Learning Latent Space Models with Angular Constraints

Pengtao Xie [1 2] **Yuntian Deng** [3] **Yi Zhou** [4] **Abhimanu Kumar** [5] **Yaoliang Yu** [6] **James Zou** [7] **Eric P. Xing** [2]

Abstract

The large model capacity of latent space models (LSMs) enables them to achieve great performance on various applications, but meanwhile renders LSMs to be prone to overfitting. Several recent studies investigate a new type of regularization approach, which encourages components in LSMs to be diverse, for the sake of alleviating overfitting. While they have shown promising empirical effectiveness, in theory why larger "diversity" results in less overfitting is still unclear. To bridge this gap, we propose a new diversity-promoting approach that is both theoretically analyzable and empirically effective. Specifically, we use near-orthogonality to characterize "diversity" and impose angular constraints (ACs) on the components of LSMs to promote diversity. A generalization error analysis shows that larger diversity results in smaller estimation error and larger approximation error. An efficient ADMM algorithm is developed to solve the constrained LSM problems. Experiments demonstrate that ACs improve generalization performance of LSMs and outperform other diversity-promoting approaches.

1. Introduction

Latent space models (LSMs), such as sparse coding (Olshausen & Field, 1997), topic models (Blei et al., 2003) and neural networks, are widely used in machine learning to extract hidden patterns and learn latent representations of data. An LSM consists of a set of *components*. Each component aims at capturing one latent pattern and is parameterized by a weight vector. For instance, in a topic model (Blei et al., 2003), the components are referred to as *topics*, aiming at discovering the semantics underlying documents. Each topic is associated with a weight vector. The modeling power of LSMs can be very large when the number of components is large and the dimension of weight vectors is high. For instance, in the LightLDA (Yuan et al., 2015) topic model, the number of topics is 1 million and the dimension of topic vector is 50000, resulting in a topic matrix with 50 billion parameters. The vast model capacity of LSMs enables them to flexibly adapt to the complex patterns underlying data and achieve great predictive performance therefrom.

While highly expressive, LSMs are prone to overfitting, because of their large amount of model parameters. A key ingredient to successfully train LSMs is regularization, which imposes certain control over the model parameters to reduce model complexity and improve the generalization performance on unseen data. Many regularizers have been proposed, including ℓ_2 regularization, ℓ_1 regularization (Tibshirani, 1996), nuclear norm (Recht et al., 2010), Dropout (Srivastava et al., 2014) and so on.

Recently, a new type of regularization approaches (Yu et al., 2011; Zou & Adams, 2012a; Xie et al., 2015; 2016a; Rodríguez et al., 2016), which aim at encouraging the weight vectors of components in LSMs to be "diverse", are emerging. Zou & Adams (2012b) apply Determinantal Point Process (Kulesza & Taskar, 2012) to encourage the topic vectors in LDA to be "diverse". Bao et al. (2013) develop a softmax regularizer to promote incoherence among hidden units in neural network. Xie et al. (2015) propose an angle-based regularizer to "diversify" the weight vectors in Restricted Boltzmann Machine (RBM). While these approaches have demonstrated promising effectiveness on a wide range of empirical studies, in theory how they reduce overfitting is still unclear. One intuitive explanation could be: promoting diversity imposes a structural constraint on model parameters, which reduces the model capacity of LSMs and therefore alleviates overfitting. However, how to make this formal is challenging. In this paper, we aim to bridge this gap, by proposing a diversity-promoting approach that is both empirically effective and theoretically analyzable. We use near-orthogonality to represent "diversity" and propose to learn LSMs with angular

[1]Machine Learning Department, Carnegie Mellon University [2]Petuum Inc. [3]School of Engineering and Applied Sciences, Harvard University [4]College of Engineering and Computer Science, Syracuse University [5]Groupon Inc. [6]School of Computer Science, University of Waterloo [7]Department of Biomedical Data Science, Stanford University. Correspondence to: Pengtao Xie <pengtaox@cs.cmu.edu>, Eric P. Xing <eric.xing@petuum.com>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

constraints (ACs) where the angle between components is constrained to be close to $\frac{\pi}{2}$, which hence encourages the components to be close to orthogonal (therefore "diverse"). Using sparse coding and neural network as study cases, we analyze how ACs affect the generalization performance of these two LSMs. The analysis shows that the more close to $\frac{\pi}{2}$ the angles are, the smaller the estimation error is and the larger the approximation error is. The best tradeoffs of these two errors can be explored by properly tuning the angles. We develop an alternating direction method of multipliers (ADMM) (Boyd et al., 2011) algorithm to solve the angle-constrained LSM (AC-LSM) problems. In various experiments, we demonstrate that ACs improve the generalization performance of LSMs and outperform other diversity-promoting regularization approaches.

The major contributions of this work are:

- We propose a new approach to promote diversity in LSMs, by imposing angular constraints (ACs) on components, for the sake of alleviating overfitting.
- We perform theoretical analysis on how ACs affect the generalization error of two exemplar LSMs: sparse coding and neural networks.
- We develop an efficient ADMM algorithm to solve the AC-LSM problems.
- Empirical evaluation demonstrates that ACs are very effective in reducing overfitting and outperforms other diversity-promoting approaches.

The rest of the paper is organized as follows. Section 2 reviews related works. Second 3 introduces the angle-constrained LSMs and Section 4 gives the theoretical analysis. Section 5 presents experimental results and Section 6 concludes the paper.

2. Related Works

Diversity-promoting learning of latent space models has been widely studied recently. Ramirez et al. (2010) define a regularizer based on squared Frobenius norm to encourage the dictionary in sparse coding to be incoherent. Zou & Adams (2012a) use the determinantal point process (DPP) (Kulesza & Taskar, 2012) to encourage the location vectors in Gaussian mixture model (GMM) and topics in latent Dirichlet allocation (Blei et al., 2003) to be "diverse". Given m vectors $\{\mathbf{w}_j\}_{j=1}^m$, DPP is defined as $\log\det(\mathbf{G})$. $\mathbf{G}$ is a kernel matrix where $G_{ij} = k(\mathbf{w}_i, \mathbf{w}_j)$ and $k(\cdot,\cdot)$ is a kernel function. $\det(\mathbf{G})$ is the volume of the parallelepiped formed by $\{\phi(\mathbf{w}_j)\}_{j=1}^m$, where $\phi(\cdot)$ denotes the reproducing kernel feature map associated with kernel k. Vectors with larger volume are considered to be more diverse since they are more spread out. Xie (2015) develop an angle-based regularizer to encourage the weight vectors of hidden units in restricted Boltzmann machine to be close to orthogonal. The non-obtuse angle between each pair of weight vectors is measured and the regularizer is defined as the mean of these angles minus their variance. A larger mean encourages the vectors to have larger angles overall and a smaller variance encourages the vectors to be evenly different from each other. Xie (2015) apply this regularizer to encourage the projection vectors in distance metric learning to be diverse and show that promoting diversity can reduce model size without sacrificing modeling power. Besides frequentist-style regularization, diversity-promoting learning is also investigated in Bayesian learning where the components are random variables. Affandi et al. (2013) apply DPP as a repulsive prior to encourage the location vectors in GMM to be far apart. Xie et al. (2016a) propose a mutual angular prior that has an inductive bias towards vectors having larger angles.

In the literature of neural networks, many works have studied the "diversification" of hidden units. Le et al. (2010) apply a strict-orthogonality constraint over the weight parameters to make the hidden units uncorrelated (therefore "diverse"). In practice, this hard constraint might be too restrictive and hurts performance, as we will confirm in experiments. Bao et al. (2013) propose a softmax regularizer to encourage the weight vectors of hidden units to have small cosine similarity. Cogswell et al. (2015) propose to decorrelate hidden activations by minimizing their covariance. In convolutional neural networks (CNNs) where the number of activations is much larger than that of weight parameters, this regularizer is computationally prohibitive since it is defined over activations rather than weights. Henaff et al. (2016) perform a study to show that random orthogonal initialization of the weight matrices in recurrent neural networks improves its ability to perform long-memory tasks. Xiong et al. (2016) propose a structured decorrelation constraint which groups hidden units and encourages units within the same group to have strong connections during the training procedure and forces units in different groups to learn nonredundant representations by minimizing the cross-covariance between them. Rodríguez et al. (2016) show that regularizing negatively correlated features inhibits effective diversity and propose a solution which locally enforces feature orthogonality. Chen et al. (2017) propose a group orthogonal CNN which leverages side information to learn diverse feature representations. Mao et al. (2017) impose a stochastic decorrelation constraint based on covariance to reduce the co-adaptation of hidden units. Xie et al. (2017) define a kernel-based regularizer to promote diversity and analyze how it affects the generalize performance of neural networks (NNs). It is unclear how to generalize the analysis to other LSMs.

Diversity-promoting learning has been investigated in non-LSM models as well. In multi-class classification, Malkin & Bilmes (2008) encourage the coefficient vectors of different classes to be diverse by maximizing the determi-

nant of the covariance matrix of the coefficient vectors. In classifiers ensemble, Yu et al. (2011) develop a regularizer to encourage the coefficient vectors of support vector machines (SVMs) to have small cosine similarity and analyze how this regularizer affects the generalization performance. The analysis is specific to SVM ensemble. It is unclear how to generalize it to latent space models.

3. Methods

In this section, we propose Angle-Constrained Latent Space Models (AC-LSMs) and present an ADMM algorithm to solve them.

3.1. Latent Space Models with Angular Constraints

An LSM consists of m components and these components are parameterized by vectors $\mathcal{W} = \{\mathbf{w}_j\}_{j=1}^m$. Let $\mathcal{L}(\mathcal{W})$ denote the objective function of this LSM. Similar to (Bao et al., 2013; Xie et al., 2015; Rodríguez et al., 2016), we use angle to characterize diversity: the components are considered to be more diverse if they are close to being orthogonal, i.e., their angles are close to $\frac{\pi}{2}$. To encourage this, we require the absolute value of cosine similarity between each pair of components to be less than a small value τ, which leads to the following angle-constrained LSM (AC-LSM) problem

$$\begin{array}{ll} \min\limits_{\mathcal{W}} & \mathcal{L}(\mathcal{W}) \\ s.t. & 1 \leq i < j \leq m, \frac{|\mathbf{w}_i \cdot \mathbf{w}_j|}{\|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2} \leq \tau \end{array} \tag{1}$$

The parameter τ controls the level of near-orthogonality (or diversity). A smaller τ indicates that the vectors are more close to being orthogonality, and hence are more diverse. As will be shown later, representing diversity using the angular constraints facilitates theoretical analysis and is empirically effective as well.

3.2. Case Studies

In this section, we apply the ACs to two LSMs.

Sparse Coding Given a set of data samples $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x} \in \mathbb{R}^d$, sparse coding (SC) (Olshausen & Field, 1997) aims to use a set of "basis" vectors (referred to as *dictionary*) $\mathcal{W} = \{\mathbf{w}_j\}_{j=1}^m$ to reconstruct the data samples. Each data sample $\mathbf{x}$ is reconstructed by taking a sparse linear combination of the basis vectors $\mathbf{x} \approx \sum_{j=1}^m \alpha_j \mathbf{w}_j$ where $\{\alpha_j\}_{j=1}^m$ are the linear coefficients (referred to as *sparse codes*) and most of them are zero. The reconstruction error is measured using the squared ℓ_2 norm $\|\mathbf{x} - \sum_{j=1}^m \alpha_j \mathbf{w}_j\|_2^2$. To achieve sparsity among the coefficients, ℓ_1-regularization is utilized: $\sum_{j=1}^m |\alpha_j|_1$. To avoid the degenerated case where most coefficients are zero and the basis vectors are of large magnitude, ℓ_2-regularization is applied to the basis vectors: $\|\mathbf{w}_j\|_2^2$. Putting these pieces together, we learn the basis vectors and sparse codes (denoted by $\mathcal{A}$) by minimizing the following objective function: $\mathcal{L}(\mathcal{W}, \mathcal{A}) = \frac{1}{2}\sum_{i=1}^n (\|\mathbf{x}_i - \sum_{j=1}^m \alpha_{ij}\mathbf{w}_j\|_2^2 + \lambda_1 \sum_{j=1}^m |\alpha_{ij}|_1) + \lambda_2 \sum_{j=1}^m \|\mathbf{w}_j\|_2^2$. Applying ACs to the basis vectors, we obtain the following AC-SC problem:

$$\begin{array}{ll} \min_{\mathcal{W},\mathcal{A}} & \mathcal{L}(\mathcal{W}, \mathcal{A}) \\ s.t. & 1 \leq i < j \leq m, \frac{|\mathbf{w}_i \cdot \mathbf{w}_j|}{\|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2} \leq \tau \end{array} \tag{2}$$

Neural Networks In a neural network (NN) with L hidden layers, each hidden layer l is equipped with $m^{(l)}$ units and each unit i is connected with all units in layer $l-1$. Hidden unit i at layer l is parameterized by a weight vector $\mathbf{w}_i^{(l)}$. These hidden units aim at capturing latent features underlying data. Applying ACs to the weight vectors of hidden units, we obtain the following AC-NN problem

$$\begin{array}{ll} \min\limits_{\mathcal{W}} & \mathcal{L}(\mathcal{W}) \\ s.t. & \forall\, 1 \leq l \leq L, 1 \leq i < j \leq m^{(l)}, \frac{|\mathbf{w}_i^{(l)} \cdot \mathbf{w}_j^{(l)}|}{\|\mathbf{w}_i^{(l)}\|_2 \|\mathbf{w}_j^{(l)}\|_2} \leq \tau \end{array}$$

where $\mathcal{W}$ denotes weight vectors in all layers and $\mathcal{L}(\mathcal{W})$ is the objective function of this NN.

3.3. Algorithm

In this section, we develop an ADMM-based algorithm to solve the AC-LSM problem. To make it amenable for optimization, we first factorize each weight vector $\mathbf{w}$ into its ℓ_2 norm $g = \|\mathbf{w}\|_2$ and direction $\widetilde{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$. Under such a factorization, $\mathbf{w}$ can be reparameterized as $\mathbf{w} = g\widetilde{\mathbf{w}}$, where $g > 0$ and $\|\widetilde{\mathbf{w}}\|_2 = 1$. Then the problem defined in Eq.(1) can be transformed into

$$\begin{array}{ll} \min\limits_{\widetilde{\mathcal{W}},\mathcal{G}} & \mathcal{L}(\widetilde{\mathcal{W}}, \mathcal{G}) \\ s.t. & \forall j, g_j \geq 0, \|\widetilde{\mathbf{w}}_j\|_2 = 1 \\ & \forall i \neq j, |\widetilde{\mathbf{w}}_i \cdot \widetilde{\mathbf{w}}_j| \leq \tau \end{array} \tag{3}$$

where $\widetilde{\mathcal{W}} = \{\widetilde{\mathbf{w}}_j\}_{j=1}^m$ and $\mathcal{G} = \{g_j\}_{j=1}^m$. We solve this new problem by alternating between $\widetilde{\mathcal{W}}$ and $\mathcal{G}$. Fixing $\widetilde{\mathcal{W}}$, the problem defined over $\mathcal{G}$ is: $\min_{\mathcal{G}} \mathcal{L}(\mathcal{G})$ $s.t.$ $\forall j, g_j \geq 0$, which can be solved using projected gradient descent. Fixing $\mathcal{G}$, the sub-problem defined over $\widetilde{\mathcal{W}}$ is

$$\begin{array}{ll} \min\limits_{\widetilde{\mathcal{W}}} & \mathcal{L}(\widetilde{\mathcal{W}}) \\ s.t. & \forall j, \|\widetilde{\mathbf{w}}_j\|_2 = 1 \\ & \forall i \neq j, |\widetilde{\mathbf{w}}_i \cdot \widetilde{\mathbf{w}}_j| \leq \tau \end{array} \tag{4}$$

which we solve using an ADMM algorithm. There are $R = m(m-1)$ pairwise constraints $|\widetilde{\mathbf{w}}_i \cdot \widetilde{\mathbf{w}}_j| \leq \tau$. For the r-th constraint, let $p(r)$ and $q(r)$ be the index of the first and second vector respectively, i.e., the r-th constraint is $|\widetilde{\mathbf{w}}_{p(r)} \cdot \widetilde{\mathbf{w}}_{q(r)}| \leq \tau$. First, we introduce auxiliary variables $\{\mathbf{v}_1^{(r)}\}_{r=1}^R$ and $\{\mathbf{v}_2^{(r)}\}_{r=1}^R$, to rewrite the problem in

Eq.(4) into an equivalent form. For each pairwise constraint: $|\widetilde{\mathbf{w}}_{p(r)} \cdot \widetilde{\mathbf{w}}_{q(r)}| \leq \tau$, we introduce two auxiliary vectors $\mathbf{v}_1^{(r)}$ and $\mathbf{v}_2^{(r)}$, and let $\widetilde{\mathbf{w}}_{p(r)} = \mathbf{v}_1^{(r)}$, $\widetilde{\mathbf{w}}_{q(r)} = \mathbf{v}_2^{(r)}$, $\|\mathbf{v}_1^{(r)}\|_2 = 1$, $\|\mathbf{v}_2^{(r)}\|_2 = 1$, $|\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)}| \leq \tau$. To this end, we obtain the following problem

$$\begin{aligned} \min_{\widetilde{\mathcal{W}},\mathcal{V}} \quad & \mathcal{L}(\widetilde{\mathcal{W}}) \\ s.t. \quad & \forall j, \|\widetilde{\mathbf{w}}_j\|_2 = 1 \\ & \forall r, \widetilde{\mathbf{w}}_{p(r)} = \mathbf{v}_1^{(r)}, \widetilde{\mathbf{w}}_{q(r)} = \mathbf{v}_2^{(r)} \\ & \forall r, \|\mathbf{v}_1^{(r)}\|_2 = 1, \|\mathbf{v}_2^{(r)}\|_2 = 1, |\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)}| \leq \tau \end{aligned}$$

where $\mathcal{V} = \{(\mathbf{v}_1^{(r)}, \mathbf{v}_2^{(r)})\}_{r=1}^R$. Then we define the augmented Lagrangian, with Lagrange multipliers $\mathcal{Y} = \{(\mathbf{y}_1^{(r)}, \mathbf{y}_2^{(r)})\}_{r=1}^R$ and parameter ρ

$$\begin{aligned} \min_{\widetilde{\mathcal{W}},\mathcal{V},\mathcal{Y}} \quad & \mathcal{L}(\widetilde{\mathcal{W}}) + \sum_{r=1}^R (\mathbf{y}_1^{(r)} \cdot (\widetilde{\mathbf{w}}_{p(r)} - \mathbf{v}_1^{(r)}) \\ & + \mathbf{y}_2^{(r)} \cdot (\widetilde{\mathbf{w}}_{q(r)} - \mathbf{v}_2^{(r)}) + \tfrac{\rho}{2}\|\widetilde{\mathbf{w}}_{p(r)} - \mathbf{v}_1^{(r)}\|_2^2 \\ & + \tfrac{\rho}{2}\|\widetilde{\mathbf{w}}_{q(r)} - \mathbf{v}_2^{(r)}\|_2^2) \\ s.t. \quad & \forall j, \|\widetilde{\mathbf{w}}_j\|_2 = 1 \\ & \forall r, \|\mathbf{v}_1^{(r)}\|_2 = 1, \|\mathbf{v}_2^{(r)}\|_2 = 1, |\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)}| \leq \tau \end{aligned}$$

which can be solved by alternating between $\widetilde{\mathcal{W}}$, $\mathcal{V}$, $\mathcal{Y}$.

Solve $\widetilde{\mathcal{W}}$ The sub-problem defined over $\widetilde{\mathcal{W}}$ is

$$\begin{aligned} \min_{\widetilde{\mathcal{W}}} \quad & \mathcal{L}(\widetilde{\mathcal{W}}) + \sum_{r=1}^R (\mathbf{y}_1^{(r)} \cdot \widetilde{\mathbf{w}}_{p(r)} + \mathbf{y}_2^{(r)} \cdot \widetilde{\mathbf{w}}_{q(r)} \\ & + \tfrac{\rho}{2}\|\widetilde{\mathbf{w}}_{p(r)} - \mathbf{v}_1^{(r)}\|_2^2 + \tfrac{\rho}{2}\|\widetilde{\mathbf{w}}_{q(r)} - \mathbf{v}_2^{(r)}\|_2^2) \quad (5) \\ s.t. \quad & \forall j, \|\widetilde{\mathbf{w}}_j\|_2 = 1 \end{aligned}$$

For sparse coding, we solve this sub-problem using coordinate descent. At each iteration, we update $\widetilde{\mathbf{w}}_j$ by fixing the other variables. Please refer to the supplements for details. For neural network, this sub-problem can be solved using projected gradient descent which iteratively performs the following three steps: (1) compute the gradient of $\widetilde{\mathbf{w}}_j$ using backpropagation; (2) perform a gradient descent update of $\widetilde{\mathbf{w}}_j$; (3) project each vector onto the unit sphere: $\widetilde{\mathbf{w}}_j \leftarrow \widetilde{\mathbf{w}}_j / \|\widetilde{\mathbf{w}}_j\|_2$.

Solve $\mathbf{v}_1^{(r)}, \mathbf{v}_2^{(r)}$ The corresponding sub-problem is

$$\begin{aligned} \min_{\mathbf{v}_1^{(r)},\mathbf{v}_2^{(r)}} \quad & -\mathbf{y}_1^{(r)} \cdot \mathbf{v}_1^{(r)} - \mathbf{y}_2^{(r)} \cdot \mathbf{v}_2^{(r)} \\ & + \tfrac{\rho}{2}\|\widetilde{\mathbf{w}}_{p(r)} - \mathbf{v}_1^{(r)}\|_2^2 + \tfrac{\rho}{2}\|\widetilde{\mathbf{w}}_{q(r)} - \mathbf{v}_2^{(r)}\|_2^2 \\ s.t. \quad & \|\mathbf{v}_1^{(r)}\|_2 = 1, \|\mathbf{v}_2^{(r)}\|_2 = 1, \\ & \mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)} \leq \tau, -\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)} \leq \tau \end{aligned}$$

Let $\gamma_1, \gamma_2, \lambda_1 \geq 0, \lambda_2 \geq 0$ be the KKT multipliers associated with the four constraints in this sub-problem. According to the KKT conditions, we have

$$-\mathbf{y}_1^{(r)} + \rho(\mathbf{v}_1^{(r)} - \widetilde{\mathbf{w}}_{p(r)}) + 2\gamma_1 \mathbf{v}_1^{(r)} + (\lambda_1 - \lambda_2)\mathbf{v}_2^{(r)} = 0 \quad (6)$$

$$-\mathbf{y}_2^{(r)} + \rho(\mathbf{v}_2^{(r)} - \widetilde{\mathbf{w}}_{q(r)}) + 2\gamma_2 \mathbf{v}_2^{(r)} + (\lambda_1 - \lambda_2)\mathbf{v}_1^{(r)} = 0 \quad (7)$$

We solve these two equations by examining four cases.

Case 1 First, we assume $\lambda_1 = 0, \lambda_2 = 0$, then $(\rho + 2\gamma_1)\mathbf{v}_1^{(r)} = \mathbf{y}_1^{(r)} + \rho\widetilde{\mathbf{w}}_{p(r)}$ and $(\rho + 2\gamma_2)\mathbf{v}_2^{(r)} = \mathbf{y}_2^{(r)} + \rho\widetilde{\mathbf{w}}_{q(r)}$. According to the primal feasibility $\|\mathbf{v}_1^{(r)}\|_2 = 1$ and $\|\mathbf{v}_2^{(r)}\|_2 = 1$, we know

$$\mathbf{v}_1^{(r)} = \frac{\mathbf{y}_1^{(r)} + \rho\widetilde{\mathbf{w}}_{p(r)}}{\|\mathbf{y}_1^{(r)} + \rho\widetilde{\mathbf{w}}_{p(r)}\|_2}, \ \mathbf{v}_2^{(r)} = \frac{\mathbf{y}_2^{(r)} + \rho\widetilde{\mathbf{w}}_{q(r)}}{\|\mathbf{y}_2^{(r)} + \rho\widetilde{\mathbf{w}}_{q(r)}\|_2}$$

Then we check whether the constraint $|\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)}| \leq \tau$ is satisfied. If so, then $\mathbf{v}_1^{(r)}$ and $\mathbf{v}_2^{(r)}$ are the optimal solution.

Case 2 We assume $\lambda_1 > 0$ and $\lambda_2 = 0$, then

$$(\rho + 2\gamma_1)\mathbf{v}_1^{(r)} + \lambda_1 \mathbf{v}_2^{(r)} = \mathbf{y}_1^{(r)} + \rho\widetilde{\mathbf{w}}_{p(r)} \quad (8)$$

$$(\rho + 2\gamma_2)\mathbf{v}_2^{(r)} + \lambda_1 \mathbf{v}_1^{(r)} = \mathbf{y}_2^{(r)} + \rho\widetilde{\mathbf{w}}_{q(r)} \quad (9)$$

According to the complementary slackness condition, we know $\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)} = \tau$. For the vectors on both sides of Eq.(8), taking the square of their ℓ_2 norm, we get

$$(\rho + 2\gamma_1)^2 + \lambda_1^2 + 2(\rho + 2\gamma_1)\lambda_1\tau = \|\mathbf{y}_1^{(r)} + \rho\widetilde{\mathbf{w}}_{p(r)}\|_2^2 \quad (10)$$

Similarly, from Eq.(9), we get

$$(\rho + 2\gamma_2)^2 + \lambda_1^2 + 2(\rho + 2\gamma_2)\lambda_1\tau = \|\mathbf{y}_2^{(r)} + \rho\widetilde{\mathbf{w}}_{q(r)}\|_2^2 \quad (11)$$

Taking the inner product of the two vectors on the left hand sides of Eq.(8,9), and that on the right hand sides, we get

$$\begin{aligned} & (2\rho + 2\gamma_1 + 2\gamma_2)\lambda_1 + ((\rho + 2\gamma_1)(\rho + 2\gamma_2) + \lambda_1^2)\tau \\ & = (\mathbf{y}_1^{(r)} + \rho\widetilde{\mathbf{w}}_{p(r)})^\top (\mathbf{y}_2^{(r)} + \rho\widetilde{\mathbf{w}}_{q(r)}) \end{aligned} \quad (12)$$

Solving the system of equations consisting of Eq.(10-12), we obtain the optimal values of γ_1, γ_2 and λ_1. Plugging them into Eq.(8) and Eq.(9), we obtain a solution of $\mathbf{v}_1^{(r)}$ and $\mathbf{v}_2^{(r)}$. Then we check whether this solution satisfies $-\mathbf{v}_1^{(r)} \cdot \mathbf{v}_2^{(r)} \leq \tau$. If so, this is an optimal solution.

In Case 3, we discuss $\lambda_1 = 0$, $\lambda_2 > 0$. In Case 4, we discuss $\lambda_1 > 0$, $\lambda_2 > 0$. The corresponding problems can be solved in a similar way as Case 2. Please refer to the supplements for details.

Solve $\mathbf{y}_1^{(r)}, \mathbf{y}_2^{(r)}$ We simply perform the following:

$$\mathbf{y}_1^{(r)} = \mathbf{y}_1^{(r)} + \rho(\widetilde{\mathbf{w}}_{p(r)} - \mathbf{v}_1^{(r)}) \tag{13}$$

$$\mathbf{y}_2^{(r)} = \mathbf{y}_2^{(r)} + \rho(\widetilde{\mathbf{w}}_{q(r)} - \mathbf{v}_2^{(r)}) \tag{14}$$

Compared with a vanilla backpropagation algorithm, the major extra cost in this ADMM algorithm comes from solving the $R = m(m-1)$ pairs of vectors $\{\mathbf{v}_1^{(r)}, \mathbf{v}_2^{(r)}\}_{r=1}^R$. Solving each pair incurs $O(m)$ cost. The R pairs bring in a total cost of $O(m^3)$. Such a cubic cost is also incurred in other decorrelation methods such as (Le et al., 2010; Bao et al., 2013). In practice, m is typically less than 1000. This $O(m^3)$ cost does not substantially bottleneck computation, as we will validate in experiments.

4. Analysis

In this section, we discuss how the parameter τ which controls the level of diversity affects the generalization performance of sparse coding and neural network.

4.1. Sparse Coding

Following (Vainsencher et al., 2011), we assume the data example $\mathbf{x} \in \mathbb{R}^d$ and basis vector $\mathbf{w} \in \mathbb{R}^d$ are both of unit length, and the linear coefficient vector $\mathbf{a} \in \mathbb{R}^m$ is at most k sparse, i.e., $\|\mathbf{a}\|_0 \leq k$. The estimation error of dictionary $\mathcal{W}$ is defined as

$$L(\mathcal{W}) = \mathbb{E}_{\mathbf{x}\sim p^*}[\min_{\|\mathbf{a}\|_0 \leq k} \|\mathbf{x} - \sum_{j=1}^m a_j \mathbf{w}_j\|_2]. \tag{15}$$

Let $\widetilde{L}(\mathcal{W}) = \frac{1}{n}\sum_{i=1}^n \min_{\|\mathbf{a}\|_0 \leq k} \|\mathbf{x}_i - \sum_{j=1}^m a_j \mathbf{w}_j\|_2$ be the empirical reconstruction error on n samples. We have the following theorem.

Theorem 1 *Assume $\tau < \frac{1}{k}$, with probability at least $1-\delta$:*

$$L(\mathcal{W}) \leq \widetilde{L}(\mathcal{W}) + \sqrt{\frac{dm \ln \frac{4\sqrt{n}k}{1-k\tau}}{2n}} + \sqrt{\frac{\ln 1/\delta}{2n}} + \sqrt{\frac{4}{n}}. \tag{16}$$

Note that the right hand side is an increasing function w.r.t τ. As expected, a smaller τ (implying more diversity) would induce a lower estimation error bound.

4.2. Neural Network

The generalization error of a hypothesis f represented with a neural network is defined as $L(f) = \mathbb{E}_{(\mathbf{x},y)\sim p^*}[\ell(f(\mathbf{x}), y)]$, where p^* is the distribution of input-output pair $(\mathbf{x}, y)$ and $\ell(\cdot,\cdot)$ is the loss function. The training error is $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^n \ell(f(\mathbf{x}^{(i)}), y^{(i)})$, where n is the number of training samples. Let $f^* \in \operatorname{argmin}_{f\in\mathcal{F}} L(f)$ be the true risk minimizer and $\hat{f} \in \operatorname{argmin}_{f\in\mathcal{F}} \hat{L}(f)$ be the empirical risk minimizer. We aim to analyze the generalization error $L(\hat{f})$ of the empirical risk minimizer $\hat{f}$. $L(\hat{f})$ can be decomposed into $L(\hat{f}) = L(\hat{f}) - L(f^*) + L(f^*)$, where $L(\hat{f}) - L(f^*)$ is the estimation error and $L(f^*)$ is the approximation error.

For simplicity, we start with a "simple" fully connected network with one hidden layer of m units, used for univariate regression (one output unit) with squared loss. Analysis for more complicated fully connected NNs with multiple hidden layers can be achieved in a straightforward way by cascading our analysis for this "simple" fully connected NN. Let $\mathbf{x} \in \mathbb{R}^d$ be the input vector and y be the response value. For simplicity, we assume $\max\{\|\mathbf{x}\|_2, |y|\} \leq 1$. Let $\mathbf{w}_j \in \mathbb{R}^d$ be the weights connecting the j-th hidden unit with input units, with $\|\mathbf{w}_j\|_2 \leq C$.

Let $\boldsymbol{\alpha}$ be a vector where α_j is the weight connecting hidden unit j to the output unit, with $\|\boldsymbol{\alpha}\|_2 \leq B$. We assume the activation function $h(t)$ applied on the hidden units is Lipschitz continuous with constant L. Commonly used activation functions such as rectified linear $h(t) = \max(0,t)$, tanh $h(t) = (e^t - e^{-t})/(e^t + e^{-t})$, and sigmoid $h(t) = 1/(1+e^{-t})$ are all Lipschitz continuous with $L = 1, 1, 0.25$, respectively. Consider the hypothesis set

$$\mathcal{F} = \{\mathbf{x} \mapsto \sum_{j=1}^m \alpha_j h(\mathbf{w}_j^\top \mathbf{x}) \mid \|\boldsymbol{\alpha}\|_2 \leq B, \|\mathbf{w}_j\|_2 \leq C, \\ \forall i \neq j, |\mathbf{w}_i \cdot \mathbf{w}_j| \leq \tau \|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2\}.$$

The estimation error given in Theorem 2 below indicates how well the algorithm is able to learn from the samples.

Theorem 2 *Let the activation function h be L-Lipschitz continuous and the loss $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$. Then, with probability at least $1-\delta$:*

$$L(\hat{f}) - L(f^*) \leq \frac{\gamma^2\sqrt{2\ln(4/\delta)} + 4\gamma B(2CL + |h(0)|)\sqrt{m}}{\sqrt{n}} \tag{17}$$

where $\gamma = 1 + BCL\sqrt{(m-1)\tau + 1} + \sqrt{m}B|h(0)|$.

Note that γ, hence the above bound on estimation error, decreases as τ becomes smaller. The bound goes to zero as n (sample size) goes to infinite. The inverse square root dependence on n matches existing results (Bartlett & Mendelson, 2003). We note that it is straightforward to extend our bound to any bounded Lipschitz continuous loss ℓ.

The approximation error indicates how capable the hypothesis set $\mathcal{F}$ is to approximate a target function $g = \mathbb{E}[y|\mathbf{x}]$, where the error is measured by $\min_{f\in\mathcal{F}} \|f - g\|_{L^2}$ and $\|f - g\|_{L^2}^2 = \int (f(\mathbf{x}) - g(\mathbf{x}))^2 P(\mathrm{d}\mathbf{x})$. Following (Barron, 1993), we assume the target function g satisfies certain smoothness condition that is expressed in the first moment of its Fourier representation: $\int \|\boldsymbol{\omega}\|_2 |\tilde{g}(\boldsymbol{\omega})| \mathrm{d}\boldsymbol{\omega} \leq B/2$

where $\tilde{g}(\boldsymbol{\omega})$ is the Fourier representation of $g(\mathbf{x})$. For such function the following theorem states its approximation error. (In order to derive explicit constants we restrict h to be the sigmoid function, but other Lipschitz continuous activation function can be similarly handled.)

Theorem 3 *Let $C > 1$, $m \leq 2(\lfloor \frac{\frac{\pi}{2}-\theta}{\theta} \rfloor + 1)$, where $\theta = \arccos(\tau)$, and $h(t) = 1/(1+e^{-t})$. Then, there is a function $f \in \mathcal{F}$ such that*

$$\|f - g\|_{L^2} \leq B(\tfrac{1}{\sqrt{m}} + \tfrac{1+2\ln C}{C}) + \\ + 2\sqrt{m}BC\sin(\tfrac{\min(3m\theta,\pi)}{2}). \quad (18)$$

This theorem implies that whether to use the angular constraint (AC) or not has a significant influence on the approximate error bound: without using AC ($\tau = 1$), the bound is a decreasing function of m (the number of hidden units); using AC ($\tau < 1$), the bound increases with m. This striking phrase-change indicates the impact of AC. Given a fixed m, the bound decreases with τ, implying that a stronger regularization (smaller τ) incurs larger approximation error. When $\tau = 1$, the second term in the bound vanishes and the bound is reduced to the one in (Barron, 1993), which is a decreasing function of m (and C, the upper bound on the weights). When $\tau < 1$, the second term increases with m up to the upper bound $2(\lfloor \frac{\frac{\pi}{2}-\theta}{\theta} \rfloor + 1)$. This is because a larger number of hidden units bear a larger difficulty in satisfying the pairwise ACs, which causes the function space $\mathcal{F}$ to shrink rapidly; accordingly, the approximation power of $\mathcal{F}$ decreases quickly.

The analysis in the two theorems shows that τ incurs a tradeoff between the estimation error and the approximation error: decreasing τ reduces the estimation error and enlarges the approximation error. Since the generalization error is the sum of the estimation error and the approximation error, τ has an optimal value to yield the minimal generalization error.

5. Experiments

In this section, we present experimental results. Due to space limit, we put some results into supplements.

5.1. Sparse Coding

Following (Yang et al., 2009), we applied sparse coding for image feature learning. We used three datasets in the experiments: Scenes-15 (Lazebnik et al., 2006), Caltech-256 (Griffin et al., 2007) and UIUC-Sport (Li & Fei-Fei, 2007). For each dataset, five random train/test splits are performed and the results are averaged over the five runs. We extract pixel-level dense SIFT (Lowe, 2004) features where the step size and patch size are 8 and 16 respectively. On top of the SIFT features, we use sparse coding methods to learn a dictionary and represent each SIFT feature into a sparse code. To obtain image-level features, we apply max-pooling (Yang et al., 2009) and spatial pyramid matching (Lazebnik et al., 2006; Yang et al., 2009) over the pixel-level sparse codes. Then a linear SVM is applied to classify the images. We compare with other diversity-promoting regularizers including determinant of covariance matrix (DCM) (Malkin & Bilmes, 2008), cosine similarity (CS) (Yu et al., 2011), determinantal point process (DPP) (Kulesza & Taskar, 2012; Zou & Adams, 2012b), InCoherence (IC) (Bao et al., 2013) and mutual angles (MA) (Xie et al., 2015). We use 5-fold cross validation to tune τ in $\{0.3, 0.4, \cdots, 1\}$ and the number of basis vectors in $\{50, 100, 200, \cdots, 500\}$. The parameter ρ in ADMM is set to 1.

	Scene	Caltech	Sports
SC	83.6 ± 0.2	42.3 ± 0.4	87.4 ± 0.5
DCM-SC	85.4 ± 0.5	44.7 ± 0.8	89.6 ± 0.1
CS-SC	84.8 ± 0.6	45.4 ± 0.5	88.3 ± 0.3
DPP-SC	84.6 ± 0.3	43.5 ± 0.3	88.1 ± 0.2
IC-SC	85.5 ± 0.1	43.9 ± 0.7	90.2 ± 0.7
MA-SC	86.1 ± 0.5	45.6 ± 0.4	89.7 ± 0.4
AC-SC	**86.5** ± 0.7	**46.1** ± 0.3	**90.9** ± 0.3

Table 1. Classification accuracy (%) on three datasets.

Table 1 shows the classification accuracy on three datasets, from which we can see that compared with unregularized SC, AC-SC greatly improves performance. For example, on the Sports dataset, AC improves the accuracy from 87.4% to 90.9%. This suggests that AC is effective in reducing overfitting and improving generalization performance. Compared with other diversity-promoting regularizers, AC achieves better performance, demonstrating its better efficacy in promoting diversity.

5.2. Neural Networks

We evaluate AC on three types of neural networks: fully-connected NN (FNN) for phone recognition (Hinton et al., 2012), CNN for image classification (Krizhevsky et al., 2012), and RNN for question answering (Seo et al., 2017). In the main paper, we report results on four datasets: TIMIT[1], CIFAR-10[2], CNN (Hermann et al., 2015), Daily Mail (Hermann et al., 2015). Please refer to the supplements for results on other datasets.

FNN for Phone Recognition The TIMIT dataset contains a total of 6300 sentences (5.4 hours), divided into a training set (462 speakers), a validation set (50 speakers) and a core test set (24 speakers). We used the Kaldi (Povey et al., 2011) toolkit to train the monophone system which was utilized to do forced alignment and to get labels for speech frames. The toolkit was also utilized to preprocess the data into log-filter banks. Among methods based on FNN, Karel's recipe in Kaldi achieves state

[1]https://catalog.ldc.upenn.edu/LDC93S1

[2]https://www.cs.toronto.edu/ kriz/cifar.html

Network	Error
Segmental NN (Abdel-Hamid et al., 2013)	21.9
MCRBM (Dahl et al., 2010)	20.5
DSR (Tang et al., 2015)	19.9
Rectifier NN (Tóth, 2013)	19.8
DBN (Srivastava et al., 2014)	19.7
Shallow CNN (Ba & Caruana, 2014)	19.5
Structured DNN (Yang et al., 2016)	18.8
Posterior Modeling (Prabhavalkar et al., 2013)	18.5
Kaldi	18.53
CS-Kaldi	18.48
IC-Kaldi	18.46
MA-Kaldi	18.51
DC-Kaldi	18.50
AC-Kaldi	18.41
CTC (Graves et al., 2013)	18.4
RNN Transducer (Graves et al., 2013)	17.7
Attention RNN (Chorowski et al., 2015)	17.6
CTC+SCRF (Lu et al., 2017)	17.4
Segmental RNN (Lu et al., 2016)	17.3
RNNDrop (Moon et al., 2015)	16.9
CNN (Tóth, 2014)	16.7

Table 2. Phone error rate (%) on the TIMIT test set.

of the art performance. We apply AC to the FNN in this recipe. The inputs of the FNN are the FMLLR (Gales, 1998) features of the neighboring 21 frames, which are mean centered and normalized to have unit variance. The number of hidden layers is 4. Each layer has 1024 hidden units. Stochastic gradient descent (SGD) is used to train the network. The learning rate is set to 0.008. We compare with four diversity-promoting regularizers: CS, IC, MA and DeCorrelation (DC) (Cogswell et al., 2015). The regularization parameter in these methods are tuned in $\{10^{-6}, 10^{-5}, \cdots, 10^{5}\}$. The β parameter in IC is set to 1.

Table 2 shows state of the art phone error rate (PER) on the TIMIT core test set. Methods in the first panel are mostly based on FNN, which perform less well than Kaldi. Methods in the third panel are all based on RNNs which in general perform better than FNN since they are able to capture the temporal structure in speech data. In the second panel, we compare AC with other diversity-promoting regularizers. Without regularization, the error is 18.53%. With AC, the error is reduced to 18.41%, which is very close to a strong RNN-based baseline Connectionist Temporal Classification (CTC) (Graves et al., 2013). Besides, AC outperforms other regularizers.

CNN for Image Classification The CIFAR-10 dataset contains 32x32 color images from 10 categories, with 50,000 images for training and 10,000 for testing. We used 5000 training images as the validation set to tune hyperparameters. The data is augmented by first zero-padding the images with 4 pixels on each side, then randomly cropping the padded images to reproduce 32x32 images. We apply AC to wide residual network (WideResNet) (Zagoruyko & Komodakis, 2016) where the depth is set to 28 and the width is set to 10. SGD is used for training, with epoch number 200, initial learning rate 0.1, minibatch size 128, Nesterov momentum 0.9, dropout probability 0.3 and weight decay 0.0005. The learning rate is dropped by 0.2 at 60, 120 and 160 epochs. The performance is the median of 5 runs. We compare with CS, IC, MA, DC and an Orthogonality-Promoting (OP) regularizer (Rodríguez et al., 2016).

Network	Error
Maxout (Goodfellow et al., 2013)	9.38
NiN (Lin et al., 2013)	8.81
DSN (Lee et al., 2015)	7.97
Highway Network (Srivastava et al., 2015)	7.60
All-CNN (Springenberg et al., 2014)	7.25
ResNet (He et al., 2016)	6.61
ELU-Network (Clevert et al., 2015)	6.55
LSUV (Mishkin & Matas, 2015)	5.84
Fract. Max-Pooling (Graham, 2014)	4.50
WideResNet (Huang et al., 2016)	3.89
CS-WideResNet	3.81
IC-WideResNet	3.85
MA-WideResNet	3.68
DC-WideResNet	3.77
OP-WideResNet	3.69
AC-WideResNet	3.63
ResNeXt (Xie et al., 2016b)	3.58
PyramidNet (Huang et al., 2016)	3.48
DenseNet (Huang et al., 2016)	3.46
PyramidSepDrop (Yamada et al., 2016)	3.31

Table 3. Classification error (%) on CIFAR-10 test set

Table 3 shows state of the art classification error on the test set. Compared with the unregularized WideResNet which achieves an error of 3.89%, applying AC reduces the error to 3.63%. AC achieves lower error than other regularizers.

LSTM for Question Answering We apply AC to long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) network, which is a type of RNN. Given the input $\mathbf{x}_t$ at timestamp t, LSTM produces a hidden state $\mathbf{h}_t$ based on the following transition equations:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^{(i)}\mathbf{x}_t + \mathbf{U}^{(i)}\mathbf{h}_{t-1} + \mathbf{b}^{(i)}) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^{(f)}\mathbf{x}_t + \mathbf{U}^{(f)}\mathbf{h}_{t-1} + \mathbf{b}^{(f)}) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^{(o)}\mathbf{x}_t + \mathbf{U}^{(o)}\mathbf{h}_{t-1} + \mathbf{b}^{(o)}) \\
\mathbf{c}_t &= \mathbf{i}_t \odot \tanh(\mathbf{W}^{(c)}\mathbf{x}_t + \mathbf{U}^{(c)}\mathbf{h}_{t-1} + \mathbf{b}^{(c)}) + \mathbf{f}_t \odot \mathbf{c}_{t-1} \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

where $\mathbf{W}$s are $\mathbf{U}$s are gate-specific weight matrices. On the row vectors of each weight matrix, the AC is applied. The LSTM is used for a question answering (QA) task on two datasets: CNN and DailyMail (Hermann et al., 2015),

	CNN		DailyMail	
	Dev	Test	Dev	Test
Hermann et al. (2015)	61.6	63.0	70.5	69.0
Hill et al. (2015)	63.4	6.8	-	-
Kadlec et al. (2016)	68.6	69.5	75.0	73.9
Kobayashi et al. (2016)	71.3	72.9	-	-
Sordoni et al. (2016)	72.6	73.3	-	-
Trischler et al. (2016)	73.4	74.0	-	-
Chen et al. (2016)	73.8	73.6	77.6	76.6
Cui et al. (2016)	73.1	74.4	-	-
Shen et al. (2016)	72.9	74.7	77.6	76.6
BIDAF	76.31	76.94	80.33	79.63
CS-BIDAF	76.43	77.10	80.37	79.71
IC-BIDAF	76.41	77.21	80.49	79.83
MA-BIDAF	76.49	77.09	80.42	79.74
DC-BIDAF	76.35	77.15	80.38	79.67
AC-BIDAF	76.62	77.23	80.65	79.88
Dhingra et al. (2016)	77.9	77.9	81.5	80.9
Dhingra et al. (2017)	79.2	78.6	–	–

Table 4. Accuracy (%) on the two QA datasets

each containing a training, development and test set with 300k/4k/3k and 879k/65k/53k examples respectively. Each example consists of a passage, a question and an answer. The question is a cloze-style task where an entity is replaced by a placeholder and the goal is to infer this missing entity (answer) from all the possible entities appearing in the passage. The LSTM architecture and experimental settings follow the Bidirectional Attention Flow (BIDAF) (Seo et al., 2017) model, which consists of the following layers: character embedding, word embedding, contextual embedding, attention flow, modeling and output. LSTM is applied in the contextual embedding and modeling layer. Character embedding is based on one-dimensional convolutional neural network, where the number of filters is set to 100 and the width of receptive field is set to 5. In LSTM, the size of hidden state is set to 100. Optimization is based on AdaDelta (Zeiler, 2012), where the minibatch size and initial learning rate are set to 48 and 0.5. The model is trained for 8 epochs. Dropout (Srivastava et al., 2014) with probability 0.2 is applied. We compare with four diversity promoting regularizers: CS, IC, MA and DC.

Table 4 shows state of the art accuracy on the two datasets. As can be seen, after applying AC to BIDAF, the accuracy is improved from 76.94% to 77.23% on the CNN test set and from 79.63% to 79.88% on the DailyMail test set. Among the diversity-promoting regularizers, AC achieves the highest accuracy.

5.3. Sensitivity to Parameter τ

In the theoretical analysis presented in Section 4, we have shown that the parameter τ which controls the level of near-

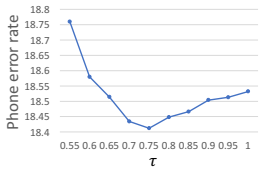

Figure 1. Phone error rate on TIMIT, under varying τ

	TIMIT	CIFAR-10	CNN
No regularization	1.1	6.3	69.4
CS	1.2	6.8	74.8
IC	1.2	6.7	76.1
MA	1.3	7.0	78.6
DC	1.5	7.6	82.9
OP	–	6.8	–
AC	1.3	7.1	79.7

Table 5. Average runtime (hours)

orthogonality (or diversity) incurs a tradeoff between estimation error and approximation error. In this section, we provide an empirical verification, using FNN on TIMIT as a study case. Figure 1 shows how the phone error rates vary on the TIMIT core test set. As can be seen, the lowest test error is achieved under a moderate τ ($= 0.75$). Either a smaller or a larger τ degrades the performance. This empirical observation is aligned with the theoretical analysis that the best generalization performance is achieved under a properly chosen τ. When τ is close to 0, the hidden units are close to orthogonality, which yields much poorer performance. This confirms that the strict-orthogonality constraint proposed by (Le et al., 2010) is too restrictive and is less favorable than a "soft" regularization approach.

5.4. Computational Time

We compare the computational time of neural networks under different regularizers. Table 5 shows the total runtime time of FNNs on TIMIT and CNNs on CIFAR-10 with a single GTX TITAN X GPU, and the runtime of LSTM networks on the CNN dataset with 2 TITAN X GPUs. Compared with no regularization, AC incurs a 18.2% extra time on TIMIT, 12.7% on CIFAR-10 and 14.8% on CNN. The runtime of AC is comparable to that under other diversity-promoting regularizers.

6. Conclusions

In this paper, we propose Angled-Constrained Latent Space Models (AC-LSMs) that aim at promoting diversity among components in LSMs for the sake of alleviating overfitting. Compared with previous diversity-promoting methods, AC has two benefits. First, it is theoretically analyzable: the generalization error analysis shows that larger diversity leads to smaller estimation error and larger approximation error. Second, it is empirically effective, as validated in various experiments.

Acknowledgements

We would like to thank the anonymous reviewers for the suggestions and comments that help to improve this work a lot, and thank Yajie Miao for helping with some of the experiments. P.X and E.X are supported by National Institutes of Health P30DA035778, R01GM114311, National Science Foundation IIS1617583, DARPA FA872105C0003 and Pennsylvania Department of Health BD4BH4100070287.

References

Abdel-Hamid, Ossama, Deng, Li, Yu, Dong, and Jiang, Hui. Deep segmental neural networks for speech recognition. *INTERSPEECH*, 2013.

Affandi, Raja Hafiz, Fox, Emily, and Taskar, Ben. Approximate inference in continuous determinantal processes. In *Advances in Neural Information Processing Systems*, pp. 1430–1438, 2013.

Ba, Jimmy and Caruana, Rich. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.

Bao, Yebo, Jiang, Hui, Dai, Lirong, and Liu, Cong. Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6980–6984. IEEE, 2013.

Barron, Andrew R. Universal approximation bounds for superpositions of a sigmoidal function. *Information Theory, IEEE Transactions on*, 1993.

Bartlett, Peter L and Mendelson, Shahar. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

Blei, David M, Ng, Andrew Y, and Jordan, Michael I. Latent dirichlet allocation. *the Journal of machine Learning research*, 2003.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 2011.

Chen, Danqi, Bolton, Jason, and Manning, Christopher D. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*, 2016.

Chen, Yunpeng, Jin, Xiaojie, Feng, Jiashi, and Yan, Shuicheng. Training group orthogonal neural networks with privileged information. *arXiv preprint arXiv:1701.06772*, 2017.

Chorowski, Jan K, Bahdanau, Dzmitry, Serdyuk, Dmitriy, Cho, Kyunghyun, and Bengio, Yoshua. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pp. 577–585, 2015.

Clevert, Djork-Arné, Unterthiner, Thomas, and Hochreiter, Sepp. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. *ICLR*, 2015.

Cui, Yiming, Chen, Zhipeng, Wei, Si, Wang, Shijin, Liu, Ting, and Hu, Guoping. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.

Dahl, George, Mohamed, Abdel-rahman, Hinton, Geoffrey E, et al. Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in neural information processing systems*, pp. 469–477, 2010.

Dhingra, Bhuwan, Liu, Hanxiao, Cohen, William W, and Salakhutdinov, Ruslan. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*, 2016.

Dhingra, Bhuwan, Yang, Zhilin, Cohen, William W, and Salakhutdinov, Ruslan. Linguistic knowledge as memory for recurrent neural networks. *arXiv preprint arXiv:1703.02620*, 2017.

Gales, Mark JF. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2):75–98, 1998.

Goodfellow, Ian J, Warde-Farley, David, Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Maxout networks. *ICML*, 2013.

Graham, Benjamin. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649. IEEE, 2013.

Griffin, Gregory, Holub, Alex, and Perona, Pietro. Caltech-256 object category dataset. 2007.

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.

Henaff, Mikael, Szlam, Arthur, and LeCun, Yann. Orthogonal rnns and long-memory tasks. *arXiv preprint arXiv:1602.06662*, 2016.

Hermann, Karl Moritz, Kocisky, Tomas, Grefenstette, Edward, Espeholt, Lasse, Kay, Will, Suleyman, Mustafa, and Blunsom, Phil. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1693–1701, 2015.

Hill, Felix, Bordes, Antoine, Chopra, Sumit, and Weston, Jason. The goldilocks principle: Reading children's books with explicit memory representations. *ICLR*, 2015.

Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George E, Mohamed, Abdel-rahman, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 2012.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, Gao, Liu, Zhuang, Weinberger, Kilian Q, and van der Maaten, Laurens. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Kadlec, Rudolf, Schmid, Martin, Bajgar, Ondrej, and Kleindienst, Jan. Text understanding with the attention sum reader network. *ACL*, 2016.

Kobayashi, Sosuke, Tian, Ran, Okazaki, Naoaki, and Inui, Kentaro. Dynamic entity representation with max-pooling improves machine reading. In *Proceedings of NAACL-HLT*, pp. 850–855, 2016.

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

Kulesza, Alex and Taskar, Ben. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 2012.

Lazebnik, Svetlana, Schmid, Cordelia, and Ponce, Jean. Beyond bags of features: Spatial pyramid matching for recognizing scene categories. In *CVPR*, 2006.

Le, Quoc V, Ngiam, Jiquan, Chen, Zhenghao, Chia, Daniel, Koh, Pang Wei, and Ng, Andrew Y. Tiled convolutional neural networks. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pp. 1279–1287. Curran Associates Inc., 2010.

Lee, Chen-Yu, Xie, Saining, Gallagher, Patrick, Zhang, Zhengyou, and Tu, Zhuowen. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pp. 562–570, 2015.

Li, Li-Jia and Fei-Fei, Li. What, where and who? classifying events by scene recognition. In *ICCV*, 2007.

Lin, Min, Chen, Qiang, and Yan, Shuicheng. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Lowe, David G. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

Lu, Liang, Kong, Lingpeng, Dyer, Chris, Smith, Noah A, and Renals, Steve. Segmental recurrent neural networks for end-to-end speech recognition. *arXiv preprint arXiv:1603.00223*, 2016.

Lu, Liang, Kong, Lingpeng, Dyer, Chris, and Smith, Noah A. Multi-task learning with ctc and segmental crf for speech recognition. *arXiv preprint arXiv:1702.06378*, 2017.

Malkin, Jonathan and Bilmes, Jeff. Ratio semi-definite classifiers. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 4113–4116. IEEE, 2008.

Mao, Fengling, Xiong, Wei, Du, Bo, and Zhang, Lefei. Stochastic decorrelation constraint regularized auto-encoder for visual recognition. In *International Conference on Multimedia Modeling*, pp. 368–380. Springer, 2017.

Mishkin, Dmytro and Matas, Jiri. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

Moon, Taesup, Choi, Heeyoul, Lee, Hoshik, and Song, Inchul. Rnndrop: A novel dropout for rnns in asr. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pp. 65–70. IEEE, 2015.

Olshausen, Bruno A and Field, David J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Povey, Daniel, Ghoshal, Arnab, Boulianne, Gilles, Burget, Lukas, Glembek, Ondrej, Goel, Nagendra, Hannemann, Mirko, Motlicek, Petr, Qian, Yanmin, Schwarz, Petr, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

Prabhavalkar, Rohit, Sainath, Tara N, Nahamoo, David, Ramabhadran, Bhuvana, and Kanevsky, Dimitri. An evaluation of posterior modeling techniques for phonetic

recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 7165–7169. IEEE, 2013.

Ramirez, Ignacio, Sprechmann, Pablo, and Sapiro, Guillermo. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3501–3508. IEEE, 2010.

Recht, Benjamin, Fazel, Maryam, and Parrilo, Pablo A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Rodríguez, Pau, Gonzàlez, Jordi, Cucurull, Guillem, Gonfaus, Josep M, and Roca, Xavier. Regularizing cnns with locally constrained decorrelations. *arXiv preprint arXiv:1611.01967*, 2016.

Seo, Minjoon, Kembhavi, Aniruddha, Farhadi, Ali, and Hajishirzi, Hannaneh. Bidirectional attention flow for machine comprehension. *ICLR*, 2017.

Shen, Yelong, Huang, Po-Sen, Gao, Jianfeng, and Chen, Weizhu. Reasonet: Learning to stop reading in machine comprehension. *arXiv preprint arXiv:1609.05284*, 2016.

Sordoni, Alessandro, Bachman, Philip, Trischler, Adam, and Bengio, Yoshua. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*, 2016.

Springenberg, Jost Tobias, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, Martin. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Srivastava, Nitish, Hinton, Geoffrey E, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

Srivastava, Rupesh Kumar, Greff, Klaus, and Schmidhuber, Jürgen. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Tang, Hao, Wang, Weiran, Gimpel, Kevin, and Livescu, Karen. Discriminative segmental cascades for feature-rich phone recognition. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pp. 561–568. IEEE, 2015.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Tóth, László. Phone recognition with deep sparse rectifier neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6985–6989. IEEE, 2013.

Tóth, László. Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 190–194. IEEE, 2014.

Trischler, Adam, Ye, Zheng, Yuan, Xingdi, and Suleman, Kaheer. Natural language comprehension with the epireader. *arXiv preprint arXiv:1606.02270*, 2016.

Vainsencher, Daniel, Mannor, Shie, and Bruckstein, Alfred M. The sample complexity of dictionary learning. *Journal of Machine Learning Research*, 12(Nov):3259–3281, 2011.

Xie, Bo, Liang, Yingyu, and Song, Le. Diversity leads to generalization in neural networks. *AISTATS*, 2017.

Xie, Pengtao. Learning compact and effective distance metrics with diversity regularization. In *ECML*, 2015.

Xie, Pengtao, Deng, Yuntian, and Xing, Eric P. Diversifying restricted boltzmann machine for document modeling. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2015.

Xie, Pengtao, Zhu, Jun, and Xing, Eric. Diversity-promoting bayesian learning of latent variable models. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 59–68, 2016a.

Xie, Saining, Girshick, Ross, Dollár, Piotr, Tu, Zhuowen, and He, Kaiming. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016b.

Xiong, Wei, Du, Bo, Zhang, Lefei, Hu, Ruimin, and Tao, Dacheng. Regularizing deep convolutional neural networks with a structured decorrelation constraint. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 519–528. IEEE, 2016.

Yamada, Yoshihiro, Iwamura, Masakazu, and Kise, Koichi. Deep pyramidal residual networks with separated stochastic depth. *arXiv preprint arXiv:1612.01230*, 2016.

Yang, J, Ragni, Anton, Gales, Mark JF, and Knill, Kate M. Log-linear system combination using structured support vector machines. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 8, pp. 1898–1902, 2016.

Yang, Jianchao, Yu, Kai, Gong, Yihong, and Huang, Thomas. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

Yu, Yang, Li, Yu-Feng, and Zhou, Zhi-Hua. Diversity regularized machine. In *IJCAI*, 2011.

Yuan, Jinhui, Gao, Fei, Ho, Qirong, Dai, Wei, Wei, Jinliang, Zheng, Xun, Xing, Eric Po, Liu, Tie-Yan, and Ma, Wei-Ying. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1351–1361. ACM, 2015.

Zagoruyko, Sergey and Komodakis, Nikos. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zeiler, Matthew D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Zou, James Y. and Adams, Ryan P. Priors for diversity in generative latent variable models. In *NIPS*, 2012a.

Zou, James Y and Adams, Ryan P. Priors for diversity in generative latent variable models. In *Advances in Neural Information Processing Systems*, pp. 2996–3004, 2012b.