Maximizing Margin Lower Bound in Deep Nonlinear Networks

Abstract

If the data is linearly separable, support vector machines are usually employed to estimate parameters which produce maximum margin in the input space. In cases where the data is not linearly separable, kernel tricks can be used to transform the data and SVMs are then applied over the transformed data to estimate parameters which produce maximum margin in the transformed space. In this work, we theoretically explain a method involving induced 2-norm regularization to learn parameters of a deep hierarchical model such that the lower bound on the margin in the input space of the model is maximized. We provide a minimax and probabilistic interpretation of induced 2-norm regularization along with experimental results over synthetic datasets proving the practical consistency of our theory.

1. INTRODUCTION

The ability of a learning algorithm to predict correctly on previously unseen data can be considered as one of the most significant criterion in designing supervised learning techniques due to the limited availability of labelled data. Models that utilize margin while learning from training data has been of special interest to machine learning and statistics community as the generalization error bounds can be computed in many cases based on the algorithm parameters and margin terms. The most important factor to be considered in design of such models is to achieve an optimal decision boundary having maximum margin in their input space. With models linear in parameters, it is possible to obtain the optimal decision boundary in a linearly separable data by using support vector machines (Cortes & Vapnik, 1995). However, a large number of datasets are not linearly separable and to deal with this issue kernel methods were employed to transform the data so that it becomes linearly separable. An SVM applied over the transformed data obtains maximum margin decision boundary in the transformed space but may generalize badly on the unseen data due to a sub-optimal decision boundary in the input space. For more than a decade, deep neural networks are being employed for function approximation and have shown high generalization in various classification and regression tasks (He et al., 2016; Szegedy et al., 2015; Krizhevsky et al., 2012). However, the decision boundaries learnt by these networks may not have maximum margin in the input space. In this work, we present,

- A formal and general theory dealing with a method to increase margin in the input space of nonlinear hierarchical models - We derive a lower bound on the margin and formulate an optimization objective for its maximization which involves induced 2-norm regularization.
- A minimax and probabilistic interpretation of induced 2-norm regularization.
- Experiments over the synthetic datasets which prove that the induced 2-norm regularization achieves better guarantee of margin in comparison to F-norm or no regularization.

2. RELATED WORK

The problem of increasing margin in the input space of deep hierarchical networks have not been directly dealt in many papers before. The work which is most closely related to our work is presented in (An et al., 2015), where the authors constrain the fully connected layers to be contraction mappings and have the final layer in the hierarchy to be an SVM. They prove that the margin obtained by the model in the input space will be atleast equal to the margin obtained by the model in the feature space acting as the input of the SVM layer. However, their method has two major limitations - the procedure of clipping the singular values of weight matrices which are greater than 1 in order to make the fully connected layer a contraction mapping is sub-optimal and the recomputation of the weight matrices using the clipped singular values may represent an unwanted shift in the parameter space away from the optimum value of parameters. (Szegedy et al., 2013) devised a procedure for the generation of adversarial examples by adding parameters and loss dependent noise to the training examples (noise that brings an example closer to the learnt decision boundary and possibly placing it on the other side of

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the boundary) and showed that the classes predicted by the underlying network for the generated adversarial examples are incorrect. They claimed that networks having weight matrices with better spectral stability may bring robustness to additive noise and thus may lead to correct classification of adversarial examples. They also claimed that the fully connected and convolutional layers with lower value of lipschitz constants tend to have underlying weight matrices with better spectral stability. (Bruna & Mallat, 2012) showed that a linear mapping with a finite and lower value of the lipschitz constant has better invariance to small deformation in its input than a mapping with higher value of lipschitz constant. In our work, we show that the minimization of the lipschitz constants of weight matrices is, in fact, directly proportional to the maximization of the margin in the input space.

In section 3, we present relevant definitions and obtain a lower bound on the margin in the input space of a nonlinear deep hierarchical network. It will be evident from the expression of the lower bound that the minimization of the lipschitz constants of the transformation and activation layers in the network is directly proportional to maximization of the lower bound on the margin in the input space of the network. We will also derive the lipschitz constants of fully connected and convolution layers as well as commonly used activation layers. It will turn out that the lipschitz constant of a fully connected layer is equal to the induced 2-norm of the underlying weight matrix if the metric defined on its input and output space is 2-norm. From now on, by 2-norm of a tensor, we refer to its induced 2-norm. In section 4, we present a probabilistic and minimax optimization based interpretation of the induced 2-norm regularization. Then, in section 5, we present details of our experiments on synthetic data and relevant table and figures showing the practical consistency of our theory. Finally, in section 6, we conclude our work and briefly discuss about future work.

3. PROPOSED METHOD

We start by revisiting the definition of lipschitz continuous mappings which is as follows.

Definition 1. *Given two metric spaces (X, d_X) and (Y, d_Y) where d_X denotes the metric on the set X and d_Y denotes the metric on the set Y, a function $\psi : X \to Y$ is said to be lipschitz continuous if there exists a real constant $K \geq 0$ such that, for all x_1 and x_2 in X,*

$$d_Y(\psi(x_1), \psi(x_2)) \leq K d_X(x_1, x_2) \tag{1}$$

Any such K is referred to as a Lipschitz constant for the function ψ. The smallest constant is sometimes called the (best) Lipschitz constant and we denote it by K^*.

Unless specified, by Lipschitz constant we always mean the best Lipschitz constant. Now, we give a general definition of a hierarchical network that we will be using throughout this work.

Definition 2. *We define a hierarchical network of depth P as a function $h : X \times \Theta \to Y$ where X is the set of inputs, Y is the set of possible outputs and Θ is the set of parameters. The function h can also be written as a composition of functions i.e. $h \equiv g_P \circ f_P \circ g_{P-1} \circ f_{P-1} \circ \ldots \circ g_1 \circ f_1$ where $f_i \equiv f_i(a_i, \theta_i)$ is a function which takes inputs $a_i \in A_i$, the set of inputs and $\theta_i \in \Theta_i$, the set of parameters for function f_i and produces output $z_{i+1} \in Z_{i+1}$. The function $g_i \equiv g_i(z_{i+1})$ takes input in $z_{i+1} \in Z_{i+1}$ and produces output $a_{i+1} \in A_{i+1}$. Here $A_1 = X$, $A_{P+1} = Y$ and $\Theta = \bigcup_{i=1}^{P} \Theta_i$.*

Although, the functions f_i and g_i, $i \in \{1, 2, 3, \ldots, P-1, P\}$ can be any mapping from $A_i \times \Theta_i$ to Z_{i+1} and from Z_{i+1} to A_{i+1} respectively, however, to draw parallels with deep neural networks (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014), we restrict ourselves to the fully connected layer and convolution (linear) transformation layers as mappings representing f_i and linear, relu, sigmoid and tanh (possibly nonlinear) activation layers as mappings representing g_i. In the definition 2, a_1 acts as the input layer of the network and a_{P+1} acts as the output layer of the network, z_i, $i \in \{2, 3, \ldots, P, P+1\}$ act as the output of the transformation layers of the network and a_i, $i \in \{2, 3, \ldots P, P+1\}$ act as the activation of z_i.

From now on we focus on the networks specific to the task of classification in which case the set $Y = \mathbb{R}^C$ where C is the number of classes and for a $y \in Y$, y_i denotes the i_{th} element of vector y which represents the score of the corresponding input belonging to the class i. Higher the value of y_i, more likely the input belongs to the class i. We now define the set of boundary points in the input space of each transformation and activation layer in a hierarchical network.

Definition 3. *Suppose we have a hierarchical network h of depth P as described in the definition 2 and the output set of the network is $Y = \mathbb{R}^C$ where C is the number of classes, then for a $\theta \in \Theta$ we define the set of boundary points in the input space of f_k and in the input space of g_k, $k \in \{1, 2, 3, \ldots, P-1, P\}$ as*

$$B_{f_k}(\theta) = \{a_k \,|\, y_i = y_j,\, a_k \in A_k,\; y = g_P \circ f_P \ldots \circ f_{k+1} \circ g_k \circ f_k(a_k, \theta)\} \tag{2}$$

$$B_{g_k}(\theta) = \{z_{k+1} \,|\, y_i = y_j,\, z_{k+1} \in Z_{k+1},\; y = g_P \circ f_P \ldots \circ f_{k+1} \circ g_k(z_{k+1})\} \tag{3}$$

where

$$i = \underset{k\in\{0,1,\ldots,C-1\}}{\operatorname{argmax}} y_k$$
$$j = \underset{k\in\{0,1,\ldots,C-1\}-\{i\}}{\operatorname{argmax}} y_k$$

Note that $B_{f_1} = B_h$ represents the set of boundary points in the input space of the network h. Also note that for every $a_k \in B_{f_k}(\theta)$ and for $k^{'} > k$, there exists $a_{k'} = g_{k'-1} \circ f_{k'-1} \ldots \circ g_k \circ f_k(a_k, \theta)$ which lies in $B_{f_{k'}}(\theta)$ and for $k^{'} \geq k$ there exists $z_{k'+1} = f_{k'} \circ g_{k'-1} \ldots \circ g_k \circ f_k(a_k, \theta)$ which lies in $B_{g_{k'}}(\theta)$. Similarly, for every $z_{k+1} \in B_{g_k}(\theta)$ and for $k^{'} \geq k$, there exists $a_{k'+1} = g_{k'} \circ f_{k'} \ldots \circ f_{k+1} \circ g_k(z_{k+1}, \theta)$ which lies in $B_{f_{k'+1}}(\theta)$ and for $k^{'} > k$ there exists $z_{k'+1} = f_{k'} \circ g_{k'-1} \ldots \circ f_{k+1} \circ g_k(z_{k+1}, \theta)$ which lies in $B_{g_{k'}}(\theta)$.

Now, we define the margin in the input space of each transformation and activation layer of a hierarchical network being used to model some dataset.

Definition 4. *Given a dataset $\{x_i, t_i\}_{i=1}^N$ where x_i is the input and t_i is the corresponding true output which is a one-hot vector representing the class of x_i and suppose we use a hierarchical network h of depth P as described in the definition 2 to approximate the mapping from x_i to t_i with parameters $\theta \in \Theta$, then with the set of boundary points $B_{f_k}(\theta)$ and $B_{g_k}(\theta)$ as in the definition 3, the margin produced by θ in the input space of f_k and in the input space of g_k, $k \in \{1, 2, 3, \ldots, P-1, P\}$ are given by,*

$$\delta_{f_k}(\theta) = \min_{i\in[N]} \inf_{x\in B_{f_k}(\theta)} d_{A_k}(g_{k-1} \ldots \circ g_1 \circ f_1(x_i), x) \quad (4)$$

$$\delta_{g_k}(\theta) = \min_{i\in[N]} \inf_{x\in B_{g_k}(\theta)} d_{Z_{k+1}}(f_k \circ g_{k-1} \ldots \circ g_1 \circ f_1(x_i), x) \quad (5)$$

Here $g_0(x) = x$ and d_{A_k} and $d_{Z_{k+1}}$ are metrics defined on the sets A_k and Z_{k+1} respectively. Also, note that $\delta_h(\theta) = \delta_{f_1}(\theta)$ which is the margin in the input space of the network.

The following theorem derives an expression of the lower bound on the margin in the input space of a transformation layer of a deep hierarchical network given the margin produced by the network in the some subsequent transformation or activation layer ahead in the hierarchy.

Theorem 1. *Given a dataset $\{x_i, t_i\}_{i=1}^N$ where x_i is the input and t_i is the corresponding true output which is a one-hot vector representing the class of x_i and suppose we use a hierarchical network of depth P as described in the definition 2 to approximate the mapping from x_i to t_i with parameters $\theta \in \Theta$ and the margin in the input space of g_k be $\delta_{g_k}(\theta)$ and in the input space of f_k be $\delta_{f_k}(\theta)$ as in the definition 4. Assuming that the best lipschitz constant of f_k is L_{f_k} and that of g_k is L_{g_k} then the following conditions hold for every $k^{'} \geq k$,*

$$\delta_{f_k}(\theta) \geq \frac{\delta_{g_{k'}}(\theta)}{\prod_{i=k}^{i=k'} L_{f_i} . \prod_{j=k}^{j=k'-1} L_{g_j}} \quad (6)$$

$$\delta_{f_k}(\theta) \geq \frac{\delta_{f_{k'}}(\theta)}{\prod_{i=k}^{i=k'-1} L_{f_i} . \prod_{j=k}^{j=k'-1} L_{g_j}} \quad (7)$$

Proof. We prove the condition in Eq. 6 and the proof of the condition in Eq. 7 follows similarly.
From the definition 1, if a function $\psi : X \to Y$ has the best Lipschitz constant K^* then for all $a, b \in X$,

$$d_Y(\psi(a), \psi(b)) \leq K^* d_X(a, b)$$
$$\Rightarrow d_X(a, b) \geq \frac{d_Y(\psi(a), \psi(b))}{K^*} \quad (8)$$

From the definition 4, for some $i^* \in [N]$ and $x^* \in B_{f_k}(\theta)$, we have,

$$\delta_{f_k}(\theta) = d_{A_k}(g_{k-1} \ldots \circ f_2 \circ g_1 \circ f_1(x_{i^*}), x^*) \quad (9)$$

Using Eq. 8, we get,

$$\begin{aligned}
& d_{A_k}(g_{k-1} \ldots \circ f_2 \circ g_1 \circ f_1(x_{i^*}), x^*) \\
& \geq \frac{d_{Z_{k+1}}(f_k \circ g_{k-1} \ldots \circ f_2 \circ g_1 \circ f_1(x_{i^*}), f_k(x^*))}{L_{f_k}} \\
& \geq \frac{d_{A_{k+1}}(g_k \circ f_k \circ g_{k-1} \ldots \circ f_2 \circ g_1 \circ f_1(x_{i^*}), g_k \circ f_k(x^*))}{L_{f_k} L_{g_k}} \\
& \vdots \\
& \geq \frac{\begin{array}{c} d_{Z_{k'+1}}(f_{k'} \circ g_{k'-1} \ldots \circ g_k \circ f_k \circ g_{k-1} \ldots \\ \ldots \circ g_1 \circ f_1(x_{i^*}), f_{k'} \circ g_{k'-1} \ldots \circ g_k \circ f_k(x^*)) \end{array}}{\prod_{i=k}^{i=k'} L_{f_i} . \prod_{j=k}^{j=k'-1} L_{g_j}} \\
& \geq \min_{i'\in[N]} \inf_{x'\in B_{g_{k'}}(\theta)} \frac{\begin{array}{c} d_{Z_{k'+1}}(f_{k'} \circ g_{k'-1} \ldots \circ g_k \circ \\ f_k \circ g_{k-1} \ldots \circ g_1 \circ f_1(x_{i'}), x') \end{array}}{\prod_{i=k}^{i=k'} L_{f_i} . \prod_{j=k}^{j=k'-1} L_{g_j}} \\
& = \frac{\delta_{g_{k'}}(\theta)}{\prod_{i=k}^{i=k'} L_{f_i} . \prod_{j=k}^{j=k'-1} L_{g_j}}
\end{aligned} \quad (10)$$

The last equation follows from the Eq. 5 and from the definition 3 from which we know that a boundary point in the

input space of a layer maps to a boundary in the input space of a subsequent layer, i.e.,

$$x^* \in B_{f_k}(\theta) \Rightarrow f_{k'} \circ g_{k'-1} \ldots \circ g_k \circ f_k(x^*) \in B_{g_{k'}}(\theta) \tag{11}$$

□

The following corollary gives an expression of the lower bound on the margin in the input space of a hierarchical network given the margin produced by the network in the input space of the last transformation layer.

Corollary 1. *From the definition 2 and the Th. 1, for a hierarchical network h with depth P,*

$$\delta_h(\theta) \geq \frac{\delta_{f_P}(\theta)}{\prod_{i=1}^{i=P-1} L_{f_i} . \prod_{j=1}^{j=P-1} L_{g_j}} \tag{12}$$

Parameters θ which produce maximum margin in the input space of a hierarchical network which in mathematical terms mean that they maximize $\delta_h(\theta)$, and at the same time produce maximum training and validation accuracy are the most robust parameters to the noise in the test data. Although, deriving an exact formula for the margin in the input space of the nonlinear network is a non trivial task, here, we have derived a lower bound on the margin (Cor. 1) in terms of the margin in the input space of last transformation layer (f_P) and the lipschitz constants of the transformation and activation layers constituting the network. Now, we derive the values of the lipschitz constants of commonly used transformation and activation layers.

Theorem 2. *Consider the linear transformation performed by a fully connected layer, $f(x) = Ax + b$ where $x \in X = \mathbb{R}^n$ is an n dimensional input vector, A is an $m \times n$ weight matrix and b is an m dimensional bias vector. The output of this transformation $f(x) \in Y = \mathbb{R}^m$. Assuming that the metric defined on X and Y is $\|\cdot\|$ then the best lipschitz constant K^* of this transformation is $\|A\|$.*

Proof. From the definition 1, for all $x_1, x_2 \in X$,

$$\begin{aligned} K^* &= \sup_{x_1 \neq x_2} \frac{\|f(x_1) - f(x_2)\|}{\|x_1 - x_2\|} \\ &= \sup_{x_1 \neq x_2} \frac{\|Ax_1 + b - Ax_2 - b\|}{\|x_1 - x_2\|} \\ &= \sup_{x_1 \neq x_2} \frac{\|A(x_1 - x_2)\|}{\|x_1 - x_2\|} \\ &= \sup_{r \neq 0} \frac{\|Ar\|}{\|r\|} \\ &= \|A\| \end{aligned} \tag{13}$$

□

As described in (Szegedy et al., 2013), if W denotes the 4-tensor implementing a convolution layer with C input features, D output features, support $N \times N$ and spatial stride Δ,

$$Wx = \left\{ \sum_{c=1}^{C} x_c \star w_{c,d}(n_1\Delta, n_2\Delta); d = 1, \ldots, D \right\} \tag{14}$$

where x_c denotes the c-th input feature image, and $w_{c,d}$ is the spatial kernel corresponding to feature c and output feature d. The induced norm of W using Parseval's formula, is given by,

$$\|W\| = \sup_{\xi \in [0, N\Delta^{-1})^2} \|A(\xi)\| \tag{15}$$

where $A(\xi)$ is a $D \times (C.\Delta^2)$ matrix whose rows are $\forall d \in \{1, \ldots D\}$, $A(\xi)_d = \left(\Delta^{-2}\widehat{w_{c,d}}(\xi + l.N.\Delta^{-1}); c \in \{1, \ldots, C\}, l \in \{0, \ldots, \Delta - 1\}^2\right)$ and $\widehat{w_{c,d}}$ is the 2-D Fourier transform of $w_{c,d}$:

$$\widehat{w_{c,d}}(\xi) = \sum_{u \in [0,N)^2} w_{c,d}(u) \exp\left(\frac{-2\pi i(u.\xi)}{N^2}\right) \tag{16}$$

Now, we derive the lipschitz constants for the linear, relu, tanh, and sigmoid activations.

Theorem 3. *Consider the activation functions $g_1(x) = x$, $g_2(x) = \max(x, 0)$, $g_3(x) = \frac{1}{1+\exp(-x)}$ and $g_4(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$, all of which take scalar input in $\mathbb{R}$ and produce scalar output in $\mathbb{R}$. Then, the best lipschitz constants $K_1^* = 1, K_2^* = 1, K_3^* = 0.25$ and $K_4^* = 1$ are given by,*

Proof.

$$K_1^* = \sup_{x_1 \neq x_2} \frac{|x_1 - x_2|}{|x_1 - x_2|} = 1 \tag{17}$$

$$\begin{aligned} K_2^* &= \sup_{x_1 \neq x_2} \frac{|\max(x_1, 0) - \max(x_2, 0)|}{|x_1 - x_2|} \\ &= 1, \text{ when } x_1 > 0 \wedge x_2 > 0 \end{aligned} \tag{18}$$

$$\begin{aligned} K_3^* &= \sup_{x_1 \neq x_2} \frac{|\frac{1}{1+\exp(-x_1)} - \frac{1}{1+\exp(-x_2)}|}{|x_1 - x_2|} \\ &= \sup_x |\nabla_x(\frac{1}{1+\exp(-x)})| \\ &= 0.25 \end{aligned} \tag{19}$$

$$
\begin{aligned}
K_4^* &= \sup_{x_1 \neq x_2} \frac{|\frac{\exp(x_1)-\exp(-x_1)}{\exp(x_1)+\exp(-x_1)} - \frac{\exp(x_2)-\exp(-x_2)}{\exp(x_2)+\exp(-x_2)}|}{|x_1 - x_2|} \\
&= \sup_x |\nabla_x (\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)})| \\
&= 1
\end{aligned} \tag{20}
$$

□

Now, with the aim to have high accuracy over the input dataset and at the same time maximize the margin in the input space of a hierarchical network, we formulate an optimization problem which minimizes the prediction error along with maximization of the expression in the Cor. 1. This will result in the maximization of the margin in the input space of the last transformation layer and minimization the lipschitz constants of the remaining transformation layers. Note that the lipschitz constants of the activation layers (linear, relu, sigmoid and tanh) are constant numbers (Th. 3) and hence does not affect the optimization objective. On the other hand, lipschitz constants of the transformation layers (fully connected and convolution layers) are dependent on the parameters (Th. 2, Eq. 14−16) and thus affect the optimization objective. Moreover, to minimize the prediction error and maximize the margin in the input space of the last transformation layer, we use an SVM as the last transformation layer by having a squared hinge loss (Crammer & Singer, 2001; Rosasco et al., 2004; Moore & DeNero) between the predicted output and the ground truth. To minimize the lipschitz constants corresponding to the transformation layers we minimize the induced norm of their underlying tensors (Th. 2, Eq. 14−16). Note that we assume 2-norm (euclidean metric) on the input space of each transformation layer which reduces norm of the tensors to their induced 2-norms.

Given a dataset $\{x_i, t_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^n$ in case of vector inputs or $x_i \in \mathbb{R}^{n_1 \times n_2}$ in case of image (matrix) inputs and $t_i \in \mathbb{R}^C$ is a one-hot class vector where C is the number of classes and $t_{ij} = 1$ implies that i-th input belongs to class j. Suppose we use a hierarchical network of depth P, h, to approximate the mapping between x_i and t_i. Let the tensors W_j, $j \in 1, 2, \ldots P$ represent the tensors corresponding to the transformation layers f_j where we assume that f_P is always a fully connected transformation layer and g_P is always a linear activation layer. For a convolution transformation layer, the tensor is 4 dimensional and for a fully connected transformation layer the tensor is 2 dimensional. The set of parameters θ can then be defined as the union of W_j, b_j, $j \in 1, 2, \ldots P$ where b_j represent the bias used in the j-th transformation layer f_j. The optimization objective that we aim to minimize is thus,

$$
L(\theta) = \frac{1}{N} \sum_{i=1}^{N} l(h(x_i, \theta), t_i) + \lambda_P \|W_P\|_F^2 + \sum_{j=1}^{P-1} \lambda_j L_{f_j}^2 \tag{21}
$$

in which

$$
\begin{aligned}
L_{f_j}^2 &= \|W_j\|_2^2 = \|W_j \hat{u_j}^*\|_2^2 \\
\text{where } \hat{u_j}^* &= \underset{\substack{u_j \\ \|u_j\|_2 = 1}}{\operatorname{argsup}} \|W_j u_j\|_2
\end{aligned} \tag{22}
$$

In the above equation, l is the squared hinge loss which for the prediction vector p and the ground truth one-hot class vector t is defined as,

$$
\begin{aligned}
l(p, t) &= \max(0, 1 + \max_{k \neq u} p_k - p_u)^2, \\
\text{where } t_u &= 1 \wedge t_v = 0, \forall v \neq u
\end{aligned} \tag{23}
$$

Note that the squared hinge loss combined with the F-norm regularization of the weight tensor in the last transformation layer along with linear activation in last activation layer acts as an SVM with the output of penultimate activation layer as its input. As explained in (Crammer & Singer, 2001), the minimization of hinge loss (and therefore its squared variant) implies minimization of *1 - margin* . However, as can be verified from Eq. 23 evaluation of *1 - margin* contains an implicit maximization step. Hence hinge loss minimization can be interpreted as the minimization of an entity *1- margin* which is evaluated using an implicit maximization step. We make use of this argument in section 4.

F-norm regularization (more popularly known as $l2$-regularization) (Moore & DeNero; Ng, 2004) have been extensively studied and has a bayesian interpretation of standard normal prior over the weight tensors. It, being an upper bound on the 2-norm, can serve as a replacement to the 2-norm of the tensors in the above optimization objective. In this work we aim to regularize by using the 2-norm of the tensors itself. We follow an approach similar to alternate optimization where we first compute a $\hat{u_j}^*$ which maximizes $\|W_j u_j\|_2$ with the constraint $\|u_j\|_2 = 1$ and then replace the term $\|W_j\|_2$ with $\|W_j \hat{u_j}^*\|_2$, thereafter using the conventional training of hierarchical networks using backpropagation of errors over a batch of training examples (more details of training in section 5). We repeat these two steps for a fixed number of iterations and analyze the results (section 5). There can be various ways to compute $\hat{u}_j^*$, but we follow a stochastic gradient descent approach which starts with a u_j initialized with uniform random numbers in $[-1, 1]$, normalize it to get $\hat{u}_j$, compute the gradient of $-\|W_j \hat{u}_j\|_2^2$ (serves as the objective to minimize) with respect to u_j using chain rule of derivatives and finally update

u_j using the computed gradient. After some fixed number of iterations, we get u_j^* which we normalize to finally get $\hat{u_j}^*$.

4. PROBABILISTIC AND MINIMAX INTERPRETATION OF INDUCED 2-NORM REGULARIZATION

The bayesian interpretation of F-norm regularization over a tensor states that each element of the tensor independently and identically apriori follows a standard normal distribution. Similarly, the induced 2-norm regularization of a 2-tensor, W of size $m \times n$ where W_i denotes the i-th row of W, with regularization parameter λ can be viewed as a prior distribution over the 2-tensors given by,

$$\begin{aligned} p(W|\lambda) &= \exp(-\lambda \|W\|_2^2) \\ &= \exp(-\lambda \sup_{\substack{u \\ \|u\|_2=1}} \|Wu\|_2^2) \\ &= \exp(-\lambda \sup_{\substack{u \\ \|u\|_2=1}} \sum_{i=1}^{m} W_i u u^T W_i^T) \\ &= \inf_{\substack{u \\ \|u\|_2=1}} \exp(-\lambda \sum_{i=1}^{m} W_i u u^T W_i^T) \\ &= \inf_{\substack{\Lambda \\ \Lambda=uu^T \\ \|u\|_2=1}} \prod_{i=1}^{m} \exp(-\lambda W_i \Lambda W_i^T) \\ &\propto \inf_{\substack{\Lambda \\ \Lambda=uu^T \\ \|u\|_2=1}} \prod_{i=1}^{m} \mathcal{N}(W_i|0,\Lambda) \end{aligned} \tag{24}$$

where its negative log is given by,

$$\begin{aligned} -\log p(W|\lambda) &= \sup_{\substack{\Lambda \\ \Lambda=uu^T \\ \|u\|_2=1}} \sum_{i=1}^{m} \lambda W_i \Lambda W_i^T \\ &= \sup_{\substack{u \\ \|u\|_2=1}} \sum_{i=1}^{m} \lambda W_i u u^T W_i^T \end{aligned} \tag{25}$$

Therefore, induced 2-norm regularization of a 2-tensor puts a normal prior with mean 0 and precision Λ over the rows of the 2-tensor where the precision matrix is chosen from the set of matrices $\{uu^T | u^T u = 1\}$ in such a way the combined likelihood of rows of the 2-tensors is minimized. In our experiments, we update this precision matrix after every batch update. Computing the MAP estimate of the parameters involve the minimization of squared hinge loss (a risk based on parameters) and the negative log of a prior which is a *least favorable normal prior* with mean 0 and restrictions over precision matrix, over rows of 2-tensor, the expression of both of which involve a maximization step (Eq. 23 and Eq. 25). Therefore, the 2-norm regularization of 2-tensors in transformation layers along with minimization of squared hinge loss can be interpreted as a *minimax classifier*. This is in contrast to F-norm regularization in which the negative log of prior probability does not involve any maximization step and thus the corresponding formulation is not minimax. Based on the theory of minimax optimization, we claim that the predictions made by a network trained with squared hinge loss and 2-norm regularization are robust to the additive noise in the input and we aim to verify this claim in our future work.

5. Experiments and Analysis

To verify our claim of better guarantee of margin (higher lower bound on margin) in the input space using induced 2-norm regularization as compared to F-norm and no regularization, we conducted experiments over 4 synthetic datasets with low-dimensional inputs. The details of the generation of examples in these datasets is provided in the supplementary material. We used a network with an input layer of dimension equal to the dimension of examples in the dataset, 3 subsequent hidden layers comprising of 32 units each and finally an output layer of dimension equal to the number of classes. Fully connected transformation layers with relu activation layers (last transformation layer has linear activation layer) are applied over the input layer and the subsequent hidden layers to get the next hidden layers and output layer. The weight tensor in the last transformation layer was regularized with F-norm with the regularization parameter of $\lambda_P = 0.01$ and the weight tensors in other transformation layers were regularized with either of induced 2-norm, F-norm or no-norm with a regularization parameter of $\lambda_j = 0.001$, $j \in \{1, 2, 3\}$ for all synthetic datasets (Eq. 21). The network was trained for 200 epochs and with a minibatch size of 256 with squared hinge loss (Rosasco et al., 2004; Crammer & Singer, 2001) and adam gradient descent (Kingma & Ba, 2014). We show the final decision boundaries learnt in the input space and the value of the lower bound on margin in the inputs space for each dataset as the training progresses (with a value of zero when train or validation accuracy is not 100%) which is computed by computing the margin in the input space of the last transformation layer (it is possible to obtain this margin because the last activation layer is linear) and then dividing it by the product of 2-norms of weight tensors occurring before the last transformation layer (Cor. 1). We present the results in Fig. 1. The maximum value of the lower bound (when the training and validation accuracy is 100%) for each synthetic dataset corresponding to each

regularizer is shown in Table 1. Table 1 and plots in Fig. 1 are consistent with our claim of better guarantee of margin in the input space with induced 2-norm regularization as compared to F-norm and no regularization. From Fig. 2, it is easy to observe that the decision boundaries with induced 2-norm and F-norm regularization are much closer to the optimal decision boundary (one having maximum margin in the input space) as compared to the decision boundary obtained without any regularization, however, visually, the difference between their decision boundaries is marginal.

	Regularizer		
Dataset	**None**	**F-norm**	**Induced 2-norm**
1	0.055	0.063	**0.128**
2	0.037	0.090	**0.204**
3	0.005	0.018	**0.055**
4	0.047	0.074	**0.13**

Table 1. Comparison of maximum value of lower bound on margin in the input space of the nonlinear hierarchical network for each synthetic dataset (see Fig. 2) corresponding to induced 2-norm, F-norm and no regularizers.

6. Conclusions and Future Work

In this work, we formally derived a lower bound on the margin in the input space of a general deep hierarchical network, each layer of which can represent any (possibly nonlinear) mapping, in terms of the margin in the input space of the last layer and the best lipschitz constants of the intermediate layers (Th. 1 and Cor. 1) . We showed that in case of deep neural networks which are commonly composed of alternative linear transformation layers (fully connected and convolution) and activation layers (linear, tanh, relu, sigmoid), maximization of the derived lower bound is equivalent to the minimization of hinge loss (or its squared variant) (Crammer & Singer, 2001) and the induced 2-norms of the tensors in the underlying linear transformation layers (Th. 2) with the assumption that the metric defined on the input space and the output space of the subsequent mappings is 2-norm. We then described the implementation of induced 2-norm regularization (section 3) and the similarity of the resulting optimization with minimax optimization (section 4). Finally, we trained a neural network for the task of classification on synthetic datasets with induced 2-norm, F-norm and no regularization with same settings of hyperparameters and verified that the induced 2-norm regularization gives the best guarantee of margin in the input space in comparison to F-norm and no regularization (Table 1). A limitation of induced 2-norm regularization of a tensor W is that it requires recomputation of x such that $\|x\|_2 = 1$ which maximizes $\|Wx\|_2$, after ever batch which is currently preventing us from testing it over real datasets. We plan to devise a computationally efficient approach which can reflect small changes in W due to the backpropagation of errors, in the corresponding x. As claimed in (Szegedy et al., 2013), the networks with lower value of the lipschitz constants of the underlying layers (which we showed is equivalent to lower value of 2-norm of the tensors in those layers if the metric defined in the input and output space is 2-norm (Th. 2)) have better spectral stability and are robust to additive noise and thus are claimed to correctly classify adversarial examples. We aim to verify this claim in our future work. Note that it is extremely challenging to compute the exact margin in the input space of a nonlinear hierarchical network and hence no comparison can be made over the actual margin obtained in the input space with each of the regularizers.

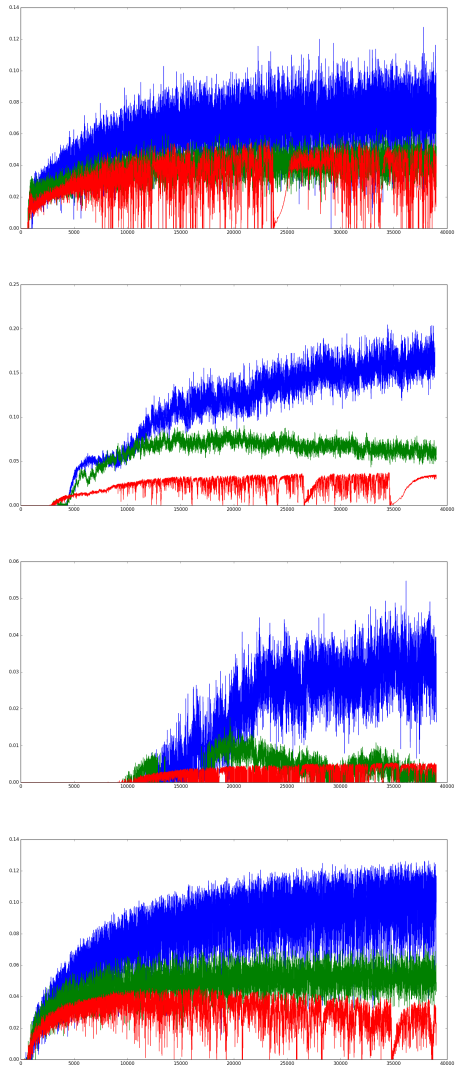
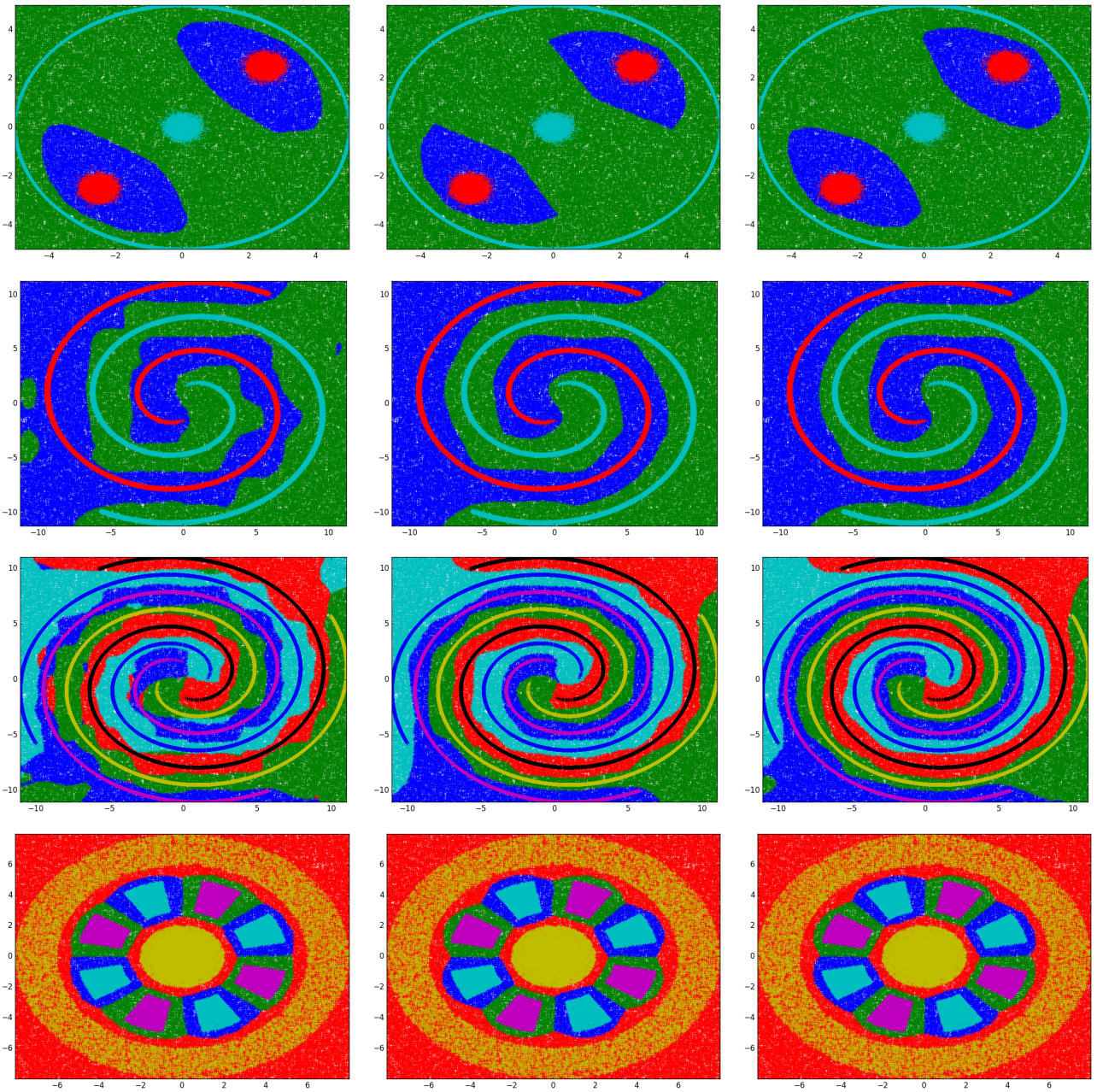


Figure 1. Plots of lower bound on margin in the input space of the 4 synthetic datasets (see Fig. 2) vs iteration number using induced 2-norm (blue), F-norm (green) and no (red) regularization.



(a) (b) (c)

Figure 2. Decision boundary in the input space of 4 synthetic datasets with (a) no regularization, (b) F-norm regularization and (c) induced 2-norm regularization. Refer the supplementary material for a script to generate these datasets and their plots without decision boundaries.

References

An, Senjian, Hayat, Munawar, Khan, Salman H, Bennamoun, Mohammed, Boussaid, Farid, and Sohel, Ferdous. Contractive rectifier networks for nonlinear maximum margin classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2515–2523, 2015.

Bruna, Joan and Mallat, Stéphane. Invariant scattering convolution networks. *CoRR*, abs/1203.1513, 2012. URL http://arxiv.org/abs/1203.1513.

Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL http://dx.doi.org/10.1007/BF00994018.

Crammer, Koby and Singer, Yoram. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec): 265–292, 2001.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

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

Moore, Robert and DeNero, John. L1 and l2 regularization for multiclass hinge loss models.

Ng, Andrew Y. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 78. ACM, 2004.

Rosasco, Lorenzo, De Vito, Ernesto, Caponnetto, Andrea, Piana, Michele, and Verri, Alessandro. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian J., and Fergus, Rob. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.