Interpretable White-box Deep Networks

CSCI-699: Theory of Machine Learning

Presenter: Zheyi Zhu, Jingmin Wei. Nov 13, 2023

Outline

- Motivations
- Rate Reduction
- ReduNet for Optimizing Rate Reduction
- White-box Transformers (CRATE)
- Conclusions

Motivations

Why White-box?

Motivation: A gap between practice (directly use CE) and theory (how it works).

$$\min_{\boldsymbol{\theta}\in\Theta} \mathrm{CE}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) \doteq -\mathbb{E}[\langle \boldsymbol{y}, \log[f(\boldsymbol{x}, \boldsymbol{\theta})]\rangle] \approx -\frac{1}{m}\sum_{i=1}^{m}\langle \boldsymbol{y}^i, \log[f(\boldsymbol{x}^i, \boldsymbol{\theta})]\rangle.$$

Objective:

- Make the representation of data easy to use.
- Understand the complicated mapping of deep neural nets.

Hence, interpret deep networks from the principles of data compression and discriminative representation.

Good Data Representation

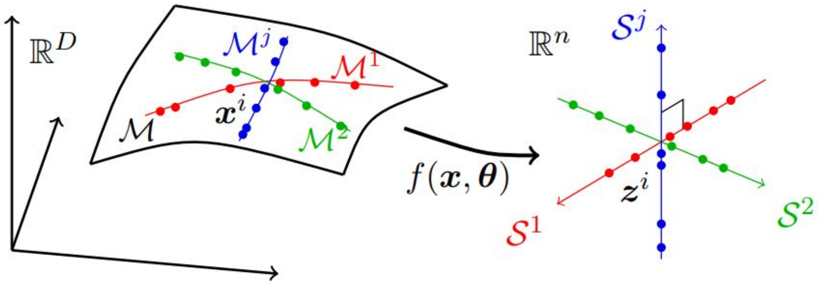
Assume the mixed data lies on low dimension of sub manifolds M.

Try to make that representation of data easy to use.

What is good representation?

For the data between different classes: highly incoherent.

For the data within the same class: stay close together.



Rate Reduction

Rate Distortion

What is the volume spanned by all the features? Consider feature mapping:

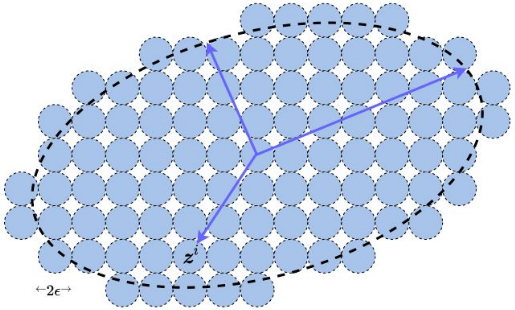
$$\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{D \times m} \xrightarrow{f(\boldsymbol{x}, \theta)} \boldsymbol{Z}(\theta) = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m] \in \mathbb{R}^{d \times m}.$$

The average coding length per sample (rate) subject to a distortion ϵ:

$$R(\boldsymbol{Z}, \epsilon) \doteq \frac{1}{2} \log \det \left(\boldsymbol{I} + \frac{n}{m\epsilon^2} \boldsymbol{Z}\boldsymbol{Z}^* \right).$$

This gives the number of binary bits in order to save the data (Z matrix) in the computer.

From the figure, it means how many ϵ balls can be packed into the volume of the space (volume of whole / volume of each ball)



$\mathrm{vol}(\boldsymbol{Z})$

Chan, Kwan Ho Ryan, et al. "ReduNet: A white-box deep network from the principle of maximizing rate reduction." *The Journal of Machine Learning Research* 23.1 (2022): 4907-5009.

Rate Distortion

What is the volume spanned by individual classes? For data with multiple classes (subsets):

$$\boldsymbol{Z} = \boldsymbol{Z}_1 \cup \boldsymbol{Z}_2 \cup \cdots \cup \boldsymbol{Z}_k.$$

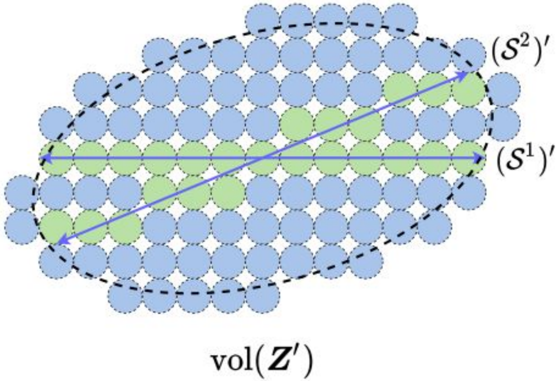
Π encodes the membership of m samples in k classes.

$$\boldsymbol{\Pi} = \{\boldsymbol{\Pi}^j \in \mathbb{R}^{m \times m}\}_{j=1}^{k}$$

With the partition, the average number of bits per sample

(coding rate) can be written:

$$R_c(\boldsymbol{Z}, \epsilon \mid \boldsymbol{\Pi}) \doteq \sum_{j=1}^{k} \frac{\operatorname{tr}(\boldsymbol{\Pi}^j)}{2m} \log\det\left(\boldsymbol{I} + \frac{n}{\operatorname{tr}(\boldsymbol{\Pi}^j)\epsilon^2} \boldsymbol{Z}\boldsymbol{\Pi}^j \boldsymbol{Z}^*\right).$$



Rate Reduction

Maximize the difference between the space of all features and the average rate for individual classes:

$$\Delta R(\boldsymbol{Z}, \boldsymbol{\Pi}, \epsilon) = \underbrace{\frac{1}{2} \log\det\left(\boldsymbol{I} + \frac{d}{m\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top\right)}_{R} - \underbrace{\sum_{j=1}^{k} \frac{\mathrm{tr}(\boldsymbol{\Pi}_j)}{2m} \log\det\left(\boldsymbol{I} + \frac{d}{\mathrm{tr}(\boldsymbol{\Pi}_j)\epsilon^2}\boldsymbol{Z}\boldsymbol{\Pi}_j\boldsymbol{Z}^\top\right)}_{R^c}.$$

$$\Delta R(\boldsymbol{Z}, \boldsymbol{\Pi}, \epsilon) \doteq R(\boldsymbol{Z}, \epsilon) - R_c(\boldsymbol{Z}, \epsilon \mid \boldsymbol{\Pi}).$$

Expand features: make different classes as different as possible.

Compress each class: all the features belongs to the same class as small as possible.

Maximizing Rate Reduction

First term (green spheres + blue spheres): how many sphere ϵ ball can pack into the space.

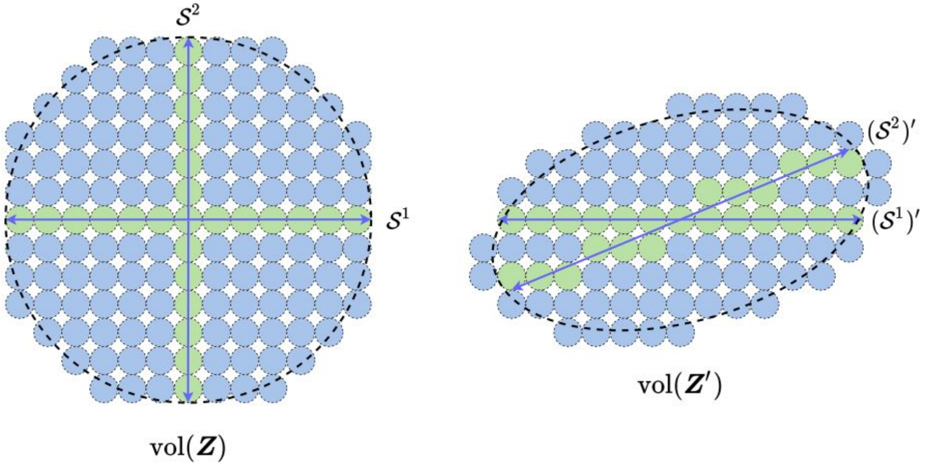
Second term (green spheres): only need the number of green balls to pack the data.

$$\max_{\theta} \quad \Delta R\big(\boldsymbol{Z}(\theta), \boldsymbol{\Pi}, \epsilon\big) = R(\boldsymbol{Z}(\theta), \epsilon) - R^c(\boldsymbol{Z}(\theta), \epsilon \mid \boldsymbol{\Pi}),$$
$$\text{subject to} \quad \|\boldsymbol{Z}_j(\theta)\|_F^2 = m_j, \ \boldsymbol{\Pi} \in \Omega.$$

Rate reduction: blue spheres.

New objective function:

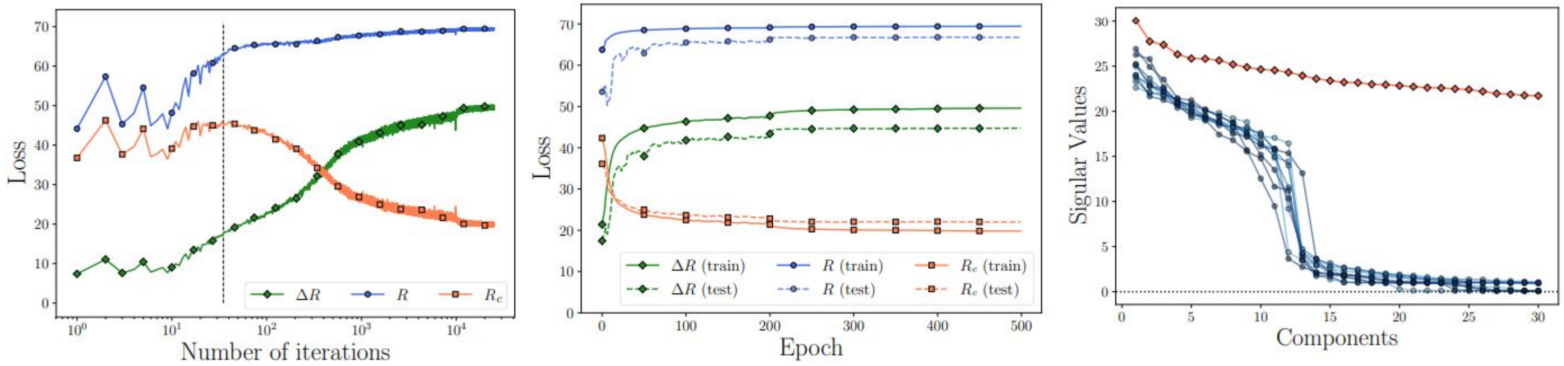
Max the Coding Rate Reduction, MCR².



Yu, Yaodong, et al. "Learning diverse and discriminative representations via the principle of maximal coding rate reduction." *Advances in Neural Information Processing Systems* 33 (2020): 9422-9434.

Experiments

ResNet-18 on CIFAR-10. Replace CE with Rate Reduction (MCR^2):



(a) Evolution of $R, R_c, \Delta R$ during the training process.

(b) Training loss versus testing loss.

(c) PCA: (**red**) overall data; (**blue**) individual classes.

Classification results with features learned with labels corrupted at different levels:

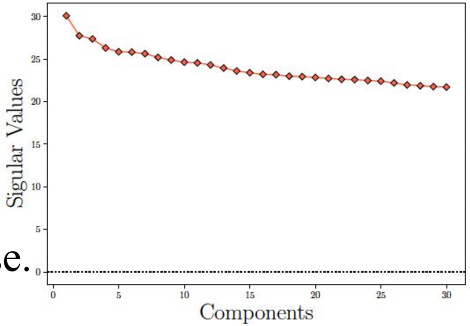
	Ratio=0.0	Ratio=0.1	Ratio=0.2	Ratio=0.3	Ratio=0.4	Ratio=0.5
CE Training	0.939	0.909	0.861	0.791	0.724	0.603
MCR^2 Training	**0.940**	**0.911**	**0.897**	**0.881**	**0.866**	**0.843**

USC

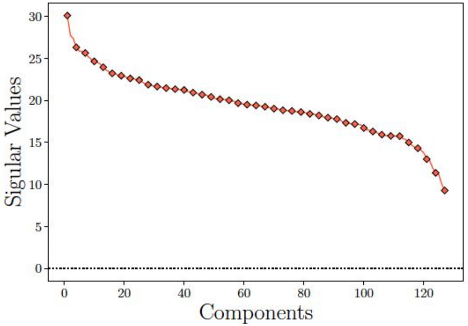
Experiments

Comparison:

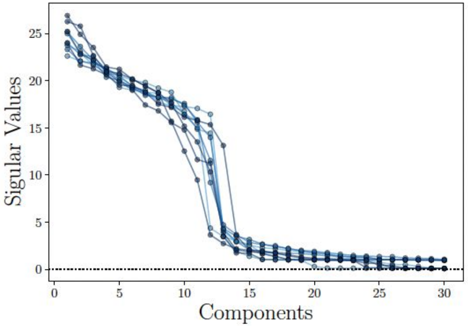
MCR² is more diverse.



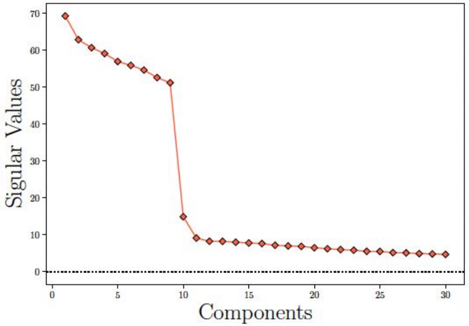
(a) PCA: MCR^2 training learned features for overall data (first 30 components).



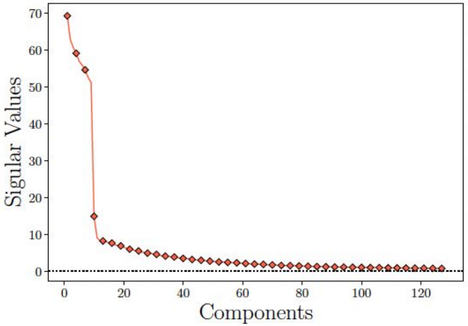
(b) PCA: MCR^2 training learned features for overall data.



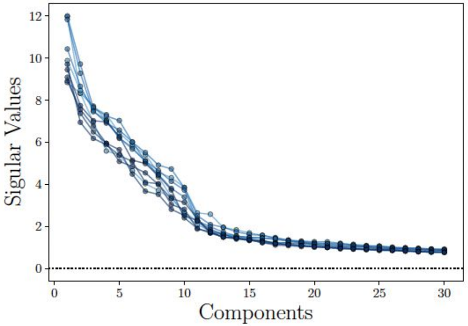
(c) PCA: MCR^2 training learned features for every class.



(d) PCA: cross-entropy training learned features for overall data (first 30 components).



(e) PCA: cross-entropy training learned features for overall data.



(f) PCA: cross-entropy training learned features for every class.

USC

ReduNet for Optimizing Rate Reduction

USC

Projected Gradient Ascent for Rate Reduction

We cannot directly optimize the function, since it's non-convex:

$$\Delta R(\boldsymbol{Z}, \boldsymbol{\Pi}, \epsilon) = R(\boldsymbol{Z}, \epsilon) - R_c(\boldsymbol{Z}, \epsilon \mid \boldsymbol{\Pi})$$
$$\doteq \underbrace{\frac{1}{2} \log \det \left(\boldsymbol{I} + \alpha \boldsymbol{Z} \boldsymbol{Z}^* \right)}_{R(\boldsymbol{Z}, \epsilon)} - \underbrace{\sum_{j=1}^{k} \frac{\gamma_j}{2} \log \det \left(\boldsymbol{I} + \alpha_j \boldsymbol{Z} \boldsymbol{\Pi}^j \boldsymbol{Z}^* \right)}_{R_c(\boldsymbol{Z}, \epsilon \mid \boldsymbol{\Pi})},$$

So we use PGA to optimize:

$$\boldsymbol{Z}_{\ell+1} \propto \boldsymbol{Z}_\ell + \eta \cdot \left. \frac{\partial \Delta R}{\partial \boldsymbol{Z}} \right|_{\boldsymbol{Z}_\ell} \quad \text{s.t.} \quad \boldsymbol{Z}_{\ell+1} \subset \mathbb{S}^{n-1}, \; \ell = 1, 2, \ldots,$$

Projected Gradient Ascent for Rate Reduction

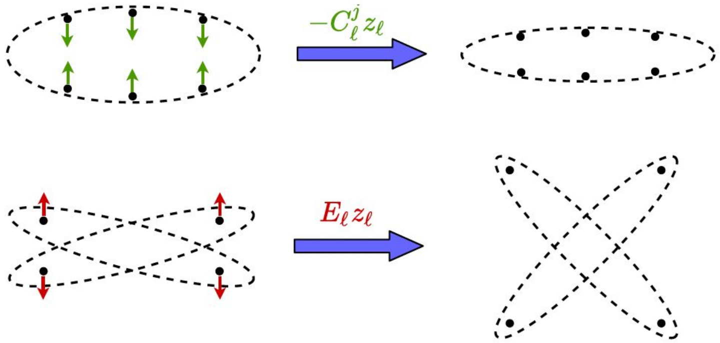
The derivatives are:

$$\frac{1}{2}\frac{\partial \log\det(\boldsymbol{I}+\alpha \boldsymbol{Z}\boldsymbol{Z}^*)}{\partial \boldsymbol{Z}}\bigg|_{\boldsymbol{Z}_\ell} = \underbrace{\alpha(\boldsymbol{I}+\alpha \boldsymbol{Z}_\ell \boldsymbol{Z}_\ell^*)^{-1}}_{\boldsymbol{E}_\ell \in \mathbb{R}^{n\times n}} \boldsymbol{Z}_\ell,$$

$$\frac{1}{2}\frac{\partial\left(\gamma_j \log\det(\boldsymbol{I}+\alpha_j \boldsymbol{Z}\boldsymbol{\Pi}^j \boldsymbol{Z}^*)\right)}{\partial \boldsymbol{Z}}\bigg|_{\boldsymbol{Z}_\ell} = \gamma_j \underbrace{\alpha_j(\boldsymbol{I}+\alpha_j \boldsymbol{Z}_\ell \boldsymbol{\Pi}^j \boldsymbol{Z}_\ell^*)^{-1}}_{\boldsymbol{C}_\ell^j \in \mathbb{R}^{n\times n}} \boldsymbol{Z}_\ell \boldsymbol{\Pi}^j.$$

Hence the gradient is:

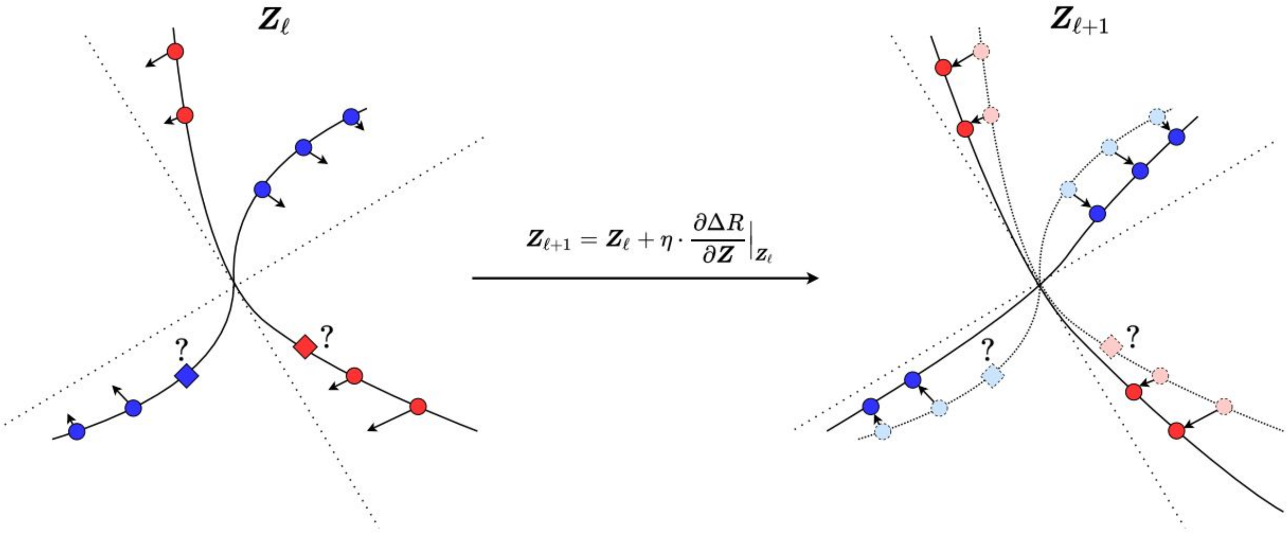
$$\frac{\partial \Delta R}{\partial \boldsymbol{Z}}\bigg|_{\boldsymbol{Z}_\ell} = \underbrace{\boldsymbol{E}_\ell}_{\text{Expansion}} \boldsymbol{Z}_\ell - \sum_{j=1}^{k} \gamma_j \underbrace{\boldsymbol{C}_\ell^j}_{\text{Compression}} \boldsymbol{Z}_\ell \boldsymbol{\Pi}^j.$$



Projected Gradient Ascent for Rate Reduction

Goal: try to push the data into orthogonal subspaces.

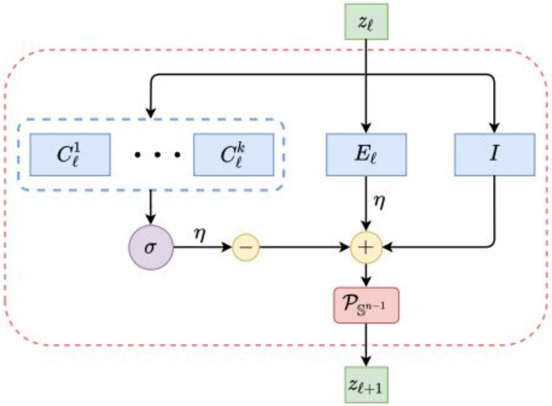
Locally, compute the gradient flow, and push the data along the gradient flow.



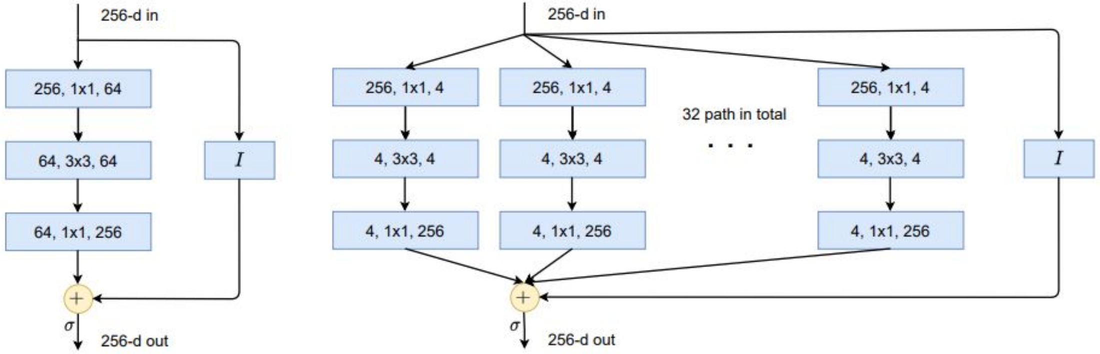
ReduNet for Optimizing Rate Reduction

One layer of the ReduNet: one PGA iteration.

$$\boldsymbol{z}_{\ell+1} \propto \boldsymbol{z}_\ell + \eta \cdot \underbrace{\left[\boldsymbol{E}_\ell \boldsymbol{z}_\ell + \boldsymbol{\sigma}\left([\boldsymbol{C}_\ell^1 \boldsymbol{z}_\ell, \dots, \boldsymbol{C}_\ell^k \boldsymbol{z}_\ell]\right)\right]}_{g(\boldsymbol{z}_\ell, \boldsymbol{\theta}_\ell)} \quad \text{s.t.} \quad \boldsymbol{z}_{\ell+1} \in \mathbb{S}^{d-1}$$



(a) **ReduNet**



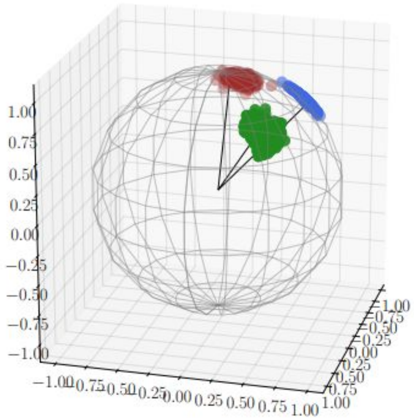
(b) **ResNet** and **ResNeXt**.

 Chan, Kwan Ho Ryan, et al. "ReduNet: A white-box deep network from the principle of maximizing rate reduction." *The Journal of Machine Learning Research* 23.1 (2022): 4907-5009.

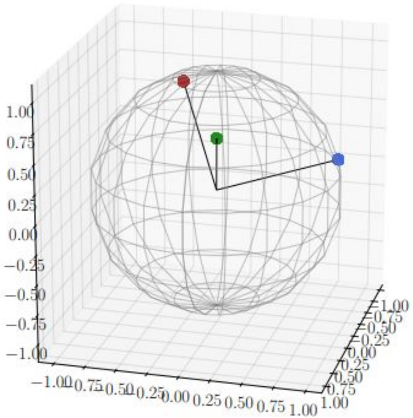
Learning Mixture of Gaussians

Left: original samples X and ReduNet features Z = f(Z, θ) for 3D Mixture of Gaussians.

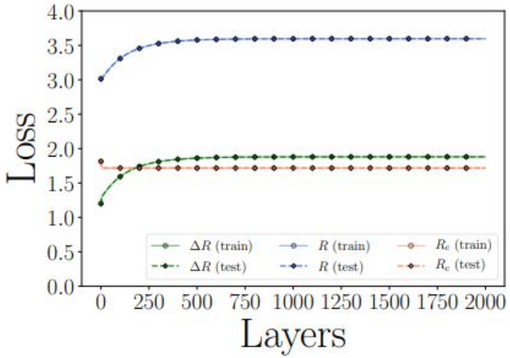
Right: plots for the progression of values of the rates.



(a) $\boldsymbol{X}_{\text{train}}$ $(3D)$



(b) $\boldsymbol{Z}_{\text{train}}$ $(3D)$



(c) Loss $(3D)$

White-box Transformers (CRATE)

White-Box Transformer (CRATE)

Can we interpret **transformers** via rate reduction?

Ultimate Goal: **compress** and **sparsify** representations of large-scale real-world vision datasets

Multi-head self-attention operator: a gradient descent step to compress the token sets by minimizing their lossy coding rate

Subsequent multi-layer perceptron: attempting to sparsify the representation

Unified objective function: sparse rate reduction

Yu, Yaodong, et al. "White-Box Transformers via Sparse Rate Reduction." *arXiv preprint arXiv:2306.01129* (2023).

Unified Objective Function - Sparse Rate Reduction

$$\max_{f\in\mathcal{F}} \mathbb{E}_{\boldsymbol{Z}}\big[\Delta R(\boldsymbol{Z};\boldsymbol{U}_{[K]}) - \lambda\|\boldsymbol{Z}\|_0\big] = \max_{f\in\mathcal{F}} \mathbb{E}_{\boldsymbol{Z}}\big[R(\boldsymbol{Z}) - R^c(\boldsymbol{Z};\boldsymbol{U}_{[K]}) - \lambda\|\boldsymbol{Z}\|_0\big] \text{ s.t. } \boldsymbol{Z} = f(\boldsymbol{X})$$

$$R(\boldsymbol{Z}) \doteq \frac{1}{2}\operatorname{logdet}\left(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}^*\boldsymbol{Z}\right) = \frac{1}{2}\operatorname{logdet}\left(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^*\right)$$

$$R^c(\boldsymbol{Z};\boldsymbol{U}_{[K]}) = \sum_{k=1}^{K} R(\boldsymbol{U}_k^*\boldsymbol{Z}) = \frac{1}{2}\sum_{k=1}^{K}\operatorname{logdet}\left(\boldsymbol{I} + \frac{p}{N\epsilon^2}(\boldsymbol{U}_k^*\boldsymbol{Z})^*(\boldsymbol{U}_k^*\boldsymbol{Z})\right)$$

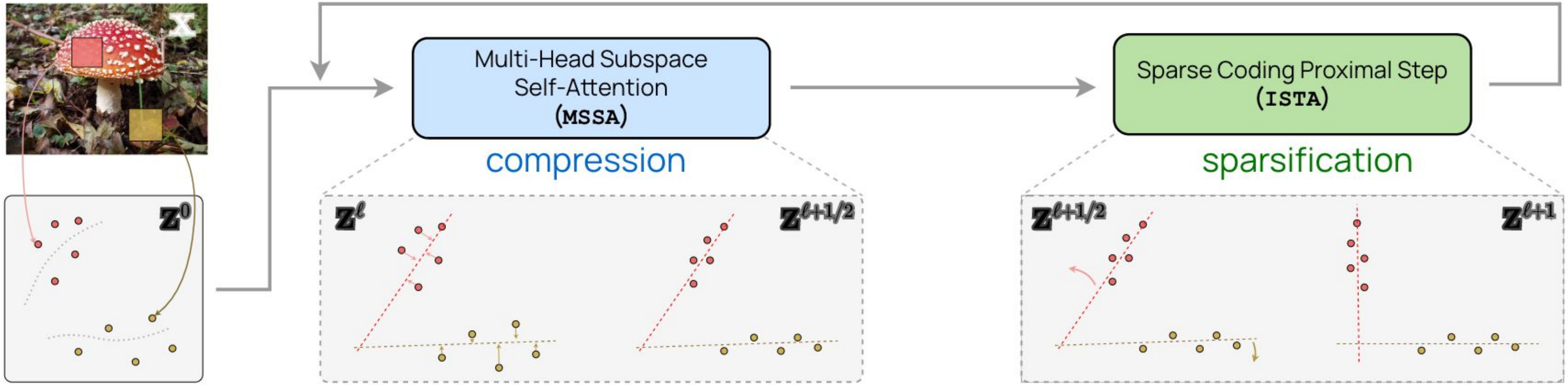
$\boldsymbol{U}_{[K]} = (\boldsymbol{U}_k)_{k=1}^{K}$ bases for supporting subspaces for the mixture-of-Gaussians model at each layer

---

$$f\colon \boldsymbol{X} \xrightarrow{f^0} \boldsymbol{Z}^0 \rightarrow \cdots \rightarrow \boldsymbol{Z}^\ell \xrightarrow{f^\ell} \boldsymbol{Z}^{\ell+1} \rightarrow \cdots \rightarrow \boldsymbol{Z}^L = \boldsymbol{Z}$$

$$\boldsymbol{Z}^{\ell+1} = f^\ell(\boldsymbol{Z}^\ell)$$

‘Main Loop’ of the White-box Transformers Design



Compression (multi-head self-attention):

transform the data to low dimensional subspaces by minimizing the coding rate

Sparsification (mlp):

sparse coding against a global dictionary

Minimizing Coding Rate Reduction

$$R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]}) = \sum_{k=1}^{K} R(\boldsymbol{U}_k^* \boldsymbol{Z}) = \frac{1}{2} \sum_{k=1}^{K} \operatorname{logdet}\left(\boldsymbol{I} + \frac{p}{N\epsilon^2} (\boldsymbol{U}_k^* \boldsymbol{Z})^* (\boldsymbol{U}_k^* \boldsymbol{Z})\right)$$

$$\nabla_{\boldsymbol{Z}} R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]}) = \frac{p}{N\epsilon^2} \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^* \boldsymbol{Z} \left(\boldsymbol{I} + \frac{p}{N\epsilon^2} (\boldsymbol{U}_k^* \boldsymbol{Z})^* (\boldsymbol{U}_k^* \boldsymbol{Z})\right)^{-1}$$

$$\boldsymbol{Z}^{\ell+1/2} = \boldsymbol{Z}^{\ell} - \kappa \nabla_{\boldsymbol{Z}} R^c(\boldsymbol{Z}^{\ell}; \boldsymbol{U}_{[K]}) \approx \left(1 - \kappa \cdot \frac{p}{N\epsilon^2}\right) \boldsymbol{Z}^{\ell} + \kappa \cdot \frac{p}{N\epsilon^2} \cdot \texttt{MSSA}(\boldsymbol{Z}^{\ell} \mid \boldsymbol{U}_{[K]})$$

where MSSA is defined through an SSA operator as:

$$\texttt{SSA}(\boldsymbol{Z} \mid \boldsymbol{U}_k) \doteq (\boldsymbol{U}_k^* \boldsymbol{Z}) \operatorname{softmax}((\boldsymbol{U}_k^* \boldsymbol{Z})^* (\boldsymbol{U}_k^* \boldsymbol{Z})), \quad k \in [K],$$

$$\texttt{MSSA}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]}) \doteq \frac{p}{N\epsilon^2} \cdot [\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K] \begin{bmatrix} \texttt{SSA}(\boldsymbol{Z} \mid \boldsymbol{U}_1) \\ \vdots \\ \texttt{SSA}(\boldsymbol{Z} \mid \boldsymbol{U}_K) \end{bmatrix}.$$

Optimizing the Remaining Terms

$$\max_{\boldsymbol{Z}} [R(\boldsymbol{Z}) - \lambda \|\boldsymbol{Z}\|_0] = \min_{\boldsymbol{Z}} \left[\lambda \|\boldsymbol{Z}\|_0 - \frac{1}{2} \operatorname{logdet}\left(\boldsymbol{I} + \frac{d}{N\epsilon^2} \boldsymbol{Z}^* \boldsymbol{Z} \right) \right]$$

$$R(\boldsymbol{Z}) \doteq \frac{1}{2} \operatorname{logdet}\left(\boldsymbol{I} + \frac{d}{N\epsilon^2} \boldsymbol{Z}^* \boldsymbol{Z} \right) = \frac{1}{2} \operatorname{logdet}\left(\boldsymbol{I} + \frac{d}{N\epsilon^2} \boldsymbol{Z} \boldsymbol{Z}^* \right)$$

---

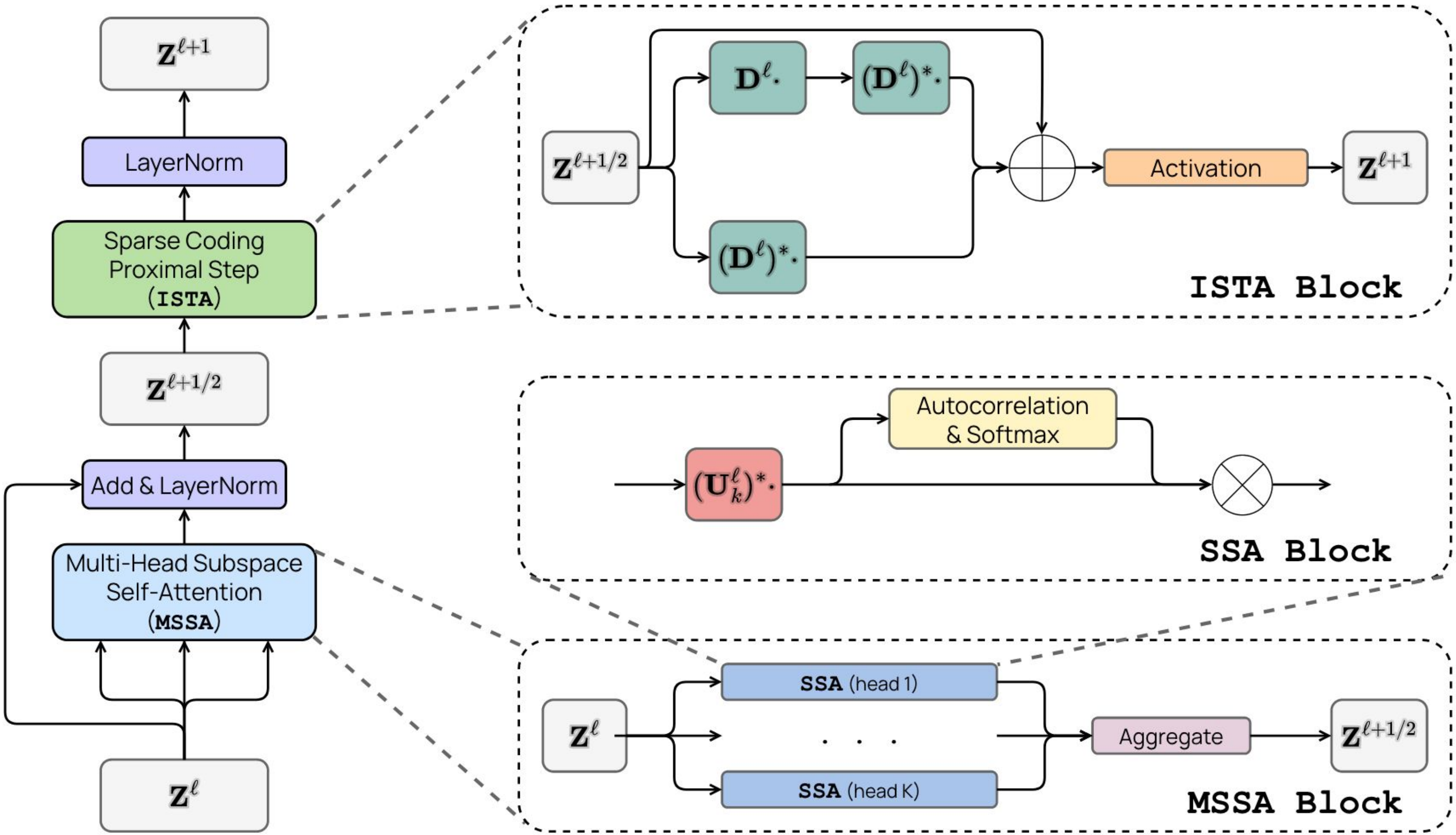
orthogonal dictionary $\boldsymbol{D} \in \mathbb{R}^{d \times d}$ $\quad \boldsymbol{D}^* \boldsymbol{D} \approx \boldsymbol{I}_d \quad \boldsymbol{Z}^{\ell+1/2} = \boldsymbol{D} \boldsymbol{Z}^{\ell+1}$

$$R(\boldsymbol{Z}^{\ell+1}) \approx R(\boldsymbol{D}\boldsymbol{Z}^{\ell+1}) = R(\boldsymbol{Z}^{\ell+1/2})$$

$$\boldsymbol{Z}^{\ell+1} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_0 \quad \text{subject to} \quad \boldsymbol{Z}^{\ell+1/2} = \boldsymbol{D}\boldsymbol{Z}$$

Unrolled proximal gradient descent step:

$$\boldsymbol{Z}^{\ell+1} = \operatorname{ReLU}(\boldsymbol{Z}^{\ell+1/2} + \eta \boldsymbol{D}^*(\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}\boldsymbol{Z}^{\ell+1/2}) - \eta\lambda \mathbf{1}) \doteq \texttt{ISTA}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D})$$



$\mathbf{Z}^{\ell+1}$
LayerNorm
Sparse Coding Proximal Step (ISTA)
$\mathbf{Z}^{\ell+1/2}$
Add & LayerNorm
Multi-Head Subspace Self-Attention (MSSA)
$\mathbf{Z}^{\ell}$
$\mathbf{Z}^{\ell+1/2}$
$\mathbf{D}^{\ell}\cdot$
$(\mathbf{D}^{\ell})^{*}\cdot$
$(\mathbf{D}^{\ell})^{*}\cdot$
Activation
$\mathbf{Z}^{\ell+1}$
ISTA Block
$(\mathbf{U}_k^{\ell})^{*}\cdot$
Autocorrelation & Softmax
SSA Block
$\mathbf{Z}^{\ell}$
SSA (head 1)
. . .
SSA (head K)
Aggregate
$\mathbf{Z}^{\ell+1/2}$
MSSA Block

USC

Experiment

Table 1: Top 1 accuracy of CRATE on various datasets with different model scales when pre-trained on ImageNet. For ImageNet/ImageNetReaL, we directly evaluate the top-1 accuracy. For other datasets, we use models that are pre-trained on ImageNet as initialization and the evaluate the transfer learning performance via fine-tuning.

Datasets	CRATE-T	CRATE-S	CRATE-B	CRATE-L	ViT-T	ViT-S
# parameters	6.09M	13.12M	22.80M	77.64M	5.72M	22.05M
ImageNet	66.7	69.2	70.8	71.3	71.5	72.4
ImageNet ReaL	74.0	76.0	76.5	77.4	78.3	78.4
CIFAR10	95.5	96.0	96.8	97.2	96.6	97.2
CIFAR100	78.9	81.0	82.7	83.6	81.8	83.2
Oxford Flowers-102	84.6	87.1	88.7	88.3	85.1	88.5
Oxford-IIIT-Pets	81.4	84.9	85.3	87.4	88.5	88.6

Conclusions

Conclusions

Rate reduction can help make better data representation. It expand the volume of all features and compress the volume for individual classes.

We can construct interpretable White-box Neural Nets via maximizing rate reduction.

The network is constructed in a forward fashion (with iterative optimization) instead of backward propagation. All the parameters are automatically initialized by the forward construction.

We can construct interpretable White-box Transformers via rate reduction and sparsification.

The multi-head self-attention layer can be viewed as a gradient descent to compress the token sets by minimizing their lossy coding rate, and the multi-layer perceptron layer can be viewed as attempting to sparsify the data representation.

Thanks for Listening

CSCI-699: Theory of Machine Learning

Interpretable White-box Deep Networks

Presenter: Zheyi Zhu, Jingmin Wei