integrated manner. Convolutional nets are trained *end-to-end* to map pixel images to outputs. Their main advantage is that they can learn optimal shift-invariant local feature detectors and build representations that are robust to geometric distortions of the input image. The exact specifications of the network we use are given in section 3.2.

3. Experiments

The model and architecture described in the previous section was trained on 3 databases of face images, and tested on 2 of those databases. We will discuss the databases in detail and then explain the training protocol and architecture.

3.1. Datasets and Data Processing

The first round of training and testing was done with a relatively small dataset of 400 images from the AT&T Database of Faces [1]. The dataset contains 10 images each of 40 subjects, with variations in lighting, facial expression, accessories, and head position. Each image is 112x92 pixels, gray scale, and closely cropped to include the face only. See figure 4.

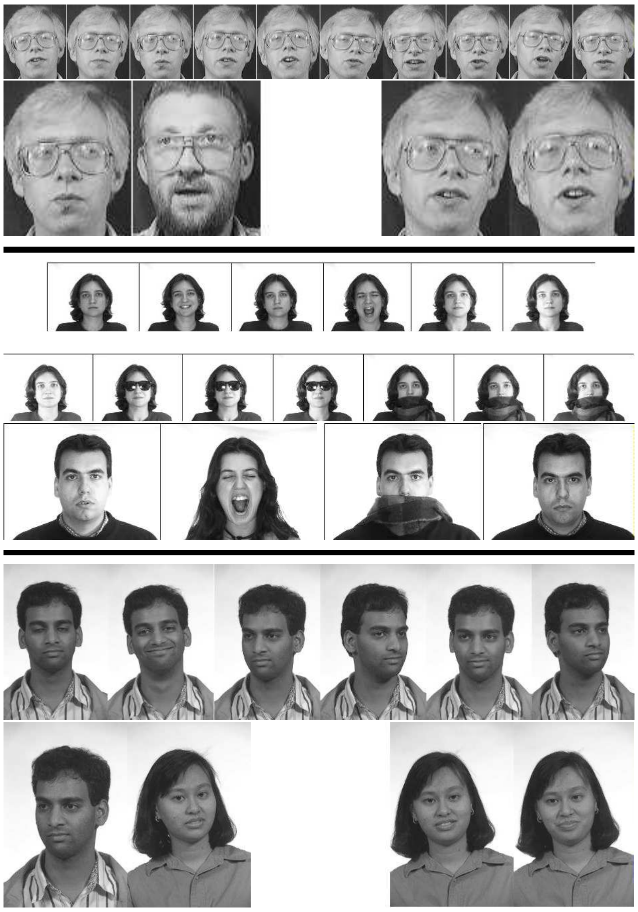

Figure 4. Top: Images from AT&T dataset. Middle: Images from the AR dataset. Bottom: Images from FERET dataset. Each graphic shows a genuine pair, an impostor pair and images from a typical subject.

There was no need to pre-process the images for size or lighting normalization, since one of the stated goals was to train an architecture that would be resilient to such variations. However, we did reduce the resolution of the images to 56x46 using 4x4 subsampling.

The second set of training and testing experiments was performed by combining two datasets: the AR Database of Faces, created at Purdue University and publicly available [11], and a subset of the grayscale Feret Database [2]. Image pairs from both of these datasets were used in training, but only images from the AR dataset were used for testing.

The AR dataset comprises 3,536 images of 136 subjects with 26 images per subject. Each 26-image set for a subject is made up of 2 sets of 13 images taken 14 days apart. Within each set of 13 images, there are 4 images with expression variations, 3 images with lighting variations, 3 images with dark sunglasses and lighting variations, and 3 images with face-obscuring scarves and lighting variations. This dataset is extremely challenging because of the wide variation of appearance between images of a single subject. Examples can be seen in figure 4. Since the faces were not well centered in the images, a simple correlation-based centering algorithm was applied. The images were then cropped and reduced to 56x46 pixels. Although the centering was sufficient for the purposes of cropping, there remained substantial variations in head position in many images.

The Feret Database, distributed by the National Institute of Standards and Technology, comprises 14,051 images collected from 1,209 subjects. We used a subset of the full database solely for training. Our subset consists of 1122 images, that is, 6 images each of 187 subjects. The only preprocessing was cropping and subsampling to 56x46 pixels.

Partitioning For the purpose of generating a test set consisting of images of the subjects that are not seen by the machine during training, we split the datasets into two disjoint sets, namely SET1 and SET2. Each image in each of these sets was paired up with every other image in that set to generate the maximum number of genuine pairs and impostor pairs.

For the AT&T data, SET1 consisted of 350 images of first 35 subjects and SET2 consisted of 50 images of last 5 subjects. This way a total of 3500 genuine and 119000 impostor pairs were generated from SET1 and 500 genuine and 2000 impostor pairs were generated from SET2. Training was done using only the image pairs generated from SET1. Testing (verification) was done using the image pairs from SET2 and the unused image pairs from SET1.

For the AR/Feret data, SET1 contained all the Feret images and 2,496 images from 96 subjects in the AR database. SET2 contained the 1,040 images from the remaining 40 subjects in the AR database. Taking all combinations of 2 images resulted in 71,628 genuine and 11,096,376 impostor pairs. The actual training set that was used contained 140,000 image pairs that were evenly split between genuine and impostor. The testing set was drawn from the 1,081,600 pairs in SET2. Thus, only subjects that had not been seen in training were used for testing.

3.2. Training Protocol and Architecture

Siamese Architecture The Siamese framework comprises two identical networks and one cost module. The input to the system is a pair of images and a label. The images are passed through the sub-networks, yielding two outputs which are passed to the cost module which produces the scalar energy as discussed in section 2.3. The loss function combines the label with energy. The gradient of the loss function with respect to the parameter vector controlling both subnets is computed using back-propagation. The parameter vector is updated with a stochastic gradient method using the sum of the gradients contributed by the two subnets.

The first set of experiments, using the small AT&T dataset, explored 6 different sub-net architectures: one 2-layer fully-connected neural network and five convolutional networks that varied in number and size of layers and convolution kernel size. Based on those experiments, the second set of experiments focused on a single convolutional network architecture. We only describe the best-performing architecture in the following sections. C_x denotes a convolutional layer, S_x denotes a sub-sampling layer, and F_x denotes a fully connected layer, where x is the layer index. The basic architecture is $C_1 - S_2 - C_3 - S_4 - C_5 - F_6$,

- C_1. Feature maps: 15; Size 50x40; Kernel size: 7x7. Trainable parameters: 750; Connections: 1500000. Fully Connected with the input.
- S_2. Feature maps: 15; Size: 25x20; Field of view: 2x2. Trainable parameters: 30; Connections: 37500.
- C_3. Feature maps: 45; Size: 20x15; Kernel size: 6x6. Trainable parameters: 7128 ; Connections: 2139600. Partially connected to S_2. The exact connections are in a pattern similar to that used in [8]. The motivation behind this was to break symmetry, thereby pushing the feature maps to extract and learn different features.
- S_4. Feature maps: 45; Size: 5x5; Field of view: 4x3. Trainable parameters: 100; Connections: 16250.
- C_5. Feature maps: 250; Size 1x1; Kernel size: 5x5. Trainable connections: 312750. Fully connected to S_4.
- F_6. Number of units: 50. Trainable parameters: 12550; Connections: 12550.

Training Protocol Training requires two sets of data: the training set, for actually learning the weights of the system, and the validation set, for testing the performance of the system during training. Periodical performance evaluation with the validation set allows us to control over-fitting.

Training the network was done with pairs of images taken from SET1. One half of the image pairs were genuine and one half were impostor, produced by randomly pairing images of different subjects. The validation set was composed of 1500 image pairs, taken from the unused pairs of SET1, and in the same 50% genuine, 50% impostor ratio as the training set.

The performance of the network was measured by a calculation of the percentage of impostor pairs accepted (FA),

	AT&T		AR/Purdue	
	Val	Test	Val	Test
Number of Subjects	35	5	96	40
Images/Subject	10	10	26	26
Images/Model	–	5	–	13
No. Genuine Images	500	500	750	500
No. Impostor Images	500	4500	750	4500

	False Accept		
	10%	7.5%	5%
AT&T (Test)	0.00	1.00	1.00
AT&T (Validation)	0.00	0.00	0.25
AR (Test)	11	14.6	19
AR (Validation)	0.53	0.53	0.80

Table 1. Above: Details of the validation and test sets for the two datasets. Below: False reject percentage for different false accept percentages.

Figure 5. Internal state of the convolutional network for a particular example.

and the percentage of genuine pairs rejected (FR). This calculation was made by measuring the norm of the difference between the outputs of a pair, then picking a threshold value that sets a given trade-off between the FA and FR percentages.

4. Testing and Results

Figure 5 shows the internal state of the convolutional network for a particular test image. The first layer extracts various types of local gradient features, as well as smooth features.

The system was tested for a face verification scenario. The system is given an image and asked to confirm the claimed identity of the subject in that image. We perform verification by comparing the test image with a gaussian model of images of the claimed subject. The method is discussed below.

4.1. Verification

Testing (verification) was done on a test set of size 5000. It consisted of 500 genuine and 4500 impostor pairs. For the AT&T experiments the test images were from 5 subjects unseen in training. For AR/Feret experiments the test images were from 40 unseen subjects in the more difficult AR database.

The output from one of the subnets of the Siamese network is a feature vector of the input image of the subject. We assume that the feature vectors of each subject's image form a multivariate normal density. A model is constructed of each subject by calculating the mean feature vector and the variance-covariance matrix using the feature vectors generated from the first five images of each subject.

The likelihood that a test image is genuine, ρ_{genuine}, is found by evaluating the normal density of the test image on the model of the concerned subject. The likelihood of a test image being an impostor, ρ_{impostor}, is assumed to be a constant whose value is estimated by calculating the average ρ_{genuine} value of all the impostor images of the concerned subject. The probability that the given image is genuine is given by

$$\mathrm{Prob(genuine)} = \frac{\rho_{\mathrm{genuine}}}{\rho_{\mathrm{genuine}} + \rho_{\mathrm{impostor}}}$$

The values of the percentage of falsely rejected images and the falsely accepted images are plotted for all possible values of the threshold probability. The optimal threshold probability is the value that partitions the test set into genuine and impostor pairs and minimizes FA and FR rates.

The verification rates obtained from testing the AT&T database and the AR/Purdue database are strikingly different (see table 1 and figures 6 and 7), underlining the differences in difficulty in the two databases. The AT&T dataset is relatively small, and our system required only 5000 training samples to achieve very high performance on the test set. The AR/Purdue dataset is very large and diverse, with huge variations in expression, lighting, and added occlusions. Our higher error rates reflect this level of difficulty.

5. Conclusion and Outlook

We present a general discriminative method for learning complex similarity metrics. The method is best suited for classification or verification scenarios where the number of classes is very large, and/or where examples of all the classes are not available at the time of training. We illustrate the method with a face verification application.

We propose a loss function and show that minimizing this function causes the system to approach the desired behavior. Our loss function is discriminative in the sense that it drives the system to make the right decision, but does not cause it to produce probability estimates. The method is

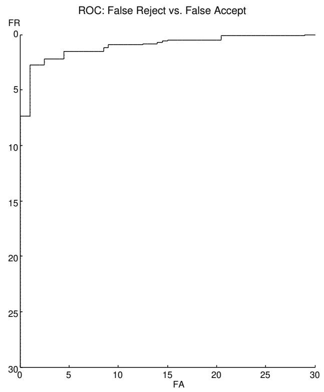

Figure 6. AT&T dataset: percent false reject vs. false accept.

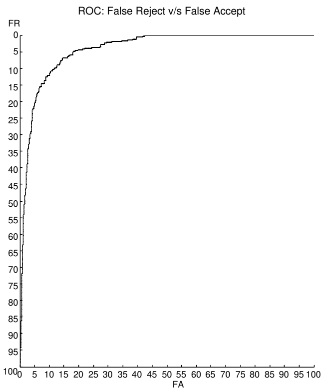

Figure 7. AR/Purdue dataset: percent false reject vs. false accept.

different from probabilistic density models in that there is no attempt to estimate a density for each class in the input space. This gives us an additional amount of flexibility in the choice of $G_W(X)$, because we do need to worry about normalization. We chose to use a convolutional network architecture which exhibits robustness to geometric variations of the input, thereby reducing the need for accurate registration of the face images.

Trainable similarity metrics have numerous applications beyond the one described in this paper. Among other things, they can be used to build invariant kernel functions with which to build Support Vector Machines and other kernel-based models [17].

References

[1] http://www.uk.research.att.com/facedatabase.html.

[2] http://www.itl.nist.gov/iad/humanid/feret/.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. PAMI, Special Issue on Face Recognition*, 19(7), July 1997.

[4] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah. Signature verification using a siamese time delay neural network. J. Cowan and G. Tesauro (eds) *Advances in Neural Information Processing Systems*, 1993.

[5] M. Hsuan Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. *In Proc. of the 2000 IEEE International Conference on Image Processing (ICIP)*, 1:37–40, September 2000.

[6] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion-invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(3):300–311, 1993.

[7] S. Lawrence, C. Lee Giles, A. Chung Tsoi, and A. D. Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks*, 1997.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Y. LeCun and F. Jie Huang. Loss functions for discriminative training of energy-based models. *AI-stats*, 2005.

[10] A. M. Martinez. Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):748–763, 2002.

[11] A. M. Martinez and R. Benavente. The ar face database. *CVC Technical Report*, 24, June 1998. http://rvl1.ecn.purdue.edu/ aleix/aleix_face_DB.html.

[12] S. Rizvi, P. J. Phillips, and H. Moon. The feret verification testing protocol for face recognition algorithms. *Technical Report NISTIR* 6,281, *Nat'l Inst. Standards and Technology*, 1998. http://www.nist.gov/itl/div894/894.03/pubs.html#face.

[13] Y. Rubner, L. J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimenional scaling, and color-based image retrieval. *Proc. DARPA Image Understanding Workshop*, pages 661–668, May 1997.

[14] G. Shakhnarovich and B. Moghaddam. *Handbook of Face Recognition*, chapter Face Recognition in Subspaces. Springer-Verlag, 2004.

[15] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290.

[16] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[17] P. Vincent and Y. Bengio. A neural support vector network architecture with adaptive kernels. *In Proc. of the International Joint Conference on Neural Networks*, 5, July 2000.

[18] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, 2003.

[19] P. Y. Simard, Y. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition – tangent distance and tangent propagation. *International Journal of Imaging Systems and Technology*, 11(3), 2000.