Do Neural Network Weights Account for Classes Centers?
Ioannis Kansizoglou, Loukas Bampis, and Antonios Gasteratos, Senior Member, IEEE

Abstract—The exploitation of deep neural networks (DNNs) as descriptors in feature learning challenges enjoys apparent popularity over the past few years. The above tendency focuses on the development of effective loss functions that ensure both high feature discrimination among different classes, as well as low geodesic distance between the feature vectors of a given class. The vast majority of the contemporary works rely their formulation on an empirical assumption about the feature space of a network’s last hidden layer, claiming that the weight vector of a class accounts for its geometrical center in the studied space. This article at hand follows a theoretical approach and indicates that the aforementioned hypothesis is not exclusively met. This fact raises stability issues regarding the training procedure of a DNN, as shown in our experimental study. Consequently, a specific symmetry is proposed and studied both analytically and empirically that satisfies the above assumption, addressing the established convergence issues. More specifically, the aforementioned symmetry suggests that all weight vectors are unit, coplanar, and their vector summation equals zero. Such a layout is proven to ensure a more stable learning curve compared against the corresponding ones succeeded by popular models in the field of feature learning.

Index Terms—Deep neural networks (DNNs), discriminative feature learning, geometric algebra, symmetrical layer.

I. INTRODUCTION

EFFECTIVE data representation into a compact metric space occupies prominent place in the recent challenges of computer science (CS) [1], [2]. To that end, effort is concentrated on mapping each pattern of the sensory input into a space displaying specific and known metric properties through a suitably designed method. Such a method may derive either from traditional feature engineering or from a learning-based algorithm [3], [4]. In the first case, the output representation is usually obtained via a mathematical projection rule, such as principal component analysis [5] and linear discriminant analysis [6], or a specific extraction scheme making use of a set of predefined handcrafted rules [7]–[10]. In contrast, the second case employs a machine learning algorithm in order to discover salient features from raw data [11]–[13].

Due to the complex nature of real-world sensory inputs in present problems, the manual definition of descriptive features becomes increasingly unreliable, leaving no option but to engage machine learning schemes. Typical algorithms are support vector machines (SVMs) [11] and deep neural networks (DNNs) [13], [14], with DNNs forming the leading choice in feature extraction for cascade [15], [16] and fusion tasks [17], [18] given their proven efficacy over the past years. Yet, in order for a learning algorithm to ensure robust representation capacity, a set of techniques needs to be applied during its training phase. The above techniques form the main research subject of feature or representation learning [1]. The studied assessment criteria for a method are mainly focused on its capability to map similar patterns into close geodesic distance in the output metric space while keeping high distance between unlike inputs. The above properties are broadly known as high intraclass compactness and interclass discrepancy [19], where class refers to the set of all the database instances from a common label category. Hence, the mutual satisfaction of the above two criteria is incorporated into the optimization goal of the adopted learning algorithm [19], [20].

The most prevailing approaches in the field of feature learning have been developed in image retrieval problems and more specifically in the challenge of face verification [21]. The adopted convolutional neural network (CNN) architectures are trained with an enhanced version of the original softmax loss, which embodies the two representation learning criteria, by shaping a space with specific geometrical properties. Given a training step, each feature vector, formed by the activations of the last hidden layer, is forced to approximate the center of its target class. Based on this rule, an ample variety of approaches have been developed outweighing prior works in the field [19], [20], [22]. However, in order to materialize a representation for each class into a computationally effective loss function, the weight vector of each class is utilized, assuming that it coincides with the respective center. The above practice shapes a hypothesis, which is from now on referred to as $\mathcal{H}$.

The present study sheds light on the empirical supposition $\mathcal{H}$ since it is not exclusively met for arbitrary distributions of the weight vectors, introducing an implicit error in the training procedure. As it will be extensively discussed in our empirical
study, the implementation of existing works that adopt $\mathcal{H}$ led us to several convergence issues. The above refer both to the stochastic ability of a DNN to converge, as well as to the uncertainty of resulting in an acceptable performance level after the training procedure. Moving further, a specific symmetry regarding those vectors in the studied feature space is proposed, in order to satisfy $\mathcal{H}$. Eventually, we proceed with empirical findings about the impact of the added error during the training procedure of a DNN. Accordingly, the contributions of this work can be summarized as follows:

1) refutability of the common hypothesis ($\mathcal{H}$), implying the coincidence between the weight vector of a class and its corresponding geometrical center in the studied space;
2) introduction and implementation details of a symmetrical layout for the weight vectors that always satisfies $\mathcal{H}$, securing equivalence with the class center;
3) empirical study indicating the stability issues introduced by widespread methods that adopt $\mathcal{H}$ and their confrontation through the proposed symmetry;
4) comparative results against state-of-the-art feature learning techniques regarding performance and time complexity.

The remainder of this article is structured as follows. In Section II, we discuss typical approaches in the field of discriminative feature learning that denote the assumption $\mathcal{H}$, while Section III clearly states the motivation behind the current work. Then, Sections IV and V theoretically prove the inconsistency between a class’s weight vector and its geometrical center for an arbitrary distribution of the weights as well as propose a novel symmetrical layout. Section VI provides a detailed description regarding the implementation of the above symmetry. Subsequently, Section VII displays several experiments, proving the efficiency of the proposed layer and illustrating the way that the implicit error of the initial supposition affects the DNN’s convergence. In the last section, the acquired conclusions are collected, suggesting concise concepts that can be employed in representation learning tasks.

II. RELATED WORK

In this section, a comprehensive review is presented, focusing on the exploitation of DNNs in feature learning challenges. Consequently, individual attention is paid to recent trends in verification tasks as well as the theoretical background that forms the basis of the proposed methods.

A. Neural-Based Feature Learning

The original implementation of the softmax loss has been progressively proved insufficient for challenging verification tasks that require high feature discrimination. Motivated by this fact, a host of methods have been developed, proposing the enhancement of the common loss with advanced restrictive conditions. The initial step, introduced with the concept of Siamese networks, suggests the exploitation of two or more identical configurations of a DNN during the training procedure [23]. Hence, the extracted feature vectors can be compared, by feeding two or more samples, passing their similarity score to a suitable loss function under a pairwise learning procedure. In such a procedure, the common cross-entropy loss cannot be implemented, leading to the utilization of novel objective functions, with triplet [24] and contrastive loss [25] forming the most typical ones. At each training step, the objective of triplet loss refers to: 1) a distance minimization component between a specific training sample and another one of the same class (positive pair) and 2) a distance maximization component between the same sample and a sample of another class (negative pair) [26]. Aiming at an improved interclass separation, the contrastive loss compares the similarity score of a positive pair against a negative one for each training sample [25].

B. Enhanced-Softmax Loss Functions

Despite their efficiency, Siamese networks displayed two main disadvantages, i.e., increased training duration, due to pairwise learning, and opaque outputs since they do not provide a probabilistic distribution, such as the softmax loss. Thus, the research community has resorted to alternative solutions that exploit softmax loss and enhance its discrimination capacity. Accordingly, the center loss proposed the penalization of a feature vector by means of its Euclidean distance from the center of its target class [27]. During training, the center of each class was calculated and updated by the respective feature vectors of the minibatch, adding a considerable computational cost in the procedure. Two variations of center loss, i.e., the island and range loss, were developed in order to enhance discrimination performance [28], [29]. On the one hand, island loss suggested for the first time, apart from the compression of the feature vectors of the same class, the mutual digression between the centers of the classes [28]. On the other hand, range loss constitutes the first attempt to address the imbalances of training data in the challenge of face recognition [29].

In an effort to reduce the required computational complexity, the idea of applying more rigorous constraints directly in the softmax loss, rather than providing complementary losses, was introduced. On this account, large margin softmax ($L$-Softmax) encouraged both discrimination criteria in the common cross-entropy loss, by adjusting an angular margin that scales the angle between a feature and the weight vector of its target class [30]. Thereby, the weight vector comprises the reference point of a class or, equally, the desired orientation for the feature vectors. Instead of an angular constraint, $L_2$-Softmax suggested the feature vectors’ normalization, in order for them to lie on the surface of a hypersphere with configurable radius, boosting the discrimination performance [31]. Combining both angular and norm constraints, SphereFace proposed an improved version of the original softmax loss also known as $A$-Softmax [20]. SphereFace was also the first method that coped with stability issues, requiring the supervision of a simple softmax loss mainly during the initial steps of the training procedure. CosFace inserted an additive cosine margin directly to the target logits of the CNN as an attempt to improve stability [22]. Subsequently, ArcFace succeeded an even simpler and simultaneously efficient approach that adopts an additive angular margin [19].
However, all of the above methods keep the same strategy of applying an angular margin between the feature and the weight vector of the target class, assuming that this is its optimal orientation [19], [20], [22]. An entirely different approach constitutes the implementation of the angular margin astride the decision boundaries between the classes [32]. Considering the above, the reader can conclude that all existing works are exclusively based on \( \mathcal{H} \), apart from the last one proposed by us. The main argument here focuses on the empirical nature of this assumption that allows plenty of space for doubt. One can easily wonder whether there are cases that \( \mathcal{H} \) is not valid and, in such a case, how it can be prevented.

III. MOTIVATION

Keeping in mind the stated hypothesis \( \mathcal{H} \), we first present the inspiration behind the current work and formulate our established concern. For visualization purposes, we proceed with a typical analysis in a 2-D feature space \( \mathcal{F}_1 \subset \mathbb{R}^2 \), training a vanilla CNN on the MNIST database [33] using a two-unit hidden layer before the output one. The training procedure lasted for ten epochs using an SGD optimizer with a learning rate of 0.01. The produced weight vectors’ layout is assessed in Fig. 1.

More specifically, given the resulting weight vectors \( \hat{w}_i \in \mathcal{F}_1, i = 1, 2, \ldots, 10 \), and the set of possible embeddings \( \hat{e}(j) = \frac{1}{\sqrt{\pi/180}}, j \in \mathbb{N}^{<360} \) with \( \hat{e}(j) \in \mathcal{F}_1 \) in polar coordinates, we calculate the dot product output values

\[
l_i(j) = \max_i (\hat{w}_i \cdot \hat{e}(j)).
\]

Also, \( \forall j \), we keep the index \( i \) that corresponds to the maximum value, denoting the prevailing class. As proved in a previous work of ours [34], each class ends up occupying a convex angular subspace in \( \mathcal{F}_1 \), while the extreme points of \( l_i(j) \), calculated by

\[
dl_i(j)/dj = 0
\]

coincide with the orientation of the weight vectors \( \hat{w}_i \), as shown in Fig. 1(a). By further applying the common softmax activation function and keeping its maximum value, identified as

\[
s_i(j) = \max_{\hat{w}_i} \left( \sum_{k=1}^{10} e^{\hat{w}_i \cdot \hat{e}_j} \right)
\]

we produce the regarding softmax distribution, as shown in Fig. 1(b). By calculating the orientations \( j \) that account for the extreme points of \( s_i(j) \) from

\[
ds_i(j)/dj = 0
\]

we find out that it diverges from the equivalent ones from the dot product. Bearing our results in mind, the reader can understand the error introduced by the use of \( \hat{w}_i \) as the point of reference in a loss function, especially considering a space \( \mathcal{F} \) of increased dimensionality as well as the randomness of \( \hat{w}_i \) during the initialization of a training procedure.

IV. RULES IN HIGH-DIMENSIONAL SPACES

We proceed with several findings that enable the study of an arbitrary feature space \( \mathcal{F}_d \subset \mathbb{R}^{d+1}, d \in \mathbb{N}^>1 \).

Lemma 1: Given the vectors \( \vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n, n \in \mathbb{N}^>1 \) with equal norms in the 2-D plane \( P \) such that \( (\vec{b}_i, \vec{b}_{i+1}) = 2\pi/n, \forall i = 1, \ldots, n-1 \), then

\[
\sum_{i=1}^n \vec{b}_i = \vec{0}.
\]

Proof: The above is trivial to be shown through the proof by contradiction. Let us suppose the vectors \( \vec{b}_i = O\vec{V}_i \) of equal norm in \( P \), with \( i = 1, 2, \ldots, n \) and their sum \( \sum_{i=1}^n \vec{b}_i = \vec{0} \), where \( \vec{0} \) also lies in \( P \). If we rotate the plane \( P \) or equally all vectors \( \vec{b}_i \) with respect to their common origin \( O \) by an angle of \( 2\pi/n \), then their layout does not change, implying that their sum should also remain unchanged. However, \( \vec{0} \) is also rotated by \( 2\pi/n \), indicating that the initial sum changes, except for the case that \( \vec{0} = \vec{0} \). A geometrical interpretation of the above is shown in Fig. 2.
Lemma 2: Given the equation
\[
\sum_{k=0}^{n-1} \sin(x - 2k\pi/n)e^{\cos(x - 2k\pi/n)} = 0, \quad n \in \mathbb{N}^+ \tag{6}
\]
x_r = 2r\pi/n, r \in \mathbb{N}^{+n} are roots of (6) in [0, 2\pi).

Proof: Without loss of generality, let us consider the root x_r = 2r\pi/n, with r = N^{(1,\alpha/2)}. It is trivial to show that the term for k = r_0 in (6) is equal to zero. Then, (6) becomes
\[
\sum_{k=0}^{n-1} \sin(2r_0\pi/n - 2k\pi/n)e^{\cos(2r_0\pi/n - 2k\pi/n)} = 0
\]
and the lemma holds. In case that the total terms of (6) have been eradicated and the lemma cases for even and odd values of I, respectively, we arrive at (2) m \sin \pi/2 + (\pi/2), with m = N - n + 1, m_1 and k_m = n - 1 - m_1, respectively, \forall m_1 \in \mathbb{N}^{(n-2-n-1)/2}.

\[
\mathcal{I}^+_m = \sin(2(r_0 - 2r_0 - 1 - m_1)\pi/n) = \sin(-2(r_0 + m_1 + 1)\pi/n)e^{\cos(2(r_0 + m_1 + 1)\pi/n)}.
\]
\[
\mathcal{I}^-m = \sin(2(r_0 - n - 1 + m_1)\pi/n) = \sin(2(r_0 + m_1 + 1)\pi/n)e^{\cos(2(r_0 + m_1 + 1)\pi/n)}.
\]

Hence, \mathcal{I}^+_m + \mathcal{I}^-m = 0, \forall m_1 \in \mathbb{N}^{(n-2-n-1)/2}. We discern two cases for even and odd values of n - 2r_0 - 1. In the first case, the total terms of (6) have been eradicated and the lemma holds. In case that n - 2r_0 - 1 is odd, then a singular term remains that is the center of \{0, n - 2r_0 - 1\} or, equally, for m = (n - 2r_0 - 2)/2 = (-r_0 + 1) + n/2. Note that
\[
k_m = 2r_0 + 1 + (-r_0 - 1 + n/2) = r_0 + n/2
\]
and
\[
k_m = n - 1 - (-r_0 - 1 + n/2) = r_0 + n/2.
\]
However, for this term, we have
\[
\sin(2(r_0 - (r_0 + n/2))\pi/n)e^{\cos(2(r_0 - (r_0 + n/2))\pi/n)} = \sin(-\pi)e^{\cos(-\pi)} = 0
\]
concluding that our statement holds for the second case as well. Finally, following the similar procedure, it is trivial to show that the same conclusion holds for n/2 < r_0 < n.

Lemma 3: If two unit vectors a, b ∈ \mathcal{F}_d with a common origin O as well as a bivector B that passes through their summation vector s = a + b, then the projections of a and b onto B have equal norms and angular distances from s.

Proof (Based on Rules of Clifford Algebra [35]): From [36], \(a_{||B} = (a - B)b^{-1}\) and \(b_{||B} = (b - B)b^{-1}\). Given that B passes through s, we know that
\[
sB = s \cdot B + s \wedge B = s \cdot B. \tag{14}
\]
For the addition of the collinear with B vectors, we have
\[
a_{||B} + b_{||B} = (a - B)b^{-1} + (b - B)b^{-1} = \frac{(a + b - B)b^{-1}}{s} = sB^{-1} = s. \tag{15}
\]
Moreover, for the projections of a and b on s, we write
\[
a_{1||B} = a \cdot s = a \cdot (a + b) = a \cdot a + a \cdot b = 1 + a \cdot b \quad b_{1||B} = b \cdot s = b \cdot (a + b) = b \cdot a + b \cdot b = 1 + a \cdot b
\]
\[
\implies a_{1||B} = a \cdot s = b \cdot s = b_{1||B}. \tag{16}
\]
Hence
\[
a \cdot s = b \cdot s
\]
\[
\implies (a_{1||B} + b_{1||B}) \cdot s = (b_{1||B} + b_{1||B}) \cdot s
\]
\[
\implies a_{1||B} \cdot s + a_{1||B} \cdot s = b_{1||B} \cdot s + b_{1||B} \cdot s
\]
\[
\implies a_{1||B} \cdot s = b_{1||B} \cdot s \implies (a_{1||B} - b_{1||B}) \cdot s = 0 \tag{17}
\]
suggesting perpendicularity between \(a_{1||B} - b_{1||B}\) and s. However, those vectors constitute the diagonals of a parallelogram \(P\) on B with sides \(a_{1||B}\) and \(b_{1||B}\). The above perpendicularity between the diagonals drops \(P\) to the degenerate case of a rhombus with sides \(a_{1||B}\) and \(b_{1||B}\), thus ensuring that the following conditions hold.

1. \((a_{1||B}, b_{1||B})\) bisection by s.
2. \(\|a_{1||B}\| = \|b_{1||B}\|\).

An illustration of Lemma 3 is shown in Fig. 3.

V. \(H\) AS A NONEXCLUSIVE CONDITION

In this section, we begin by demonstrating the falsifiability of \(H\) in the input space of the output layer \(\mathcal{F}_d \subset \mathbb{R}^{d+1}\), where \(d \in \mathbb{N}^+\) the dimensionality both of the embeddings.
and the weight vectors. Then, we proceed with the definition of a specific symmetrical layout in $F_d$ that produces proven consistency with $\mathcal{H}$. In our following analysis, we consider an arbitrary number of target classes $n \in \mathbb{N}^+$. The studied criterion, which leads to the maximization of $\mathcal{H}$, refers to the maximization of the $i$th class’s softmax output

$$S_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$  
(18)

with $z_j = \hat{\omega}_j \cdot \hat{\epsilon}, \hat{\epsilon} \in F_d$ the embedding, and $\hat{\omega}_j \in F_d$ the $j$th class’s weight vector. Then, the criterion demands the maximization of $S_i$ for the case that $\hat{\epsilon}$ coincides with the target class’s weight vector $\hat{\omega}_i$. According to [34], the output of the $j$th neuron can be written as $z_j = \|\hat{\omega}_j\| \cos(\theta - \phi_j)$, where $\hat{\omega}_j$ denotes the projection of $\hat{\omega}_j$ on the common plane of $\hat{\omega}_i$ and $\hat{\epsilon}$. Then

$$\frac{dS_i}{d\theta} = 0 \implies \frac{d}{d\theta} \left( \frac{e^{z_i}(\theta)}{\sum_{j=1}^{n} e^{z_j}(\theta)} \right) = 0$$

$$\implies \frac{d e^{z_i}(\theta)}{d\theta} \sum_{j=0}^{n-1} e^{z_j}(\theta) - e^{z_i}(\theta) \sum_{j=0}^{n-1} \frac{d e^{z_j}(\theta)}{d\theta} = 0$$

$$\implies \sum_{j=0}^{n-1} \frac{dz_i(\theta)}{d\theta} e^{z_i}(\theta) - \sum_{j=0}^{n-1} \frac{dz_j(\theta)}{d\theta} e^{z_j}(\theta) = 0$$

$$\implies \sum_{j=0}^{n-1} \left( \frac{dz_i(\theta)}{d\theta} - \frac{dz_j(\theta)}{d\theta} \right) e^{z_i}(\theta) e^{z_j}(\theta) = 0.$$  
(19)

We want $S_i$ to maximize for $\hat{\epsilon} = \hat{\omega}_i$, where $\theta = \phi_i$ and \(d\bar{z}_i(\theta)/d\theta|_{\theta=\phi_i} = -\|\hat{\omega}_i\| \sin(\theta - \phi_i)|_{\theta=\phi_i} = 0$. Hence, (19) becomes

$$-\sum_{j=0}^{n-1} \frac{dz_j(\theta)}{d\theta} e^{z_i}(\theta) e^{z_j}(\theta) = 0 \implies \sum_{j=0}^{n-1} \frac{dz_j(\theta)}{d\theta} e^{z_j}(\theta) = 0.$$  
(20)

Equation (20) shapes the mathematical formulation of the studied criterion.

A. Refutability of $\mathcal{H}$

At this stage, it is sufficient to determine a layout of the weight vectors, which leads to the falsification of $\mathcal{H}$. Let us consider $n$ weight vectors $\hat{\omega}_i \in F_d$ such that the following conditions hold.

1) A.I: All lie in a 2-D plane $P$.
2) A.II: $\|\hat{\omega}_i\| = 1, \forall i = 0, 1, \ldots, n - 1$.
3) A.III: $(\hat{\omega}_i, \hat{\omega}_{i+1}) = \pi/n, \forall i = 0, 1, \ldots, n - 2$.

This is shown in Fig. 4. By working on the common plane $P$ of A.I as well as considering A.II, we ensure that $\|\hat{\omega}_j\| = 1, \forall j \in \mathbb{N}^{<n}$ in (20). Moreover, according to A.III and by considering the reference vector $\hat{\omega}_0$, we have $(\hat{\omega}_0, \hat{\omega}_1) = j\pi/n, \forall j \in \mathbb{N}^{<n}$. Elaborating more, we can express the dot product of each weight vector $\hat{\omega}_j$, as a function of the weight of reference $\hat{\omega}_0$, as follows:

$$z_0 = \hat{\omega}_0 \cdot \hat{\epsilon} = \cos \theta$$
$$z_j = \hat{\omega}_j \cdot \hat{\epsilon} = \cos(\theta - j\pi/n) \quad \forall j \in \mathbb{N}^{<n}.$$  

Hence, (20) ends up to

$$\sum_{j=0}^{n-1} \sin(\theta - j\pi/n) e^{\cos(\theta - j\pi/n)} = 0.$$  
(21)

However, in case that $\hat{\epsilon}$ and $\hat{\omega}_0$ coincide, indicating that $\theta = 0$, we have

$$\sum_{j=0}^{n-1} \sin(j\pi/n) e^{\cos(j\pi/n)} > 0 \quad \forall n \in \mathbb{N}^+2.$$  
(22)

The above leads us to the conclusion that $\mathcal{H}$ is not satisfied given an arbitrary distribution of the weight vectors in $F_d$.

B. Proposed Symmetry in $F_d$

In turn, we define a specific symmetrical layout of the last layer’s weights in $F_d$, which ensures that those weights account for the classes’ centers for any number of target classes $n$ and feature vectors’ dimension $d$. More specifically, we examine the case of $n$ weight vectors $\hat{\omega}_i$ such that the following conditions hold.

1) B.I: All lie in a 2-D plane $P$.
2) B.II: $\|\hat{\omega}_j\| = 1, \forall i \in \mathbb{N}^{<n}$.
3) B.III: $(\hat{\omega}_i, \hat{\omega}_{i+1}) = 2\pi/n, \forall i \in \mathbb{N}^{<n}$.

The is shown in Fig. 5. In Fig. 6, the proposed symmetry is displayed in $F_d \subseteq \mathbb{R}^3$ for $n = 3$ and $n = 4$.

Considering the above properties, we examine the required condition to relate class centrality with its weight vector. Given that $\hat{\epsilon} = \hat{\omega}_i$, there is no specific plane defined by $\hat{\epsilon}$ and $\hat{\omega}_i$. Ergo, we are free to work on the common plane $P$ of B.I, ensuring consistency with B.II since $\|\hat{\omega}_j\| = 1, \forall j \in \mathbb{N}^{<n}$ in (20). Moreover, elaborating B.III, we can express
that lie on the dot product of each weight vector \( \bar{w}_j \), as a function of the weight of reference \( \bar{w}_0 \), as follows:

\[
z_0 = \bar{w}_0 \cdot \bar{e} = \cos \theta, \\
z_j = \bar{w}_j \cdot \bar{e} = \cos(\theta - 2j\pi/n) \quad \forall j \in \mathbb{N}^{[1,n]}.
\]

Consequently, (20) ends up to the equation

\[
\sum_{j=0}^{n-1} \sin(\theta - 2j\pi/n) e^{\cos(\theta - 2j\pi/n)} = 0. 
\]

According to Lemma 2, the weight vectors \( \bar{w}_j \) satisfy (23), thus indicating that they constitute extremum points of \( S_i \), \( \forall i \in \mathbb{N}^{<n} \). Following Lemma 1, we can further conclude that the following conditions hold:

1. \( B.IV: \sum_{i=1}^{n} \bar{w}_i = \bar{0}. \)

At this stage, we check the validity of (20) on a random plane \( P' \) that passes through the weight vector \( \bar{w}_i \). Let \( \bar{w}_{i-m} \) and \( \bar{w}_{i+m}, m \in \mathbb{N}^{[1,(n-1)/2]} \) be the weight vectors astride \( \bar{w}_i \) that lie on \( P \). Then, \( \bar{w}_{i+m} - \bar{w}_{i-m} = 2m\pi/n \), \( \bar{w}_{i+m} - \bar{w}_{i} = 2m\pi/n \), and

\[
\frac{dz_{i-m}(\theta)}{d\theta} e^{z_{i-m}(\theta)} + \frac{dz_{i+m}(\theta)}{d\theta} e^{z_{i+m}(\theta)} \\
= \frac{\|\bar{w}_{i-m}\| \sin(\theta - \phi_{i-m}) e^{\cos(\theta - \phi_{i-m})}}{\|\bar{w}_{i+m}\| \sin(\theta - \phi_{i+m}) e^{\cos(\theta - \phi_{i+m})}} \\
+ \frac{\|\bar{w}_{i+m}\| \sin(\theta - \phi_{i+m}) e^{\cos(\theta - \phi_{i+m})}}{\|\bar{w}_{i-m}\| \sin(\theta - \phi_{i-m}) e^{\cos(\theta - \phi_{i-m})}}.
\]

Moreover, the summation vector \( \bar{w}_{i-m} + \bar{w}_{i+m} \) is parallel to \( \bar{w}_i \). According to Lemma 3, the projections of \( \bar{w}_{i-m} \) and \( \bar{w}_{i+m} \) on \( P' \) have equal norms and are also bisected by \( \bar{w}_i \). Hence, \( \|\bar{w}_{i-m}\| = \|\bar{w}_{i+m}\| = \|\bar{w}_i\| \) and \( \phi_i - \phi_{i-m} = -(\phi_i - \phi_{i+m}) = \phi_i \). For \( \theta = \phi_i \), (24) becomes

\[
\|\bar{w}_m\| \sin(\phi_m) e^{\cos(\phi_m)} + \|\bar{w}_m\| \sin(-\phi_m) e^{\cos(-\phi_m)} \\
= \|\bar{w}_m\| \sin(\phi_m) e^{\cos(\phi_m)} - \|\bar{w}_m\| \sin(\phi_m) e^{\cos(\phi_m)} = 0.
\]

That is, the vectors that present equal angle astride \( \bar{w}_i \) eradicate themselves in the sum of (20). In case their total number is odd, the remaining one is the counterbalancing vector \( -\bar{w}_i \), a term that also equals zero, as shown in Lemma 2. Eventually, the proposed symmetry secures maximization of \( S_i \) at the orientation of \( \bar{w}_i \) from any possible direction.

**VI. IMPLEMENTATION DETAILS**

This section focuses on the implementation of the proposed symmetry in the last layer of a CNN. The code has been developed using PyTorch 1.4 [39], supporting GPU-enabled operations on an NVIDIA GeForce GTX 1060, 6 GB.

According to Section V-B, regardless of the number of classes and the feature space dimensionality, the whole symmetry is placed on a 2-D plane on which all the weights of the last layer lie. Hence, since the definition of such a plane requires two orthogonal vectors, and the trainable parameters of the layer are reduced to the vectors \( \hat{u}_1, \hat{u}_2 \in \mathcal{F}_d \). The above two vectors are free in terms of orientation and scale. In an effort to develop a layer compatible with the supported operations in DL frameworks, such as PyTorch, we follow the Gram–Schmidt orthogonalization [40]. Hence, we calculate the orthonormal vectors \( \hat{n}_1, \hat{n}_2 \in \mathcal{F}_d \) that form the basis of the rotation plane. More specifically, \( \hat{n}_1 \) is the \( l_2 \)-normalized vector of \( \hat{u}_1 \) and \( \hat{n}_2 \) is orthonormal to \( \hat{n}_1 \). Consequently, the proposed symmetrical layout is produced by rotating the initial vector \( \hat{n}_1 \) by \( 2\pi i/n \), \( \forall i \in \mathbb{N}^{<n} \). The above rotations are conducted in parallel, shaping a GPU-enabled operation. The code regarding the implementation of the layer is provided in Appendix, while the whole pipeline for training a CNN is available online.¹

Similar to the most methods in the field of neural-based feature learning, the introduced layer includes the scalar parameter \( \sigma \), which refers to the radius of the hypersphere in \( \mathcal{F}_d \). This parameter can be either predefined or it can be learned during the training procedure of the DNN [41].

**VII. EXPERIMENTAL STUDY**

In this section, we proceed with the application of the proposed layer in the broadly known challenge of image classification, exploiting the benchmark database CIFAR-10 (C10) [37]. At this point, we should state that the main and only requirement of the above layer refers to the exploitation of the softmax loss during training. However, it is independent of the DNN architecture, in terms of number and type of layers. Consequently, as long as the last layer is a common classification layer, the sensory input can be either an image, a feature vector, or a point cloud [34]. The main reason for focusing on images is owed to the fact that it is the main field extensively occupied with the task of deep feature learning, thus adopting the discussed hypothesis \( \mathcal{H} \). To that end, we compare the introduced layer against the benchmark FC layer in terms of performance, demanding at least competitive classification results. Consequently, an empirical study is demonstrated to highlight the stability inconsistencies observed in the field’s state-of-the-art methods, i.e., SphereFace [20] and ArcFace [19], and prove the superiority of ours with respect to the training behavior.

**A. Symmetrical Layout Convergence**

In order to evaluate the convergence of the layer, we utilize the widely known ResNet-18 architecture [38] and conduct several experiments for different values of \( \sigma \) on C10. To ensure fairness, the training procedure of all experiments lasts 160 epochs with a batch size of 256. The stochastic gradient descent (SGD) optimizer is employed, using momentum 0.9, weight decay \( 10^{-4} \), and initial learning rate at 0.1 that decays by an order of magnitude at 50% and 75% of the total duration.

¹https://github.com/ IoannisKansizoglou/Symmetrical-Feature-Space
In Fig. 7, the training curves of the introduced layer on C10 for $\sigma = 8, 16, 32,$ and 64 are demonstrated. In addition, the corresponding curves for the common FC layer are included to visualize the differences between the two approaches. Note that the experiments with the FC layer are also conducted following the same training setup. At first, we highlight the competitive performance of our layer compared against the FC one for all the investigated values of $\sigma$. In the cases of $\sigma = 16$ and 32, the succeeded evaluation accuracy exceeds by $\approx 2.5\%$ the benchmark one, as shown in Table I. However, the main argument of the current work does not focus on the enhancement of the classification accuracy, but rather on showing that the proposed method sustains the state-of-the-art performance. Furthermore, by paying attention to the training curves in Fig. 7, we understand that the symmetrical layer converges with a slower rate. The above fact is highly anticipated, due to the restrictions inserted to the trainable parameters ensuring the desired layout. We denote that the contribution of this layer lies upon the satisfaction of $\mathcal{H}$, meaning that it is the optimal way to apply a feature learning technique, such as $\text{SphereFace}$ and $\text{ArcFace}$.

In Fig. 8, the characteristic vector of the plane of symmetry $P$ is monitored. More specifically, in Fig. 8(a), we present the angle between two successive snapshots of the above vector during the training procedure, thus summarizing the variation of its orientation in degrees ($^\circ$). The curve of the corresponding training loss is also included. For a more representative comparison, Fig. 8(b) shows the above curves after normalization in $[0, 1]$.

### B. Stability Issues and Comparative Study

The aim of this section is to demonstrate in practice the issues introduced by the common hypothesis $\mathcal{H}$, which is adopted both in $\text{SphereFace}$ [20] and $\text{ArcFace}$ [19], as stated in Section II. Hence, we proceed with a grid search methodology for the scalar parameter of $\text{ArcFace}$ with $\sigma$ in $\{4, 8, 16, 32, 64\}$, which is adopted by the proposed method as well. $\text{SphereFace}$ is trained only for $\sigma = 1$ based on the available implementation of the layer in PyTorch. Each experiment is conducted three times under exactly the same setup of parameters. The training setup follows the one proposed in Section VII-A.

The obtained results for $\text{ArcFace}$ are shown in Table II. The notation $x$ is employed to represent the cases, where the training loss diverges from finite values. In any other case, the best evaluation accuracy (in %) is kept. By examining Table II, the reader can discover the unstable and simultaneously different behavior of such a method even under identical training and hyperparameter setup. We argue that the above fact is

| Method          | $\sigma$ | Accuracy (%) |
|-----------------|---------|--------------|
| FC layer        | -       | 79.77        |
| $\text{SphereFace}$ [20] | -       | 79.34        |
| $\text{ArcFace}$ [19]  | 64      | 73.71        |
| Ours            | 8       | 81.53        |
| Ours            | 16      | 82.37        |
| Ours            | 32      | 82.22        |
| Ours            | 64      | 80.74        |

Fig. 8. With purple, the variation of the plane’s ($P$) angle (in degrees $^\circ$) is illustrated during training the symmetrical layer on C10, while the corresponding training loss is depicted with magenta. In (a), both curves are presented in their actual scale, while in (b), they are normalized to further demonstrate their correlation.
TABLE II
TRAINING RESULTS UNDER DIFFERENT EXPERIMENTAL SETUPS OF ArcFace [19], AS OBTAINED BY USING ResNet-18 ON C10

| σ   | m  | Accuracy1 (%) | Accuracy2 (%) | Accuracy3 (%) |
|-----|----|---------------|---------------|---------------|
| 4   | 0.1| x             | x             | x             |
| 8   | 0.1| x             | x             | x             |
| 16  | 0.1| x             | 10.00         | x             |
| 64  | 0.1| 20.02         | 73.71         | x             |

Fig. 9. Training curves of ArcFace [19] (σ=64 and m=0.1) for the training (green) and evaluation (red) sets of C10, as obtained by using ResNet-18. (a) Softmax accuracy. (b) Softmax loss.

TABLE III
TRAINING RESULTS OF SphereFace [20], AS OBTAINED BY USING ResNet-18 ON C10

| σ   | Accuracy1 (%) | Accuracy2 (%) | Accuracy3 (%) |
|-----|---------------|---------------|---------------|
| 1   | 47.31         | 79.34         | x             |

Fig. 10. Training curves of SphereFace [20] (σ=1) for the training (green) and evaluation (red) sets of C10, as obtained by using ResNet-18. (a) Softmax accuracy. (b) Softmax loss.

C. Time Complexity

As a final assessment, we focus on time efficiency, ascertaining that the proposed implementation adds no particular complexity compared against a simple fully connected (FC) solution. To that end, we exploit the ResNet-18 architecture and change only the last hidden layer of the network. We proceed with four different versions of the network: 1) using a vanilla FC layer; 2) incorporating SphereFace; 3) incorporating ArcFace; and 4) applying the proposed symmetrical layer. Thus, by maintaining the same architecture, the computed time differences are owed exactly to the complexity of the investigated layers. We train each version on C10 and each experimentation is conducted three times adopting the training procedure described in Section VII-A. In Table IV, the mean and standard deviation values of the corresponding seconds per epoch are computed among the three repetitions of the same experiment, using the FC layer, SphereFace, ArcFace, and ours. As it was anticipated, the proposed layer adds a marginal time delay to the vanilla FC layer (about 0.35 sec/epoch) when training on C10. However, it remains noticeably lower from the complexity introduced by the corresponding layers of SphereFace and ArcFace. Knowing the above, we ensure that the proposed implementation is suitable for large-scale applications.

VIII. CONCLUSION

In drawing to a close, this article at hand deals with the empirical assumption adopted by the vast majority of recent methods in the field of neural-based feature learning. The aforementioned hypothesis (H) claims the coincidence between the last FC layer’s weight vector with the geometrical center of the corresponding class. Following a theoretical approach, we prove the refutability of H, given a random distribution of the weight vectors in the feature space \( \mathcal{F}_d \), which refers to the input space of the output layer. Consequently, a specific symmetrical layout for the weight vectors is proposed, which is proven to satisfy H for any dimensionality \( d \) of space \( \mathcal{F}_d \).
Proceeding with our empirical study, the implementation of the proposed symmetry is described, which can be easily adopted as a custom layer in widespread deep learning frameworks, such as PyTorch. The code of the layer is openly provided in Appendix. Then, several experiments are conducted to demonstrate the convergence capabilities of the layer. We further demonstrate that the required training duration is similar to the one of a common FC layer. The achieved accuracy values of the proposed layer compared against the FC one as well as two of the most widespread models in the field are also at competitive and occasionally beneficiary levels, i.e., SphereFace and ArcFace. Finally, the impact of the false supposition in stability issues is empirically demonstrated within the above methods in the field, by evaluating their achieved evaluation accuracy between multiple repetitions of the same training setup.

As part of future work, we aim to exploit the formulation of the proposed symmetrical layout in a feature learning task, such as the challenge of face verification. In specific, the combination of the proposed layer with the rules of state-of-the-art schemes, such as ArcFace, can be investigated, as an attempt to enhance the feature learning capacity of a DNN. In addition, by starting from lower dimensions of the feature space $F_d$, we aim at investigating a formal definition that provides the equivalent of a class center for an arbitrary layout of the weight vectors.

### Appendix

See Listing 1.

### References

[1] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[2] G. Zhong, L.-N. Wang, X. Ling, and J. Dong, “An overview on data representation learning: From traditional feature learning to recent deep learning,” *J. Finance Data Sci.*, vol. 2, no. 4, pp. 265–278, Dec. 2016.

[3] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, “Image matching from handcrafted to deep features: A survey,” *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 1–57, 2020.

[4] H. Liang, X. Sun, Y. Sun, and Y. Gao, “Text feature extraction based on deep learning: A review,” *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, pp. 1–12, Dec. 2017.

[5] F. R. S. K. Pearson, “LIII. On lines and planes of closest fit to systems of points in space,” *London, Edinburgh, Dublin Philosoph. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

[6] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Aug. 1936.

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[9] F. Eyben, M. Wöllmer, and B. Schuller, “Opensmile: The Munich versatile and fast open-source audio feature extractor,” in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 1459–1462.

[10] K. A. Tsintotas, L. Bampis, and A. Gasteratos, “Probabilistic appearance-based place recognition through bag of tracked words,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1737–1744, Apr. 2019.

[11] M. Norouzi and D. J. Fleet, “Minimal loss hashing for compact binary codes,” in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 353–360.

[12] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *Proc. Sci. Conf.*, Aug. 2014, pp. 372–378.

[13] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, “Deep representation learning in speech processing: Challenges, recent advances, and future trends,” 2020, arXiv:2001.00378.

[14] W. Guo, J. Wang, and S. Wang, “Deep multimodal representation learning: A survey,” *IEEE Access*, vol. 7, pp. 63373–63394, 2019.

[15] J. C. Caicedo and S. Lazebnik, “Active object localization with deep reinforcement learning,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2488–2496.

[16] V. Balaska, L. Bampis, I. Kansizoglou, and A. Gasteratos, “Enhancing satellite semantic maps with ground-level imagery,” *Robot. Auton. Syst.*, vol. 139, May 2021, Art. no. 103760.

[17] S. Antol et al., “VQA: Visual question answering,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.
[18] I. Kansizoglou, L. Bampis, and A. Gasteratos, “An active learning paradigm for online audio-visual emotion recognition,” IEEE Trans. Affect. Comput., early access, Dec. 20, 2019, doi: 10.1109/TAFFC.2019.2961089.

[19] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4690–4699.

[20] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 212–220.

[21] L. Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep face recognition: A survey,” in Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI), Oct. 2018, pp. 471–478.

[22] H. Wang et al., “CosFace: Large margin cosine loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5265–5274.

[23] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in Proc. Int. Conf. Mach. Learn. Deep Learn. Workshop, 2015, pp. 1–30.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 815–823.

[25] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2006, pp. 1735–1742.

[26] I. Kansizoglou, L. Bampis, and A. Gasteratos, “HASEparation: Hyperplane-assisted softmax loss for discriminative face verification,” in Proc. 25th ACM Int. Conf. Multimedia, Oct. 2017, pp. 1041–1049.

[27] I. Kansizoglou, L. Bampis, and A. Gasteratos, “HA Separator: Hyperplane-assisted softmax,” in Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2020, pp. 519–526.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[29] I. Kansizoglou, L. Bampis, and A. Gasteratos, “Deep feature space: A geometrical perspective,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Jul. 7, 2021, doi: 10.1109/TPAMI.2021.3094625.

[30] W. K. Clifford, “On the classification of geometric algebras,” in Mathematical Papers, R. Tucker, Ed. London, U.K. MacMillian, 1882, pp. 397–401.

[31] J. Suter. (2003). Geometric Algebra Primer. [Online]. Available: http://www.jaapzutier.com/paper/ga_primer.pdf

[32] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 7, 2009.

[33] H. Wang et al., “CosFace: Large margin cosine loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 5265–5274.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[36] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” 2019, arXiv:1912.01703.

[37] E. Schmidt, “Zur theorie der linearen und nichtlinearen Integralgleichungen. III. Teil,” Mathematische Annalen, vol. 65, no. 3, pp. 370–399, Sep. 1908.

[38] R. Ranjan, C. D. Castillo, and R. Chellappa, “L2-constrained softmax loss for discriminative face verification,” 2017, arXiv:1703.09507.

Ioannis Kansizoglou received the Diploma degree from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2017, and the Ph.D. degree in deep representation learning and computer vision from the Democritus University of Thrace (DUTH), Komotini, Greece, in 2021.

He is currently a Post-Doctoral Fellow with the Laboratory of Robotics and Automation (LRA), Department of Production and Management Engineering, DUTH. His research interests include deep representation learning, emotion analysis, and human–robot interaction.

Loukas Bampis received the Diploma degree in electrical and computer engineering and the Ph.D. degree in machine vision and embedded systems from the Democritus University of Thrace (DUTH), Komotini, Greece, in 2013 and 2019, respectively.

He is currently an Assistant Professor with the Laboratory of Mechatronics and Systems Automation (MeSA Lab), Department of Electrical and Computer Engineering, DUTH. His work has been supported through several research projects funded by the European Space Agency, the European Commission, and the Greek Government. His main research interests include real-time robot localization and place recognition techniques using hardware accelerators.

Dr. Bampis is an Associate Editor at Electronics Letters and has served as a reviewer for various international journals and conferences related to robotics and machine vision.

Antonios Gasteratos (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees from the Department of Electrical and Computer Engineering, Democritus University of Thrace (DUTH), Komotini, Greece, in 1994 and 1998, respectively.

He is currently a Professor and the Head of the Department of Production and Management Engineering, DUTH. He is also the Director of the Laboratory of Robotics and Automation (LRA), DUTH, where he teaches the courses of robotics, automatic control systems, electronics, mechatronics, and computer vision. He has authored or coauthored more than 220 articles in books, journals, and conferences. His research interests include mechatronics and robot vision.

Dr. Gasteratos is a fellow of IET. He has served as a reviewer for numerous scientific journals and international conferences. He is a Subject Editor of Electronics Letters and an Associate Editor of the International Journal of Optomechatronics and he has organized/coorganized several international conferences.