Efficient Maximal Coding Rate Reduction by Variational Forms

-Summary-

Introduction

- Arising new objective for classification : MCR^2(Maximizing the coding rate reduction)

- Problem of CE :
  - Learning based on CE will leads to **neural collapse** : when CE → 0, the representations of each class at penultimate layer collapse to a single point, suppressing within-class variability. (Common symptom on penultimate layer)

- To alleviate this phenomenon, [Yu et al., 2020] suggested $\boldsymbol{MCR^2}$ **objective** that encourages the latent representation of the entire training set to occupy as much volume as possible, while enforcing each class to occupy as little space as possible.
- Empirically, and theoretically, it was shown that the latent representations will be a low dimensional linear subspace with the subspace orthogonal to each other.

Introduction

- But, calculating MCR^2 is intractable as the # of class k increases (due to $\log det$). This paper tries to resolve this problem by suggesting variational form of MCR^2.

- Original MCR^2 objective :

$$\max_{\theta} \Delta R(\boldsymbol{Z}_\theta) \equiv R(\boldsymbol{Z}_\theta) - R_c(\boldsymbol{Z}_\theta, \boldsymbol{\Pi}) \quad \text{s.t. } \boldsymbol{Z}_\theta \in \mathcal{S}, \quad \text{where}$$

$$R(\boldsymbol{Z}_\theta) = \frac{1}{2} \log\det\left(\boldsymbol{I} + \alpha \boldsymbol{Z}_\theta \boldsymbol{Z}_\theta^\top\right), \quad \text{and}$$

$$R_c(\boldsymbol{Z}_\theta, \boldsymbol{\Pi}) = \sum_{j=1}^{k} \frac{\gamma_j}{2} \log\det\left(\boldsymbol{I} + \alpha_j \boldsymbol{Z}_\theta \mathrm{Diag}(\boldsymbol{\Pi}_j) \boldsymbol{Z}_\theta^\top\right)$$

where $Z = [f_\theta(X_1), \dots f_\theta(X_m)] \in \mathbb{R}^{d\times m}$, $X = [X_1, \dots X_m] \in \mathbb{R}^{D\times m}$
and $\Pi \in \mathbb{R}^{m\times k}$ = class membership matrix
[$\Pi_{i,j} = p(X_i \text{ is in class } j)$]

$k = \#\ of\ classes$
$m = \#\ of\ samples$
$X_i = i\ th\ sample$
$d = feature\ dimension$
$D = input\ dimension$
$\Pi = membership\ martix$
$\Pi_j = jth\ column\ of\ \Pi$

Description of some terms :

1. $\alpha = \frac{d}{m\epsilon^2}$
2. $\alpha_j = \frac{d}{<1,\Pi_j>\epsilon^2}$
3. $\gamma_j = \frac{<1,\Pi_j>}{m}$

Introduction

- Original MCR^2 objective :

$$\max_{\theta} \Delta R(\boldsymbol{Z}_\theta) \equiv R(\boldsymbol{Z}_\theta) - R_c(\boldsymbol{Z}_\theta, \boldsymbol{\Pi}) \quad \text{s.t. } \boldsymbol{Z}_\theta \in \mathcal{S}, \quad \text{where}$$

$$R(\boldsymbol{Z}_\theta) = \frac{1}{2} \log\det\left(\boldsymbol{I} + \alpha \boldsymbol{Z}_\theta \boldsymbol{Z}_\theta^\top\right), \quad \text{and}$$

$$R_c(\boldsymbol{Z}_\theta, \boldsymbol{\Pi}) = \sum_{j=1}^{k} \frac{\gamma_j}{2} \log\det\left(\boldsymbol{I} + \alpha_j \boldsymbol{Z}_\theta \mathrm{Diag}(\boldsymbol{\Pi}_j) \boldsymbol{Z}_\theta^\top\right)$$

- $R(Z_\theta)$ (expansion term) : <u>captures the dimension (or volume) of the space spanned by Z_θ</u>
(≅ required bits to encode Z_θ with assumption of MN
- $R_c(Z_\theta, \Pi)$ (compression term) : <u>measures the sum of the dimensi-ons (or volumes) of the data from each class</u>
(≅ required bits to encode Z_θ with assumption of mixture (~Π) of MN)

$\boldsymbol{k = \#\ of\ classes}$
$\boldsymbol{m = \#\ of\ samples}$
$\boldsymbol{X_i = i\ th\ sample}$
$\boldsymbol{d = feature\ dimension}$
$\boldsymbol{D = input\ dimension}$
$\boldsymbol{\Pi = membership\ martix}$
$\boldsymbol{\Pi_j = jth\ column\ of\ \Pi}$

Description of some terms :

1. $\alpha = \frac{d}{m\epsilon^2}$
2. $\alpha_j = \frac{d}{<1,\Pi_j>\epsilon^2}$
3. $\gamma_j = \frac{<1,\Pi_j>}{m}$

Introduction

- What does Original MCR^2 objective do?
  - **<u>While maximizing the overall volume of the embedded features (1st term), we want to compress the volume of embedded features from each class (2nd term)</u>**
  - (In other perspective, it can be seen as minimizing the # of bits to encode Z when Π is known while maximizing the # of bits to encode Z when we don't have mixture assumption)

- But, calculating $R_c(Z_\theta, \Pi)$ involves k computations of $\log det$, which becomes intractable as # of classes $\uparrow$.
- They cleverly modify their terms into variational form to avoid this problem.

Variational formulation of MCR^2

- Observe following fact and theorems :

1. Identity equation for PSD matrix M and $c \geq 0$:

$$\log\det(\boldsymbol{I} + c\boldsymbol{M}) = \sum_{i=1}^{r} \log(1 + c\sigma_i(\boldsymbol{M}))$$

2. Variational form of spectral functions :

Theorem 2.1 (Adapted from [16]) *For any matrix $\boldsymbol{X}$, let r denote the rank of $\boldsymbol{X}$, let $\sigma_i(X)$ denote the i^{th} singular value of $\boldsymbol{X}$, and define*

$$H(\boldsymbol{X}) = \sum_{i=1}^{r} h(\sigma_i(\boldsymbol{X})).$$

for some function h. If h is a concave, non-decreasing function on $[0, \infty)$ with $h(0) = 0$, then the following holds

$$H(\boldsymbol{X}) = \min_{\boldsymbol{U},\boldsymbol{V}:\boldsymbol{U}\boldsymbol{V}^\top=\boldsymbol{X}} \sum_i h(\|\boldsymbol{U}_i\|_2 \|\boldsymbol{V}_i\|_2),$$

where $(\boldsymbol{U}_i, \boldsymbol{V}_i)$ denotes the i^{th} columns of $(\boldsymbol{U}, \boldsymbol{V})$. Note also that $(\boldsymbol{U}, \boldsymbol{V})$ can have an arbitrary number of columns $(\geq r)$ provided $\boldsymbol{U}\boldsymbol{V}^\top = \boldsymbol{X}$.

$\Rightarrow$

Proposition 3.1 *Let $\boldsymbol{M}$ be any real positive semi-definite matrix and let $c \geq 0$ be any non-negative scalar. Then the following holds:*

$$-\log\det(\boldsymbol{I} + c\boldsymbol{M}) = \max_{\boldsymbol{U}:\,\boldsymbol{U}\boldsymbol{U}^\top=\boldsymbol{M}} -\sum_i \log\left(1 + c\|\boldsymbol{U}_i\|_2^2\right). \tag{3}$$

Further, if $\bar{\boldsymbol{U}}\boldsymbol{S}\bar{\boldsymbol{U}}^\top = \boldsymbol{M}$ is a SVD of $\boldsymbol{M}$ then $\boldsymbol{U}^* = \bar{\boldsymbol{U}}\boldsymbol{S}^{1/2}$ is a solution to the above problem.

Variational formulation of MCR^2

- Using proposition 3.1, we can optimization variable $\{U^{(j)}\}_{j=1}^{k}$ to replace $Z_\theta Diag(\Pi_j)Z_\theta^T$:

$$\max_{\theta} \Delta R(\boldsymbol{Z}_\theta) =$$
$$\max_{\theta,\{\boldsymbol{U}^{(j)}\}_{j=1}^{k}} \frac{1}{2}\log\det\left(\boldsymbol{I} + \alpha\sum_{j=1}^{k}\boldsymbol{U}^{(j)}(\boldsymbol{U}^{(j)})^\top\right)$$
$$-\sum_{j=1}^{k}\frac{\gamma_j}{2}\sum_{i}\log\left(1+\alpha_j\left\|\boldsymbol{U}_i^{(j)}\right\|_2^2\right)$$
$$\text{s.t. } \forall j, \boldsymbol{U}^{(j)}(\boldsymbol{U}^{(j)})^\top = \boldsymbol{Z}_\theta \mathrm{Diag}(\boldsymbol{\Pi}_j)\boldsymbol{Z}_\theta^\top \text{ and } \boldsymbol{Z}_\theta \in \mathcal{S}.$$

Note :

q is larger than k (usually $q \cong 2k$)

Notation :

$\mathcal{S}$ = matrix sets where their columns are l_2 normalized.

where $U^{(j)} = \Gamma Diag(A_j)^{\frac{1}{2}}$, $\Gamma \in \mathbb{R}^{d\times q} \cap \mathcal{S}$ (dictionary with unit l_2 normalized columns), $A_j \in \mathbb{R}_+^q$ (non-negative encoding vector)

- Due to this reparameterization, $\Gamma Diag(A_j)\Gamma^T = U^{(j)}U^{(j)^T}$ and $\left\|U_i^{(j)}\right\|_2^2 = A_{i,j}$ holds

Variational formulation of MCR^2

- In addition to this reparameterization, they add regularization term for this reparameterization :

$$M(Z_\theta, \Gamma, A) = \sum_{j=1}^{k} \frac{1}{\gamma_j} \left\| Z_\theta Diag(\Pi_j) Z_\theta^T - \Gamma Diag(A_j) \Gamma^T \right\|_F^2$$

- Their final proposed formulation ($V - MCR^2$) is following :

$$\max_{\theta, \Gamma \in \mathbb{R}^{d \times q} \cap \mathcal{S}, \boldsymbol{A} \in \mathbb{R}_+^{q \times k}} R^v(\Gamma, \boldsymbol{A}) - R_c^v(\boldsymbol{A}) - \frac{\mu}{2m} M(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A})$$

$$\text{where} \quad R^v(\Gamma, \boldsymbol{A}) = \frac{1}{2} \log\det \left(\boldsymbol{I} + \alpha \sum_{j=1}^{k} \Gamma \mathrm{Diag}(\boldsymbol{A}_j) \Gamma^\top \right),$$

$$R_c^v(\boldsymbol{A}) = \sum_{j=1}^{k} \frac{\gamma_j}{2} \sum_{l=1}^{q} \log\left(1 + \alpha_j \boldsymbol{A}_{l,j}\right),$$

$$M(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A}) = \sum_{j=1}^{k} \frac{1}{\gamma_j} \left\| \boldsymbol{Z}_\theta \mathrm{Diag}(\boldsymbol{\Pi}_j) \boldsymbol{Z}_\theta^\top - \Gamma \mathrm{Diag}(\boldsymbol{A}_j) \Gamma^\top \right\|_F^2$$

Note :

The penalization term M can be interpreted a class-balanced low-rank LASSO

$\because$ it imposes $\Gamma Diag(A_j)\Gamma^T \rightarrow Z_\theta Diag(\Pi_j) Z_\theta^T$ where $\Gamma \in \mathbb{R}^{d \times q}, A_j \in \mathbb{R}_+^q, Z_\theta \in \mathbb{R}^{d \times k}, \Pi_j \in \mathbb{R}_+^k$ and $q > k$.

Variational formulation of MCR^2

- Now, the complexity of calculating R_C term is changed from $O\left(k \min\{d^3, m^3\}\right)$ to $O(qk)$.

- *Note : complexity for calculating M is $O(kq^2)$, which is also tractable compared to original R_C terms.*

- Then, how to optimize this terms?
  - 1st step : optimize Γ, A by stable GA (using Lipschitz upper-bounded lr) using whole objective.
  - 2nd step : optimize θ by naïve GD using only $M(Z_\theta, \Gamma, A)$
  - 3rd step : Guide Γ, A by explicit solution : SVD of $Z_\theta Diag\left(\Pi_j\right) Z_\theta^T$ (called 'latching')

Variational formulation of MCR^2

- Overall algorithm :

Algorithm 1 Variational MCR^2 Training

Input: data $\boldsymbol{X}$, labels $\boldsymbol{Y}$, featurizer $f_\theta(\cdot)$, `latch-freq`, step sizes $(\nu_\theta, \nu_\Gamma, \nu_{\boldsymbol{A}})$
Initialize $\boldsymbol{A}, \Gamma \leftarrow \text{latching}(\boldsymbol{X}, \boldsymbol{Y}, f_\theta)$
for `iter` $= 0, 1, \ldots, n-1$ **do**
  Get $\boldsymbol{Z}_\theta = f_\theta(\boldsymbol{X})$ and membership matrices $\boldsymbol{\Pi}$
  Get $\ell_{\text{V-MCR}^2}(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A})$
  Compute $L_{\boldsymbol{A}}, L_\Gamma$
  $\Gamma \leftarrow \Gamma + \frac{\nu_\Gamma}{L_\Gamma} \nabla_\Gamma \ell_{\text{V-MCR}^2}(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A})$
  $\boldsymbol{A} \leftarrow \boldsymbol{A} + \frac{\nu_{\boldsymbol{A}}}{L_{\boldsymbol{A}}} \nabla_{\boldsymbol{A}} \ell_{\text{V-MCR}^2}(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A})$
  Project $\boldsymbol{A} \leftarrow \text{ReLU}(\boldsymbol{A})$
  Project $\Gamma_l \leftarrow \frac{1}{\|\Gamma_l\|_2} \Gamma_l \quad \forall l \in [q]$
  Recompute $M(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A})$
  $\theta \leftarrow \theta - \nu_\theta \nabla_\theta (M(\boldsymbol{Z}_\theta, \Gamma, \boldsymbol{A}))$
  if `iter` mod `latch-freq` $= 0$ **then**
    $\boldsymbol{A}, \Gamma \leftarrow \text{latching}(\boldsymbol{X}, \boldsymbol{Y}, f_\theta)$
  end if
end for
return f_θ

Algorithm 2 Latching

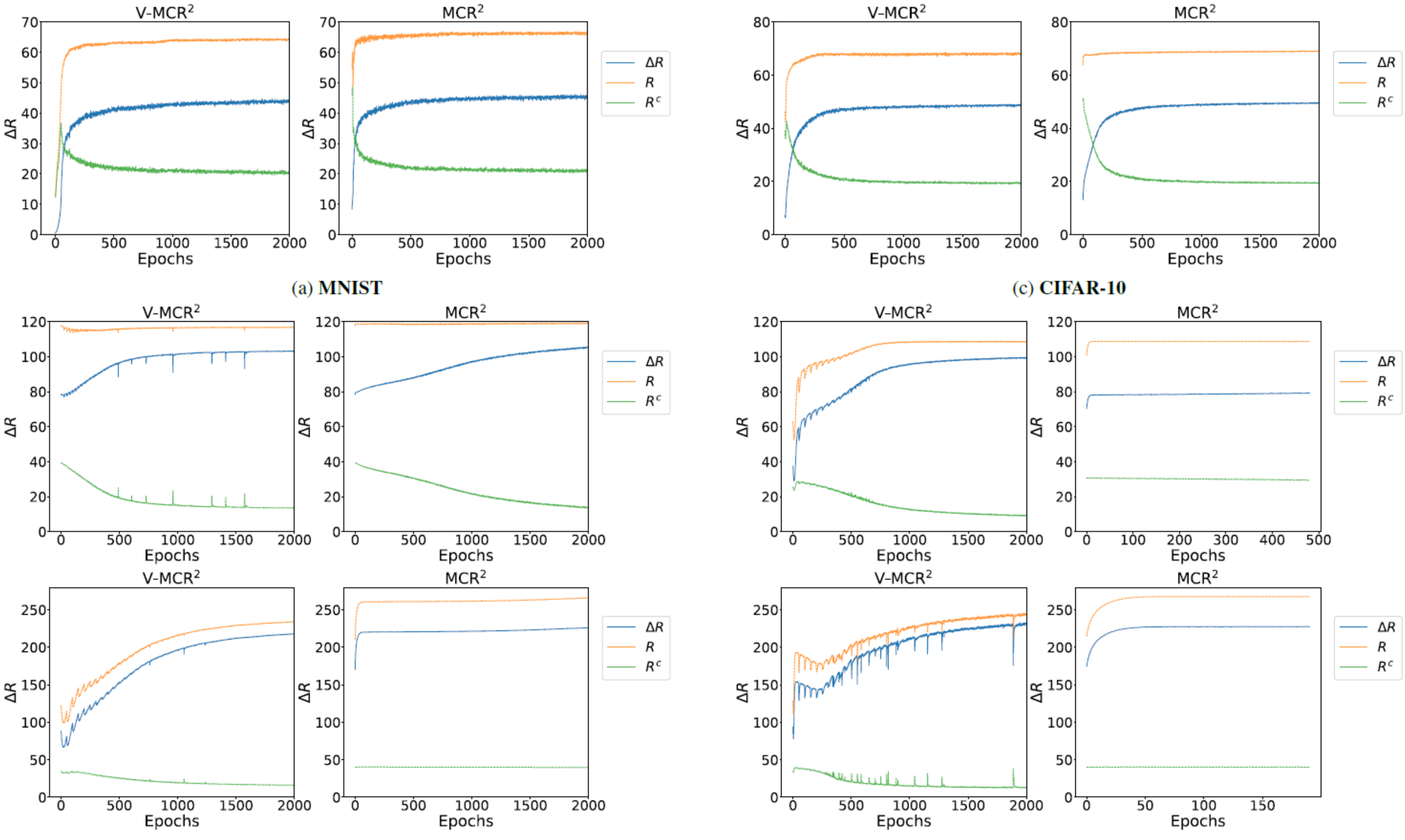
Input: data $\boldsymbol{X}$, labels $\boldsymbol{Y}$, featurizer $f_\theta(\cdot)$
Get $\boldsymbol{Z}_\theta = f_\theta(\boldsymbol{X}) \in \mathbb{R}^{d \times m}$ and membership $\boldsymbol{\Pi} \in \mathbb{R}^{m \times k}$
$\boldsymbol{A} \leftarrow \boldsymbol{0} \in \mathbb{R}^{q \times k}$ (assume q is divisible by k)
$\Gamma \leftarrow \boldsymbol{0} \in \mathbb{R}^{d \times q}$
for $j = 1, \ldots, k$ **do**
  Get $\boldsymbol{U}\text{Diag}(\boldsymbol{\sigma})\boldsymbol{V}^\top = \text{SVD}(\boldsymbol{Z}_\theta \text{Diag}(\boldsymbol{\Pi}_j) \boldsymbol{Z}_\theta^\top)$
  $s \leftarrow q/k$
  $\Gamma[:, (j-1)*s : j*s] = \boldsymbol{U}[:, 0 : s]$ % python indexing
  $\boldsymbol{A}[(j-1)*s : j*s, j] = \boldsymbol{\sigma}[0 : s]$ % python indexing
end for
return $\boldsymbol{A}, \Gamma$

1. ***Recall that optimal*** $U^{(j)} = SVD(Z_\theta Diag(\Pi_j) Z_\theta)$ ***when*** $q = k$
2. ***When the latching is applied, we get*** s^{th} ***order PC for each*** $U^{(j)}$***:***
   Especially, $U^{(j)}_{l+j*s} = \sigma_l U_l$ ***for each*** $j \in [k]$ ***and*** $l \in [s]$ ***(o.w 0 vector)***
   where $U_l = l\ th$ ***column of*** U ***(from SVD of*** $Z_\theta Diag(\Pi_j) Z_\theta^T$***)***
3. ***According to paper, it helps to improve convergence (although it is quite computationally expensive ~*** $O(kd^3)$ ***)***

Experiments of $V - MCR^2$

- Then, ΔR of $V - MCR^2$ is similar for original MCR^2 when k and d is small.
- However, it requires some time to have similar ΔR when k and d is large.

Convergence of training ΔR



(a) **MNIST**

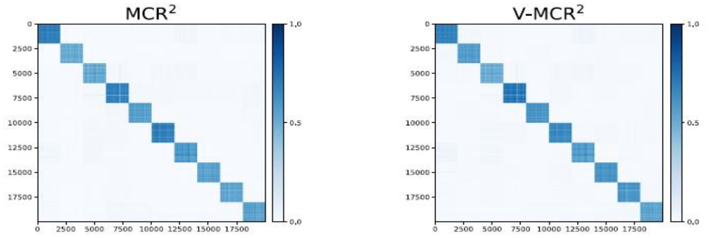
(c) **CIFAR-10**

(b) **CIFAR-100** $d = 100$ (top), $d = 500$ (bottom)

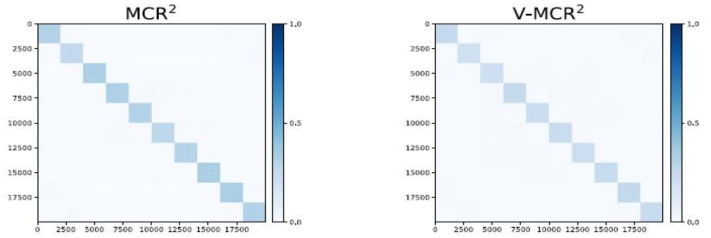
(d) **Tiny ImageNet** $d = 200$ (top), $d = 500$ (bottom)

Experiments of $V - MCR^2$

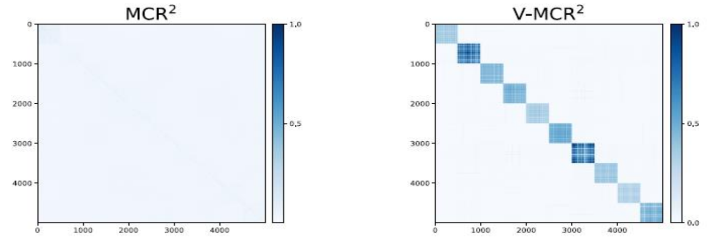
- Surprisingly $V - MCR^2$ shows better representation learning compared to original MCR^2:



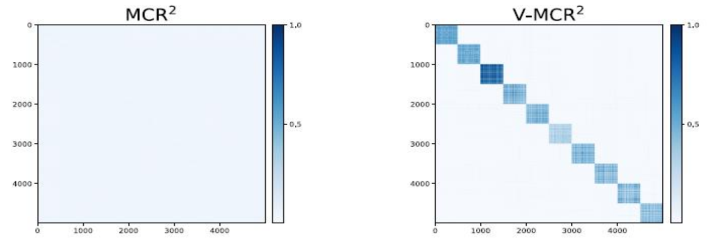
(a) MNIST



(b) CIFAR-10



(c) CIFAR-100



(d) Tiny ImageNet

Dataset	Objective	Training ΔR	Test Accuracy
MNIST	MCR^2	44.6429	0.9785
	V-MCR^2	44.2117	**0.9788**
	CE	-	0.9738
CIFAR-10	MCR^2	49.40	0.8956
	V-MCR^2	48.43	**0.8997**
	CE	-	0.8665
CIFAR-100	MCR^2	226.0519	0.2421
	V-MCR^2	218.0185	**0.5872**
	CE	-	0.5840
Tiny ImageNet 200	MCR^2	227.6468	0.1319
	V-MCR^2	231.1538	**0.2665**
	CE	-	0.1907

Inner product of representations (Left)

Comparison of classification performance (Center)

Note (left) : Sort the columns of Z_θ by classes and calculate absolute value of inner product $|Z_\theta^T Z_\theta|$

$V - MCR^2$ related thoughts

- Although $V - MCR^2$ is approximation of MCR^2, it shows better representation learning performance in the criteria : (seemingly more direct criteria compared to InfoMax principle)

  high quality representation ⇔ if points from different classes lie on separate, orthogonal subspaces, and the union these subspaces span as many dimensions as possible.

1. What optimization process result in better representation learning ?
   - The common phenomenon on $V - MCR^2$ is that R^c term is further decreased compared to MCR^2, which directly contributed the orthogonality of subspaces
   - It seems that reparameterization plays key role in further minimizing the R^c term.

$V - MCR^2$ related thoughts

2. The original MCR^2 is heading for low dimensional ***linear*** subspaces that classifies well.
   - Hence, it will obviously improve downstream performance under linear evaluation protocol if the learned subspaces reflect the representation well enough.

3. Again, **the assumption that $\boldsymbol{z \sim MN}$ or $\boldsymbol{Z \sim mixture\ of\ MN}$ is not guaranteed here.**
   - If the assumption holds, it would be better to just use EM algorithm on learned representation. (EM algorithm is well-known for good performance on grouping mixture of MN)
   - To avoid this assumption, we may be able to use variational bounds for entropy of $p(z)$ to replace R or R_c term. ($R \sim H(Z)$, $R_c \sim H(Z|Y)$)
   - Before this, why this assumption can be assumed in very complex data set???
   - Or we can design our architecture for f_θ to enforce the learned representation to follow mixture of MN.