Maximally Compact and Separated Features with Regular Polytope Networks

Federico Pernici, Matteo Bruni, Claudio Baecchi and Alberto Del Bimbo
MICC – Media Integration and Communication Center
University of Florence – Italy
{federico.pernici, matteo.bruni, claudio.baecchi, alberto.delbimbo}@unifi.it

Abstract

Convolutional Neural Networks (CNNs) trained with the Softmax loss are widely used classification models for several vision tasks. Typically, a learnable transformation (i.e. the classifier) is placed at the end of such models returning class scores that are further normalized into probabilities by Softmax. This learnable transformation has a fundamental role in determining the network internal feature representation.

In this work we show how to extract from CNNs features with the properties of maximum *inter-class separability and* maximum *intra-class compactness by setting the parameters of the classifier transformation as not trainable (i.e. fixed). We obtain features similar to what can be obtained with the well-known "Center Loss" [1] and other similar approaches but with several practical advantages including maximal exploitation of the available feature space representation, reduction in the number of network parameters, no need to use other auxiliary losses besides the Softmax.*

Our approach unifies and generalizes into a common approach two apparently different classes of methods regarding: discriminative features, pioneered by the Center Loss [1] and fixed classifiers, firstly evaluated in [2].

Preliminary qualitative experimental results provide some insight on the potentialities of our combined strategy.

1. Introduction

Convolutional Neural Networks (CNNs) together with the Softmax loss have achieved remarkable successes in computer vision, improving the state of the art in image classification tasks [3, 4, 5, 6]. In classification all the possible categories of the test samples are also present in the training set and the predicted labels determine the performance. As a result, the Softmax with Cross Entropy loss is widely adopted by many classification approaches due to its simplicity, good performance and probabilistic interpretation. In other applications like face recognition [7] or human body reidentification [8] test samples are not known in advance and recognition at test time is performed according to learned features based on their distance.

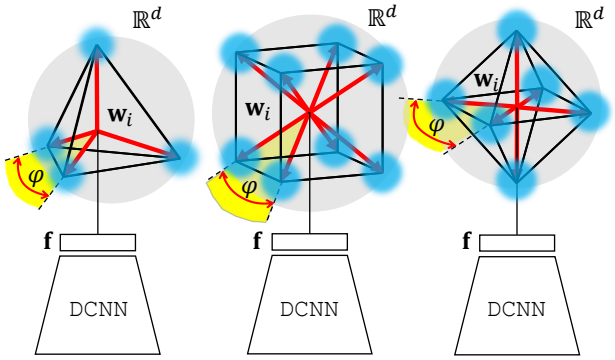

Figure 1: Margin Regular Polytope Networks (Margin-RePoNets). Features with *maximal* inter-class separability and intra-class compactness are shown (light blue). These are determined combining fixed classifiers derived from regular polytopes [9] with a recently developed margin loss [10]. Maximal features separation is obtained by setting the classifier weights $\mathbf{w}_i$ according to values following the symmetrical of configuration regular polytopes (red). Maximal compactness is obtained by setting the margin between the features at the maximum allowed (i.e. φ).

The underlying assumption in this learning scenario is that images of the same identity (person) are expected to be closer in the representation space, while different identities are expected to be far apart. Or equivalently, the learned features having low intra-class distance and large inter-class distance are successful at modeling novel unseen identities and for this reason such features are typically defined "discriminative". Specifically, the Center Loss, firstly proposed in [1], has been proved to be an effective method to compute discriminative features. The method learns a center determined as the average of features belonging to the same class. During training, the centers are updated by minimiz-

ing the distances between the deep features and their corresponding class centers. The CNN is trained under the joint supervision of the Softmax loss and the Center Loss by balancing the two supervision signals. Intuitively, the Softmax loss forces the deep features of different classes to be separable while the Center Loss attracts the features of the same class to their centers achieving compactness.

Despite its usefulness, the Center Loss has some limitations: the feature centers are extra parameters stored outside the network that are not jointly optimized with the network parameters. Indeed, they are updated with an autoregressive mean estimator that tracks the underlying representation changes at each step. Moreover, when a large number of classes must be learned, mini-batches do not provide enough samples for a correct estimation of the mean. Center Loss also requires a balancing between the two supervision losses which typically requires a search over the balancing hyper-parameter.

Some works have successfully addressed all the issues described above importing intra-class feature compactness directly into the Softmax loss. This class of methods, including [11, 12, 10, 13, 14], avoids the need of an auxiliary loss (as in the Center Loss) with the possibility of including a margin between the class decision boundaries, all in a single Softmax loss.

Other successful works follow a nearly opposite strategy by removing the final classification layer and learn directly a distance evaluated on image pairs or image triplets as shown in [15] and in [16] respectively. Despite the performance results, carefully designed pair and triplet selection is required to avoid slow convergence and instability.

Except for few recent cases [17, 18, 9] inter-class separability and compactness are always enforced in a local manner without considering global inter-class separability and intra-class compactness. For this purpose, the work [18] uses an auxiliary loss for enforcing global separability. The work [17] use an auxiliary loss similar to [18] for enforcing global separability and a further margin loss to enforce compactness. The work [9] uses a fixed classifier in which the parameters of the final transformation implementing the classifier are *not* subjected to learning and are set with values taken from coordinate vertices of a regular polytope. This avoids optimizing for maximal separation as in [17] and [18] since regular polytopes naturally provide distributed vertices (i.e. the classifier weights) at equal angles maximizing the available space.

In this paper we address *all* those limitations including global inter-class separability and compactness in a maximal sense without the need of any auxiliary loss. This is achieved by exploiting the Regular Polytope fixed classifiers (RePoNets) proposed in [9] and improving their feature compactness according to the additive angular margin described in [10]. As illustrated in Fig. 1, the advantage of the proposed combination is the capability of generating global maximally separated and compact features (shown in light blue) angularly centered around the vertices of polytopes (i.e. the classifier fixed weights shown in red). The same figure further illustrates the three basic types of features that can be learned. Although, there are infinite regular polygons in $\mathbb{R}^2$ and 5 regular polyedra in $\mathbb{R}^3$, there are only three regular polytopes in $\mathbb{R}^d$ with $d \geq 5$, namely the d-Simplex, the d-Cube and the d-Orthoplex.

In particular, the angle φ subtended between a class weight and its connected class weights is constant and maximizes inter-class separability in the available space. The angle φ is further exploited to obtain the maximal compactness by setting the angular margin between the features to φ (i.e. the maximum allowed margin). The advantage of our formulation is that the margin is no longer an hyperparameter that have to be searched since it is obtained from a closed form solution.

2. Related Work

Fixed Classifiers. Empirical evidence, reported in [19], firstly shows that a CNN with a fixed classification layer does not worsen the performance on the CIFAR10 dataset. A recent paper [2] explores in more detail the idea of excluding the classification parameters from learning. The work shows that a fixed classifier causes little or no reduction in classification performance for common datasets (including ImageNet) while allowing a noticeable reduction in trainable parameters, especially when the number of classes is large. Setting the last layer as not trainable also reduces the computational complexity for training as well as the communication cost in distributed learning. The described approach sets the classifier with the coordinate vertices of orthogonal vectors taken from the columns of the Hadamard[1] matrix. Although the work uses a fixed classifier, the properties of the generated features are not explored. A major limitation of this method is that, when the number of classes is greater than the dimension of the feature space, it is not possible to have mutually orthogonal columns and therefore some of the classes are constrained to lie in a common subspace causing a reduction in classification performance.

Recently [9] improves in this regard showing that a novel set of unique directions taken from regular polytopes overcomes the limitations of the Hadamard matrix. The work further shows that the generated features are stationary at training time and coincide with the equiangular spaced vertices of the polytope. Being evaluated for classification the method does not enforce feature compactness. We extend this work by adding recent approaches to explicitly enforce

[1]The Hadamard matrix is a square matrix whose entries are either +1 or 1 and whose rows are mutually orthogonal.

feature compactness by constraining features to lie on a hypersphere [12] and to have a margin between other features [10].

Fixed classifiers have been recently used also for not discriminative purposes. The work [20] trains a neural network in which the last layer has fixed parameters with predefined points of a hyper-sphere (i.e. a spherical lattice). The work aims at learning a function to build an index that maps real-valued vectors to a uniform distribution over a d-dimensional sphere to preserve the neighborhood structure in the input space while best covering the output space. The learned function is used to make high-dimensional indexing more accurate.

Softmax Angular Optimization. Some papers train DCNNs by direct angle optimization [14, 21, 12]. From a semantic point of view, the angle encodes the required discriminative information for class recognition. The wider the angles the better the classes are separated from each other and, accordingly, their representation is more discriminative. The common idea of these works is that of constraining the features and/or the classifier weights to be unit normalized. The works [22], [23] and [12] normalize both features and weights, while the work [11] normalizes the features only and [14] normalizes the weights only. Specifically, [11] also proposes adding a scale parameter after feature normalization based on the property that increasing the norm of samples can decrease the Softmax loss [24].

From a statistical point of view, normalizing weights and features is equivalent to considering features distributed on the unit hypersphere according to the von Mises-Fisher distribution [23] with a common concentration parameter (i.e. features of each class have the same compactness). Under this model each class weight represents the mean of its corresponding features and the scalar factor (i.e. the concentration parameter) is inversely proportional to their standard deviations. Several methods implicitly follow this statistical interpretation in which the weights act as a summarizer or as parameterized prototype of the features of each class [12, 14, 25, 26, 27]. Eventually, as conjectured in [12] if all classes are well-separated, they will roughly correspond to the means of features in each class.

In [9] the fixed classifiers based on regular polytopes produce features exactly centered around their fixed weights as the training process advances. The work globally imposes the largest angular distances between the class features before starting the learning process without an explicit optimization of the classifier or the requirement of an auxiliary loss as in [18] and [17]. The works [18, 17] add a regularization loss to specifically force the classifier weights during training to be far from each other in a global manner. These works including [9] draw inspiration from a well-known problem in physics – the Thomson problem [28] – where given K charges confined to the surface of a sphere, one seeks to find an arrangement of the charges which minimizes the total electrostatic energy. Electrostatic force repels charges each other inversely proportional to their mutual distance. In [18] and [17] global equiangular features are obtained by adding to the standard categorical Cross-Entropy loss a further loss inspired by the Thomson problem while [9] builds directly an arrangement for global separability and compactness by considering that minimal energies are often concomitant with special geometric configurations of charges that recall the geometry of regular polytopes in high dimensional spaces [29].

3. Regular Polytope Networks with Additive Angular Margin Loss

In Neural Networks the representation for an input sample is the feature vector $\mathbf{f}$ generated by the penultimate layer, while the last layer (i.e. the classifier) outputs score values according to the inner product as:

$$z_i = \mathbf{w}_i^\top \cdot \mathbf{f} \tag{1}$$

for each class i, where $\mathbf{w}_i$ is the weight vector of the classifier for the class i. In the final loss, the scores are further normalized into probabilities via the Softmax function.

Since the values of z_i can be also expressed as $z_i = \mathbf{w}_i^\top \cdot \mathbf{f} = \|\mathbf{w}_i\| \, \|\mathbf{f}\| \cos(\theta)$, where θ is the angle between $\mathbf{w}_i$ and $\mathbf{f}$, the score for the correct label with respect to the other labels is obtained by optimizing $\|\mathbf{w}_i\|$, $\|\mathbf{f}\|$ and θ. According to this, feature vector directions and weight vector directions align simultaneously with each other at training time so that their average angle is made as small as possible. In [9] it is shown that if classifier weights are excluded from learning, they can be regarded as fixed angular references to which features align. In particular, if the fixed weights are derived from the three regular polytopes available in $\mathbb{R}^d$ with $d \geq 5$, then their symmetry creates angular references to which class features centrally align. More formally, let $\mathbf{X} = \{(x_i, y_i)\}_{i=1}^N$ be the training set containing N samples, where x_i is the image input to the CNN and $y_i \in \{1, 2, \cdots, K\}$ is the label of the class that supervises the output of the DCNN. Then, the Cross Entropy loss can be written as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{e^{\mathbf{w}_{y_i}^\top \mathbf{f}_i + \mathbf{b}_{y_i}}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^\top \mathbf{f}_i + \mathbf{b}_j}} \right), \tag{2}$$

where $\mathbf{W} = \{\mathbf{w}_j\}_{j=1}^K$ are the fixed classifier weight vectors for the K classes. Only three polytopes exist in every dimensionality and are: the d-Simplex, the d-Orthoplex and the d-Cube from which three classifiers can be defined as follow:

$$\mathbf{W}_s = \left\{ e_1, e_2, \ldots, e_{d-1}, \alpha \sum_{i=1}^{d-1} e_i \right\}, \tag{3}$$

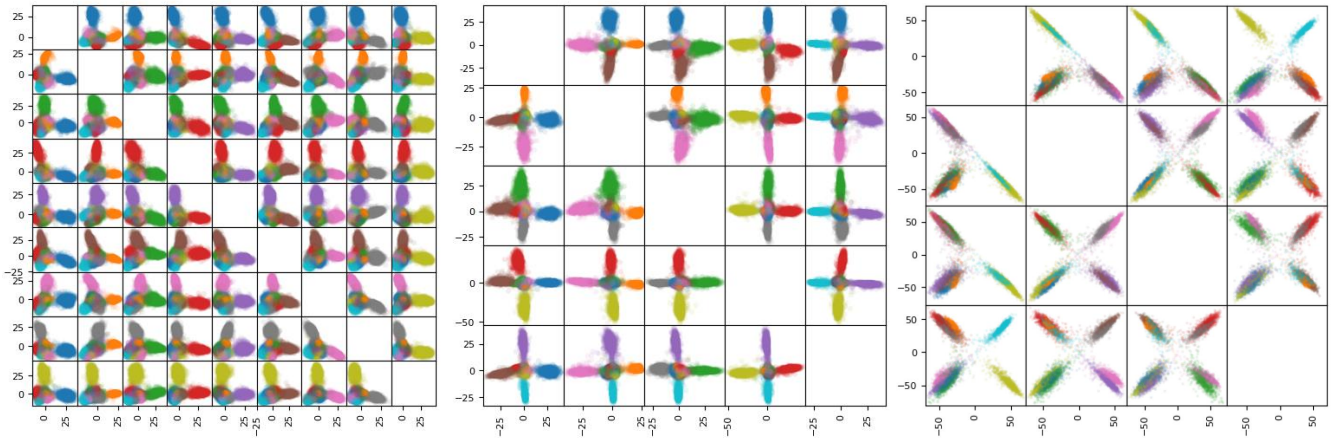

Figure 2: The distribution of features learned from the MNIST dataset using the RePoNet classifiers. Features are shown (from left to right) with a scatter plot matrix for the d-Simplex, d-Orthoplex and d-Cube classifier respectively. It can be noticed that features are distributed following the symmetric vertex configuration of polytopes. Although features are maximally separated, their compactness is limited.

$$\mathbf{W}_o = \{\pm e_1, \pm e_2, \ldots, \pm e_d\}, \tag{4}$$

$$\mathbf{W}_c = \left\{ \mathbf{w} \in \mathbb{R}^d : \left[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}} \right]^d \right\}, \tag{5}$$

where $\alpha = \frac{1-\sqrt{d+1}}{d}$ in Eq.3 and e_i with $i \in \{1, 2, \ldots, d-1\}$ in Eqs.3 and 4 denotes the standard basis in $\mathbb{R}^{d-1}$. The final weights in Eq.3 are further shifted about the centroid, the other two are already centered around the origin. Such sets of weights represent the vertices of the generalization of the tetrahedron, octahedron and cube respectively, to arbitrary dimension d. The weights are further unit normalized ($\hat{\mathbf{w}}_j = \frac{\mathbf{w}_j}{\|\mathbf{w}_j\|}$) and the biases are set to zero ($\mathbf{b}_j = 0$). According to this, Eq. 2 simplifies to:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{e^{\hat{\mathbf{w}}_{y_i}^\top \mathbf{f}_i}}{\sum_{j=1}^{K} e^{\hat{\mathbf{w}}_j^\top \hat{\mathbf{f}}_i}} \right). \tag{6}$$

Although, Eq. 6 directly optimizes for small angles, only partial intra-class compactness can be enforced. Fig.2 shows (from left to right) the distribution of features learned from the MNIST dataset with the three different classifiers. The features are displayed as a collection of points, each having the activation of one feature coordinate determining the position on the horizontal axis and the value of the other feature coordinate activation determining the position on the vertical axis. All the pairwise scatter plots of the feature activation coordinates are shown and feature classes are color coded. The size of the scatter plot matrices follows the size of the feature dimensionality d of each fixed classifier which can be determined according to the number of classes K as:

$$d = K - 1, \qquad d = \lceil \log_2(K) \rceil, \qquad d = \left\lceil \frac{K}{2} \right\rceil, \tag{7}$$

respectively. The scatter plot matrices therefore result in the following dimensions: 9×9, 5×5 and 4×4 respectively. As evidenced from the figure, the features follow the symmetric and maximally separated vertex configurations of their corresponding polytopes. This is due to the fact that each single pairwise scatter plot is basically a parallel projection onto the planes defined by pairs of multidimensional axes. According to this, features assume a $\curlywedge$, $+$, and $\times$ shaped configuration for the d-Simplex, d-Orthoplex and d-Cube respectively. Although maximal separation is achieved, the intra-class average distance is large and therefore not well suited for recognition purposes.

The plotted features are obtained training the so called LeNet++ architecture [1]. The network is a modification of the LeNet architecture [30] to a deeper and wider network including parametric rectifier linear units (pReLU) [31]. The network is learned using the Adam optimizer [32] with a learning rate of 0.0005, the convolutional parameters are initialized following [33] and the mini-batch size is 512.

To improve compactness keeping the global maximal feature separation we follow [13, 12] normalizing the features and multiplying them by a scalar κ: $\hat{\mathbf{f}}_i = \frac{\mathbf{f}_i}{\|\mathbf{f}_i\|}\kappa$. The loss in Eq.2 can be therefore rewritten as:

$$\begin{aligned} \mathcal{L} &= -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{e^{\kappa \hat{\mathbf{w}}_{y_i}^\top \hat{\mathbf{f}}_i}}{\sum_{j=1}^{K} e^{\kappa \hat{\mathbf{w}}_j^\top \hat{\mathbf{f}}_i}} \right) \\ &= -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{e^{\kappa \cos(\theta_{y_i})}}{\sum_{j=1}^{K} e^{\kappa \cos(\theta_j)}} \right) \end{aligned} \tag{8}$$

The equation above minimizes the angle θ_{y_i} between the fixed weight corresponding to the label y_i and its associated feature. The equation can be interpreted as if features are realizations from a set of K von Mises-Fisher distributions having a common concentration parameter κ. Under this parameterization $\hat{\mathbf{w}}$ is the mean direction on the hypersphere and κ is the concentration parameter. The greater the value of κ the higher the concentration of the distribution around the mean direction $\hat{\mathbf{w}}$ and the more compact the features. This value has already been discussed sufficiently in several previous works [12, 11]. In this paper, we directly fixed it to 30 and will not discuss its effect anymore.

To obtain maximal compactness the additive angular margin loss described in [10] is exploited. According to this, Eq.8 becomes:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{\kappa\cos(\theta_{y_i}+m)}}{e^{\kappa\cos(\theta_{y_i}+m)}+\sum\limits_{\substack{j=1\\ j\neq y_i}}^{n} e^{\kappa\cos(\theta_j)}}\right), \tag{9}$$

where the scalar value m is an angle in the normalized hypersphere introducing a margin between class decision boundaries. The loss of Eq. 9 together with the fixed classifier weights of Eqs. 3, 4, 5 allows learning discriminative features without using any auxiliary loss other than the Softmax.

The advantage of our formulation is that m is no longer an hyperparameter that have to be searched. Indeed, the loss above when used with RePoNet classifiers is completely interpretable and the margin m can be set according to the angle φ subtended between a class weight and its connected class weights as illustrated in Fig.1. For each of the three RePoNet fixed classifiers the angle φ can be analytically determined as [9]:

$$\varphi_s = \arccos\left(-\frac{1}{d}\right), \tag{10}$$

$$\varphi_o = \frac{\pi}{2}, \tag{11}$$

$$\varphi_c = \arccos\left(\frac{d-2}{d}\right), \tag{12}$$

respectively, where d is the feature space dimension size. Fig. 3 shows the effect of setting:

$$m = \varphi.$$

In the figure we draw a schematic 2D diagram to show the effect of the margin m on pushing the class decision boundary to achieve feature compactness. In the standard case of a learnable classifier, as shown in Fig. 3 *(left)*, the value φ is not known in advance, it varies from class to class and features are not guaranteed to distribute angularly centered around their corresponding weights. Therefore, m cannot be set in an interpretable way. Contrarily, in the case proposed in this paper and shown in Fig. 3 *(right)*, the value φ is constant and known in advance, therefore by setting $m = \varphi$, the class decision boundaries are maximally pushed to compact features around their fixed weights. This because the Softmax boundary (from which the margin is added) is exactly in between the two weights $\mathbf{w}_1$ and $\mathbf{w}_2$. According to this, the features generated by the proposed method are not only *maximally separated* but also *maximally compact* (i.e. maximally discriminative).

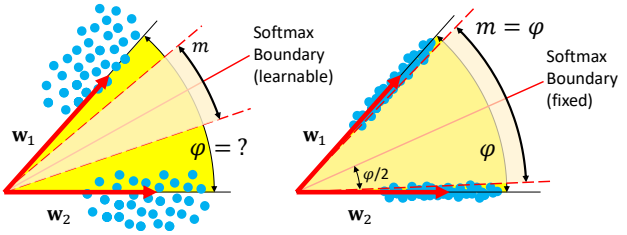

Figure 3: Maximally compact feature learning with RePoNet fixed classifiers and the angular margin loss. *Left*: In a standard learnable classifier the decision boundaries (dashed lines) defined by the angular margin m do not push features to their respective weights uniformly (red arrows). *Right*: In RePoNet classifiers the margin can be analytically determined ($m = \varphi$) so that the decision boundaries maximally push the features closer to their respective fixed weight.

4. Exploratory Results

Experiments are conducted with the well-known MNIST and EMNIST [34] datasets. MNIST contains $50,000$ training images and $10,000$ test images. The images are in grayscale and the size of each image is 28×28 pixels. There are 10 possible classes of digits. The EMNIST dataset (balanced split) holds $112,800$ training images, $18,800$ test images and has 47 classes including lower/upper case letters and digits.

Fig. 4 shows a visual comparison between the features generated by the RePoNet fixed classifiers (*left column*) and by a standard CNN baseline with learnable classifiers (*right column*). Both approaches are trained according to the loss of Eq. 9 and have exactly the same architecture, training settings and embedding feature dimension used in Fig. 2. Results are presented with a scatter plot matrix. Although the two methods achieve substantially the same classification accuracy (i.e. 99.45% and 99.47% respectively), it can be noticed that the learned features are different. Specifically, Margin-RePoNet follows the exact configuration geometry of their related polytopes. Features follow very precisely their relative $\curlywedge$, $+$, and $\times$ shapes therefore achieving maximal separability. The standard baselines with learn-

able classifiers (Fig. 4 *left column*) achieve good but non maximal separation between features. However, as the embedding dimension decreases, as in Fig. 4(c), the separation worsens.

This effect is particularly evident in more difficult datasets. Fig.5 shows the same visual comparison using the EMNIST dataset where some of the 47 classes are difficult to be correctly classified due to their inherent ambiguity. Fig. 5 shows the scatter plot matrix of the d-Cube classifier (*left*) compared with its learnable classifier baseline (*right*) in dimension $d = 6$. Although also in this case they both achieved the same classification accuracy (i.e. 88.31% and 88.39%), the features learned by the baseline are neither well separated nor compact.

Finally, in Fig. 6 we show the L_2 normalized features (typically used in recognition) of both the training (*top*) and test set (*bottom*) for the same experiment shown in Fig. 5. Class features in this case correctly follow the vertices of the six-dimensional hypercube since all the parallel projections defined by each pairwise scatter plot result in the same unit square centered at the origin.

5. Conclusion

We have shown how to extract features from Convolutional Neural Networks with the desirable properties of maximal separation and maximal compactness in a global sense. We used a set of fixed classifiers based on regular polytopes and the additive angular margin loss. The proposed method is very simple to implement and preliminary exploratory results are promising.

Further implications may be expected in large face recognition datasets with thousands of classes (as in [7]) to obtain maximally discriminative features with a significant reduction in: the number of model parameters, the feature size and the hyperparameters to be searched.

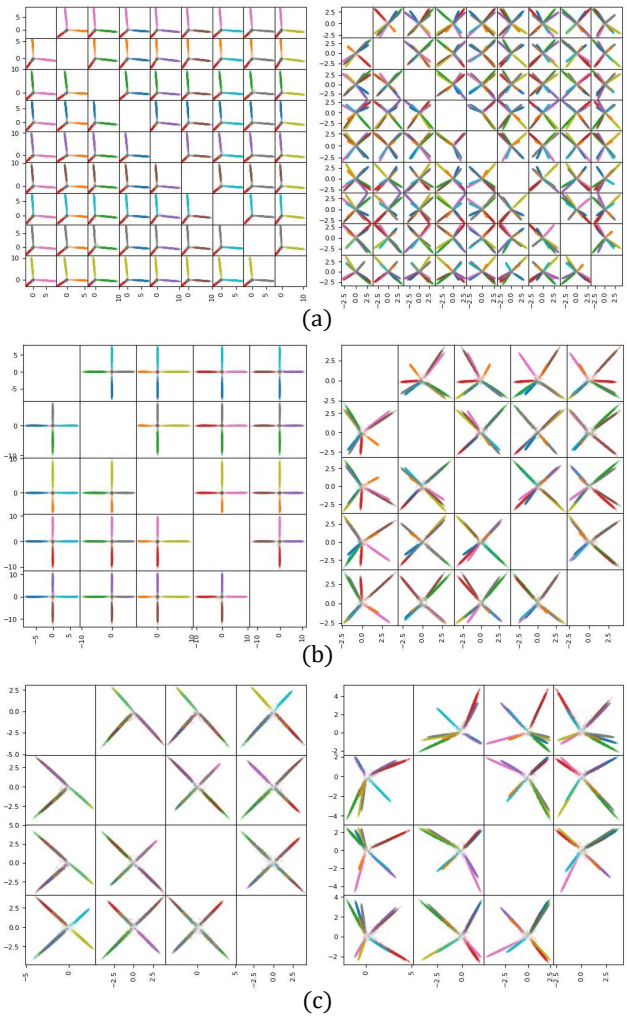

Figure 4: The distribution of MNIST learned features using the proposed method (*Left*) and learned using a standard trainable classifier (*Right*). The scatter plot highlights the maximal separability and compactness of the extracted features for the (a) d-Simplex, (b) d-Orthoplex and (c) d-Cube classifiers. Class features are color coded. As the feature space dimension decreases standard baselines have difficulty in obtaining inter-class separation.

References

[1] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 1, 4

[2] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[5] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, July 2017. 1

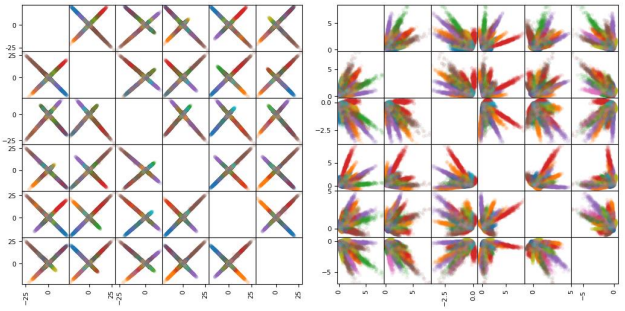

Figure 5: The distribution of EMNIST (balanced split) learned features. *Left*: Features learned using the d-Cube fixed classifier with the additive angular margin loss. *Right*: Features learned using a standard trainable classifier with the additive angular margin loss. In both cases the feature dimension is 6 and the classification accuracy is comparable. Maximal separability and compactness are evident in our approach.

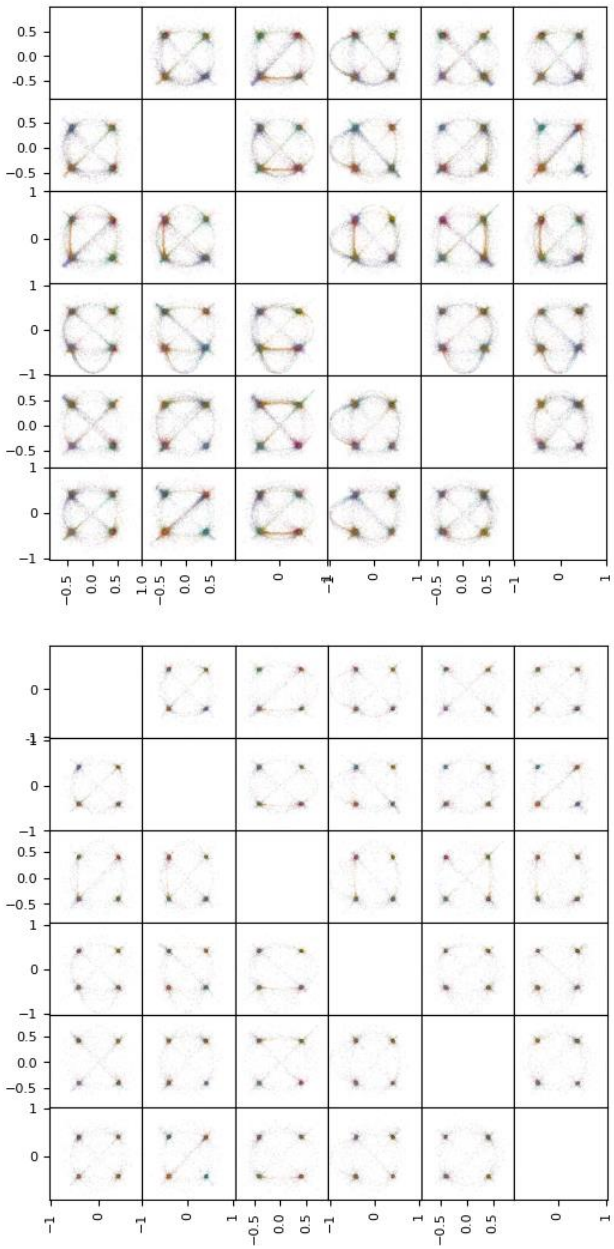

Figure 6: The distribution of the EMNIST *normalized* learned features shown in 5 *(Left)*. (*Top*) training-set. (*Bottom*) test-set (best viewed in electronic version).

[6] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018. 1, 6

[8] Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1062–1071, 2018. 1

[9] Federico Pernici, Matteo Bruni, Claudio Baecchi, and Alberto Del Bimbo. Fix your features: Stationary and maximally discriminative embeddings using regular polytope (fixed classifier) networks. *arXiv preprint arXiv:1902.10441*, 2019. 1, 2, 3, 5

[10] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 5

[11] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 2, 3, 5

[12] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: l 2 hypersphere embedding for face verification. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1041–1049. ACM, 2017. 2, 3, 4, 5

[13] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face

recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 2, 4

[14] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 2, 3

[15] S Chopra, R Hadsell, and Y LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 2

[16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2

[17] Ming-Ming Cheng Kai Zhao, Jingyi Xu. Regularface: Deep face recognition via exclusive regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3

[18] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. *NIPS*, 2018. 2, 3

[19] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *International Conference on Learning Representations (ICLR)*, 2017. 2

[20] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *International Conference on Learning Representations (ICLR)*, 2019. 3

[21] Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M. Rehg, and Le Song. Decoupled networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[22] Yu Liu, Hongyang Li, and Xiaogang Wang. Learning deep features via congenerous cosine loss for person recognition. *arXiv preprint: 1702.06890*, 2017. 3

[23] Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017. 3

[24] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Feature incay for representation regularization. *arXiv preprint arXiv:1705.10284*, 2017. 3

[25] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. 3

[26] Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[27] Zhirong Wu, Alexei A. Efros, and Stella X. Yu. Improving generalization via scalable neighborhood component analysis. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[28] Joseph John Thomson. XXIV. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(39):237–265, 1904. 3

[29] J Batle, Armen Bagdasaryan, M Abdel-Aty, and S Abdalla. Generalized thomson problem in arbitrary dimensions and non-euclidean geometries. *Physica A: Statistical Mechanics and its Applications*, 451:237–250, 2016. 3

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 4

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 4

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV*, pages 1026–1034, 2015. 4

[34] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 2921–2926, 2017. 5