**Deep Hierarchical-Hyperspherical Learning (DH$^2$L)**

**Anonymous authors**

Paper under double-blind review

**ABSTRACT**

Regularization is known to be an inexpensive and reasonable solution to alleviate the overfitting problem arising in inference models, including deep neural networks. In this paper, we propose hierarchical regularization which preserves the semantic structure of a sample distribution. At the same time, this regularization promotes diversity by imposing a distance between parameter vectors enlarged within semantic structures. To generate evenly distributed parameters which is considered less redundant, we constrain them to lie on hierarchical hyperspheres. To define a hierarchical parameter space, we propose to reformulate the topology space with multiple hyperspheres. On each hypersphere, a projection is parameterized by two individual parameters. Since maximizing a groupwise and pairwise distance between points on hypersphere is nontrivial (generalized Thomson problem), we propose a new discrete metric integrated with a continuous metric. The proposed method shows improved generalization performance on extensive experiments using publicly available datasets (CIFAR-10, CIFAR-100, CUB200-2011, and Stanford Cars), especially when the number of super-classes is large.

1 **INTRODUCTION**

Diversity promoting learning has been widely adopted via enlarging pairwise distances \cite{Xie2018, Xie2017a, Liu2018}, increasing orthogonality \cite{Xie2018}, reducing covariance between parameters \cite{Xie2017b}, or reducing correlation on feature \cite{Cogswell2016} to improve generalization performance. Among them, diversity promoting regularization \cite{Xie2017a, Liu2018} by enforcing large diversity between projection parameters achieves a reasonable performance without modifying the model structure. While optimizing the objective function with a covariance matrix in \cite{Xie2017b, Liu2018} is nontrivial, the diversity promoting regularization via minimizing energy of parameters of deep neural networks has been proposed \cite{Liu2018} in a simpler way. By minimizing a pairwise distance between parameters on a hypersphere with the known metric, they achieved the improved generalization performance.

Following an efficient regularization on the hypersphere, we explore further this direction with three main concepts (hierarchical and hyperspherical learning with discrete metrics).

1) **Why hierarchical learning?** Hierarchical inference explains human intelligence. In \cite{Kurzweil2013}, it states that “the neocortex contains about 300 million very general pattern recognizers, arranged in a hierarchy”. Applying the hierarchy of multiple classes based on semantic taxonomy is a natural choice to devise machine intelligence. Effectiveness of the hierarchical learning can be found in \cite{Verma2012}.

2) **Why hyperspherical learning?** Hypersphere is the set of points at the equidistance (radius) from a given point (centroid) in a certain dimensional space. Due to the denominator in the unit-length normalization ($\frac{\mathbf{w}}{||\mathbf{w}||}$, $\mathbf{w} \in \mathbb{R}^{d+1}$), the angular distance defined on the hypersphere converges when the magnitude of $\mathbf{w}$ goes infinity while Euclidean distance goes infinity. Due to this bounded property, a hierarchical structure with multiple separated hyperspheres can be defined.

3) **Why discrete metric learning?** If vector points form discontinuous series with discrete representation (e.g. multi-dimensional binary or ternary), they are isolated from each other with a certain margin. This property may fit with a disconnected/groupwise manifold space problem. We note that making points to be equidistributed where a pairwise distance is maximized is a nontrivial task. As
Figure 1: Multiple (hyper)spheres as quotient spaces of a topology space on Euclidean space might be found by gluing process with identifying points. Those separated hyperspheres are assumed to be under the quotient space conditions (Tu, 2010). Within an individual (hyper)sphere, projection parameters in deep neural networks preserve a hierarchical structure. The space can be formed in a series: (a), (b), (c), and (d)

points in a discrete metric space are finite and isolated, search efforts to satisfy those constraints can be reduced.

In this paper, we propose to apply a hierarchical structure to parameter regularization on the multiple groupwise hyperspheres. In order to find an appropriate metric in this space, we explore a discrete angular metric. We examine the proposed method on extensive experimental setups in terms of datasets and deep network models.

2 MULTIPLE SEPARATED HYPERSPHERES

Samples observed from the real world may be on disconnected manifolds. In other words, the disjoint union of those manifolds can generate the global manifold (Lee, 2000). In this section, we decompose the one space into multiple spaces (manifolds) and re-define the spaces in a hierarchical point of view.

2.1 DISCONNECTED MANIFOLD VIA EQUIVALENT RELATIONS

Since it is not suitable to measure a pairwise distance between high dimensional vectors which have the hierarchical structure in the same space, we construct another identification space where sub-spaces (sub-manifolds) are isolated (via equivalence relation (Tu, 2010)). Denote $d$-sphere $S^d$ to be the set of points that satisfies $S^d = \{ w \in \mathbb{R}^{d+1} : ||w|| = 1 \}$. We construct multiple separated hyperspheres using multiple identifying relations. In Figure 1, we use a center parameter and a surface parameter to define a projection parameter vector $w$ which constitutes a hypershpere space.

2.2 PRIOR DISTRIBUTION AND REGULARIZATION

To make parameter vectors uniformly distributed on the unit hypersphere, parameters are sampled from the Gaussian normal distribution (Muller, 1959; Harman & Lacko, 2010). This is because the normal distribution is spherically symmetric (Muller, 1959). In a Bayesian point of view, neural networks with Gaussian priors are known to induce $l^2$-norm regularization (Vladimirova et al., 2019). From two evidences, we know that enforcing parameters to have the Gaussian prior is important for hyperspherical learning in neural networks. Note that a parameter which is calculated from the difference arithmetic operation with two parameters on the normal Gaussian distribution is on the normal difference distribution.

3 METHOD

In deep neural networks, the objective function $J$ with regularization $R$ in addition to a loss $L$, $J_R(W) = L(x, W) + \lambda R(W)$, is optimized to find the optimal $W$ having a near minimum loss $L$, $\arg \min W J_R(x, W)$, where $x \in \mathbb{R}^d$ denotes an input vector, $W = \{W_i \in \mathbb{R}^{d_i-1 \times c_i} : W_i = \{w_j \in \mathbb{R}^{d_j-1} \}, j = 1, \ldots, c_i, i = 1, \ldots, L\}$ denotes a set of parameter matrices (i.e. neurons/kernels), $L$ denotes the number of layers, and $\lambda > 0$ is to control the degree of the regularization. For a classification task, the cross entropy loss is used for the loss function $L$. We propose a new regularization formulation $R$ preserving a hierarchical parameter structure in Section 3.1.
With a group notation

Figure 2: A series of levelwise \((l)\) spheres with a radius \((r_l)\) is shown: (a) A radius of an overall area converges to \(r_0 = \sum_1^\infty r_k^\delta\) \(\ (= \sum_1^\infty r_0^\delta\) \(\) the sum of radius series \(\) as \(l\) goes to infinity where \(l\) denotes a level, \(r_0\) is their initial radius, and \(\delta\) is the ratio between radiuses \((r_l, r_{l-1})\). \(\frac{r_l}{r_{l-1}} < 1\). (b) The radius of the overall area is bounded to the initial radius \(r_0\) of spheres. This bears a resemblance to the process of repeat of Hypersphere packing which arranges non-overlapping spheres within a containing space. (c) 2-sphere is defined following (b) which is appropriate to model a hierarchical structure. This can be generalized with a hypersphere \(\mathbb{S}^d, d \geq 3\) in a higher dimensional space.

3.1 REGULARIZATION FOR HIERARCHICAL AND HYSPHERICAL HYPOTHESES

Denote \(w\) a parameter vector (called a projection parameter, an element of \(W\) at a certain layer) to transform a given input into an embedding space defined in a Euclidean metric space: \(x \in \mathbb{R}^{d+1} \mapsto w^T \cdot x \in \mathbb{R}\). By the definition of unit-length projection \(\frac{w}{\|w\|}\), a new parameter vector \(\hat{w}\) can be defined on \(d\)-sphere: \(\mathbb{S}^d = \{\hat{w} \in \mathbb{R}^{d+1} : \|\hat{w}\| = 1\}\) where \(\|\cdot\|\) denotes \(l^2\)-norm and a center is zero. In other word, the parameter \(\hat{w}\) can be defined by a center parameter vector \(w_c \in \mathbb{R}^{d+1}\) and a surface parameter vector \(w_s \in \mathbb{R}^{d+1}\) using an arithmetic operation: \(\hat{w} := w_s - w_c\). We define the \(d\)-sphere with the center and surface parameters: \(\mathbb{S}^d_{w_c, w_s} = \{w_s - w_c \in \mathbb{R}^{d+1} : \|w_s - w_c\| = 1\}\). For a notation simplicity, we use \(w\) instead of \(\hat{w}\) hereafter. While we consider a radius equals to 1 for simplicity, the parameter vector can have a radius \(r > 0\).

3.1.1 HIERARCHICAL PARAMETERIZATION VIA LEVELWISE AND GROUPWISE PARAMETERS

We assume that the hierarchical structure consists of a levelwise structure with a notation \((l)\) and a groupwise structure with a notation \(g\). Through defining these structures serially, we explain hierarchical parameters.

**Levelwise structure** The projection parameter \(w \in \mathbb{R}^{d+1}\) on \(\mathbb{S}^d_{w_c, w_s}\) can be defined with a levelwise notation \((l)\) as follows,

\[
\begin{align*}
\begin{aligned}
\hat{w} &= \hat{w}_s(l) - \hat{w}_c(l) \\
\hat{w}_s(l) &= \hat{w}_{s, l-1}(l) + \Delta \hat{w}_s(l) \\
\hat{w}_c(l) &= \hat{w}_{c, l-1}(l) + \Delta \hat{w}_c(l)
\end{aligned}
\end{align*}
\]  

(1)

where \(\mathbb{S}^d_{w_c, w_s}\) denotes the \(l\)-th level \(d\)-sphere centered at \(w_c(l)\). In this paper, we define the parameter in a higher dimensional space than that shown in Figure 2.

In a levelwise setting, \(w_s(l)\) and \(w_c(l)\) at the current level \((l)\) are additively represented based on a center parameter of the previous level \((l-1)\): \(w_c(l-1) + \Delta w_c(l) \mapsto w_c(l)\), where \(w_c(l-1) = \sum_{i=1}^{l-1} \Delta w_c(l)\) is accumulated and \(\Delta w_c(l)\) denotes a connecting parameter from \(w_{c, l-1}(l)\) to \(w_{c, l}(l)\). By denoting \(\Delta w_s(l)\) as \(w_s(l, l-1)\), the center vector at the \(l\)-th level is defined as, \(w_c(l) := w_{c, l-1}(l) + w_c(l-1)\), and the surface vector as \(w_s(l) := w_{s, l-1}(l) + w_s(l-1)\). Both the center parameter and the surface parameter at the current level are based on the center parameter at the previous level.\(^4\) Hence, Eq. (1) is equivalent to

\[
\begin{align*}
\hat{w}(l) &= w_{s, l-1}(l) - w_{c, l-1}(l)
\end{align*}
\]  

(2)

Note that we use a superscript \((l, l-1)\) to denote parameters at the \(l\)-th level connected from the center parameter at the \((l-1)\)-th level.

**Groupwise structure** With a group notation \(g_k\), the center parameter in Eq. (2) can be rephrased as \(w_{c, g_k}(l, l-1)\), on \(\mathbb{S}^d_{w_{c, g_k}, w_{s, g_k}}\) which denotes \(d\)-sphere of \(g_k\) group at the \(l\)-th level. Each group constitutes a group set at the \(l\)-th level \(g^{(l)} = \{g_k\}_{k=1}^{\|g(l)\|}\) where \(g^{(l)} \subseteq G^{(l)}\), \(G^{(l)}\) denotes a batch group set at the \(l\)-th level, and \(\|\cdot\|\) denotes the cardinality. The group set \(g^{(l)}\) at the \(l\)-th level is conditioned on

\(^4\)As not every sample has a child sample, it might be more reasonable to branch from representative parameter or center parameter rather than from individual projection parameters.
a group set at the \((l - 1)\)-th level, \(g^{(l-1)} := \{ g_{k}^{(l-1)} \}_{k=1}^{d} \) where \( g^{(l-1)} \subseteq G^{(l-1)} \). By their group-wise relation over levels, an adjacency indication \( P^{(l-1)}(\{ G^{(l-1)}, G^{(l)} \}) \in \{0, 1\}^{[G^{(l-1)}] \times [G^{(l)}]} \)

is given. Hence, the \( i \)-th parameter vector on \( S^{d}_{s,g_{k},l} \) is defined as follows,

\[
\mathbf{u}_{i}^{(l)} := \mathbf{w}_{g_{k},i}^{(l-1)} - \mathbf{w}_{c,g_{k}}^{(l-1)},
\]

where \( \{ \mathbf{w}_{g_{k},i}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)} \} \forall i \) is calculated based on a center vector \( \mathbf{w}_{c,g_{k}}^{(l-1)} \) at the \((l - 1)\)-th level\footnote{This can be replaced with a probability model.}, \( i = 1, \ldots, |g_{k}| \), and \(|g_{k}| \) denotes the number of surface parameter vectors in the group \( g_{k} \).

### 3.1.2 Hierarchical regularization

In this section, we define a regularization term using the hierarchical parameters defined above. A set of parameters \( \{ \mathbf{W}_{s,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)} \} \in \mathbf{W} \forall g_{k}, \forall g_{k}^{c} \) where \( \mathbf{W}_{s,g_{k}}^{(l-1)} := \{ \mathbf{w}_{s,g_{k},i}^{(l-1)} \}_{i=1}^{d} \) is an optimizing target of hierarchical regularization as follows:

\[
\mathcal{R}(\mathbf{W}) := \sum_{i} \lambda_{i} \mathcal{R}_{i}(\mathbf{W}_{s,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)}, \mathbf{P}^{(l-1)}) + \sum_{l} \mathcal{R}_{l}(\mathbf{w}_{c,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)}, \mathbf{P}^{(l-1)})
\]

where \( \mathcal{R}_{i} \) works on individual spheres \( S^{d}_{s,g_{k}} \), \( \lambda_{i} > 0 \), and \( \mathcal{R}_{l} \) aims to apply geometry-aware constraints across spheres. \( \mathcal{R}_{i} \) consists of two terms for regularization: \( i ) \mathcal{R}_{i,p} \) for projection parameters in the same group \( g_{k} \) on \( S^{d}_{s,g_{k}} \) and \( 2) \mathcal{R}_{i,c} \) for center parameters across groups in the same level on \( S^{d}_{s,g_{k}} \).

\[
\mathcal{R}_{i}(\mathbf{W}_{s,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)}) := \mathcal{R}_{i,p}(\mathbf{W}_{s,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)}) + \mathcal{R}_{i,c}(\mathbf{w}_{c,g_{k}}^{(l-1)}),
\]

where

\[
\mathcal{R}_{i,p}(\mathbf{W}_{s,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)}) := \frac{1}{G} \sum_{(g_{k},i,j) \in G_{k}} \frac{2}{G_{k}(G_{k} - 1)} \sum_{d(\mathbf{w}_{g_{k},i}^{(l-1)}, \mathbf{w}_{g_{k},j}^{(l-1)})},
\]

\[
\mathcal{R}_{i,c}(\mathbf{w}_{c,g_{k}}^{(l-1)}) := \frac{2}{C(C - 1)} \sum_{(g_{k},i,j) \in G_{k}} \sum_{d(\mathbf{w}_{c,g_{k}}^{(l-1)}, \mathbf{w}_{c,g_{k}}^{(l-1)})},
\]

where \( \mathbf{w}_{g_{k},i}^{(l-1)} := \mathbf{w}_{s,g_{k},i}^{(l-1)} - \mathbf{w}_{c,g_{k}}^{(l-1)} \), \( \mathbf{w}_{g_{k},j}^{(l-1)} \) is similarly defined), \( G = |\{ g_{k} \in g^{(l)} \}| \), \( G_{k} = |\{ i \neq j \in g_{k} \}| \), \( C = |\{ g_{k} \neq g_{k}^{c} \in g^{(l)} \}| \), and \( d(\cdot, \cdot) \) denotes a distance metric between parameters. The distance metric is defined in Section\footnote{The center vector \( \mathbf{w}_{c,g_{k}}^{(l-1)} \) indicates a representative vector of the group \( g_{k} \) at the \( l \)-th level and it is equivalent to a mean vector of \( \mathbf{u}_{i}^{(l)} \forall i; \mu(\mathbf{u}_{i}^{(l)}) = \frac{1}{|g_{k}|} \sum_{i} \mathbf{u}_{i}^{(l)} \). The center vector for the group \( g_{k} \) can be determined by a parameter vector at the \((l - 1)\)-th level using an adjust factor \( \epsilon, |\epsilon| < 1 \): \( \mathbf{w}_{c,g_{k}}^{(l-1)} = \epsilon \cdot \mathbf{w}_{g_{k},i}^{(l-1)} \), where \( \mathbf{w}_{g_{k},i}^{(l-1)} \in S^{d}_{s,g_{k}} \).} 2 We explain regularization terms formulated using these parameters and pairwise distances further in Figure\footnote{Under review as a conference paper at ICLR 2020} of Appendix. When a minibatch of inputs \( m_{x} \); a set of inputs \( \{ x_{i} \} \) is given, the regularization term becomes:

\[
\mathbb{E}(\mathcal{R}(\mathbf{W})) = \frac{1}{|m_{x}|} \sum_{m_{x}} \mathcal{R}(\mathbf{W}; m_{x}).
\]

The constraint term help construct geometry-aware relational parameters between different spheres on the same level and on the across levels. Multiple constraints are defined as \( \mathcal{C}_{i,k} := \sum_{k} \lambda_{i} \mathcal{C}_{i,k}^{(l-1)} \), where \( \mathcal{C}_{i,k}^{(l-1)} \) is the \( k \)-th constraint between parameters at the \( l \)-th and the \((l - 1)\)-th level, and \( \lambda_{k} > 0 \) is a Lagrange multiplier. We propose three constraints (defined in Appendix) in a geometric point of view.

### 3.2 Discrete and continuous angular distance metric

Discrete metric is a good fit for the above group-wise definition. We expect that points projected by parameters in a discrete metric space are isolated each other. In Figure\footnote{Under review as a conference paper at ICLR 2020} we show that discrete distances of parameter pairs have different values while their continuous distances are same. Hence, maximization of discrete distances of parameter pairs could help parameters distributed isolatedly and diversely.
Using parameter vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ in $\mathbb{R}^{d+1}$, we define a discrete distance metric using a sign function as follows:

$$D_h := \frac{1}{d} \sum_{k=0}^{d} \text{sign}(\mathbf{w}_i(k)) \cdot \text{sign}(\mathbf{w}_j(k)), \quad (8)$$

where $\text{sign}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases}$. -1 $\leq D_h \leq 1$, and $\mathbf{w} = \{\mathbf{w}(k) \mid \forall k = 1, \ldots, d+1 \in \mathbb{R}^{d+1}\}$. This is a normalized version of Hamming distance. For a ternary discrete, $\text{sign}(x) \in \{-1, 0, 1\}$ is used.

In order to consider the discrete distance as an angular distance within $[0, 1]$, normalized one is defined as $D_{h01} := -\frac{D_h+1}{2}$, where $0 \leq D_{h01} \leq 1$. The angular distance based on the above product can be rephrased as $\theta_{D_h} = D_{h01}$ where $0 \leq \theta_{D_h} \leq 1$.

As the discrete distance could underestimate the approximate model distance, we merge the above discrete distance metric with continuous angular distance metric ($\theta = \frac{1}{\pi} \arccos(\frac{\mathbf{w}_i \cdot \mathbf{w}_j}{||\mathbf{w}_i|| ||\mathbf{w}_j||})$, $0 \leq \theta \leq 1$) into a single metric. We simply use the definition of Pythagorean means which consist of the arithmetic mean (AM), the geometric mean (GM), and the harmonic mean (HM). Pythagorean means using a pair of angles are defined as follows:

$$D_{\text{AM}} := \frac{\theta_{D_h} + \theta}{2}, \quad D_{\text{GM}} := \theta_{D_h} \theta, \quad D_{\text{HM}} := \frac{d\theta_{D_h}}{\theta_{D_h} + \theta}. \quad (9)$$

In the angular distance\textsuperscript{5} using a pair of angles $\{\theta_{D_h}, \theta\}$, a reversed form $1 - D(\theta_{D_h}, \theta)$ is adopted to maximize an angle in the optimization formulation by minimization instead of $(\cdot)^{-s}$ where $s = 1, 2, \ldots$ which is used in Thomson problem that utilizes $s$-energy (Brauchart & Grabner, 2015). The cosine similarity using the pair of angles can be defined as follows:

$$D_{\text{cos(AM)}} := \cos\left(\frac{\theta_{D_h} + \theta}{2}\right), \quad D_{\text{cos(GM)}} := \cos(\theta_{D_h} \theta \pi), \quad D_{\text{cos(HM)}} := \cos\left(\frac{d\theta_{D_h}}{\theta_{D_h} + \theta \pi}\right). \quad (10)$$

then the cosine similarity functions are normalized with $\cos(\cdot)+1$ to have a value within a range $[0, 1]$. Finally, Pythagorean means of cosine similarities can be calculated as follows:

$$D_{\text{AM,cos}} := \frac{\cos(\theta_{D_h} \pi + \cos(\theta \pi) + 2}{4}, \quad D_{\text{GM,cos}} := \frac{\cos(\theta_{D_h} \pi + 1)(\cos(\theta \pi + 1)}{4}, \quad D_{\text{HM,cos}} := \frac{\cos(\theta_{D_h} \pi + 1)(\cos(\theta \pi + 1)}{\cos(\theta_{D_h} + \cos(\theta \pi) + 2}. \quad (11)$$

Metric functions defined in (9), (10), and (11) satisfy metric conditions: non-negativity, symmetry, and triangle inequality. The distance using the above metric function between any two parameter points is bounded, because the hypersphere is a compact manifold.

3.3 Gradients and Backpropagation

As the sign function is not differentiable at the value 0, we adopt alternative backpropagation function. We adopt straight-through estimator (STE) (Bengio et al., 2013) in the backward path of the neural networks for the sign function in the discrete metric. The derivative of the sign function is substituted with $1_{||\mathbf{w}|| \leq 1}$ in the backward pass, known as the saturated STE. As the derivative of $\arccos(x)$ ($\frac{-1}{\sqrt{1-x^2}}$) is undefined at the value $x = \pm 1$, we apply clamping to the cosine function to have $x \in [-0.99, 0.99]$ where $x = \cos(\theta \pi)$, and $0 \leq \theta \leq 1$.

4 Experiments

4.1 Experimental setup

Datasets We conduct experiments using four publicly available datasets including small size images (CIFAR-10 and CIFAR-100) and large size images (CUB200-2011 (Wah et al, 2011) and

\textsuperscript{5}On the other hand, the angle can be considered as a cosine similarity directly, $D_h := \cos(\theta_{D_h} \pi)$. So as to get the angular distance, it needs an arccosine function $\theta_{D_h} = \frac{1}{\pi} \arccos D_h$. In summary, for the angle distance $\theta_{D_h}$, either “$D_{h01}$” or “$D_{h01} = \frac{1}{\pi} \arccos D_h$” where $0 \leq D_{h01} \leq 1$, can be adopted.

\textsuperscript{6}In 0 $\leq \theta \leq 1$, the angle and its cosine value show an inverse relationship: $0 \leq \theta \leq 1 \Rightarrow 1 \geq \cos(\theta \pi) \geq -1$. $x = \{0.99 \cdot 1_{x>0.99}, x, -0.99 \cdot 1_{x<-0.99}\}$
Deep neural network models and training setting  We used the deep residual neural network (resnet) (He et al., 2016) with different sizes of parameters (weights) dependent on an image size of datasets. For a small size input ($32 \times 32$ pixels) of CIFAR-10 and CIFAR-100, we used resnet with smaller number of parameters (light models, resnet-20 (0.29M) and resnet-110 (1.73 M)) so as not to have redundant parameters leading to over-fitting. For a fine-grained input ($224 \times 224$ pixels) of CUB200 and Cars, the original resnet with larger number of parameters (heavy model, Resnet-18 (11.28M) and Resnet-50 (23.91M)) is used.

In training of the deep neural network with the hierarchical regularization, we assume that the global hierarchical structure is not given. Instead, stochastic or partial hierarchical structure is shown within each mini-batch by hierarchy pairs. Mini-batches, 512 for light models and 256 for heavy models, are used in the SGD optimizer. We applied the hierarchical regularization in the FC layer of the resnet. Even though SGD is known as an unbiased estimation, a stochastic hierarchical structure could affect the overall approximation performance upon the class distribution within the mini-batch. Settings in more detail are provided in Appendix.

Baseline setting  In terms of the regularization, we used three reference setting settings, denoted ‘baseline’, ‘baseline+$l^2$', and ‘E’ for a reference. First, in ‘baseline’, we examined neural networks without regularization. Then, in ‘baseline+$l^2$’, we added $l^2$-norm minimization based regularization (which is equivalent to a weight decay in SGD setting (Zhang et al. [2019])) via $\lambda_f \sum_k ||w_k||$, where $w_k \in W$ and $\lambda_f > 0$ to ‘baseline’. Lastly, in ‘E’, energy minimization (Liu et al., [2018]), diversity promoting regularization without hierarchical information, via $\lambda_e \sum_{i\neq j} d(w_i, w_j)$, where $d(\cdot, \cdot)$ is a pairwise distance and $\lambda_e > 0$, is added to baseline+$l^2$. When the proposed hierarchical regularization (Eq. [5]) is applied to the FC layer, we denoted it as ‘H’. Hence, the regularization is incrementally adopted to the baseline: $l^2$+$E$+$H$. In terms of a distance metric, we compared the proposed metric with other continuous metrics such as Euclidean distance with the unit-length projection (‘N-euclidean2’, $\sum_{i\neq j} \frac{w_i}{||w_i||} \cdot \frac{w_j}{||w_j||} - \frac{w_i}{||w_i||} \cdot \frac{w_j}{||w_j||}$ angular distance (‘angular2’, $\sum_{i\neq j} \arccos \left( \frac{w_i \cdot w_j}{||w_i|| ||w_j||} \right)$) where ‘2’ is from Riesz s-energy shown higher accuracy, and cosine similarity (‘cosine’, $\sum_{i\neq j} \frac{w_i \cdot w_j}{||w_i|| ||w_j||}$).

4.2 Results

Low resolution object classification  In Table[1], we showed test accuracy (%) using different metrics with energy minimization regularization without the hierarchical regularization$^7$ along a weight decay ($l^2$-norm) using CIFAR-10 and resnet-20. The ‘baseline’ method is examined without ($\times$ in Table[1]) or with ($\circ$ in Table[1]) the weight decay ($l^2$). The discrete angular metric based regularization ($D_h^{\text{Ter}}$ (ternary code), $D_h^{\text{Bin}}$ (binary code), $D_{\text{cos(AM)}}$, and $D_{\text{cos(HM)}}$) on the baseline can improve the generalization performance in term of test accuracy compared to the other continuous metrics such as N-euclidean2, angular2, and cosine. Due to the unit-length projection in N-euclidean2, their performance is comparable to that other angular metrics. $D'$ denotes an arccosine adopted discrete distance used as $D_h'$ (see footnote 4 in Section[3.2]). Regularization methods are applied over all layers except Batch Normalization (BN) layers. As explained in [van Laarhoven, 2017], [Zhang et al., 2019], $l^2$-norm minimization based regularization (weight decay) showed much improvement due to their effective learning rate in SGD.

As shown in Table[2], the regularization with proposed metrics shows better accuracy performance than that of the baseline on both resnet-20 and resnet-110 using CIFAR-100 dataset. Comparing to the baseline+$l^2$, with energy minimization ‘E’ regularization, metrics such as $D_h^{\text{Ter}}$, $D_{\text{cos(AM)}}$, and $D_{\text{cos(HM)}}$ showed better performance than other metrics. If the proposed hierarchical ‘H’ regular-

\footnotesize

\begin{itemize}
  \item[\textbullet] which is available at the pytorch library
  \item[\textbullet] $\sum_{i\neq j} (2 - 2 \frac{w_i \cdot w_j}{||w_i|| ||w_j||})^{-1}$
  \item[\textbullet] This is because that hierarchy information is not included in CIFAR-10.
\end{itemize}
binary discretization for proposed angular metrics (resnet-20 and resnet-110. As the binary metric shows a better performance than that ternary, we improvement than that other combinations.  

energy minimization regularization. In this experiment, a combination GM and HM shows a better performance than that the baseline applying only 'H') affect the generalization performance using resnet-20 and CIFAR-100 dataset. The cases applying hierarchical regularization showed better performance than that the baseline applying only energy minimization regularization. In this experiment, a combination GM and HM shows a better improvement than that other combinations.

Table 1: Test accuracy (%) using different metrics on energy minimization without hierarchy information along $l^2$-norm minimization (weight decay) on CIFAR-10 and resnet-20.

<table>
<thead>
<tr>
<th>Metric \ db</th>
<th>Resnet-18</th>
<th>Resnet-50</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>72.17</td>
<td>74.21</td>
</tr>
<tr>
<td>baseline+$l^2$</td>
<td>72.29</td>
<td>74.05</td>
</tr>
<tr>
<td>N-euclidean2</td>
<td>72.61 / 75.99</td>
<td>73.49 / 76.14</td>
</tr>
<tr>
<td>angular2</td>
<td>72.43 / 76.11</td>
<td>73.55 / 76.66</td>
</tr>
<tr>
<td>cosine</td>
<td>72.12 / 75.64</td>
<td>73.26 / 76.85</td>
</tr>
<tr>
<td>$D_{E}^{c}$</td>
<td>90.67 / 92.48</td>
<td></td>
</tr>
<tr>
<td>$D_{H}^{c}$</td>
<td>90.67 / 92.48</td>
<td></td>
</tr>
<tr>
<td>$D_{HM}^{c}$</td>
<td>90.84 / 92.93</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Test accuracy (%) using different metrics on energy minimization ('E') and hierarchical regularization ('H') on CIFAR-100: 'E' / 'E+H' respectively.

<table>
<thead>
<tr>
<th>Metric \ db</th>
<th>resnet-20</th>
<th>resnet-110</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>63.86</td>
<td>62.02</td>
</tr>
<tr>
<td>baseline+$l^2$</td>
<td>68.03</td>
<td>72.90</td>
</tr>
<tr>
<td>N-euclidean2</td>
<td>67.59 / 68.65</td>
<td>73.95 / 73.96</td>
</tr>
<tr>
<td>angular2</td>
<td>67.83 / 67.76</td>
<td>74.40 / 73.89</td>
</tr>
<tr>
<td>cosine</td>
<td>68.11 / 68.45</td>
<td>73.37 / 73.37</td>
</tr>
<tr>
<td>$D_{E}^{c}$</td>
<td>68.44 / 68.68</td>
<td></td>
</tr>
<tr>
<td>$D_{H}^{c}$</td>
<td>68.52 / 68.69</td>
<td></td>
</tr>
<tr>
<td>$D_{AM}$</td>
<td>68.58 / 68.86</td>
<td></td>
</tr>
<tr>
<td>$D_{EAM}$</td>
<td>68.58 / 68.60</td>
<td></td>
</tr>
<tr>
<td>$D_{HM}^{c}$</td>
<td>68.62 / 68.65</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Test accuracy (%) using different metrics on energy minimization ('E') and hierarchical regularization ('H') on CUB200: 'E' / 'E+H' respectively.

<table>
<thead>
<tr>
<th>Metric \ db</th>
<th>Resnet-18</th>
<th>Resnet-50</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>72.17</td>
<td>74.21</td>
</tr>
<tr>
<td>baseline+$l^2$</td>
<td>72.29</td>
<td>74.05</td>
</tr>
<tr>
<td>N-euclidean2</td>
<td>72.61 / 75.99</td>
<td>73.49 / 76.14</td>
</tr>
<tr>
<td>angular2</td>
<td>72.43 / 76.11</td>
<td>73.55 / 76.66</td>
</tr>
<tr>
<td>cosine</td>
<td>72.12 / 75.64</td>
<td>73.26 / 76.85</td>
</tr>
<tr>
<td>$D_{E}^{c}$</td>
<td>90.67 / 92.48</td>
<td></td>
</tr>
<tr>
<td>$D_{H}^{c}$</td>
<td>90.67 / 92.48</td>
<td></td>
</tr>
<tr>
<td>$D_{AM}$</td>
<td>73.04 / 76.21</td>
<td></td>
</tr>
<tr>
<td>$D_{AM}^{c}$</td>
<td>73.04 / 76.21</td>
<td></td>
</tr>
<tr>
<td>$D_{HM}^{c}$</td>
<td>90.94 / 92.69</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Test accuracy (%) using different metrics with energy minimization ('E') and hierarchical regularization ('H') on Cars: 'E' / 'E+H' respectively.

<table>
<thead>
<tr>
<th>Metric \ db</th>
<th>Resnet-18</th>
<th>Resnet-50</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>85.10</td>
<td>87.99</td>
</tr>
<tr>
<td>baseline+$l^2$</td>
<td>85.58</td>
<td>87.92</td>
</tr>
<tr>
<td>N-euclidean2</td>
<td>85.48 / 85.56</td>
<td>87.36 / 87.97</td>
</tr>
<tr>
<td>angular2</td>
<td>85.11 / 85.13</td>
<td>88.34 / 87.85</td>
</tr>
<tr>
<td>cosine</td>
<td>85.57 / 85.73</td>
<td>88.01 / 87.86</td>
</tr>
<tr>
<td>$D_{E}^{c}$</td>
<td>85.35 / 85.99</td>
<td></td>
</tr>
<tr>
<td>$D_{H}^{c}$</td>
<td>86.22 / 85.99</td>
<td></td>
</tr>
<tr>
<td>$D_{AM}$</td>
<td>85.66 / 85.66</td>
<td></td>
</tr>
<tr>
<td>$D_{AM}^{c}$</td>
<td>85.82 / 85.65</td>
<td></td>
</tr>
<tr>
<td>$D_{HM}^{c}$</td>
<td>85.35 / 85.99</td>
<td></td>
</tr>
</tbody>
</table>

Fine-grained visual categorization In this experiment, we used two large size image datasets with a fine-grained category. One is with birds (CUB200) and another is with cars (Cars) that focus on single species of objects. A hierarchy of CUB200 is defined by the academic expert on birds whereas that of Cars is categorized manually based on a name of cars by a non-expert. Moreover, the rate between the number of superclass (parent) per subclass of CUB200 (0.35) is much larger than that of Cars (0.0459) (as shown in Table 7 at Appendix). That rate of Cars is even smaller than that of CIFAR-100.

As shown in Table 3, the proposed hierarchical regularization significantly improved the test accuracy over all metrics on both Resnet-18 and Resnet-50. Compared to CUB200, as shown in Table 4, the improvement of the proposed method is not that significant for Cars dataset. This is because CUB200 dataset has not only better hierarchical categorization by the expert but also more diversified pairs between the superclass and the subclass.

Ablation study In Table 5, we showed how metrics affect the generalization performance using resnet-20 and CIFAR-100 dataset. The proposed method with different proposed metrics showed significantly improved performance compared to the baseline+$l^2$. Individual means (AM, GM, and HM) show different improvement patterns.

In Table 6, we showed how heterogeneous metrics on convolutional (conv.) layers (with the energy minimization, ‘E’) and fully connected (FC) layers (with the hierarchical ‘H’ regularization, ‘H’) affect the generalization performance using resnet-20 and CIFAR-100 dataset. The cases applying hierarchical regularization showed better performance than that the baseline applying only energy minimization regularization. In this experiment, a combination GM and HM shows a better improvement than that other combinations.
Table 5: Test accuracy (%) using different metrics on energy minimization ('E') and hierarchical regularization ('H') on CIFAR-100 and resnet-20.

<table>
<thead>
<tr>
<th>metric</th>
<th>baseline+\text{l}_2</th>
<th>\text{D}_\text{AM}</th>
<th>\text{D}_\text{GM}</th>
<th>\text{D}_\text{HM}</th>
<th>\text{D}_\text{AM,cos}</th>
<th>\text{D}_\text{GM,cos}</th>
<th>\text{D}_\text{HM,cos}</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline+\text{l}_2</td>
<td>68.03</td>
<td>68.64</td>
<td>68.70</td>
<td>68.80</td>
<td>69.24</td>
<td>68.55</td>
<td>68.77</td>
</tr>
<tr>
<td>\text{D}_\text{AM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{GM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{HM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{AM,cos}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{GM,cos}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{HM,cos}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Test accuracy (%) using heterogeneous metrics at Conv. layers and FC layer respectively, on energy minimization ('E') and hierarchical regularization ('H') on CIFAR-100 and resnet-20.

<table>
<thead>
<tr>
<th>metrics (in conv., in FC)</th>
<th>baseline+\text{l}_2</th>
<th>\text{D}_\text{GM}</th>
<th>\text{D}_\text{AM}</th>
<th>\text{D}_\text{GM}</th>
<th>\text{D}_\text{HM}</th>
<th>\text{D}_\text{AM}</th>
<th>\text{D}_\text{GM}</th>
<th>\text{D}_\text{HM}</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline+\text{l}_2</td>
<td>68.03</td>
<td>68.22</td>
<td>68.58</td>
<td>68.62</td>
<td>69.04</td>
<td>68.62</td>
<td>68.65</td>
<td>68.70</td>
</tr>
<tr>
<td>\text{D}_\text{GM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{AM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{GM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{HM}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{GM,cos}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{GM,cos}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{D}_\text{GM,cos}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Related Works

Promoting diversity in an embedding space or model parameters is an widely adopted strategy in machine learning related area to improve the generalization performance (Cogswell et al., 2016; Yang et al., 2019; Li et al., 2012; Ratzlaff & Fuxin, 2019; Xie et al., 2017b; 2018; Liu et al., 2018). This diversity based strategy can be applied in a variety of levels such as in a feature level (Cogswell et al., 2016; Xie et al., 2018), in a projection parameter level (Xie et al., 2017a; Liu et al., 2018), in a model ensemble level (Zhou et al., 2018; Ratzlaff & Fuxin, 2019), in a latent space model level (Ratzlaff & Fuxin, 2019; Liu et al., 2018), and in a generative model level (Yang et al., 2019; Ratzlaff & Fuxin, 2019). Among these approaches, some regularization methods require additional optimization effort. Enlarging pairwise distance between features (Xie et al., 2018) requires computational efforts due to their covariance matrix. In (Xie et al., 2017b), unit-eigenvalue is utilized via singular value decomposition. To optimize a direction and magnitude of the parameter vector alternatively, an alternating direction method of multipliers (ADMM) (Xie et al., 2017a) is used.

In terms of learning on a hypersphere, (Liu et al., 2017) proposed that hyperspherical convolution (SphereCov) replaces the traditional inner-product based convolution in order to conduct learning angular representation. On a hypersphere, regularization via promoting diversity of parameters has been proposed (Liu et al., 2018) based on Minimum Hyperspherical Energy.

In terms of hierarchical learning, (Nickel & Kiela, 2017; Ganea et al., 2018) proposed to apply hyperbolic space to embed the data in deep neural networks. They showed that learning representations in the hyperbolic space is effective to the data preserving a latent hierarchy compared to in the Euclidean space. Different from these works which are aimed for representation learning, our proposed method is focused on regularization learning using explicit (predefined) hierarchy information.

6 Conclusion

We proposed the regularization method, which maximizes a pairwise distance between parameters preserving a hierarchical structure. To define a hierarchical parameter space, we reformulated the topology space with multiple hyperspheres. In each hypersphere, a projection is parameterized by the surface parameter with the center parameter. By imposing a maximum distance between hierarchical parameters, diversity of parameters preserving semantic structure was promoted. For the distance metric in multiple separated spaces, we proposed a discrete metric integrated with a continuous metric. Extensive experiments using publicly available datasets (CIFAR-10, CIFAR-100, CUB200-2011, and Stanford Cars), the deep neural network with our proposed regularization method showed superior classification performance, especially when the number of super-classes is large. For further exploration in future, our proposed method can be combined with hierarchical representation learning such as hyperbolic (or Poincaré) embeddings (Nickel & Kiela, 2017; Ganea et al., 2018).

References


**APPENDIX**

**Dataset acquisition details** CIFAR-100 dataset provides a pair of superclass-subclass labels. In CUB200 dataset, the pairs can be extracted from their filename. In Cars dataset, we parsed each subclass label to one of nine superclass vehicle types, such as “Sedan”, “SUV”, “Van” and etc., following (Krause et al., 2013a).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#classes</th>
<th>#train</th>
<th>#test</th>
<th>input size</th>
<th>#samples/class</th>
<th>#super/subclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>{1, 10}</td>
<td>50,000</td>
<td>10,000</td>
<td>32 × 32</td>
<td>5000.00</td>
<td>0.1000</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>{20, 100}</td>
<td>50,000</td>
<td>10,000</td>
<td>32 × 32</td>
<td>500.00</td>
<td>0.2000</td>
</tr>
<tr>
<td>CUB200</td>
<td>{70, 200}</td>
<td>5,994</td>
<td>5,794</td>
<td>224 × 224</td>
<td>29.97</td>
<td>0.3500</td>
</tr>
<tr>
<td>Cars</td>
<td>{9, 196}</td>
<td>8,144</td>
<td>8,941</td>
<td>224 × 224</td>
<td>41.55</td>
<td>0.0459</td>
</tr>
</tbody>
</table>

**Deep neural network models and training details** First, resnet-20 (0.29M) and resnet-110 (1.73 M), which include a combination of Basic blocks with output channels [16, 30, 64] are adopted for light models. An input dimensionality of the fully connected (FC) layer (a classifier) is 64 for both resnet-20 and resnet-110. Second, heavy models are adopted such as Resnet-18 (11.28M) and

---

10Dependent on the number of classes and the corresponding center parameters, the size could variate (e.g. 11.42M for CUB200, 11.31M for Cars).
Resnet-50 (23.91M) which consists of the basic blocks (Resnet-18) or the bottleneck blocks with output channels [64, 128, 256, 512] in Conv. layers. An input dimensionality of the FC layer is 512 for Resnet-18 and 2048 for Resnet-50.

Networks are optimized with SGD for both light and heavy models: we fixed 1) the weight initialization with Random-Seed number ‘0’ in pytorch, 2) learning rate schedule [0.1, 0.01, 0.001], 3) with momentum 0.9, 4) regularization: $l^2$-norm minimization with $\lambda_f = 0.0005$, 5) energy minimization $\lambda_c = \{0.1, 1\}$, and hierarchical minimization $\lambda_l = 0.1 \times \{1, 5\}$. All the regularization is not applied to the parameters of Batch Normalization (BN) layers, since BN is also a regularization function. A bias term in the FC layer is not used. The images in training and test, images are resized to 256 size. The images is cropped with 224 × 224 size at random location in training and at center location in test. Horizontal flipping is applied in training. The light models are trained from scratch without the pretrained weights for 300 epochs. The heavy model is trained using pretrained model provided by pytorch library\textsuperscript{11} with 100 epochs. The experiments are conducted using GPU “NVIDIA TESLA P40”.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{\textbf{R}_{l,p}(W_r^{(l-1)}, W_c^{(l-1)}) = d(w_{g_i,n}, w_{g_i,n}^{(l-1)}) + d(w_{g_i,m}, w_{g_i,k}^{(l-1)}) + d(w_{g_i,k}, w_{g_i,n}^{(l-1)}) + \ldots \text{ corresponds to Eq } \text{(4)} \text{ and } \textbf{R}_{l,c}(W_c^{(l-1)}) = d(w_{c,g_i}, w_{c,g_j}^{(l-1)}) + d(w_{c,g_j}, w_{c,g_k}^{(l-1)}) + \ldots \text{ corresponds to Eq } \text{(7)}.}
\end{figure}

**Constraints in Eq. 4:** $C_l := \sum_k \lambda_k C_{l,k}$ can be given as follows:

1. **Constraint 1 ($C_1$):** This constraint describes that a radius of an inner sphere must be smaller than that of its outer sphere.
   \[ r^{(l-1)} - r^{(l)} \geq 0 \Rightarrow \|w^{(l-1)} - w^{(l)}\| = \|w_s^{(l-1)} - w_c^{(l-1)}\| - \|w_s^{(l)} - w_c^{(l)}\| \geq 0. \]

2. **Constraint 2 ($C_2$):** This constraint describes that a center of an inner sphere must be located within its outer sphere.
   \[ r^{(l-1)} - (\|w_c^{(l-1)}\| + r^{(l)}) \geq 0 \Rightarrow r^{(l-1)} - (\|w_c^{(l-1)} - w_c^{(l)}\| + r^{(l)}) = \|w_s^{(l-1)} - w_c^{(l-1)}\| - (\|w_c^{(l-1)} - w_c^{(l)}\| + \|w_s^{(l)} - w_c^{(l)}\|) \geq 0. \]

3. **Constraint 3 ($C_3$):** This constraint describes that a margin between spheres must be larger than zero.
   \[ \|w_c^{(l-1)}\|[2 - 2 \cos \theta]^{0.5} - 2r^{(l)} \geq 0 \Rightarrow \|w_c^{(l)}\|[2 - 2 \sum_{i,j} \frac{w_c^{(l-1), i} w_s^{(l), j}}{\|w_c^{(l)}\|^2}]^{0.5} - \]

\textsuperscript{11}from https://download.pytorch.org/models/resnet18-5c106cde.pth, https://download.pytorch.org/models/resnet50-19c8e357.pth respectively
\[ 2\| w_s^{(l)} - w_c^{(l)} \|, \quad \text{where} \quad \| w_c^{(l,l-1)} \|(2 - 2 \cos \theta)^{0.5} = \| w_c^{(l,l-1)} \|(r^{(l-1)} \sin \theta^2 - (r^{(l-1)} - r^{(l-1)} \cos \theta)^2)^{0.5}. \]