SEN: A Novel Feature Normalization Dissimilarity Measure for Prototypical Few-Shot Learning Networks

Van Nhan Nguyen[1,2], Sigurd Løkse[1], Kristoffer Wickstrøm[1], Michael Kampffmeyer[1], Davide Roverso[2], and Robert Jenssen[1]

[1] UiT Machine Learning Group, UiT The Arctic University of Norway
{sigurd.lokse,kristoffer.k.wickstrom,michael.c.kampffmeyer,robert.jenssen}@uit.no
[2] Analytics Department, eSmart Systems, 1783 Halden, Norway
{nhan.v.nguyen,Davide.Roverso}@esmartsystems.com

Abstract. In this paper, we equip Prototypical Networks (PNs) with a novel dissimilarity measure to enable discriminative feature normalization for few-shot learning. The embedding onto the hypersphere requires no direct normalization and is easy to optimize. Our theoretical analysis shows that the proposed dissimilarity measure, denoted the Squared root of the Euclidean distance and the Norm distance (SEN), forces embedding points to be attracted to its correct prototype, while being repelled from all other prototypes, keeping the norm of all points the same. The resulting SEN PN outperforms the regular PN with a considerable margin, with no additional parameters as well as with negligible computational overhead.

1 Introduction

Few-shot classification [8, 23, 19, 17, 6, 20] aims at adapting a classifier to previously unseen classes from just a handful of labeled examples per class. In the past few years, many approaches to few-shot classification have been proposed. These approaches can be roughly categorized as (i) learning to fine-tune approaches [6, 17]; (ii) sequence-based approaches [1, 13]; (iii) generative modeling-based approaches [29, 26]; (vi) (deep) distance metric learning-based approaches [19, 22, 27, 20]; and (v) semi-supervised approaches [3, 16]. Among these categories, distance metric learning-based approaches are typically preferred because of their simplicity and effectiveness. The basic idea of these approaches, for which the so-called Prototypical Networks (PNs) [19] are the most well-known examples, is to learn a non-linear mapping of the input into an embedding space which is commonly high-dimensional. In this space, a metric distance is defined which maps similar examples close to each other in the embedding space. Dissimilar examples are mapped to distant locations relative to each other, so that a query example can be classified by, for example, using nearest neighbor methods. Arguably one of the most commonly used distance metrics in this high dimensional embedding space is the (squared) Euclidean distance combined with a softmax function [19, 27, 3, 16].

However, even though the softmax is known to work well for closed-set classification problems, it has been shown to not be discriminative enough in problems were there are few labels relative to the number of classes [4, 15]. This has given rise to alternative loss formulations with improved discriminative ability, where high-dimensional features have been normalized explicitly to lie on a hypersphere via direct L_2 normalization [4, 15, 24]. The advantage of normalization has been theoretically analyzed in [30]. However, direct L_2 normalization leads to a non-convex loss formulation, which typically results in local minima generated by the loss function itself [30].

With the aim of performing *soft* feature normalization while preserving the convexity and the simplicity of the loss function, we equip PNs with a novel dissimilarity measure particularly suited to enable discriminative feature normalization for few-shot learning, without any direct normalization. The proposed dissimilarity measure, denoted the Squared root of the Euclidean distance and the Norm distance (SEN), replaces the Euclidean distance in PN training, with major consequences: Our theoretical analysis shows that the proposed measure explicitly forces embedded points to be attracted to the correct prototype and repelled from incorrect prototypes. Further, we provide analysis showing that SEN indeed explicitly forces all embeddings to have the same norm during training which enables the resulting SEN PN to generate a more robust embedding space. With this minimal but important modification, the SEN PN outperforms the original PN by a considerable margin and demonstrates good performance on the Mini-Imagenet [17, 23], the Fewshot-CIFAR100 (FC100) [14], and the Omniglot [9] datasets with no additional parameters as well as negligible computational overhead (a comparison of inference time is provided in the supplementary material). We furthermore experimentally show that the proposed SEN dissimilarity measure constantly outperforms the Euclidean distance in PNs with different embedding sizes as well as with different embedding networks.

2 Related Work

The literature on few-shot learning is vast; we present in this section a short summary of well-known approaches and works most relevant to our proposed approach. We refer the reader to [25] and [21] for more detailed reviews on few-shot learning.

Besides distance metric learning-based approaches, few-shot learning approaches can be categorized into (i) learning to fine-tune approaches; (ii) sequence-based approaches; (iii) generative modeling-based approaches; (iv) (deep) distance metric learning-based approaches; and (v) semi-supervised approaches. Learning to fine-tune approaches aim at learning a model's initial parameters such that it can be quickly adapted to a new task through only one or a few gradient update steps [6, 17]. These approaches typically can handle many model representations; however, they suffer from the need to fine-tune on the target problem, which makes them less appealing to few-shot learning. Sequence-based approaches formalize few-shot learning as a sequence-to-sequence problem and

leverage Recurrent Neural Networks (RNNs) with memories to address the problem [1, 13]. While appealing, these methods typically require complex RNN architectures and complicated mechanisms for storing/retrieving all the historical information of relevance, both long-term and short-term, without forgetting [20]. Generative modeling-based approaches employ adversarial training to produce additional signals/training examples to allow the classification algorithm to learn a better classifier [29, 26]. Deep distance metric learning-based approaches aim at eliminating the need for manually choosing the right distance metric (e.g., the Euclidean distance and the cosine distance) by learning not only a deep embedding network but also a deep non-linear metric (similarity function) for comparing images in the embedding space [20]. Although deep distance metric learning-based approaches can avoid the need for manually choosing the right distance metric, they are prone to overfitting and are more difficult to train compared to distance metric learning-based approaches due to the added parameters. Semi-supervised approaches utilize unlabeled data to improve few-shot learning accuracy. This is typically achieved by casting the semi-supervised few-shot learning problem as a semi-supervised clustering problem and address it by applying, for example, k-means clustering algorithms [3, 16]. We build on the distance metric learning line of work due to its simplicity and effectiveness.

Metric learning-based approaches aim to learn a non-linear mapping of the input into an embedding space and define a metric distance which maps similar examples close and dissimilar ones distant in the embedding space, so that a query example can be easily classified by, for example, using nearest neighbor methods. Some notable approaches in this line of work include Koch et al. [8], who propose to learn siamese neural networks for computing the pair-wise distance between samples. The learned distance is then used by a nearest neighbor classifier for solving the one-shot learning problem. Vinyals et al. [23] define an end-to-end differentiable nearest neighbor classifier, called matching networks, based on the cosine similarity between the support set and the query example. Snell et al. [19] propose a simple method called prototypical networks for few-shot learning based on the assumption that there exists an embedding space in which samples from each class cluster around a single prototype representation, which is simply the mean of the individual samples. Garcia and Bruna [22] argue that few-shot learning, which aims at propagating label information from labeled support examples towards unlabeled query images, can be formalized as a posterior inference over a graphical model determined by the images and labels in the support set and the query set. The authors cast posterior inference as message passing on graph neural networks and propose a graph-based model, which can be trained end-to-end, to solve the task. Wang et al. [27] propose to improve the generalization capacity of metric-based methods for few-shot learning by enforcing a large margin between the class centers. This is achieved by augmenting a large margin loss function, which is the unnormalized triplet loss [18], to the standard softmax loss function for classification.

3 Few-shot Learning

In this section, we first begin by detailing the general few-shot learning task. Next, we introduce PNs and the Euclidean distance function with special attention paid to highlight its existing challenges. Then, we describe our proposed SEN dissimilarity measure and our SEN PN model. Finally, we provide analyses on the gradient of the SEN PN's loss function and the behavior of the proposed SEN dissimilarity measure during training.

3.1 Task Description

In the traditional machine learning setting, we are typically given a dataset D. This dataset is usually split into two parts: D_{train} and D_{test}. The former is often used for training the parameters θ of the model, while the latter is typically used for evaluating its generalization. In general few-shot learning, we are dealing with meta-datasets D_{meta} containing multiple regular datasets D [17]. Each dataset $D \in D_{meta}$ has a split of D_{train} and D_{test}; however, they are usually much smaller than that of regular datasets used in the traditional machine learning setting. Let $C = \{1, \ldots, K\}$ be the set of all classes available in D_{meta}. The set C is usually split into two disjoint sets: C_{train} containing training classes and C_{test} containing unseen classes for testing, i.e., $C_{train} \cap C_{test} = \emptyset$. The meta-dataset D_{meta} is often split into two parts: The first is a meta training set $D_{meta-train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ is the feature vector of the i^{th} example, $y_i \in C_{train}$ is its corresponding label, and N is the number of training examples. The second part is a meta testing set $D_{meta-test}$. In a standard M-way K-shot classification task, the meta testing set $D_{meta-test}$ consists of a *support set* and a *query set*. The support set $S = \{(\mathbf{x}_j, y_j)\}_{j=1}^{N_S}$ contains K examples from each of the M classes from C_{test}, i.e., the number of support examples are $N_S = M \times K$ and $y_j \in C_{test}$. The *query set* contains N_Q unlabeled examples $Q = \{\mathbf{x}_j\}_{j=N_S+1}^{N_S+N_Q}$. The support set is employed by the model for learning the new task, while the query set is utilized by the model for evaluating its performance.

3.2 Prototypical Networks

Prototypical networks learn a non-linear embedding function $f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^E$ parameterized by ϕ that maps a D-dimensional feature vector of an example $\mathbf{x}_i$ to an E-dimensional embedding $\mathbf{z}_i = f_\phi(\mathbf{x}_i)$ [19]. In meta-testing, the embedding function f_ϕ is employed for mapping examples in the support set $S = \{(\mathbf{x}_j, y_j)\}_{j=1}^{N_S}$ into the embedding space. An E-dimensional representation $\mathbf{c}_k$, or *prototype*, of each class is computed by taking the mean of the embedded support points belonging to the class:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i) = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} \mathbf{z}_i, \tag{1}$$

where S_k is the support set of class k. An embedded query point $\mathbf{x}_q$ is then classified by simply finding the nearest class prototype in the embedding space.

To train PNs, the episodic training strategy proposed in [23, 17] is adopted. In particular, to train a PN for the M-way, K-shot classification task, a training episode is formed from the meta training set D_{meta_train} as follows: K examples from each of M randomly selected classes from C_{train} are sampled to form a support set $S = \{S_i\}_{i=1}^{M}$. A query set $Q = \{Q_i\}_{i=1}^{M}$ is formed by sampling from the rest of the M classes' samples. Next, for each class k, its support set $S_k \in S$ is used for computing a prototype using Equation 1. Then, a distribution over classes for each query point $\mathbf{x}_q \in Q$ based on a softmax over distances to the prototypes in the embedding space is produced:

$$p_\phi(y = k|\mathbf{x}_q) = \frac{\exp(-d(f_\phi(\mathbf{x}_q), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\phi(\mathbf{x}_q), \mathbf{c}_{k'}))}, \tag{2}$$

where $d = \mathbb{R}^E \times \mathbb{R}^E \rightarrow [0, +\infty)$ is a distance function. Based on that, the PN is trained by minimizing the negative log-probability of the true class k via Stochastic Gradient Descent (SGD):

$$J(\phi) = -\frac{1}{M} \sum_{k=1}^{M} \frac{1}{|Q_k|} \sum_{\mathbf{x}_q \in Q_k} \log p_\phi(y = k|\mathbf{x}_q). \tag{3}$$

The training is repeated with new, randomly generated training episodes until a stopping criterion is met.

PNs employ the squared Euclidean distance as the distance metric. The squared Euclidean distance between two arbitrary points $\mathbf{z} = (z_1, \ldots, z_n)$ and $\mathbf{c} = (c_1, \ldots, c_n)$ is defined as follows:

$$d_{se}(\mathbf{z}, \mathbf{c}) = \|\mathbf{z} - \mathbf{c}\|^2 = \sum_{i=1}^{n} (z_i - c_i)^2. \tag{4}$$

Although combining the softmax and the Euclidean distance has shown to give good performance for closed-set classification settings, it performs sub-optimally when few labels are available relative to the number of classes. In order to address this issue and improve the discriminative ability, new loss formulations based on feature normalization have been proposed. These tend to normalize features explicitly via L_2 normalization [15, 24, 4]. This typically results in a more compact embedding space than the Euclidean embedding space. In such an embedding space, the cosine distance is commonly chosen as the distance metric and many few-shot classification approaches [23, 17] have employed the cosine distance in the hyperspherical embedding space. The cosine distance between two arbitrary point $\mathbf{z} = (z_1, \ldots, z_n)$ and $\mathbf{c} = (c_1, \ldots, c_n)$ is defined as:

$$d_{cs} = 1 - \frac{\mathbf{z} \cdot \mathbf{c}}{\|\mathbf{z}\|\|\mathbf{c}\|} = 1 - \frac{\sum_{i=1}^{n} z_i c_i}{\sqrt{\sum_{i=1}^{n} z_i^2} \sqrt{\sum_{i=1}^{n} c_i^2}}. \tag{5}$$

However, feature normalization through hard normalization operations such as L_2 normalization leads to a non-convex loss formulation, which typically results in local minima introduced by the loss function itself [30]. Since the net-

work optimization itself is non-convex, it is important to preserve convexity in loss functions for more effective minimization.

One possible solution is to use Ring loss [30]. The Ring loss introduces an additional term to the primary loss function, which penalizes the squared difference between the norm of samples and a learned target norm value R. The modified loss function is defined as follows:

$$L = L_P + \gamma L_R, \tag{6}$$

where γ is the loss weight w.r.t to the primary loss L_P and L_R is the Ring loss, which is defined as:

$$L_R = \frac{1}{2n} \sum_{i=1}^{n} (\|f_\phi(\mathbf{x}_i)\| - R)^2. \tag{7}$$

Since the Ring loss encourages the norm of samples being value R during training instead of explicit enforcing through a hard normalization operation, the convexity in the loss function is preserved. However, the Ring loss is more difficult to train than the primary loss (e.g., the Softmax loss) due to the added term (the norm difference L_R), the added parameter (the target norm R), and the added hyperparameter (the loss weight w.r.t to the primary loss γ).

To address the shortcomings outlined above, we propose a novel dissimilarity measure for few-shot learning, called SEN. The SEN dissimilarity measure encourages the norm of samples to have the same value, in other words, force the data to lie on a scaled unit hypersphere, while preserving the convexity and the simplicity of the loss function.

3.3 SEN Dissimilarity Measure for Prototypical Networks

The SEN dissimilarity $d_s(\mathbf{z}, \mathbf{c})$ between two arbitrary points $\mathbf{z} = (z_1, \ldots, z_n)$ and $\mathbf{c} = (c_1, \ldots, c_n)$ in D-dimensional space is a combination of the standard squared Euclidean distance d_e and the squared norm distance d_n:

$$d_s(\mathbf{z}, \mathbf{c}) = \sqrt{d_e(\mathbf{z}, \mathbf{c}) + \epsilon d_n(\mathbf{z}, \mathbf{c})}, \tag{8}$$

where ϵ is a tunable balancing hyperparameter and must be chosen such that $d_e(\mathbf{z}, \mathbf{c}) + \epsilon d_n(\mathbf{z}, \mathbf{c})$ is always positive, $d_e(\mathbf{z}, \mathbf{c})$ and $d_n(\mathbf{z}, \mathbf{c})$ are defined as:

$$d_e(\mathbf{z}, \mathbf{c}) = \|\mathbf{z} - \mathbf{c}\|^2 ,$$

$$d_n(\mathbf{z}, \mathbf{c}) = (\|\mathbf{z}\| - \|\mathbf{c}\|)^2.$$

We modify the PN by replacing the Euclidean distance by our proposed SEN dissimilarity measure. We call this model SEN PN. Specifically, we replace the distance function $d(\mathbf{z}_i, \mathbf{c}_k)$ in Equation 2 by our proposed SEN dissimilarity measure $d_s(\mathbf{z}_i, \mathbf{c}_k) = \sqrt{d_e(\mathbf{z}_i, \mathbf{c}_k) + \epsilon d_n(\mathbf{z}_i, \mathbf{c}_k)}$, $\mathbf{z}_i$ is the embedding of the example $\mathbf{x}_i$, and $\mathbf{c}_k$ is the prototype of class k. For simplicity, we consider the setting in which only one query example per class is used; however, the loss function presented in this session and the analysis presented in the next section can be easily

generalized for other settings in which more than one query examples per class are used. When only one query example per class is used, the updated negative log probability loss is given as:

$$\begin{aligned} J(\phi) &= -\sum_{k} \log p_{\phi}(y_i = k|\mathbf{x}_i) \\ &= -\sum_{k} \log \frac{\exp(-d_s(\mathbf{z}_i, \mathbf{c}_k))}{\sum_{k'} \exp(-d_s(\mathbf{z}_i, \mathbf{c}_{k'}))} \\ &= \sum_{k} \left(d_s(\mathbf{z}_i, \mathbf{c}_k) + \log \sum_{k'} \exp(-d_s(\mathbf{z}_i, \mathbf{c}_{k'})) \right). \end{aligned} \tag{9}$$

The learning proceeds by minimizing $J(\phi)$ of the true class k via SGD, which is equivalent to minimizing the SEN dissimilarity measure between the query example $\mathbf{x}_i$ and its prototype $\mathbf{c}_k$: $d_s(\mathbf{z}_i, \mathbf{c}_k)$, and maximizing the SEN dissimilarity measures between the query example $\mathbf{x}_i$ and the other prototypes $\mathbf{c}_{k'}$: $d_s(\mathbf{z}_i, \mathbf{c}_{k'})$. Minimizing $d_s(\mathbf{z}_i, \mathbf{c}_k)$ pulls $\mathbf{z}_i$ to its own class and encourages embeddings of the same class to have the same norm. Maximizing $d_s(\mathbf{z}_i, \mathbf{c}_{k'})$ pushes $\mathbf{z}_i$ away from other classes; however it encourages embeddings of different classes to have different norms.

Since our goal is to force the data to lie on a scaled unit hypersphere, we define the balancing hyperparameter ϵ relative to $\mathbf{z}_i$ and $\mathbf{c}_k$ as follows:

$$\epsilon_{ik} = \begin{cases} \epsilon_p > 0 & \text{if } y_i = k \\ \epsilon_n < 0 & \text{if } y_i \neq k \end{cases}, \tag{10}$$

where i is the index of the embedding $\mathbf{z}_i$, y_i is the embedding's class label, and k is the class label of the prototype $\mathbf{c}_k$. During training, a positive epsilon ($\epsilon_{ik} = \epsilon_p > 0$) is used for computing the SEN dissimilarity measure between the query example $\mathbf{x}_i$ and its prototype $\mathbf{c}_k$, while a negative epsilon ($\epsilon_{ik} = \epsilon_n < 0$) is used for computing the SEN dissimilarity measures between the query example $\mathbf{x}_i$ and the other prototypes $\mathbf{c}_{k'}$. The negative epsilon ϵ_n will inverse the effect of the norm distance when maximizing $d_s(\mathbf{z}_i, \mathbf{c}_{k'})$. In other words, maximizing $d_s(\mathbf{z}_i, \mathbf{c}_{k'})$ with a negative epsilon ϵ_n pushes $\mathbf{z}_i$ away from other classes and encourages embeddings of all classes to have the same norm. The flexibility induced by the balancing hyperparameter ϵ_{ik} makes the SEN particularly suited to enable discriminative feature normalization in PNs.

Our proposed SEN dissimilarity measure explicitly encourages the norm of samples to have the same value during training, while preserving the convexity and the simplicity of the loss function. At test time, a positive epsilon ($\epsilon_{ik} = \epsilon_p > 0$) is used for computing all dissimilarity measures.

In the next section, we provide a theoretical analysis showing that our proposed SEN dissimilarity measure together with the special balancing hyperparameter ϵ_{ik} explicitly pulls the data to a scaled unit hypersphere during training.

3.4 Theoretical analysis

The partial derivative of the negative log probability loss $J(\phi)$ with respect to $d_s(\mathbf{z}_i, \mathbf{c}_k)$ is given by:

$$\frac{\partial J(\phi)}{\partial d_s(\mathbf{z}_i, \mathbf{c}_k)} = \sum_k (1[y_i = k] - p_\phi(y_i = k|\mathbf{x})), \tag{11}$$

where the Iverson bracket indicator function $[y_i = k]$ evaluates to 1 when $y_i = k$ and 0 otherwise. The partial derivative of the SEN dissimilarity measure $d_s(\mathbf{z}_i, \mathbf{c}_k)$ with respect to $\mathbf{z}_i$ is given by:

$$\begin{aligned}
\frac{\partial d_s(\mathbf{z}_i, \mathbf{c}_k)}{\partial \mathbf{z}_i} &= \frac{\partial \sqrt{d_e(\mathbf{z}_i, \mathbf{c}_k) + \epsilon_{ik} d_n(\mathbf{z}_i, \mathbf{c}_k)}}{\partial \mathbf{z}_i} \\
&= \frac{(\mathbf{z}_i - \mathbf{c}_k) + \epsilon_{ik}(\|\mathbf{z}_i\| - \|\mathbf{c}_k\|)\frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}}{d_s(\mathbf{z}_i, \mathbf{c}_k)} \\
&= -\frac{(\mathbf{c}_k - \mathbf{z}_i) + \epsilon_{ik}(\|\mathbf{c}_k\| - \|\mathbf{z}_i\|)\frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}}{d_s(\mathbf{z}_i, \mathbf{c}_k)} \\
&= -\frac{v(\mathbf{z}_i, \mathbf{c}_k)}{d_s(\mathbf{z}_i, \mathbf{c}_k)},
\end{aligned} \tag{12}$$

where

$$v(\mathbf{z}_i, \mathbf{c}_k) = (\mathbf{c}_k - \mathbf{z}_i) + \epsilon_{ik}(\|\mathbf{c}_k\| - \|\mathbf{z}_i\|)\frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}. \tag{13}$$

Using the chain rule, we get:

$$\begin{aligned}
\frac{\partial J(\phi)}{\partial \mathbf{z}_i} &= \frac{\partial J(\phi)}{\partial d_s(\mathbf{z}_i, \mathbf{c}_k)} \frac{\partial d_s(\mathbf{z}_i, \mathbf{c}_k)}{\partial \mathbf{z}_i} \\
&= \sum_k -\frac{1[y_i = k] - p_\phi(y_i = k|x)}{d_s(\mathbf{z}_i, \mathbf{c}_k)} v(\mathbf{z}_i, \mathbf{c}_k) \\
&= \sum_k \frac{\partial J_k(\phi)}{\partial \mathbf{z}_i}.
\end{aligned} \tag{14}$$

Thus, there is a gradient contribution from all prototypes. In particular, the gradient contribution with respect to the correct prototype, when $k = k^* = y_i$, is given by:

$$\begin{aligned}
\frac{\partial J_{k^*}(\phi)}{\partial \mathbf{z}_i} &= -\frac{1 - p_\phi(y_i = k^*|x)}{d_s(\mathbf{z}_i, \mathbf{c}_{k^*})} v(\mathbf{z}_i, \mathbf{c}_{k^*}) \\
&= -\frac{1 - p_\phi(y_i = k^*|x)}{d_s(\mathbf{z}_i, \mathbf{c}_{k^*})} v_p(\mathbf{z}_i, \mathbf{c}_{k^*}),
\end{aligned} \tag{15}$$

where

$$v_p(\mathbf{z}_i, \mathbf{c}_{k^*}) = \underbrace{(\mathbf{c}_{k^*} - \mathbf{z}_i)}_{\text{attractor}} + \underbrace{\epsilon_{ik^*}(\|\mathbf{c}_{k^*}\| - \|\mathbf{z}_i\|)\frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}}_{\text{norm equalizer}}. \tag{16}$$

The gradient contribution with respect to incorrect prototypes, when $k = k' \neq y_i$, is given by:

$$\frac{\partial J_{k'}(\phi)}{\partial \mathbf{z}_i} = -\frac{0 - p_\phi(y_i = k'|x)}{d_s(\mathbf{z}_i, \mathbf{c}_{k'})} v(\mathbf{z}_i, \mathbf{c}_{k'}) = -\frac{p_\phi(y_i = k'|x)}{d_s(\mathbf{z}_i, \mathbf{c}_{k'})} v_n(\mathbf{z}_i, \mathbf{c}_{k'}), \tag{17}$$

where

$$v_n(\mathbf{z}_i, \mathbf{c}_{k'}) = \underbrace{(\mathbf{z}_i - \mathbf{c}_{k'})}_{\text{repeller}} + \underbrace{\epsilon_{ik'}(\|\mathbf{z}_i\| - \|\mathbf{c}_{k'}\|)\frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}}_{\text{norm equalizer}}. \tag{18}$$

From the preceding analysis, we observe the following:

1. Each gradient component contains an attractor/repeller, which encourages $\mathbf{z}_i$ to move towards the correct prototype and move away from the incorrect ones.
2. From (16), it is clear that if $\|\mathbf{c}_{k^*}\| > \|\mathbf{z}_i\|$ and $\epsilon_{ik^*} > 0$, $\epsilon_{ik^*}(\|\mathbf{c}_{k^*}\| - \|\mathbf{z}_i\|)\frac{1}{\|\mathbf{z}_i\|} > 0$, such that $\|\mathbf{z}_i\|$ is encouraged to increase (and vice verca for $\|\mathbf{z}_i\| > \|\mathbf{c}_{k^*}\|$).
3. Conversely, from (18), if $\|\mathbf{c}_{k'}\| > \|\mathbf{z}_i\|$ and $\epsilon_{ik'} > 0$, $\epsilon_{ik'}(\|\mathbf{z}_i\| - \|\mathbf{c}_{k'}\|)\frac{1}{\|\mathbf{z}_i\|} < 0$ (and vice verca for $\|\mathbf{z}_i\| > \|\mathbf{c}_{k'}\|$). Thus, we need $\epsilon_{ik'} < 0$ in order to ensure similar behaviors as with the correct prototype.

Observation 2) and 3) shows that the gradient contributions with respect to the correct prototype and the incorrect ones *cooperate* in order to equalize the norms during training when $\epsilon_{ik^*} > 0$ and $\epsilon_{ik'} < 0$.

4 Experiments

To evaluate the effectiveness of the proposed SEN dissimilarity measure, we compare our proposed SEN PN approach with the original PN [19] and state-of-the-art distance metric learning-based approaches on the Mini-Imagenet [17, 23] and the Omniglot [9] dataset. Further, additional ablation studies are also performed on the Fewshot-CIFAR100 (FC100) [14] dataset.

4.1 Experimental Setup and Results

Embedding networks We utilize the same embedding network as that used by the original PN. Specifically, our network, which we refer to as 4CONV, comprises of four convolutional blocks. Each block is composed of 64 3×3 convolutional filters, a batch normalization layer, a ReLU nonlinearity, and a 2×2 max-pooling layer. To test the performance of the SEN dissimilarity measure in more general settings, we employ a more sophisticated network, the Wide Residual Network (WRN) [28], as the embedding network. We use the same network architecture proposed in [3], which is a network of depth 16 and a widening factor of 6. We train the network with both the traditional Euclidean distance (WRN PN) and the SEN dissimilarity measure (SEN WRN PN).

Model	Network	Omniglot	Mini-Imagenet
Original PN [19]	4CONV	98.9%	68.2%
Large Margin GNN [27]	4CONV	**99.2%**	67.6%
Large Margin PN [27]	4CONV	98.7%	66.8%
RN [20]	4CONV	99.1%	65.3%
Matching Nets [23]	4CONV	98.7%	60.0%
MetaGAN + RN [29]	4CONV	**99.2%**	68.6%
Semi-Supervised PN [3]	4CONV	-	65.5%
PN (ours, baseline)	4CONV	98.6%	67.8%
SEN PN (ours)	**4CONV**	98.8%	**69.8%**
Supervised WRN PN [3]	WRN	-	69.6%
Semi-Supervised WRN PN [3]	WRN	-	70.9%
WRN PN (ours)	WRN	99.2%	71.0%
SEN WRN PN (ours)	**WRN**	**99.4%**	**72.3%**

Table 1. Few-shot classification accuracy.

Hyperparameter ϵ For SEN-based models, during training, $\epsilon_p = 1.0$ is used for computing the SEN between the query example and its prototype, while $\epsilon_n = -10^{-7}$ to compute the SEN between the query example and the other prototypes. During testing, $\epsilon_p = 1.0$ is used for computing all the SEN dissimilarity measures. A discussion on how the hyperparameters ϵ_p and ϵ_n were chosen can be found in the supplementary.

Results The test results are shown in Table 1. As can be seen from Table 1, although our implementation of the PN (the baseline model) achieves 0.4 percentage points lower in terms of accuracy compared to the original implementation of the PN (67.8% vs 68.2%), the baseline model trained with the proposed SEN dissimilarity measure still outperforms the original PN by obtaining a relative increase of 2.4% and achieves an accuracy of 69.8%. In addition, the SEN WRN PN outperforms the Semi-Supervised WRN PN by a relative increase of 2% and achieves an accuracy of 72.3% with the WRN as the embedding network.

Similar trends can be observed for the Omniglot dataset, where SEN PN outperforms our PN implementation and SEN WRN PN outperforms WRN PN.

4.2 Ablation Study

To investigate the effectiveness and behavior of the proposed SEN dissimilarity measure, we conduct several ablation studies. First, we compare against the PN trained with the Euclidean distance (PN), the PN trained with the Ring loss (Ring PN), and the PN trained with the SEN dissimilarity measure (SEN PN). The test results are show in Table 2. We train the Ring PN with different values of γ, the loss weight w.r.t to the primary loss, in range $[10^{-10}, 1]$ and pick $\gamma = 10^{-7}$ since it results in the highest accuracy. R was learned during training

Model	Omniglot	Mini-Imagenet	FC100
PN	98.6%	67.8%	52.4%
Ring PN	98.7%	68.6%	52.8%
SEN PN	**98.8%**	**69.8%**	**54.6%**

Table 2. Few-shot classification accuracy on the Omniglot [9] (20-way 5-shot), the Mini-Imagenet [17, 23] (5-way 5-shot), and the FC100 [14] (5-way 5-shot) datasets.

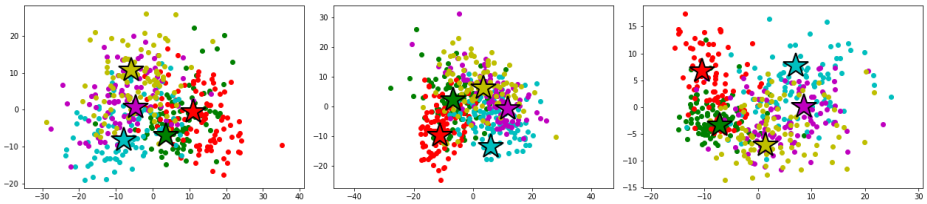

Fig. 1. 2D embeddings produced by the PN (left), the Ring PN (middle) and the SEN PN (right). The circles denote query examples, and the stars denotes prototypes.

following [30]. As can be seen from Table 2, the Ring loss improves the accuracy relative to the PN on the Mini-Imagenet dataset by 1.8%; however, it performs worse than our proposed SEN PN approach, which obtains a relative increase of 3%. Similar behavior is obtained for other few-shot learning datasets such as FC100 and Omniglot. A more thorough discussion on SEN PN vs Ring PN can be found in the supplementary.

Principal Component Analysis (PCA) We project 1600D embeddings produced by the PN, the Ring PN, and the SEN PN to 2D space using PCA and visualize the outputs (see Figure 1). As can be seen from Figure 1, the Ring loss forces the prototypes to lie on a scaled unit hypersphere; however, the prototypes produced by the Ring PN are not very well-separated compared to the ones produced by the PN. On the other hand, our proposed SEN dissimilarity measure both forces the prototypes to lie on a scaled unit hypersphere and keeps them well-separated.

Analysis of norm We plot the norm of embeddings produced by the PN, the Ring PN, and the SEN PN. As can be seen from Figure 2, the norm of embeddings produced by the PN and the Ring PN vary a lot, while the norm of embeddings produced by the SEN PN has a very consistent value. This confirms that SEN encourages all embeddings to have the same norm during training. Both the SEN and the Ring loss are adopted for explicitly enforcing their embeddings to have the same norm during the training of the PN. However, as can be seen from Figure 2, the proposed SEN dissimilarity measure is a better choice for the task than the Ring loss. This is partly due to the use of a very small gamma ($\gamma = 10^{-7}$) during training the Ring PN. In our experiments, higher gamma values do encourage the norm of embeddings to have a more consistent value; however, they cause a considerable decrease in the accuracy of the PN. This suggests that the Ring loss is not an optimal choice for enforcing feature normalization in PNs.

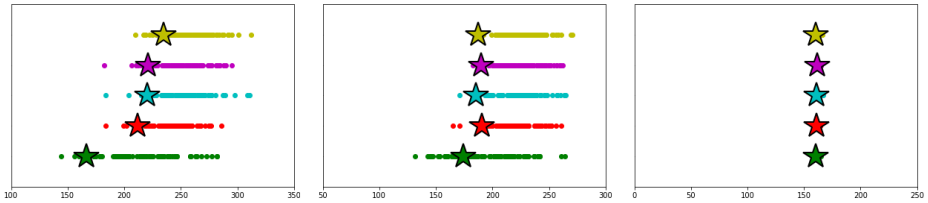

Fig. 2. The norm of embeddings produced by the PN (left), the Ring PN (middle), and the SEN PN (right). The stars denote query examples, and the diamonds denotes prototypes.

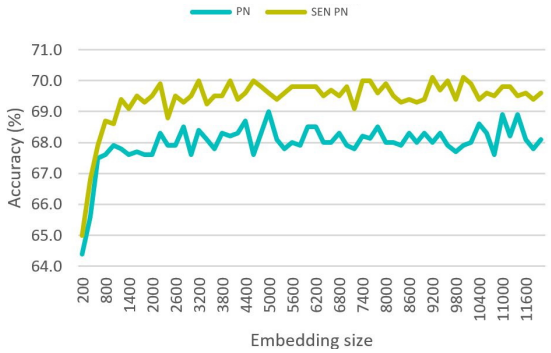

Fig. 3. The PN vs the SEN PN with different embedding sizes.

The proposed SEN dissimilarity measure; on the other hand, both encourages all embeddings to have the same norm and improves the accuracy of PNs. This indicates that the proposed SEN dissimilarity measure is a more suitable choice for feature normalization than the Ring loss in training PNs.

Analysis of embedding dimensionality We compare between the PN and the SEN PN trained with different embedding sizes (see Figure 3). As can be seen from Figure 3, in low dimensional spaces, the PN and the SEN PN perform very similarly; however, in high dimensional spaces, the SEN PN consistently outperforms the PN by a considerable margin. This suggests that the SEN dissimilarity measure is a more suitable distance metric for metric distance learning-based few-shot learning than the standard Euclidean distance in high dimensional spaces. This further explains the limited improvement on the Omniglot dataset where the embedding size is 64 compared to 1600 for the remaining datasets.

Analysis of distance We evaluate the possibility of combining the proposed SEN dissimilarity measure with other distance functions such as the Euclidean distance and the cosine distance in training PNs. Specifically, we train the PN with the SEN dissimilarity measure and test the trained model with both the Euclidean distance and the cosine distance. We compare the two tested models with the original PN, the SEN PN, and the Cosine PN (the PN trained and tested with the cosine distance). The test results are show in Table 3.

Train distance	Test distance	Omniglot	Mini-Imagenet	FC100
Cosine	Cosine	61.5%	53.3%	44.9%
Cosine	SEN	55.2%	51.4%	43.8%
Euclidean	Euclidean	98.6%	67.8%	52.4%
Euclidean	SEN	98.7%	68.5%	53.1%
SEN	**SEN**	**98.8%**	**69.8%**	**54.6%**
SEN	Euclidean	**98.8%**	68.8%	53.9%
SEN	**Cosine**	**98.8%**	**69.8%**	**54.6%**

Table 3. Test results of the PN with different distances on the Omniglot [9] (20-way 5-shot), the Mini-Imagenet [17, 23] (5-way 5-shot), and the Fewshot-CIFAR100 (FC100) [14] (5-way 5-shot) datasets.

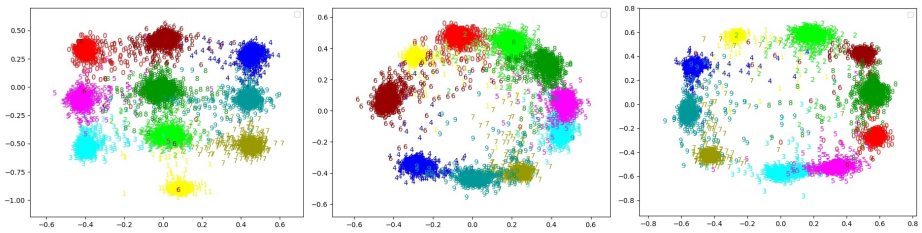

Fig. 4. 2D embeddings produced by the Siamese Baseline (left), the Siamese Ring (middle), and the Siamese SEN (right).

As can be seen from Table 3, the model trained with the SEN dissimilarity measure achieves the highest accuracy on the Mini-Imagenet, the FC100, and the Omniglot datasets when tested with either the SEN dissimilarity measure or the cosine distance. This is because the SEN dissimilarity measure explicitly forces all embeddings to have the same norm during training, and, as a result, pulling the prototypes very close to the hypersphere. For data embedded on a hypersphere, the cosine distance is a natural measure of distance [5, 2]. Experiments and discussions on alternative design choices for SEN can be found in the supplementary.

SEN beyond few-shot learning We have demonstrated that the SEN dissimilarity measure outperforms the commonly used Euclidean distance in distance metric learning-based few-shot learning with prototypical networks. In this section, we study the behaviors of the proposed SEN in combination with other metric learning-based tasks, which are based on the idea of obtaining inter-class separability and intra-class compactness. Note, due to the lack of prototypes, the SEN distance is here computed between datapoints directly. To do this, we implement the well-known Siamese network and Contrastive loss [7]. We call this model the Siamese Baseline. We augment it by replacing the Euclidean distance by our proposed SEN dissimilarity measure (Siamese SEN) and by employing Ring loss (Siamese Ring). We train the three models on the MNIST dataset [10]

for dimensionality reduction and clustering. During training the Siamese SEN, following the reasoning of Section 3.3, a positive epsilon ($\epsilon_{ik} = \epsilon_p > 0$) is used for computing the SEN dissimilarity measures between examples of the same class, and a negative epsilon ($\epsilon_{ik} = \epsilon_n < 0$) is used for computing the SEN dissimilarity measures between examples of different classes. At test time, a positive epsilon ($\epsilon_{ik} = \epsilon_p > 0$) is used for computing all dissimilarity measures.

As can be seen from Figure 4, the Siamese Ring forces all embeddings to lie on a scaled unit hypersphere; however, embeddings produced by the Siamese Ring are not as well-separated as embeddings produced by the Siamese Baseline. Our proposed SEN dissimilarity measure, on the other hand, both forces all embeddings to lie on a scaled unit hypersphere and keeps the embeddings well-separated. This suggests that SEN can also be used beyond the field of few-shot learning where distance metric learning is used and class memberships are available. In future work, other promising lines of research are to combine feature normalization with weight normalization techniques [11] and analyze their synergy, as well as to analyze the potential of SEN in other prototype-based methods [12].

5 Conclusion

In this paper, we propose a novel dissimilarity measure, called SEN, for distance metric learning-based few-shot learning by modifying the traditional Euclidean distance to attenuate the curse of dimensionality in high dimensional spaces. The SEN is a combination of the Euclidean distance and the norm distance. We extend the prototypical network by replacing the Euclidean distance by our proposed SEN dissimilarity measure, which we refer to as SEN PN. With minimal modifications, the SEN PN outperforms the original PN by a considerable margin and demonstrates good performance on the Mini-Imagenet, the FC100, and the Omniglot datasets with no additional parameters as well as negligible computational overhead. We provide analyses showing that the proposed SEN dissimilarity measure encourages the embeddings to have the same norm and enables the SEN PN to generate a hyperspherical embedding space, which is a more compact embedding space than the Euclidean space. We experimentally show that the proposed SEN dissimilarity measure consistently outperforms the Euclidean distance in PNs with different embedding sizes as well as with different embedding networks. We also show that SEN is an effective feature normalization technique not only for distance metric learning-based few-shot learning with PNs but also potentially for more general tasks, here exemplified by the Siamese network.

References

1. Adam, S., Sergey, B., Matthew, B., Daan, W., Timothy, P.L.: One-shot learning with memory-augmented neural networks. CoRR **abs/1605.06065** (2016), http://arxiv.org/abs/1605.06065
2. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. The Journal of Machine Learning Research **6**, 1345–1382 (2005)
3. Boney, R., Ilin, A.: Semi-supervised few-shot learning with prototypical networks. CoRR **abs/1711.10856** (2017), http://arxiv.org/abs/1711.10856
4. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
5. Dhillon, I.S., Fan, J., Guan, Y.: Efficient clustering of very large document collections. In: Data mining for scientific and engineering applications, pp. 357–381. Springer (2001)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1126–1135. JMLR. org (2017)
7. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 1735–1742. IEEE (2006)
8. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. vol. 2 (2015)
9. Lake, B., Salakhutdinov, R., Gross, J., Tenenbaum, J.: One shot learning of simple visual concepts. In: Proceedings of the annual meeting of the cognitive science society. vol. 33 (2011)
10. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
11. Liu, W., Lin, R., Liu, Z., Liu, L., Yu, Z., Dai, B., Song, L.: Learning towards minimum hyperspherical energy. In: Advances in neural information processing systems. pp. 6222–6233 (2018)
12. Mettes, P., van der Pol, E., Snoek, C.: Hyperspherical prototype networks. In: Advances in Neural Information Processing Systems. pp. 1485–1495 (2019)
13. Nikhil, M., Mostafa, R., Xi, C., Pieter, A.: A simple neural attentive meta-learner. CoRR **abs/1707.03141** (2017), http://arxiv.org/abs/1707.03141
14. Oreshkin, B.N., Rodriguez, P., Lacoste, A.: Tadam: task dependent adaptive metric for improved few-shot learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 719–729. Curran Associates Inc. (2018)
15. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained softmax loss for discriminative face verification. arXiv preprint arXiv:1703.09507 (2017)
16. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. CoRR **abs/1803.00676** (2018), http://arxiv.org/abs/1803.00676
17. Sachin, R., Hugo, L.: Optimization as a model for few-shot learning. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017), https://openreview.net/forum?id=rJY0-Kcll

18. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. pp. 815–823 (2015). https://doi.org/10.1109/CVPR.2015.7298682, https://doi.org/10.1109/CVPR.2015.7298682
19. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 4080–4090. Curran Associates Inc. (2017)
20. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1199–1208. IEEE (2018)
21. Vanschoren, J.: Meta-learning: A survey. CoRR **abs/1810.03548** (2018), http://arxiv.org/abs/1810.03548
22. Victor, G., Joan, B.: Few-shot learning with graph neural networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=BJj6qGbRW
23. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 3637–3645. Curran Associates Inc. (2016)
24. Wang, F., Xiang, X., Cheng, J., Yuille, A.L.: Normface: L2 hypersphere embedding for face verification. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1041–1049. ACM (2017)
25. Wang, Y., Yao, Q.: Few-shot learning: A survey. CoRR **abs/1904.05046** (2019), http://arxiv.org/abs/1904.05046
26. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7278–7286. IEEE (2018)
27. Yong, W., Xiao-Ming, W., Qimai, L., Jiatao, G., Wangmeng, X., Lei, Z., Victor, O.K.L.: Large margin few-shot learning. CoRR **abs/1807.02872** (2018), http://arxiv.org/abs/1807.02872
28. Zagoruyko, S., Komodakis, N.: Wide residual networks. CoRR **abs/1605.07146** (2016), http://arxiv.org/abs/1605.07146
29. Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: Metagan: an adversarial approach to few-shot learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 2371–2380. Curran Associates Inc. (2018)
30. Zheng, Y., Pal, D.K., Savvides, M.: Ring loss: Convex feature normalization for face recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5089–5097. IEEE (2018)