ON THE DISTRIBUTION OF PENULTIMATE ACTIVATIONS OF CLASSIFICATION NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper considers probability distributions of penultimate activations in deep classification networks. We first identify a dual relation between the activations and the weights of the final fully connected layer: learning the networks with the cross-entropy loss makes their (normalized) penultimate activations follow a von Mises-Fisher distribution for each class, which is parameterized by the weights of the final fully-connected layer. Through this analysis, we derive a probability density function of penultimate activations per class. This generative model allows us to synthesize activations of classification networks without feeding images forward through them. We also demonstrate through experiments that our generative model of penultimate activations can be applied to real-world applications such as knowledge distillation and class-conditional image generation.

1 Introduction

Deep neural networks have achieved remarkable success in image classification (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; He et al., 2016; Huang et al., 2017; Hu et al., 2018). In most of these networks, an input image is first processed by multiple layers of neurons, whose final output, called *penultimate activations*, is in turn fed to the last fully connected layer that conducts classification; these networks are typically trained in an end-to-end manner by minimizing the *cross-entropy loss*. The penultimate activations are the deepest image representation of the networks and have proven to be useful for various purposes besides classification such as image retrieval (Zhai & Wu, 2019), semantic segmentation (Noh et al., 2015), and general image description of unseen classes (Simonyan & Zisserman, 2014).

This paper studies the nature of penultimate activations in classification networks. We first identify a dual relationship between these activations and the weights of the final classification layer. Specifically, we show that minimizing the cross-entropy loss implicitly performs inference with respect to a specific generative model of the penultimate activations, which is parameterized by the final classification layer. Through this analysis, a probability density function of the penultimate activations of a class is derived; this function can be regarded as an approximate inverse of the final classification layer of the networks, and used to sample penultimate activations of a certain class.

We demonstrate by experiments that our generative model of penultimate activations can be applied to real world applications such as Knowledge Distillation (KD) (Hinton et al., 2015; Ahn et al., 2019) and class-conditional image generation (Kingma et al., 2014; Davidson et al., 2018; Miyato & Koyama, 2018). For KD, our model enables distilling knowledge from a teacher network without feeding images forward through the teacher by generating its activations directly; this new approach to KD is complementary to the standard one (Hinton et al., 2015) and more robust against domain shift. We also show that our model can be naturally integrated with a class-conditional image generation model and enhance the quality of synthesized images.

This paper is organized as follows. We present an analysis of the penultimate activations of a classification network in Section 2, which yields a probabilistic model of these activations. After reviewing our model's relation to previous work in Section 3, we apply our model of penultimate activations to the two real-world applications in Section 4. In Section 5, we conclude the paper with a discussion about limitations and future directions of our approach.

2 Analysis of the Penultimate Activations

In this section, we examine how the penultimate activations are affected by the process of minimizing the cross-entropy loss of a classification network. Our analysis shows that the statistics of the penultimate activations have a very close relation to the weights of the final classification layer, and yields a scheme to approximately invert the classification layer to infer the distribution of penultimate activations only from the layer's weights. We use this scheme in later sections to generate the penultimate activations of a classification network without feeding any data forward.

2.1 Notation

We begin with the notation we will use throughout our analysis of classification networks. Logits $\mathbf{l}$ are computed by the product of penultimate activations $\mathbf{a}$ and the weight matrix $\mathbf{W}$ of the final fully connected layer. Denote the c different d-dimensional columns of $\mathbf{W}$ as $\mathbf{w}_1, \cdots, \mathbf{w}_c \in \mathbb{R}^d$. The computation of the logits is then expressed by

$$\overbrace{\begin{pmatrix} — & \mathbf{w}_1^\top & — \\ — & \mathbf{w}_2^\top & — \\ & \vdots & \\ — & \mathbf{w}_c^\top & — \end{pmatrix}}^{\mathbf{W}^\top} \overbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{pmatrix}}^{\mathbf{a}} = \begin{pmatrix} \mathbf{w}_1^\top \mathbf{a} \\ \mathbf{w}_2^\top \mathbf{a} \\ \vdots \\ \mathbf{w}_c^\top \mathbf{a} \end{pmatrix} = \overbrace{\begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_c \end{pmatrix}}^{\mathbf{l}}. \tag{1}$$

We use y to denote the categorical random variable whose value is determined by the softmax values of l, and i to denote the corresponding ground-truth label. We make the standard assumption that the network is trained by minimizing the cross-entropy between the distributions of y and i.

Let $\bar{\mathbf{a}}, \bar{\mathbf{w}}_1, \cdots, \bar{\mathbf{w}}_c$ be unit vector normalizations of $\mathbf{a}, \mathbf{w}_1, \cdots, \mathbf{w}_c$ respectively:

$$\bar{\mathbf{a}} \triangleq \frac{\mathbf{a}}{\|\mathbf{a}\|}, \quad \bar{\mathbf{w}}_i \triangleq \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}. \tag{2}$$

Note that the normalized vectors $\bar{\mathbf{a}}$ and $\bar{\mathbf{w}}_i$ lie on the unit hypersphere $\mathbb{S}^{d-1}$.

2.2 Cross-Entropy Minimization as Inference

The cross-entropy loss can be written as an expectation over $\mathbf{a}$ and i:

$$-\mathbb{E}_{\mathbf{a},i\sim p(\mathbf{a}|i)p(i)}\left[\log \frac{\exp(\mathbf{w}_i^\top \mathbf{a})}{\sum_j \exp(\mathbf{w}_j^\top \mathbf{a})}\right] = -\mathbb{E}_{\mathbf{a},i\sim p(\mathbf{a}|i)p(i)}\left[\log \frac{\exp(\|\mathbf{a}\|\, \mathbf{w}_i^\top \bar{\mathbf{a}})}{\sum_j \exp(\|\mathbf{a}\|\, \mathbf{w}_j^\top \bar{\mathbf{a}})}\right], \tag{3}$$

where $\|\mathbf{a}\|$ acts as a datapoint-specific temperature term. Note that the joint distribution of $\mathbf{a}$ and i is factorized as $p(\mathbf{a}, i) = p(\mathbf{a}|i)p(i)$ since data depends on labels and activations are determined by data. We assume that directional statistics of $\mathbf{a}$ contain sufficient information for classification by themselves; see Section 2.3 for empirical justification of our assumption. Thus, disregarding the temperature term, Eq. (3) is simplified as

$$-\mathbb{E}_{\mathbf{a},i\sim p(\mathbf{a}|i)p(i)}\left[\log \frac{\exp(\mathbf{w}_i^\top \bar{\mathbf{a}})}{\sum_j \exp(\mathbf{w}_j^\top \bar{\mathbf{a}})}\right] = -\mathbb{E}_{\mathbf{a},i\sim p(\mathbf{a}|i)p(i)}\left[\log \frac{q(\bar{\mathbf{a}}|i)}{\sum_j q(\bar{\mathbf{a}}|j)}\right], \tag{4}$$

where $q(\bar{\mathbf{a}}|i) = \mathrm{vMF}(\bar{\mathbf{a}}; \bar{\mathbf{w}}_i, \|\mathbf{w}_i\|)$ is a von Mises-Fisher (vMF) distribution with mean direction $\bar{\mathbf{w}}_i$ and concentration $\|\mathbf{w}_i\|$. The vMF distribution, an analogue of the Gaussian distribution on the unit hypersphere, is a well-known distribution in directional statistics whose density function is

$$\mathrm{vMF}(\mathbf{x}; \mu, \kappa) = C(\kappa) \exp(\kappa \mu^\top \mathbf{x}), \tag{5}$$

where $\mu \in \mathbb{S}^{d-1}$ is mean direction, $\kappa \in [0, \infty)$ is concentration, and $C(\kappa)$ is a normalizing constant that depends only on κ. In Eq. (4), $\|\mathbf{w}_i\|$ is assumed to be constant for all i since we empirically observed that the effect of $\|\mathbf{w}_i\|$ is marginal in terms of classification performance; verification of this assumption is also given in Section 2.3.

	R-18	R-50	R-101	R-152	D-121	D-201	S-v2	RX-50	RX-101
Original	69.8	76.2	77.4	78.3	74.7	77.2	69.4	77.6	79.3
Normalized	67.1	74.7	76.2	77.5	72.5	75.5	68.4	76.9	78.9
Drop rate	-3.9%	-1.9%	-1.5%	-1.1%	-2.9%	-2.2%	-1.5%	-0.9%	-0.6%

Table 1: Performance of various classification networks before and after normalizing $\mathbf{a}$ and $\mathbf{w}_i$ in top-1 accuracy on the ImageNet validation set. R: ResNet (He et al., 2016), D: DenseNet (Huang et al., 2017), S: ShuffleNet (Ma et al., 2018), RX: ResNeXt (Xie et al., 2017).

Note that Eq. (4) is the expected negative log posterior probability of i assuming the generative model of normalized activations $q(\bar{\mathbf{a}}, i) = p(i)q(\bar{\mathbf{a}}|i)$ with uniform prior $p(i)$. Therefore, **minimizing cross-entropy loss can be seen as posterior inference with respect to our specific generative model** $q(\bar{\mathbf{a}}|i)$ for directional statistics of penultimate activations.

Minimizing the cross-entropy loss in Eq. (4) aims to maximize $q(\bar{\mathbf{a}}|i)$ for the ground-truth class i while minimizing all other $q(\bar{\mathbf{a}}|j)$. The maximization of $q(\bar{\mathbf{a}}|i)$ is expanded into

$$\min -\mathbb{E}_{\mathbf{a},i}\left[\log q(\bar{\mathbf{a}}|i)\right] = \min \mathbb{E}_{\mathbf{a},i}\left[D_{KL}(p(\bar{\mathbf{a}}|i) \,||\, q(\bar{\mathbf{a}}|i)) + H(p(\bar{\mathbf{a}}|i))\right], \tag{6}$$

where $D_{KL}(\cdot \,||\, \cdot)$ is the Kullback-Leibler (KL) divergence. Therefore, the cross-entropy loss directly encourages normalized penultimate activations $\bar{\mathbf{a}}$ for images of class i to follow our generative model $q(\bar{\mathbf{a}}|i)$ by minimizing the KL divergence between $p(\bar{\mathbf{a}}|i)$ and $q(\bar{\mathbf{a}}|i)$. We additionally have an entropy term $H(p(\bar{\mathbf{a}}|i))$ which encourages $\bar{\mathbf{a}}$ to be more concentrated, but it does not cause $p(\bar{\mathbf{a}}|i)$ to collapse to a point mass because of the KL divergence term.

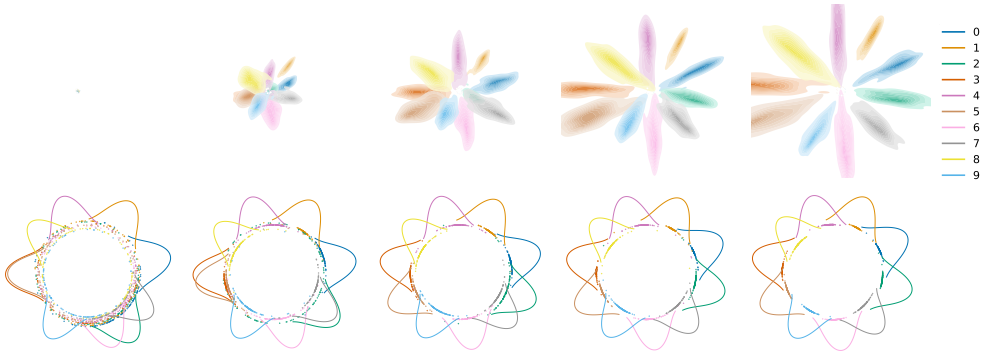
2.3 Empirical Verification

We provide empirical support for our analysis in Section 2.2 by verifying

- Our core assumption that directional statistics of penultimate activations contain sufficient information for classification by themselves,
- The conclusion of Eq. (4) and Eq. (6) that, for a trained network, $p(\bar{\mathbf{a}}|i)$ follows a von Mises-Fisher distribution for all i.

Directional statistics are sufficient for classification. When deriving the generative model of $\bar{\mathbf{a}}$ in a form of vMF distribution, the magnitudes of $\mathbf{a}$ and $\mathbf{w}_i$ are ignored as we assume that their directional statistics are already sufficient for classification. To verify our assumption, we examine how much the performance of pretrained classification networks drop by normalizing their $\mathbf{a}$ and $\mathbf{w}_i$. Specifically, we choose 9 networks pretrained for the ImageNet classification task (Russakovsky et al., 2015), and apply them to the ImageNet validation set to measure their performance. As summarized in Table 1, the performance drop by the normalization is marginal, especially when the network is deeper and more powerful. This result supports our assumption that directional statistics of $\mathbf{a}$ is sufficient for classification and the effect of $\|\mathbf{w}_i\|$ is marginal.

Penultimate activations follow vMF distributions. For qualitative verification of our assertion, we visualize in Figure 1 penultimate activations of a classification network trained on the MNIST dataset (LeCun & Cortes, 2010) and the vMF distributions derived from its final classification layer. The network consists of 4 convolution layers followed by the final fully connected layer, and is designed to produce 2-dimensional penultimate activations for ease of visualization. As shown in Figure 1, normalized penultimate activations are not well aligned with vMF distributions in early stages of training, but become grouped for each class and following the corresponding distributions tightly as training progresses. Additionally, from the density of original penultimate activations in Figure 1, one can observe that their directions are sufficient for classification as demonstrated also in the previous paragraph.



(a) Epoch 1 (b) Epoch 2 (c) Epoch 4 (d) Epoch 8 (e) Epoch 30

Figure 1: Visualization of penultimate activations and vMF distributions derived from the classification layer on the MNIST dataset. (**top**) Kernel density estimates of $\mathbf{a}$. (**bottom**) Distributions of $\bar{\mathbf{a}}$ represented by dots and vMF distributions $q(\bar{\mathbf{a}}|i)$ drawn by solid lines.

3 Related Work

Understanding deep neural networks. Understanding what a neural network learns about data is a fundamental problem in deep learning. Previous approaches have analyzed classification networks by optimizing an image to maximally activate a specific neuron (Erhan et al., 2009; Yosinski et al., 2015) or to maximize the predicted probability of a specific class (Simonyan et al., 2013). Mahendran & Vedaldi (2015) uses a similar technique to visualize the entire feature map of an image. Instead of generating representative images, Zhou et al. (2016); Selvaraju et al. (2017) locates the most salient region within an image for each class by computing a weighted average of each activation channel. While such techniques offer high-level insights into the characteristics of learned activations, there is no obvious way to use these insights to facilitate the learning of other models. In contrast, our generative model allows us to sample activations from, and quantify the relationship between, different classes. Our experiments demonstrate that these features of our model can be applied to real-world problems such as knowledge distillation and accelerating the training of a generative model. Besides the tasks we considered in this paper, our approach of modeling the behavior of a trained network has many other potential applications such as domain adaptation, anomaly detection, and uncertainty calibration.

vMF distributions in deep learning. The von Mises-Fisher (vMF) distribution is a common component of models of directional data. Mixtures of vMFs have been studied for the task of clustering such directional data (Banerjee et al., 2005; Gopal & Yang, 2014). For Bayesian inference of neural network weights, Oh et al. (2019) consider a decomposition of model weights into radial and directional components, and use vMF distributions to model the directional component. Hasnat et al. (2017) learn a vMF embedding space for deep metric learning. Such hyperspherical embedding spaces have the desirable property that the total surface area decreases as dimension increases beyond 8 (see Figure 2 of Hasnat et al. (2017)), unlike Euclidean embedding spaces for which volume increases exponentially with dimension. Kumar & Tsvetkov (2019) parameterize word embeddings as vectors on a unit hypersphere and uses the negative log likelihood of a vMF distribution as an objective, reducing the large computations involved in normalizing the softmax in sequence-to-sequence models. Our use of directional statistics differs from these previous methods in that we use it as a tool for explaining the behavior of standard classification models rather than for specialized purposes such as constructing a compact embedding space or reducing computation.

4 Applications

In this section, we demonstrate that our generative model of penultimate activations can be applied to two practical applications, knowledge distillation (Hinton et al., 2015; Ahn et al., 2019; Romero et al., 2014) and class-conditional image generation (Davidson et al., 2018).

4.1 Class-wise Knowledge Distillation

4.1.1 Algorithm Details

Our generative model is first applied to Knowledge Distillation (KD), the task of distilling knowledge from a teacher network T to a student network S (Hinton et al., 2015). Unlike most of the existing approaches, our model enables KD without feeding images forward through T by directly generating activations of a certain class. Specifically, our model is used to approximate the average prediction of T per class, which is represented as the probability of T's prediction y given class i and estimated by

$$p_T(y|i) = \int p_T(\bar{\mathbf{a}}|i)p_T(y|\bar{\mathbf{a}})\, d\bar{\mathbf{a}} \approx \frac{1}{N}\sum_{j=1}^{N} p_T(y|\bar{\mathbf{a}}_j), \tag{7}$$

where we employ Monte Carlo integration since the exact integral is intractable. Also, each $\bar{\mathbf{a}}_j$ is an i.i.d. sample from vMF($\bar{\mathbf{w}}_i, \kappa$), where κ is set to 85 for all experiments on KD.

The estimated $p_T(y|i)$ in Eq. (7) quantifies the relationship between two classes y and i that is captured by T, and is employed as a target for KD in our approach. Recall that the standard KD loss (Hinton et al., 2015) is

$$\mathcal{L}_{\text{KD}} = -\mathbb{E}_{i,\mathbf{x},\mathbf{y}\sim p(i,\mathbf{x})p_T(y|\mathbf{x})}\left[\log p_S(y|\mathbf{x})\right], \tag{8}$$

where y denotes prediction and $\mathbf{x}$ and i are data and label, respectively. The loss in Eq. (8) is designed to minimize the KL divergence between $p_T(y|\mathbf{x})$ and $p_S(y|\mathbf{x})$ for each data $\mathbf{x}$. Unlike this data-wise KD, our approach is a Class-wise KD (CKD) whose objective is

$$\mathcal{L}_{\text{CKD}} = -\mathbb{E}_{i,\mathbf{x},y\sim p(i,\mathbf{x})p_T(y|i)}\left[\log p_S(y|\mathbf{x})\right], \tag{9}$$

where the categorical distribution $p_T(y|i)$ is given by Eq. (7). Note again that while the standard KD objective in Eq. (8) requires a forward pass through the teacher network T to compute $p_T(y|\mathbf{x})$, ours in Eq. (9) utilizes the pre-computed distribution $p_T(y|i)$ without exploiting T during training of S. This property of CKD is useful especially when it is hard to conduct forward propagation through T (*e.g.*, online learning of S with limited memory and computation power) or if there is domain shift between training datasets for T and S as demonstrated by experiments in Section 4.1.4.

The overall procedures of the standard KD and our CKD are described below in Algorithm 1 and 2, respectively, where the main differences between the two methods are colored in red.

Algorithm 1 KD (Hinton et al., 2015)

Require: teacher network $\mathbf{x} \mapsto p_T(y|\mathbf{x})$
Require: student network $\mathbf{x} \mapsto p_S(y|\mathbf{x})$
while not converged **do**
  $\mathbf{x}, i \sim p(\mathbf{x}, i)$
  $p_T \Leftarrow p_T(y|\mathbf{x})$
  $p_S \Leftarrow p_S(y|\mathbf{x})$
  $\mathcal{L}_{\text{KD}} \Leftarrow -p_T \cdot \log p_S$
end while

Algorithm 2 Class-wise KD (ours)

Require: teacher network $\mathbf{x} \mapsto p_T(y|\mathbf{x})$
Require: student network $\mathbf{x} \mapsto p_S(y|\mathbf{x})$
$p_T(y|i) \Leftarrow \frac{1}{N}\sum_{j=1}^{N} p_T(y|\bar{\mathbf{a}}_j)$
while not converged **do**
  $\mathbf{x}, i \sim p(\mathbf{x}, i)$
  $p_T \Leftarrow p_T(y|i)$
  $p_S \Leftarrow p_S(y|\mathbf{x})$
  $\mathcal{L}_{\text{CKD}} \Leftarrow -p_T \cdot \log p_S$
end while

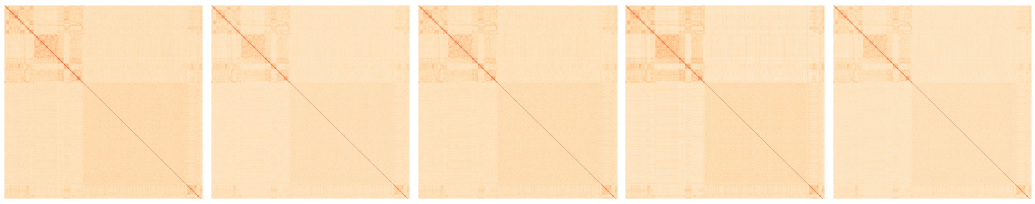
4.1.2 Qualitative Analysis on the Effect of CKD

To investigate which kind of information of T is transfered to S through CKD, we qualitatively examine penultimate activations and their generative models of the two networks on the MNIST dataset. In this experiment, T consists of 4 convolution layers followed by the final fully connected layer, and produces 2-dimensional penultimate activations. S has the same architecture but with half the number of convolution kernels. From the visualization results in Figure 2, one can observe that T and S have the same cyclic order of classes in the space of their penultimate activations. This demonstrates that Eq. (9) encourages S to follow the inter-class relationships captured by T.

Note also that the activation distributions of S are substantially more concentrated compared to those of T. This is similar with the observation of Müller et al. (2019) that embedding vectors of neural

(a) Teacher network (b) Student network trained with CKD

Figure 2: Visualization of penultimate activations and their vMF distributions on the MNIST dataset. For both of the teacher and student, (**left**) kernel density estimates of $\mathbf{a}$, (**right**) distributions of $\bar{\mathbf{a}}$ represented by dots and vMF distributions $q(\bar{\mathbf{a}}|i)$ drawn by solid lines.



(a) ResNet-18 (b) ResNet-101 (c) DenseNet-121 (d) VGG-16 (e) ResNeXt-50

Figure 3: Visualization of $\log p_T(j|i)$ of all pairs of classes calculated as Eq. (7) using 5 different networks pretrained on the ImageNet dataset. Best viewed zoomed in.

networks are more tightly clustered when they are trained with label smoothing. However, instead of the uniform label smoothing of Müller et al. (2019), our CKD objective in Eq. (9) smoothes labels by considering the inter-class relationships in T.

To further investigate whether our CKD objective in Eq. (9) can capture inter-class relations learned by large-scale neural networks, we compare those relations extracted from various Imagenet pretrained networks. Specifically, we compute $\log p_T(j|i)$ as in Eq. (7) for every pair of classes using 5 different ImageNet pretrained networks. The results are represented as 1000×1000 matrices as shown in Figure 3, where one can find that all the 5 matrices exhibit similar inter-class relations although their corresponding network architectures vary widely. This implies that the inter-class relations extracted from Eq. (7) and Eq. (9) reflect the similarity among the groups of data belonging to different classes, to which we attribute the capability of CKD to transfer knowledge from T to S.

4.1.3 Network Compression and Self Distillation

The effectiveness of CKD is first evaluated on the CIFAR-100 dataset (Krizhevsky et al., 2009) in the scenario of network compression. Following the protocol of Ahn et al. (2019), we use WRN-40-2 as T and WRN-16-2 as S, both presented in Zagoruyko & Komodakis (2016). As summarized in Table 2, CKD outperforms the baseline "Label" that does not utilize T, and is as competitive as previous distillation techniques like "FitNet" (Romero et al., 2014) and "VID-I" (Ahn et al., 2019). This result demonstrates that CKD is capable of extracting useful knowledge from T. On the other hand, the performance of CKD is worse than that of the standard "KD" (Hinton et al., 2015) since the data-wise approach can extract a larger amount of knowledge than CKD by feeding individual datapoints forward through T. However, CKD and the standard KD are complementary to each other and the performance is further enhanced by integrating them. We further demonstrate the efficacy of CKD in the self distillation scenario where S has the same architecture with T (*i.e.*, WRN-40-2). In this setting, the same tendency has been observed.

4.1.4 KD in the Presence of Domain Shift

Most KD techniques assume that T and S are trained with the same dataset or, at least, on the same domain. However, this assumption does not hold always in real world settings, *e.g.*, when the dataset used to train T is not available due to privacy issues or S is trained with streaming data that can be

	Label	Label†	KD	KD†	FitNet†	VID-I†	CKD	CKD+KD
WRN-16-2	73.3	70.4	74.1	72.9	70.9	73.3	73.7	74.0
WRN-40-2	76.2	74.2	77.7	75.8	74.3	75.3	76.6	77.7

Table 2: Top-1 test accuracy of the student networks on the CIFAR-100 dataset when using WRN-40-2 as the teacher. The scores of the approaches with † are taken directly from Ahn et al. (2019).

	Photometric Transform				Downsampling		
	0.2	0.4	0.6	0.8	$\times 0.75$	$\times 0.5$	$\times 0.25$
Label	73.53	73.37	72.01	71.24	69.46	63.29	49.69
KD	**74.05**	73.05	69.96	66.14	65.63	57.53	44.96
CKD (ours)	73.86	**73.92**	**73.55**	**71.86**	**70.32**	**63.53**	**50.02**

Table 3: Top-1 test accuracy of the student networks on the CIFAR-100 dataset with various degrees of photometric transform and image downsampling.

corrupted by sensor noises. In those cases, the quality of knowledge extracted from T in a data-wise manner could be degraded since T assumes a different data distribution from what S observes.

We argue that our CKD is more robust against domain shift since it can perform KD without taking input data explicitly. To validate the advantage of CKD, it is evaluated and compared to the standard KD (Hinton et al., 2015) on the CIFAR-100 dataset (Krizhevsky et al., 2009) while simulating domain shift. Specificlly, we consider two different types of domain shift: photometric transform and downsampling. For the photometric transform, we randomly alter brightness, contrast, and saturation of input image with 5 different degrees of alteration; degree 0 means no alteration. Also, for the image downsampling, we reduce the resolution of input image with 3 different rates ($\times 0.75, \times 0.5, \times 0.25$) using nearest neighbor interpolation. Furthermore, as in the setting of Section 4.1.3, we employ WRN-40-2 as T and WRN-16-2 as S.

In this experiment, CKD consistently enhances the performance of the baseline using only ground truth labels ("Label") while the standard KD ("KD") even deteriorates when the domain shift is significant, as summarized in Table 3. We believe this result is mainly due to the fact that the standard KD strongly depends on the data distribution. On the other hand, the knowledge captured by CKD can still be useful in the presence of domain shift since it extracts inter-class relationships directly from the weights of the final classification layer rather than relying on the data.

4.2 Class-Conditional Image Generation

We apply our generative model of penultimate activations to class-conditional image generation by modifying the Hyperspherical Variational Auto Encoder (HVAE) of Davidson et al. (2018), a latent variable model with a hyperspherical latent space. This section first describes our approach along with two baselines, then demonstrates its efficacy by experiments on the MNIST dataset.

4.2.1 Baselines and Our Approach

Baseline 1 – Hyperspherical VAE (HVAE): HVAE is a latent variable model, which first maps the given data $\mathbf{x}$ to a latent variable $\mathbf{z} \in \mathbb{S}^d$ with a prior $p(\mathbf{z})$ by an encoder $q(\mathbf{z}|\mathbf{x})$, then reconstructs $\mathbf{x}$ from $\mathbf{z}$ by a stochastic decoder $p(\mathbf{x}|\mathbf{z})$. Especially, HVAE assumes that $p(\mathbf{z})$ is a uniform distribution on the unit hypersphere $\mathbb{S}^d$. Specifically, the encoder of HVAE is trained by maximizing the following lower bound of the evidence:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})). \tag{10}$$

Baseline 2 – HVAE Conditioned by Concatenation (HVAE-C): For HVAE, a straightforward way to model the distribution of $\mathbf{x}$ while taking label i into account is to attach i to the end of

	Log-Likelihood				ELBO			
$\mathbf{z}$ dim	3	5	10	20	3	5	10	20
HVAE	−135.0	−115.3	−97.7	−95.0	−138.3	−120.3	−105.5	−106.3
HVAE-C	−139.4	−119.6	−98.4	−94.7	−141.7	−123.5	−105.5	−105.5
HVAE-L (ours)	**−133.0**	**−114.3**	**−95.6**	**−92.8**	**−136.2**	**−119.1**	**−103.6**	**−104.0**

Table 4: Comparison on the MNIST generative modeling task.

the latent vector $\mathbf{z}$. Specifically, whereas HVAE assumes that $\mathbf{x}$ is generated from $\mathbf{z}$ alone (*i.e.*, $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$), HVAE-C assumes that $\mathbf{x}$ is generated from both $\mathbf{z}$ and i (*i.e.*, $p(\mathbf{x}, \mathbf{z}, i) = p(i)p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, i)$). We train HVAE-C by maximizing the following lower bound of the evidence considering i:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x},i)}\left[\log p(\mathbf{x}|i, \mathbf{z}) + \log p(i)\right] - D_{KL}(q(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})). \tag{11}$$

Ours – HVAE Conditioned by Learned Prior (HVAE-L): Recall from Section 2.2 that we can utilize the weights of the final fully-connected layer of a classifier network to model a distribution of penultimate activations per class i. We employ this activation distribution conditioned on i as a learned prior for $\mathbf{z}$ of HVAE. This approach is similar to HVAE-C in that i is involved in the process of generating $\mathbf{x}$, but the two models differ in where i is integrated; in our HVAE-L, $\mathbf{x}$ is generated from $\mathbf{z}$ alone, where the distribution of $\mathbf{z}$ is determined by i (*i.e.*, $p(\mathbf{x}, \mathbf{z}, i) = p(i)p(\mathbf{z}|i)p(\mathbf{x}|\mathbf{z})$). HVAE-L is trained by optimizing the following objective:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x},i)}\left[\log p(\mathbf{x}|\mathbf{z}) + \log p(i)\right] - D_{KL}(q(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z}|i)). \tag{12}$$

Also, the above objective differs from that of HVAE in Eq. (10) since the two models assume different generation procedures.

4.2.2 Experiments

Our model (HVAE-L) is compared against the two baseline models (HVAE, HVAE-C) on the MNIST generative modeling task. The experimental setup including network architecture and hyperparameter setting follows directly that of Davidson et al. (2018). In addition, we ensure that the dimensionality of the latent vector $\mathbf{z}$ is the same for all the models. Also, our prior distribution for HVAE-L is learned by a classification network with the same architecture as the encoder, and its hyperparameter κ is set to 5. Quantitative results of the three models are summarized in Table 4, where our HVAE-L outperforms both of the baselines, demonstrating that our model of class-conditional activations is a useful prior for class-conditional image generation.

5 Discussion and Future Work

Our analysis focuses on standard classification networks that are trained to convergence, and our modelling assumptions do not necessarily hold in settings that don't satisfy those conditions. An obvious counterexample is a randomly initialized network: the classifier holds no information about the dataset at initialization. As another example, Hoffer et al. (2018) uses a fixed projection matrix as the final layer with no impairment on accuracy at convergence. The final layer of such a network obviously cannot reflect class relations since the weights were determined independently of data.

Next on our agenda includes (1) more precise modeling of penultimate activations of classification network than vMF distributions, and (2) investigating and implementing more applications of our activation generation model such as domain adaptation, data augmentation, and uncertainty calibration. We also plan to develop a new architecture for image classification networks based on the observations in this work.

References

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9163–9171, 2019.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.*, 6:1345–1382, December 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1088718.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *International Conference on Machine Learning*, pp. 154–162, 2014.

Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv: 1503.02531.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. *arXiv:1801.04540 [cs, stat]*, January 2018. URL http://arxiv.org/abs/1801.04540. arXiv: 1801.04540.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012.

Sachin Kumar and Yulia Tsvetkov. Von mises-fisher loss for training sequence to sequence models with continuous outputs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJlDnoA5Y7.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Proc. European Conference on Computer Vision (ECCV)*, pp. 122–138, 2018. ISBN 978-3-030-01264-9.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1528, 2015.

Changyong Oh, Kamil Adamczewski, and Mijung Park. Radial and directional posteriors for bayesian neural networks. *arXiv preprint arXiv:1902.02603*, 2019.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. *arXiv:1412.6550 [cs]*, December 2014. URL `http://arxiv.org/abs/1412.6550`. arXiv: 1412.6550.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

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

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, July 2017.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

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

Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *Proc. British Machine Vision Conference (BMVC)*, 2019.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.