Deep Neural Networks for Face Recognition: Pairwise Optimisation

E. Popova, A. Athanasopoulos, E. Ie, N. Christou, and N. Nyah

School of Computer Science & Technology, University of Bedfordshire, UK
ndifreke.nyah@study.beds.ac.uk

Abstract. There are many factors which affect the face recognition accuracy: lighting conditions, head rotations and view angles affect the reliability of face recognition and make the recognition task difficult. Recognition of multiple subjects requires to learn class boundaries, whose complexities grow exponentially. Artificial Neural Networks (ANNs) have provided efficient face recognition solutions, although the task complexity limits their performance. Multi-class and convolutional ANN models require massive computations and finding *ad-hoc* parameters in order to maximise the performance. Pairwise ANN structure has outperformed the multi-class ANNs on some face recognition tasks. We propose the pairwise optimisation for ANN, which in our experiments has required a significantly smaller number of *ad-hoc* parameters and substantially fewer computations than the multi-class and convolutional networks.

Keywords: Deep Neural Networks, Random Search Optimisation, Face Recognition

1 Introduction

Face recognition is a popular research area in Machine Learning, in which further attempts are required to improve the efficiency. Recognition algorithms are based on biometric analysis of features such as relative shape and size of face structure, distance between eyes, nose and lips. Biometric estimates are then matched with features stored in a database, as described in [12, 6].

There are many factors which affect the efficiency of face recognition algorithms. The algorithms need to deal with human faces which naturally reflect emotions and moods. Besides, human faces are often captured under variable lighting conditions, head rotations, view angles, hair styles, that makes the recognition task difficult, see e.g. [26].

For extraction of biometric features from images, methods such as Principal Component Analysis (PCA) [14], Independent Component Analysis [2], Linear Discriminant Analysis [3] have been proposed. Feasibility of using PCA for the statistical representation of facial images has been explored in [10, 29]. The PCA has been found to be efficient for representing the recognition tasks [30], although the statistical projections become unreliable if images have a poor quality.

Face recognition algorithms have been efficiently developed by using Bayesian methods which employ Markov chain Monte Carlo (MCMC) for integration over posterior parameters [25, 8, 21]. Bayesian methodology of probabilistic inference has provided highly competitive solutions, as shown in [13, 1, 20], whilst Markov models have been used for designing a highly competitive solution on the Cambridge face database [19].

The above challenges of face recognition have motivated the use of Artificial Neural Networks (ANNs) with fully-connected 3-layer structures including input, hidden, and output layers, see e.g. [31]. Such a structure has to be predefined in order to use back-propagation learning and to minimise an error function on given training data. The minimisation of the error function is achieved by adjusting neural synaptic weights. ANNs with a predefined structure are named "Shallow", see eg [24]. In contrast, "Deep" Neural Networks include multiple layers which generate new features, see e.g. [16, 23].

Integral transformation, known as convolution, defines feature maps within so-called Convolutional Neural Networks (CNNs) [15, 11]. Feature maps are typically described by 3-dimensional tensors. Each map is determined by its width, depth, and height.

It has been shown in [5] that a multi-class problem can be represented by a set of 2-class discriminators which have significantly better conditions for recognition of difficult patterns. For a c-class problem, the discriminators are trained to recognise $c(c-1)/2$ pairs of classes. A Pairwise ANN proposed in [30] has significantly outperformed the multi-class ANN on the Yale face recognition benchmark.

Pairwise ANNs reduce a multi-class problem to a set of 2-class discriminators trained to distinguish samples representing pairs of classes. Such 2-class class boundaries are significantly simpler than those of the given multi-class problem, see e.g. [30]. According to [18], the pairwise discriminators have provided outputs within the probabilistic framework for human handwriting recognition. Employing a typical 3-layer ANN for 2-class discrimination, the pairwise architecture has outperformed the multi-class ANN. The use of pairwise ANNs has been also reported to be efficient in [32]. The pairwise architecture however has not guaranteed an improvement in the accuracy if boundaries between pairs are oversimplified.

For some face recognition problems in our experiments the pairwise discrimination has minimised the training error to zero. However validation errors have still happened. We assume that the pairwise discrimination is prone to oversimplify the boundaries between patterns which are difficult for recognition.

The above motivated us to explore factors which can oversimplify the decision boundaries and find a solution to this problem. Analysis of the validation errors shows that a pairwise discriminator, trained as a 2-class ANN, makes class boundaries dependent on initialisation of ANN as well as on a distribution of the image samples used for training. Both factors arbitrarily change the location of class boundary within an area where the training error is kept to be zero, whilst the validation errors can happen. We will test this hypothesis on the Yale face

recognition benchmark available at [4], and finally we will compare our approach with alternative multi-class ANN and CNN solutions.

2 Related Work: Convolution Neural Networks

The image data received by a CNN are extracted by its convolutional layers. Pooling layers involved in CNN downsample the data in order to reduce the dimensionality of feature map, and thus to decrease processing time. Typically a pooling algorithm extracts subregions of the feature map (eg 2x2-pixel tiles) and finds the maximum value which will replace all other values in the map. There are dense (or fully connected) layers which perform classification on the features extracted by the convolutional layers and downsampled by the pooling layers. Nodes in the dense layer are connected to nodes in the previous layer.

The overfitting problem in CNN is reduced by using the Dropout technique which defines random selection of neurons at layers. Data augmentation can also mitigate the problem if some training data will be slightly transformed so as to generate variations in the data. This technique diffuses specific patterns that can exist in the training data but can be absent in the unseen data. The back-propagation learning is typically used to adjust CNN parameters so as to minimise the error function. An example of CNN described in [17] uses a 5-layer network of convolutional and pooling (or subsampling) layers. The first layer takes a $k \times k$-image convoluted by using 5×5 filter. The second layer receives a set of 4-feature maps, whose data were subsampled in a 2×2 window. Another example [28] describes a CNN providing a high recognition accuracy, which uses several layers of inhibitory neurons with different connections between layers, including those which generate feature maps.

3 Method

3.1 Pairwise Neural Networks

Let pairwise discrimination functions be fi/j, where $i = 1, \ldots, c-1$, and $j = i+1, \ldots, c$, and c is the number of classes. Fig. 1 illustrates a 3-class task represented by samples shown in Green, Red, and Blue. Components x_1 and x_2 represent the input data. The given 3-class problem is transformed to a set of 2-class discriminators $f1/2$, $f1/3$ and $f2/3$.

Discriminators fi/j are defined having a tansigmoid function with output y: $y = (1, -1)$. The output $y > 0$ for input x belonging to class i, and $y < 0$ for x belonging to class j. Each discriminator fi/j is trained independently on a set of training samples taken from the classes i and j.

The output layer includes neurons Σ_k where $k = 1, \ldots, c$. Their coefficients are assigned to be $+1$ for $k = i$ and -1 for $k = j$. The maximum $\max_{k=1}^{c}(y_k)$ determines the class to which the given input x is assigned.

Fig. 2 illustrates our approach to the problem, outlined in Section 1, associated with oversimplification of pairwise class boundaries. Each discriminant

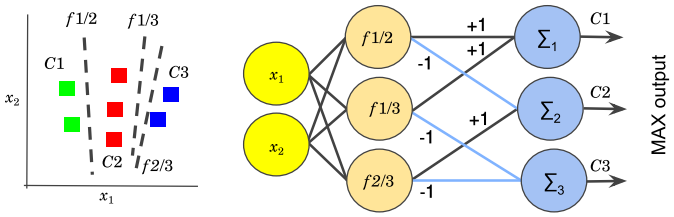

Fig. 1. Pairwise ANN for a 3-class problem.

function fi/j can be trained with zero errors as shown in Fig. 2. However the validation errors, denoted by the circles, are observed. The errors happen at random because (i) the 2-class ANNs are initialised with random weights and (ii) the validation samples are randomly distributed. Thus it is reasonable to average the outputs of discriminant functions $f_1, \ldots, f_k$ in order to reduce a side effect of the initialisation. An open question still remains: how many principal components are needed to achieve the best generalisation?

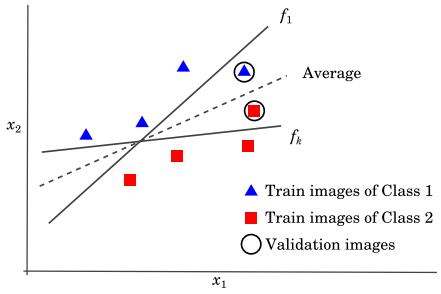

Fig. 2. Discriminant functions $f_1, \ldots, f_k$ and the expected average class boundary.

3.2 Random Search Optimisation of Hyper-Parameters

To find a solution to the above problem, we propose Algorithm 1 for search optimisation of principal components which are used by discriminant functions having parameters Θ. The algorithm makes a random change of Θ and then evaluates the proposed change in terms of entropy calculated on the training data. This search strategy is closely related to Simulated Annealing and Metropolis-Hastings algorithms, see e.g. [27, 22, 9].

The proposed algorithm uses the following settings:

(1) $[X, T]$ is the training data for a pairwise discriminator fi/j, where X are a $n \times m_{max}$ matrix of m_{max} principal components for n data samples, and $\{T_k \in \{1, -1\}\}_{k=1}^{n}$ are the labels of classes i and j, respectively.

(2) m_0 is the prior on the number of principal components, assigned to be $m_0 \sim U(m_{min}, m_{max})$, where m_{min} is the minimal number of components, and U is a uniform distribution function.

(3) v_m is the proposal deviation of the number of components.

(4) E is the entropy of an ANN model with a given number of components, specified by function $f(X, T)$.

(5) K is the number of pairwise functions to be averaged.

The algorithm makes a proposal within a given prior in order to change the number of principal components required by an ANN. The proposals are drawn from a uniform distribution U. The ANN is trained with a new set of principal components to be evaluated in terms of entropy E. The proposed change is accepted according to the following rule:

(1) accept with a probability $p = 1$, if the new entropy E' is smaller than the current entropy E

(2) accept with a probability $p < 1$, if $E' > E$.

During search, proposals are made K times, and the trained ANNs are stored in order to be finally returned as an ensemble by the algorithm.

Algorithm 1 Ensemble of ANN-based Discriminant Functions

Input: $[X, T]$, K, v_m, m_{min}, m_{max}
$ENN = cell(K)$ ▷ initialise an ensemble of ANNs
$m \sim U(m_{min}, m_{max})$ ▷ draw a number of components
$ANN = train([X, T], m)$ ▷ train an ANN with m components
$Z = sim(ANN, X)$ ▷ ANN output
$E = f(Z, T)$ ▷ Entropy
for $k \in \{1, \ldots, K\}$ **do**
    $a = U(v_m)$
    $m' \in \{m - a, m + a\}$ ▷ proposal
    $ANN' = train([X, T], m')$ ▷ train an ANN with m' components
    $Z' = sim(ANN', X)$ ▷ output of ANN'
    $E' = f(Z', T)$
    $r = \exp(E - E')$
    if $U(0, 1) < r$ **then** ▷ accept the proposal
        $m = m'$
        $E = E'$
        $ANN = ANN'$
    end if
    $ENN\{i\} = ANN$
end for
return ENN

4 Experiments

4.1 Face Recognition Benchmark

In our experiments we use the Yale Extended B database which includes 2280 images of 38 individuals, represented by 60 images, [4]. The pose and the position of the face in the image are kept constant. However, illumination conditions can significantly vary from the bright frontal illumination to the dark images, causing shadows on large part of face.

4.2 CNN setup

For our experiments CNNs have been build with the following parameters.

(1) The CNNs have been made capable to slightly change the input data by adding Gaussian noise in order to moderate the network overfitting.

(2) The first convolution layer has been designed with 64 filters.

(3) The batch normalization has been used in order to limit the range of hidden unit outputs. Padding has enhanced the process by keeping the outputs within given boundaries. This procedure is applied at the next two layers.

(4) After the first three layers, Maximum pooling is set to be (5, 5) for the feature maps. This procedure continues for the next four layers.

(5) At the final convolutional layers, flattening parameters are added in order to transform the final feature maps into a 1-dimensional vector. This allows the fully connected layers to be applied to the convolutional layers.

(6) For optimisation `Adam` algorithm has been used for averaging over the gradient second moments. Root Mean Square Propagation has been used to adjust the network weights based on the average first moments.

(7) The loss function was the cross entropy.

4.3 Pairwise ANN Setup

The Random Search Algorithm described in Section 3.2 has been applied to the benchmark problem with the following parameters:

(i) The ensemble size has been set in the range $3 \leq K \leq 7$. The training time is linearly depended on K.

(ii) The proposal deviation has been set to range $0 \leq v_m \leq 4$: setting a small v_m limits the search area, and so can reduce ensemble diversity, whilst a greater v_m reduces the acceptance rate. However a small acceptance rate is critical for the efficiency of the search strategy, see e.g. [7].

(iii) The minimal and maximal numbers of principal components have been set $m_{min} = 20$ and $m_{max} = 300$, respectively.

4.4 Performance Comparison

For comparative experiments on the benchmark problem, we run the multi-class (MANN), CNN, and the proposed Pairwise (PANN). The multi-class ANN

has been trained by the scaled conjugate gradient back-propagation method which does not require large memory when training data are large; in our case $m_{max} = 300$ components.

In our experiments, a MANN with the number of hidden neurons around 200 and 150 inputs has provided the maximal performance. A learning rate, which is another optimisation parameter, has been set to be 0.1.

For recognition of $c = 38$ persons, the proposed PANN has 435 ANN-based discriminant function fi/j. Each of these ANNs is trained to discriminate 2-class problems which are represented by $2n_1(k - 1/k) = 96$ samples, given the number of images per person, $n_1 = 60$, and $k = 5$-fold cross-validation. It has been found that the maximal performance is achieved with one hidden neuron in 2-class ANNs: such ANNs have learnt quickly.

Table 1 shows the performances of the MANN, CNN, and proposed PANN within 5-fold cross-validation. Both the CNN and PANN significantly outperform the MANN, whilst the performances of the CNN and PANN are competitive.

Table 1. Performances of the Multi-class ANN, CNN, and proposed Pairwise ANN.

MANN,%	CNN,%	PANN,%
85.2 ± 2.5	97.2 ± 1.9	97.5 ± 2.3

5 Discussion and Conclusion

There are many factors which make face recognition extremely difficult. Multi-class ANNs which employ a conventional architecture including input, hidden, and output layers have a limited ability to solve the recognition problems. The difficulties increase with the number of persons involved in the recognition. To provide the maximal performance, multi-class ANNs require to find a proper structure including the sufficient numbers of inputs and hidden neurons. The back-propagation learning rate must be also optimised. In practice these parameters are estimated *ad-hoc*, by running massive experiments.

In this paper we analysed the potential of recent convolution networks (CNN) and implemented a typical CNN structure for experiments. We have found that settings for CNN are critical for achieving the maximal recognition performance. Massive computations are required for the optimisation.

An efficient approach to solving the multi-class recognition problems is a pairwise architecture including 2-class ANNs trained to discriminate pairwise patterns. Although the number of pairwise discriminators increases exponentially with the number of classes, the network settings become significantly simpler than that for both multi-class ANN and CNNs. The decision boundary between 2 classes becomes simple, and so an ANN can quickly learn to recognise a given pair of classes.

In our comparative experiments on the Yale face recognition benchmark data, the proposed pairwise ANN has demonstrated the performance competitive with that provided by the conventional CNN. However, our approach requires a significantly less number of *ad-hoc* parameters in order to maximise the performance. It is also important that the proposed ANN does not require massive computations such as required by CNNs.

Acknowledgements. The authors would like to thank Dr Livija Jakaite, a member of the supervisory team at the School of Computer Science of University of Bedfordshire, for useful and constructive comments.

References

1. Bailey, T.C., Everson, R.M., Fieldsend, J.E., Krzanowski, W.J., Partridge, D., Schetinin, V.: Representing classifier confidence in the safety critical domain: an illustration from mortality prediction in trauma cases. Neural Computing and Applications **16**(1), 1–10 (2007)
2. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. IEEE Transactions on Neural Networks **13**(6), 1450–1464 (2002)
3. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7), 711–720 (1997)
4. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(6), 643–660 (2001)
5. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. The Annals of Statistics **26**(2), 451–471 (1998)
6. Jain, A., Ross, A.A., Nandakumar, K.: Introduction to Biometrics, 2 edn. Springer-Verlag London (2011)
7. Jakaite, L., Schetinin, V.: Feature selection for bayesian evaluation of trauma death risk. In: 14^{th} Nordic-Baltic Conference on Biomedical Engineering and Medical Physics: NBC 2008 Riga, Latvia, pp. 123–126. Springer Berlin Heidelberg (2008)
8. Jakaite, L., Schetinin, V., Maple, C.: Bayesian assessment of newborn brain maturity from two-channel sleep electroencephalograms. Computational and Mathematical Methods in Medicine pp. 1–7 (2012)
9. Jakaite, L., Schetinin, V., Schult, J.: Feature extraction from electroencephalograms for bayesian assessment of newborn brain maturity. In: 2011 24th International Symposium on Computer-Based Medical Systems (CBMS), pp. 1–6 (2011)
10. Kirby, M., Sirovich, L.: Application of the karhunen-loeve procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(1), 103–108 (1990)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems pp. 1097–1105 (2012)
12. Li, S.Z., Jain, A.: Handbook of Face Recognition, 2 edn. Springer-Verlag London (2011)
13. Moghaddam, B., Jebara, T., Pentland, A.: Bayesian face recognition. Pattern Recognition **33**(11), 1771 – 1782 (2000)

14. Moon, H., Phillips, P.J.: Computational and performance aspects of pca-based face-recognition algorithms. Perception **30**(3), 303–321 (2001)
15. Nebauer, C.: Evaluation of convolutional neural networks for visual recognition. IEEE Transactions on Neural Networks **9**(4), 685–696 (1998)
16. Nyah, N., Jakaite, L., Schetinin, V., Sant, P., Aggoun, A.: Learning polynomial neural networks of a near-optimal connectivity for detecting abnormal patterns in biometric data. In: 2016 SAI Computing Conference (SAI), pp. 409–413 (2016)
17. Phung, S.L., Bouzerdoum, A.: A pyramidal neural network for visual pattern recognition. IEEE Transactions on Neural Networks **18**(2), 329–343 (2007)
18. Price, D., Knerr, S., Personnaz, L., Dreyfus, G.: Pairwise neural network classifiers with probabilistic outputs. Neural Information Processing Systems pp. 1109–1116 (1994)
19. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: IEEE Workshop on Applications of Computer Vision, pp. 138–142 (1994)
20. Schetinin, V., Jakaite, L.: Classification of newborn EEG maturity with Bayesian averaging over decision trees. Expert Systems with Applications **39**(10), 9340–9347 (2012)
21. Schetinin, V., Jakaite, L.: Extraction of features from sleep EEG for Bayesian assessment of brain development. PLOS ONE **12**(3), 1–13 (2017)
22. Schetinin, V., Jakaite, L., Jakaitis, J., Krzanowski, W.: Bayesian decision trees for predicting survival of patients: A study on the US national trauma data bank. Computer Methods and Programs in Biomedicine **111**(3), 602 – 612 (2013)
23. Schetinin, V., Jakaite, L., Nyah, N., Novakovic, D., Krzanowski, W.: Feature extraction with GMDH-type neural networks for EEG-based person identification. International Journal of Neural Systems (2018)
24. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61**, 85–117 (2015)
25. Schönborn, S., Egger, B., Morel-Forster, A., Vetter, T.: Markov chain monte carlo for automated face image analysis. International Journal of Computer Vision **123**(2), 160–183 (2017)
26. Singh, R., Vatsa, M., Noore, A., Singh, S.K.: Age transformation for improving face recognition performance. In: A. Ghosh, R.K. De, S.K. Pal (eds.) Pattern Recognition and Machine Intelligence, pp. 576–583. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
27. Soares, S., Antunes, C.H., Arajo, R.: Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development. Neurocomputing **121**, 498 – 511 (2013). Advances in Artificial Neural Networks
28. Tivive, F., Bouzerdoum, A.: A face detection system using shunting inhibitory convolutional neural networks. In: IEEE International Joint Conference on Neural Networks, pp. 2571–2575 (2004)
29. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991)
30. Uglov, J., Jakaite, L., Schetinin, V., Maple, C.: Comparing robustness of pairwise and multiclass neural-network systems for face recognition. EURASIP Journal on Advances in Signal Processing (2008)
31. Valenti, R., Sebe, N., Gevers, T., Cohen, I.: Machine learning techniques for face analysis. In: Anonymous Machine learning techniques, pp. 159–187 (2008)
32. Yawichai, K., Kitjaidure, Y.: Multiview invariant shape recognition based on neural networks. In: 3rd IEEE Conference on Industrial Electronics and Applications, 3, pp. 1538–1542 (2008)