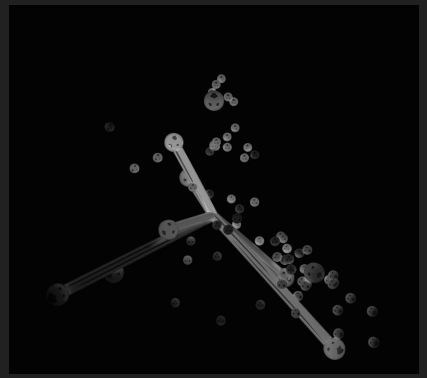
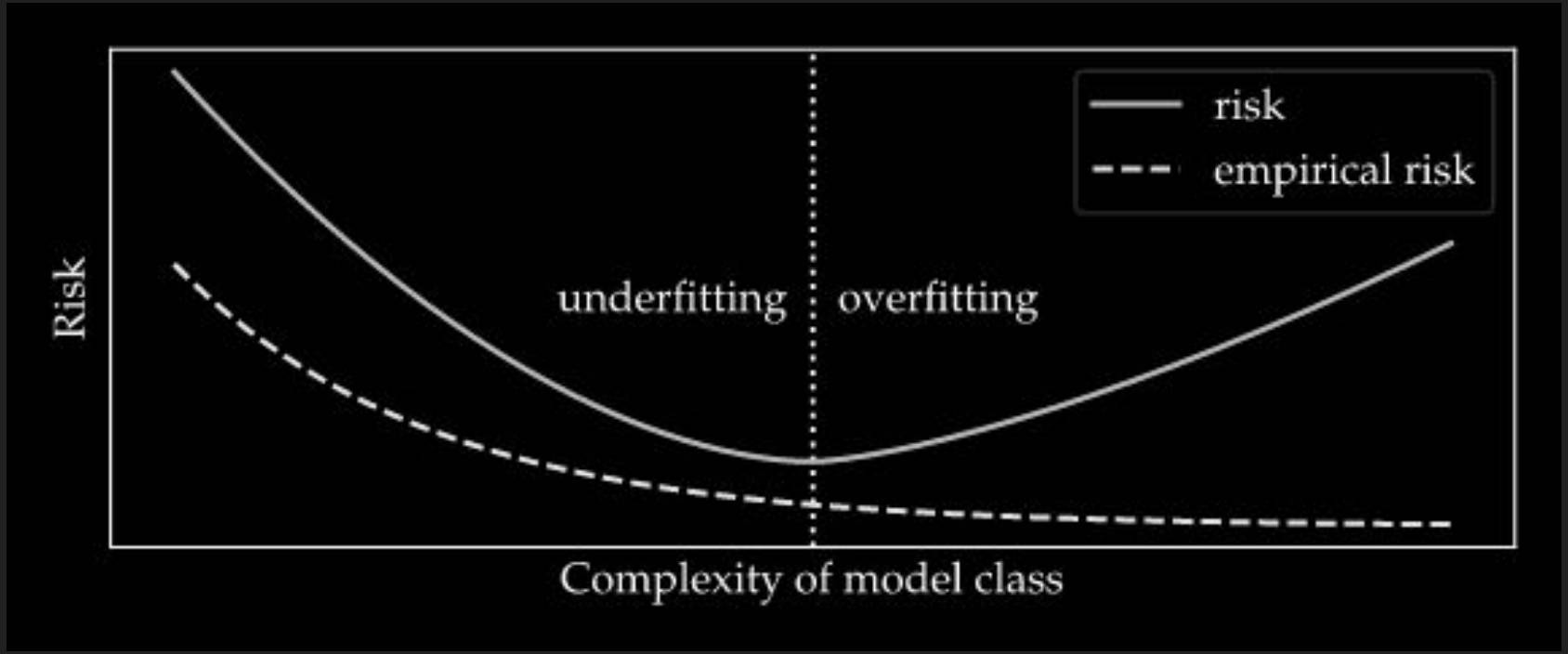
Prevalence of neural collapse during the terminal phase of deep learning training

Vardan Papyan, X. Y. Han, and David L. Donoho



Understanding Model Generalization



Overparameterized Models

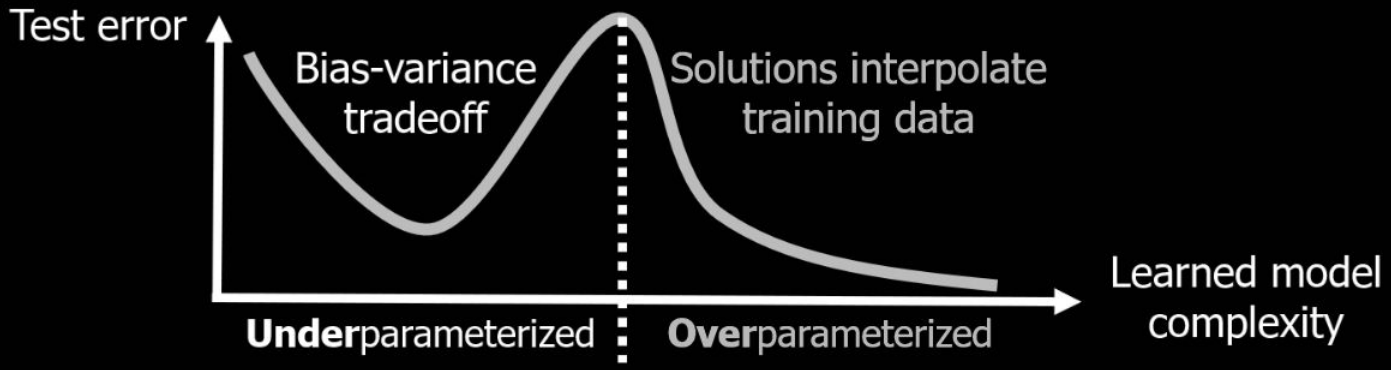
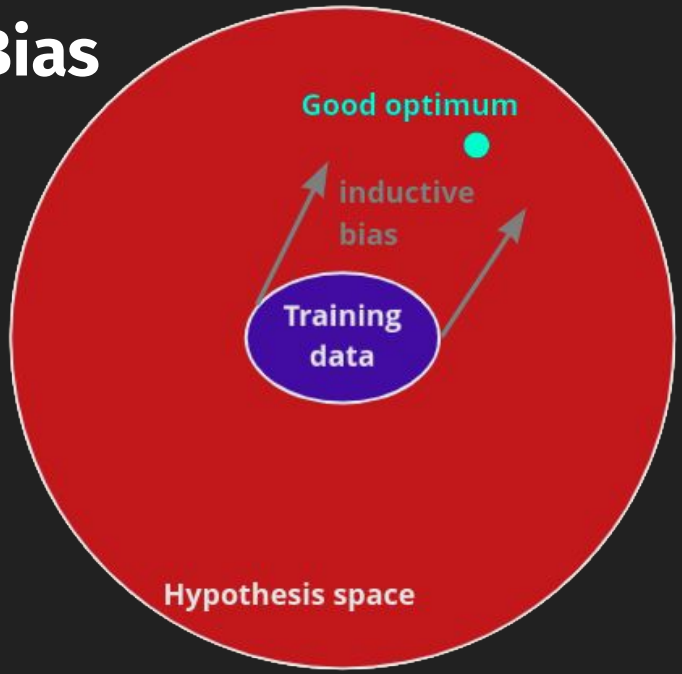
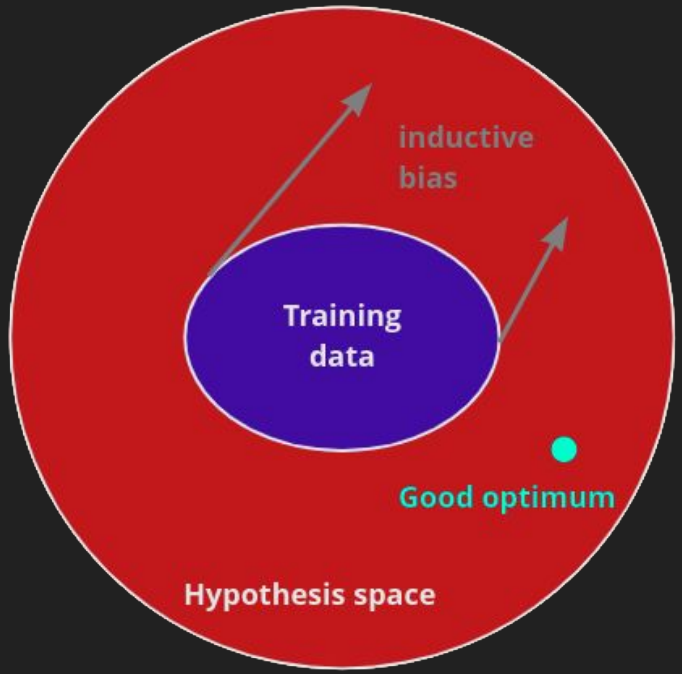


Figure 1: Double descent of test errors (i.e., generalization errors) with respect to the complexity of the learned model. TOPML studies often consider settings in which the learned model complexity is expressed as the number of (independently tunable) parameters in the model. In this qualitative demonstration, the global minimum of the test error is achieved by maximal overparameterization.

Inductive Bias





In a low data setting, right inductive bias may help to find good optimum, but in a rich data setting, it may lead to constraints that harm generalization. [Image Source]

(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

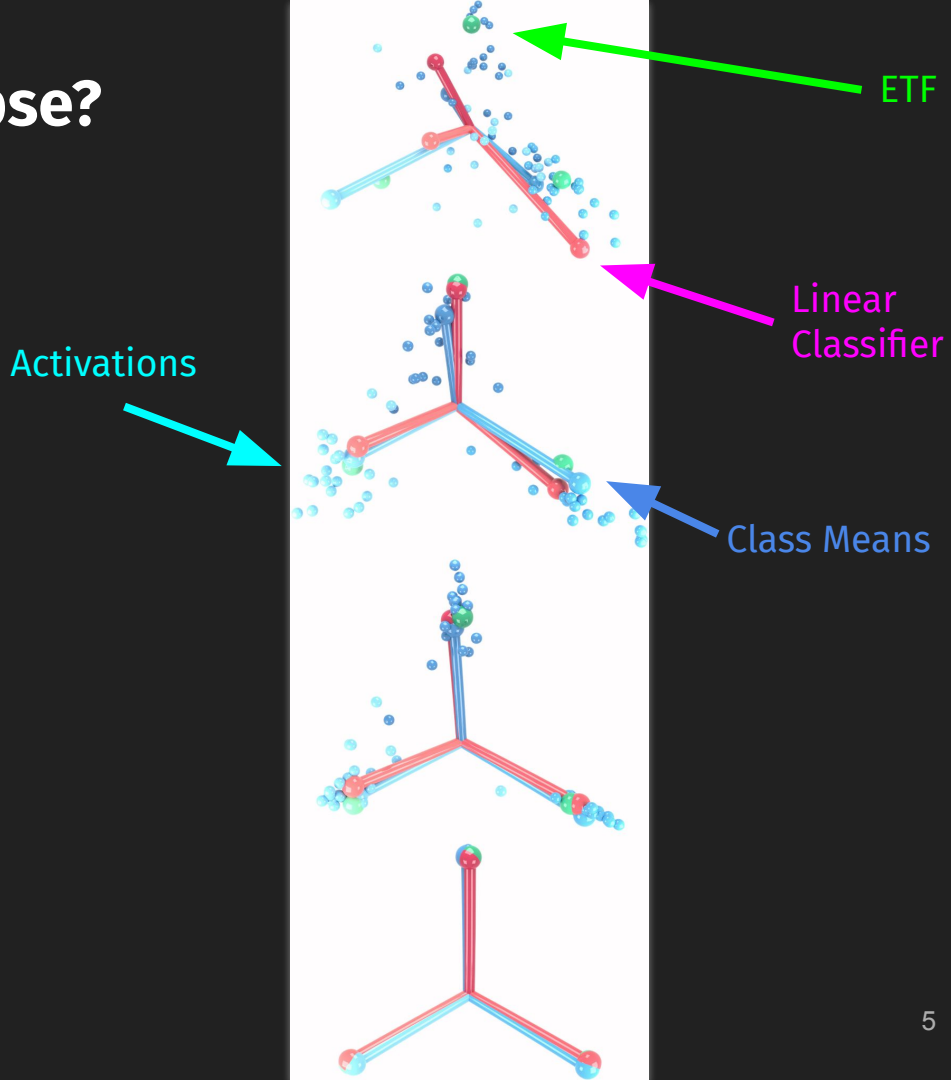
In Brief: What is Neural Collapse?

NC1: *Collapse of variability:*
For data samples belonging to the same class, their final hidden layer features ***concentrate around their class mean***. Thus, the variability of intra class features during training is lost as they collapse to a point.

NC2: *Preference towards a simplex ETF:*
The class means of the penultimate layer features tend to form a ***simplex equiangular tight frame*** (simplex ETF). A simplex ETF is a symmetric structure whose ***vertices lie on a hyper-sphere, are linearly separable and are placed at the maximum possible distance*** from each other.

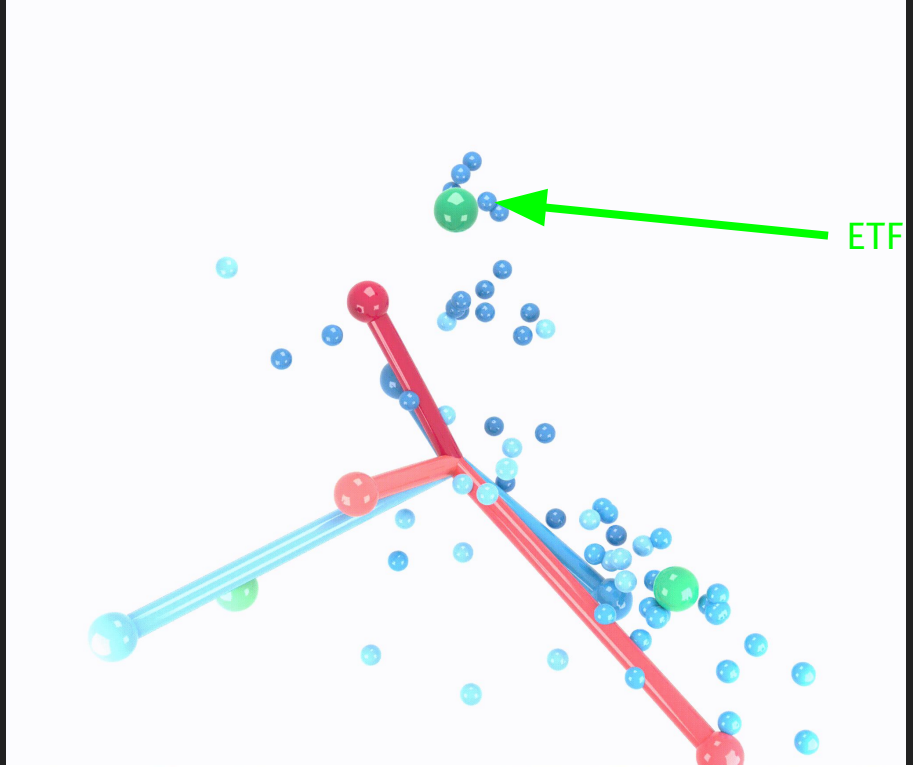
NC3: *Self-dual alignment:*
The ***columns of the last layer linear classifier matrix*** also form a simplex ETF in their dual vector space and converge to the simplex ETF (up to rescaling) of the penultimate layer features.

NC4: *Choose the nearest class mean:*
When a test point is to be classified, the last-layer classifier essentially acts as a ***nearest (train)-class mean decision rule w.r.t penultimate layer features***.



Class 0

Class 1



Final Layer

Visualization of NC. *The figure depicts, in three dimensions, NC as training proceeds from top to bottom. Green spheres represent the vertices of the standard simplex, see Definition (Simplex ETF), red ball and sticks represent linear classifiers, blue ball and sticks represent class means, and small blue spheres represent last-layer features. For all objects, we distinguish different classes via the shade of the color. As training proceeds, last-layer features collapse onto their class means (NC1), class means converge to the vertices of the simplex ETF (NC2), and the linear classifiers approach their corresponding class means (NC3). An animation can be found at https://purl.stanford.edu/br193mh4244.*

Definition: Simplex ETF

$$\boldsymbol{M}^\star = \sqrt{\frac{C}{C-1}}\left(\boldsymbol{I} - \frac{1}{C}\cancel{\mathbb{1}\mathbb{1}}^\top\right),$$

where $\boldsymbol{I} \in \mathbb{R}^{C\times C}$ is the identity matrix, and $\mathbb{1}_C \in \mathbb{R}^C$ is the ones vector. In this paper, we allow other poses, as well as rescaling, so the general simplex ETF consists of the points specified by the columns of $\boldsymbol{M} = \alpha \boldsymbol{U}\boldsymbol{M}^\star \in \mathbb{R}^{p\times C}$, where $\alpha \in \mathbb{R}_+$ is a scale factor, and $\boldsymbol{U} \in \mathbb{R}^{p\times C}$ ($p \geq C$) is a partial orthogonal matrix ($\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$).

Notation

Weights/Bias

$$\boldsymbol{W} \in \mathbb{R}^{C \times p} \quad \boldsymbol{b} \in \mathbb{R}^{C}$$

Activations from Encoder $\boldsymbol{h} = \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x})$.

Linear Classifier $\boldsymbol{W}\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{b}$.

Objective $$\min_{\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}} \sum_{c=1}^{C} \sum_{i=1}^{N} \mathscr{L}(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,c}) + \boldsymbol{b}, \boldsymbol{y}_c).$$

Notation

Train Global Mean

$$\boldsymbol{\mu}_G \triangleq \underset{i,c}{\operatorname{Ave}} \{\boldsymbol{h}_{i,c}\}$$

objective

$$\min_{\boldsymbol{\theta},\boldsymbol{W},\boldsymbol{b}} \sum_{c=1}^{C} \sum_{i=1}^{N} \mathscr{L}(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_{i,c}) + \boldsymbol{b}, \boldsymbol{y}_c).$$

Train class means

$$\boldsymbol{\mu}_c \triangleq \underset{i}{\operatorname{Ave}} \{\boldsymbol{h}_{i,c}\}, \quad c = 1, \ldots, C,$$

Train total covariance

$$\boldsymbol{\Sigma}_T \triangleq \underset{i,c}{\operatorname{Ave}} \left\{ (\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_G)(\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_G)^\top \right\},$$

Inter-class COV

$$\boldsymbol{\Sigma}_B \triangleq \underset{c}{\operatorname{Ave}} \left\{ (\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)(\boldsymbol{\mu}_c - \boldsymbol{\mu}_G)^\top \right\},$$

Intra-class COV

$$\boldsymbol{\Sigma}_W \triangleq \underset{i,c}{\operatorname{Ave}} \left\{ (\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_c)(\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_c)^\top \right\}.$$

Formalizations

(NC1) Variability collapse: $\Sigma_W \to 0$

(Recall what ____ is)

$$\Sigma_W \triangleq \underset{i,c}{\text{Ave}} \left\{ (h_{i,c} - \mu_c)(h_{i,c} - \mu_c)^\top \right\}.$$

(NC2) Convergence to Simplex ETF:

$$\left| \|\mu_c - \mu_G\|_2 - \|\mu_{c'} - \mu_G\|_2 \right| \to 0 \quad \forall\, c, c'$$

$$\langle \tilde{\mu}_c, \tilde{\mu}_{c'} \rangle \to \frac{C}{C-1}\delta_{c,c'} - \frac{1}{C-1} \quad \forall\, c, c'.$$

$$\tilde{\mu}_c = (\mu_c - \mu_G) / \|\mu_c - \mu_G\|_2$$

$$M^\star = \sqrt{\frac{C}{C-1}} \left(I - \frac{1}{C} \mathbb{1}_C \mathbb{1}_C^\top \right),$$

Formalizations

(NC3) Convergence to Self-Duality:

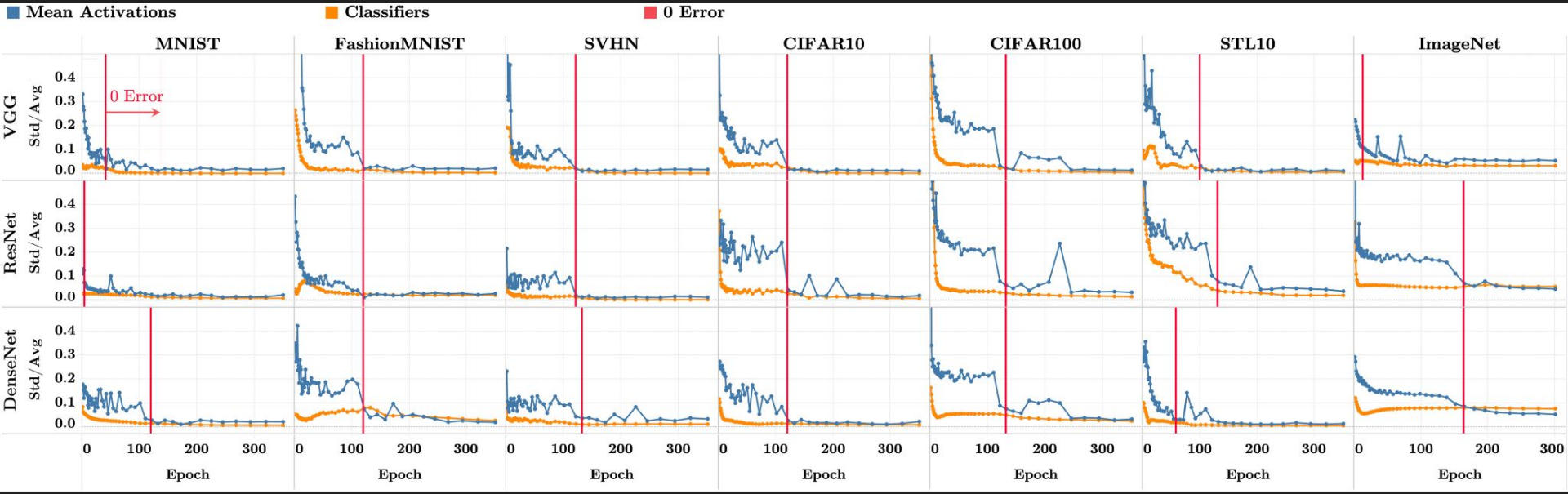
$$\left\| \frac{\boldsymbol{W}^\top}{\|\boldsymbol{W}\|_F} - \frac{\dot{\boldsymbol{M}}}{\|\dot{\boldsymbol{M}}\|_F} \right\|_F \to 0.$$

$$\dot{\boldsymbol{M}} = [\boldsymbol{\mu}_c - \boldsymbol{\mu}_G, c = 1, \ldots, C] \in \mathbb{R}^{p \times C}$$

(NC4): Simplification to Nearest Class Decision:

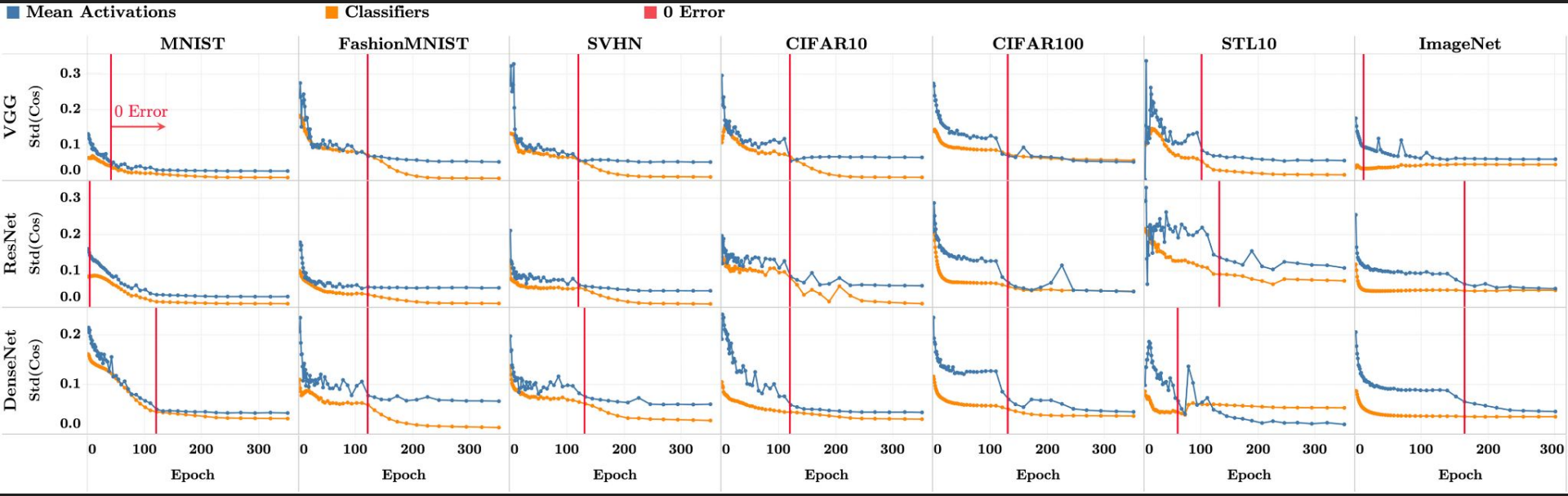
$$\arg\max_{c'} \langle \boldsymbol{w}_{c'}, \boldsymbol{h} \rangle + b_{c'} \to \arg\min_{c'} \|\boldsymbol{h} - \boldsymbol{\mu}_{c'}\|_2$$

Figure 2: Means and classifiers become equinorm



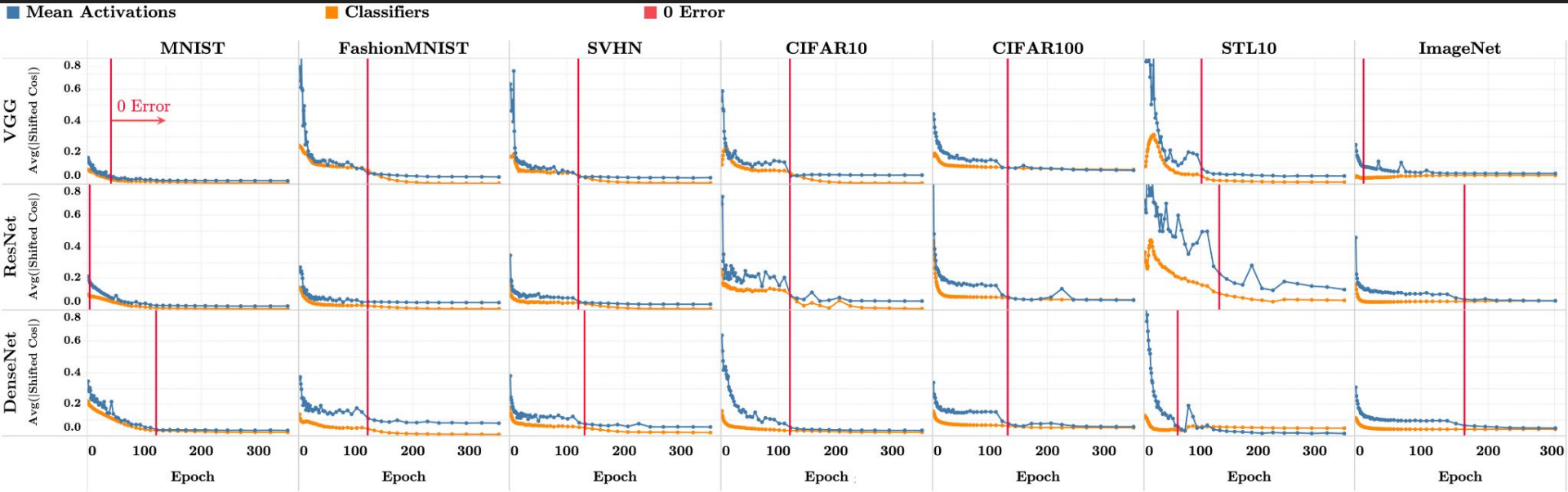
Train class means become equinorm. *The formatting and technical details are as described in Section 3. In each array cell, the vertical axis shows the coefficient of variation of the centered class-mean norms as well as the network classifiers norms. In particular, the blue lines show* $\mathrm{Std}_c\left(\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2\right) / \mathrm{Avg}_c\left(\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2\right)$ *where* $\{\boldsymbol{\mu}_c\}$ *are the class means of the last-layer activations of the training data and* $\boldsymbol{\mu}_G$ *is the corresponding train global mean; the orange lines show* $\mathrm{Std}_c\left(\|\boldsymbol{w}_c\|_2\right) / \mathrm{Avg}_c\left(\|\boldsymbol{w}_c\|_2\right)$ *where* $\boldsymbol{w}_c$ *is the last-layer classifier of the* c*th class. As training progresses, the coefficients of variation of both class means and classifiers decrease.*

Figures 3,4: Means and classifiers become maximally equiangular



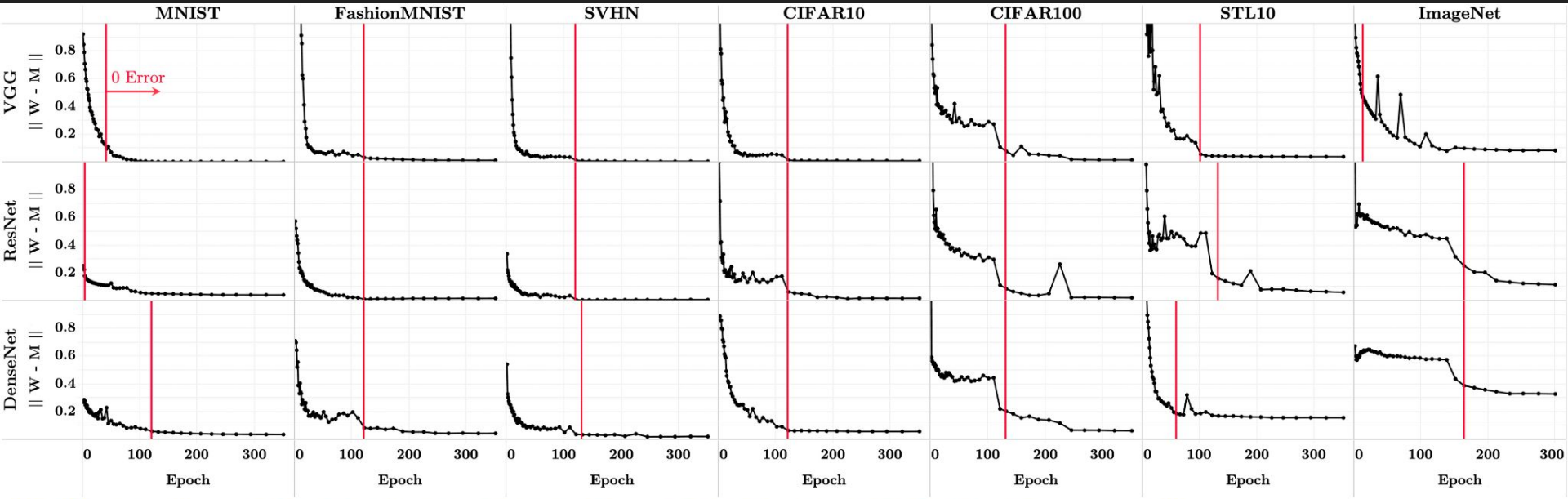
Classifiers and train class means approach equiangularity. *The formatting and technical details are as described in Section 3. In each array cell, the vertical axis shows the SD of the cosines between pairs of centered class means and classifiers across all distinct pairs of classes* c *and* c'. *Mathematically, denote* $\cos_{\mu}(c,c') = \langle \mu_c - \mu_G, \mu_{c'} - \mu_G \rangle / (\|\mu_c - \mu_G\|_2 \|\mu_{c'} - \mu_G\|_2$ *and* $\cos_{w}(c,c') = \langle w_c, w_{c'} \rangle / (\|w_c\|_2 \|w_{c'}\|_2)$ *where* $\{w_c\}_{c=1}^{C}$, $\{\mu_c\}_{c=1}^{C}$, *and* μ_G *are as in Fig. 2. We measure* $\mathrm{Std}_{c,c' \neq c}(\cos_{\mu}(c,c'))$ *(blue) and* $\mathrm{Std}_{c,c' \neq c}(\cos_{w}(c,c'))$ *(orange). As training progresses, the SDs of the cosines approach zero, indicating equiangularity.*

Figures 3, 4: Means and classifiers become maximally equiangular



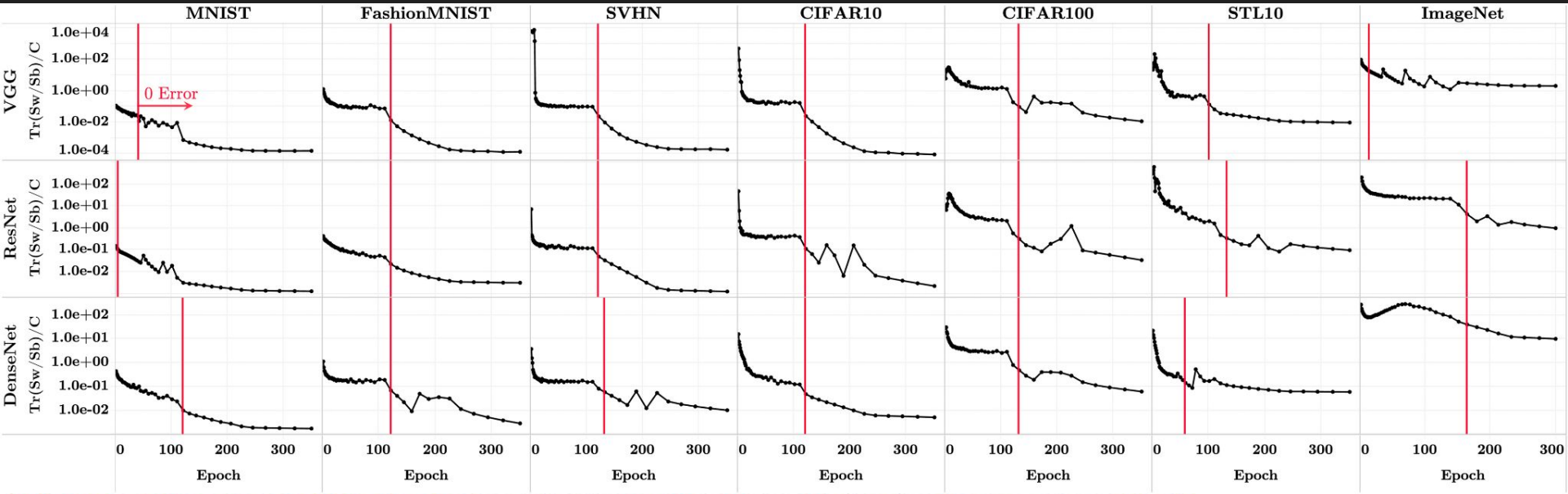
Classifiers and train class means approach maximal-angle equiangularity. *The formatting and technical details are as described in Section 3. We plot in the vertical axis of each cell the quantities* $\mathbf{Avg}_{c,c'}\left|\cos_{\mu}(c,c') + 1/(C-1)\right|$ *(blue) and* $\mathbf{Avg}_{c,c'}\left|\cos_{w}(c,c') + 1/(C-1)\right|$ *(orange), where* $\cos_{\mu}(c,c')$ *and* $\cos_{w}(c,c')$ *are as in Fig. 3. As training progresses, the convergence of these values to zero implies that all cosines converge to* $-1/(C-1)$. *This corresponds to the maximum separation possible for globally centered, equiangular vectors.*

Figure 5: Means and classifiers become self-dual



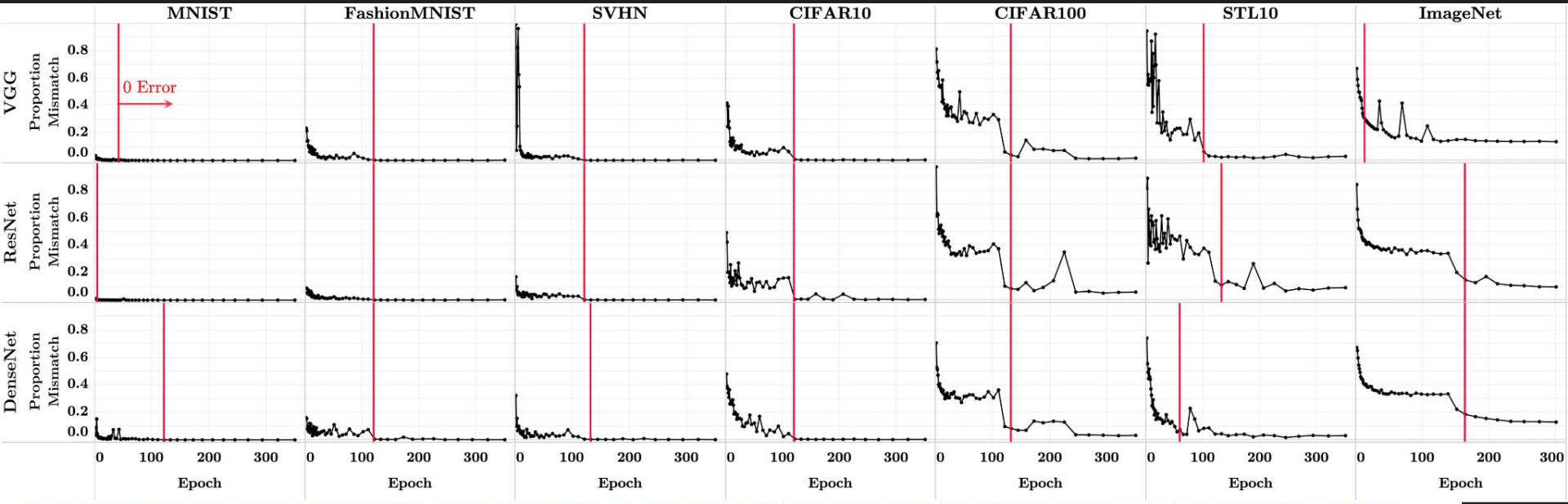
Classifier converges to train class means. *The formatting and technical details are as described in Section 3. In the vertical axis of each cell, we measure the distance between the classifiers and the centered class means, both rescaled to unit norm. Mathematically, denote* $\tilde{\boldsymbol{M}} = \dot{\boldsymbol{M}} / \left\| \dot{\boldsymbol{M}} \right\|_F$ *where* $\dot{\boldsymbol{M}} = [\boldsymbol{\mu}_c - \boldsymbol{\mu}_G : c = 1, \ldots, C] \in \mathbb{R}^{p \times C}$ *is the matrix whose columns consist of the centered train class means; denote* $\tilde{\boldsymbol{W}} = \boldsymbol{W} / \left\| \boldsymbol{W} \right\|_F$ *where* $\boldsymbol{W} \in \mathbb{R}^{C \times p}$ *is the last-layer classifier of the network. We plot the quantity* $\left\| \tilde{\boldsymbol{W}}^{\top} - \tilde{\boldsymbol{M}} \right\|_F^2$ *on the vertical axis. This value decreases as a function of training, indicating that the network classifier and the centered-means matrices become proportional to each other (self-duality).*

Figure 6: Train within-class covariance collapses



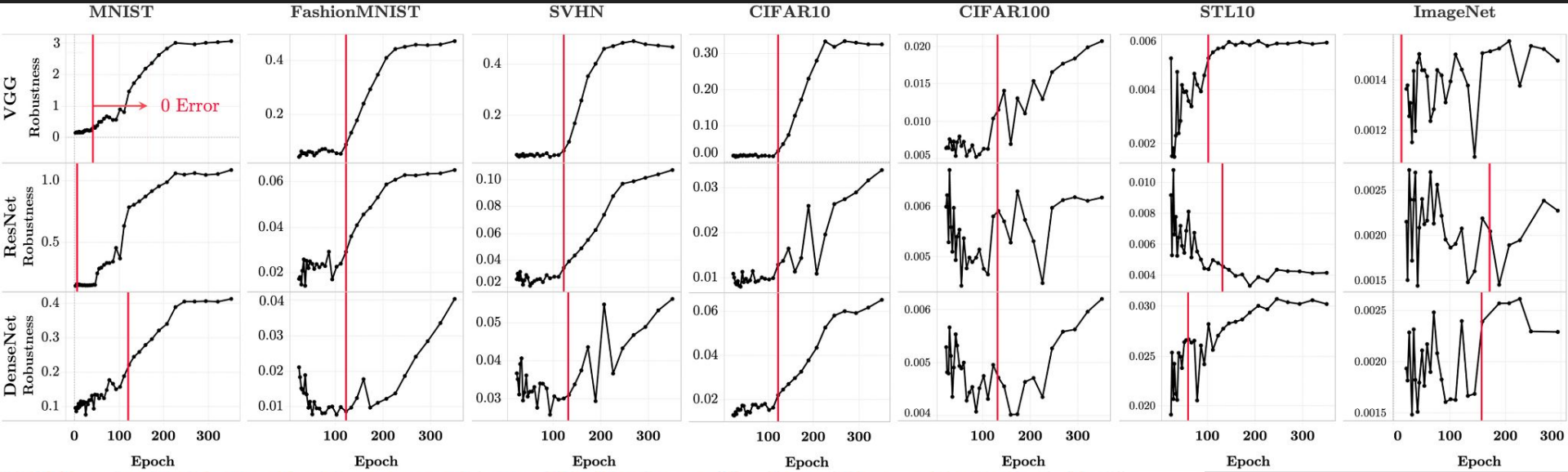
Training within-class variation collapses. *The formatting and technical details are as described in Section 3. In each array cell, the vertical axis (log scaled) shows the magnitude of the between-class covariance compared with the within-class covariance of the train activations. Mathematically, this is represented by* $\mathrm{Tr}\left(\Sigma_W \Sigma_B^{\dagger}/C\right)$ *where* $\mathrm{Tr}\left(\cdot\right)$ *is the trace operator,* Σ_W *is the within-class covariance of the last-layer activations of the training data,* Σ_B *is the corresponding between-class covariance,* C *is the total number of classes, and* $[\cdot]^{\dagger}$ *is Moore–Penrose pseudoinverse. This value decreases as a function of training—indicating collapse of within-class variation.*

Figure 7: Classifier approaches NCC



Classifier behavior approaches that of NCC. *The formatting and technical details are as described in Section 3. In each array cell, we plot the proportion of examples (vertical axis) in the testing set where network classifier disagrees with the result that would have been obtained by choosing* $\arg\min_c \|\boldsymbol{h} - \boldsymbol{\mu}_c\|_2$ *where* $\boldsymbol{h}$ *is a last-layer test activation, and* $\{\boldsymbol{\mu}_c\}_{c=1}^{C}$ *are the class means of the last-layer train activations. As training progresses, the disagreement tends to zero, showing the classifier's behavioral simplification to the nearest train class-mean decision rule.*

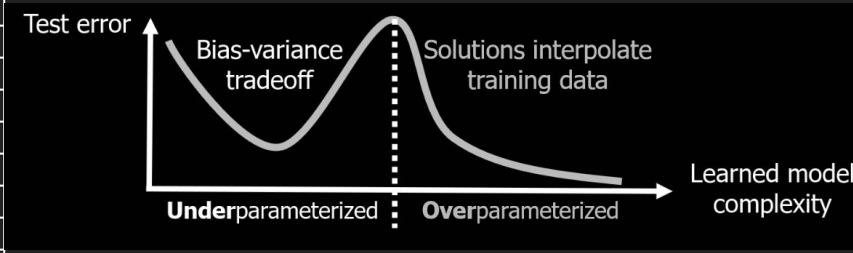
Fig 8: TPT improves robustness



Training beyond zero error improves adversarial robustness. *The formatting and technical details are as described in Section 3. For each dataset and network, we sample without replacement 100 test images —constructing for each an adversarial example using the DeepFool method proposed in ref. 23. In each array cell, we plot on the vertical axis the robustness measure,* $\mathbf{Ave}_i \|\boldsymbol{r}(\boldsymbol{x}_i)\|_2 / \|\boldsymbol{x}_i\|_2$, *from the same paper—where* $\boldsymbol{r}(\boldsymbol{x}_i)$ *is the minimal perturbation required to change the class predicted by the classifier, for a given input image* $\boldsymbol{x}_i$. *As training progresses, larger* $\boldsymbol{x}_i$. *As training progresses, larger perturbations are required to fool the deepnet. Across all array cells, the median improvement in the robustness measure in the last epoch over the first epoch achieving zero training error is 0.0252; the mean improvement is 0.2452.*

Table 1: TPT improves test error

Dataset and network	Test accuracy at zero error	Test accuracy at last epoch
MNIST		
VGG	99.40	99.56
ResNet	99.32	99.71
DenseNet	99.65	99.70
FashionMNIST		
VGG	92.92	93.31
ResNet	93.29	93.64
DenseNet	94.18	94.35
SVHN		
VGG	93.82	94.53
ResNet	94.64	95.70
DenseNet	95.87	95.93
CIFAR10		
VGG	87.85	88.65
ResNet	88.72	89.44
DenseNet	91.14	91.19
CIFAR100		
VGG	63.03	63.85
ResNet	66.19	66.21
DenseNet	77.19	76.56
STL10		
VGG	65.15	68.00
ResNet	69.99	70.24
DenseNet	67.79	70.81
ImageNet		
VGG	47.26	50.12
ResNet	65.41	64.45
DenseNet	65.04	62.38



Discussion

Fig2: Training Progress -> Variation

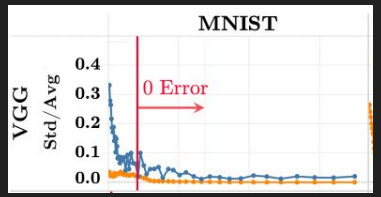


Fig3: All pairs of class means form equal-sized angles

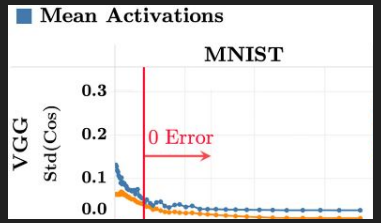


Fig4: Angles Converge to max angle

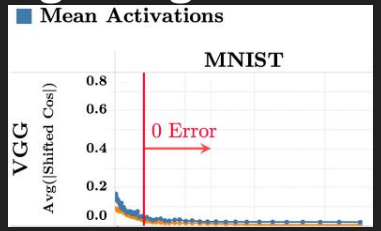
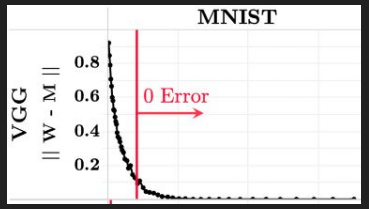


Fig5: Converge up to same ETF (Dual)



g6: Normalized Variation Converges

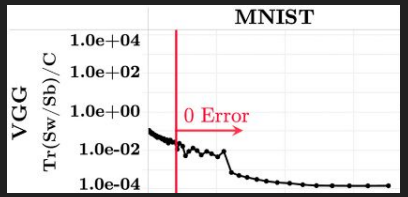
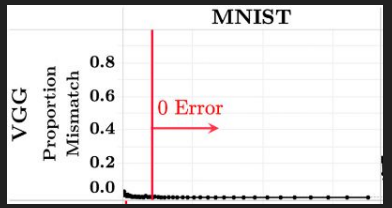
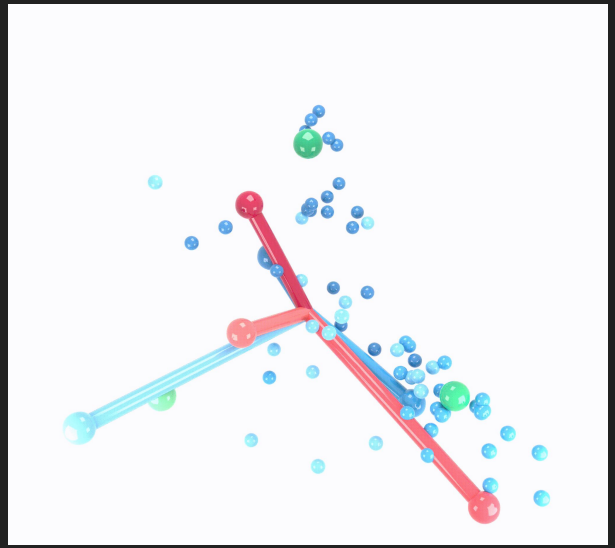


Fig7: Approach to NCC



Discussion

- Recurring themes:
  - NC continues after zero error
  - This explains TPT - induces significant changes in underlying network structure



Relation to Previous Work: Web and Lowe

- "The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis"

Proposition 1 (**Section 3 in ref. 28**). *Fix the deepnet architecture and the underlying tuning parameters* $\boldsymbol{\theta}$, *so that the activations* $\boldsymbol{h_\theta}(\boldsymbol{x})$ *involve no training and so that only the classifier weights* $\boldsymbol{W}$ *and biases* $\boldsymbol{b}$ *need to be trained. Maintain the same definitions—*$\boldsymbol{\Sigma_T}$, $\boldsymbol{\mu_c}$, $\boldsymbol{\mu_G}$, *etc.—as in Section 2. Adopting the mean squared error loss in place of the cross-entropy loss, the optimal classifier weights and biases are given by*

$$\begin{aligned} \boldsymbol{W} &= \tfrac{1}{C}\dot{\boldsymbol{M}}^\top \boldsymbol{\Sigma}_T^\dagger \\ \boldsymbol{b} &= \tfrac{1}{C}1_C - \tfrac{1}{C}\dot{\boldsymbol{M}}^\top \boldsymbol{\Sigma}_T^\dagger \boldsymbol{\mu}_G, \end{aligned} \qquad [6]$$

where † *denotes the Moore–Penrose pseudoinverse,*

$$\dot{\boldsymbol{M}} = [\boldsymbol{\mu}_c - \boldsymbol{\mu}_G, c = 1, \ldots, C] \in \mathbb{R}^{p \times C}$$

Theorem 1

Theorem 1 [**Proposition 1 +** $\overrightarrow{(\mathrm{NC1}-2)}$ **Imply** $\overrightarrow{(\mathrm{NC3}-4)}$]. *Adopt the framework and assumptions of Proposition* 1, *as well as the end state implied by* (NC1) *and* (NC2) [*i.e.*, $\overrightarrow{(\mathrm{NC1})}$ *and* $\overrightarrow{(\mathrm{NC2})}$]. *The Webb–Lowe classifier* [**6**], *in this setting, has the additional properties* $\overrightarrow{(\mathrm{NC3})}$ *and* $\overrightarrow{(\mathrm{NC4})}$.

Sketch of Proof:

By prop 1

$$\begin{aligned} \boldsymbol{W} &= \tfrac{1}{C}\dot{\boldsymbol{M}}^\top \boldsymbol{\Sigma}_B^\dagger \\ \boldsymbol{b} &= \tfrac{1}{C}1_C - \tfrac{1}{C}\dot{\boldsymbol{M}}^\top \boldsymbol{\Sigma}_B^\dagger \boldsymbol{\mu}_G. \end{aligned}$$

Inter-class variance rewrite as $\boldsymbol{\Sigma}_B = \frac{1}{C}\dot{\boldsymbol{M}}\dot{\boldsymbol{M}}^\top$

Thus

$$\begin{aligned} \boldsymbol{W} &= \dot{\boldsymbol{M}}^\top \left(\dot{\boldsymbol{M}}\dot{\boldsymbol{M}}^\top\right)^\dagger = \dot{\boldsymbol{M}}^\dagger \\ \boldsymbol{b} &= \tfrac{1}{C}1_C - \dot{\boldsymbol{M}}^\top \left(\dot{\boldsymbol{M}}\dot{\boldsymbol{M}}^\top\right)^\dagger \boldsymbol{\mu}_G = \tfrac{1}{C}1_C - \dot{\boldsymbol{M}}^\dagger \boldsymbol{\mu}_G. \end{aligned}$$

M has C-1 nonzero SVs so $\dot{\boldsymbol{M}}^\dagger = \alpha \dot{\boldsymbol{M}}^\top$

Therefore, we have

$$\begin{aligned} \boldsymbol{W} &= \alpha \dot{\boldsymbol{M}}^\top \\ \boldsymbol{b} &= \frac{1}{C}1_C - \alpha \dot{\boldsymbol{M}}^\top \boldsymbol{\mu}_G; \end{aligned}$$

This is NC3 up to rescaling.

To get NC4, the class predicted by this is

$$= \arg\min_{c'} \|\boldsymbol{h} - \boldsymbol{\mu}_{c'}\|_2.$$

Relation to Previous Work: Soudry et al.

- Constrained form of trained classifier
- With fixed activations
- Inductive bias is reason for success of deep learning
- According to this result inductive bias is even more constraining
- Modern deepnets produce linear classifiers constrained to simplex ETFs

Proposition 2 (**Theorem 7 in ref. 29**). *Let* $\mathbb{R}^{NC\times p}$ *denote the vector space spanning all last-layer activation datasets,* $\boldsymbol{H} = (\boldsymbol{h}_{i,c} : 1 \le i \le N, 1 \le c \le C)$*, and let* $\mathscr{H}$ *denote the measurable subset of* $\mathbb{R}^{NC\times p}$ *consisting of linearly separable datasets: that is, consisting of datasets* $\boldsymbol{H}$ *where, for some linear classifiers* $\{\boldsymbol{w}_c\}_{c=1}^{C}$ *(possibly depending on dataset* $\boldsymbol{H}$*), separability holds:*

$$\langle \boldsymbol{w}_c - \boldsymbol{w}'_c, \boldsymbol{h}_{i,c} \rangle \ge 1, \qquad \boldsymbol{h}_{i,c} \in \boldsymbol{H}.$$

For (Lebesgue) almost every dataset $\boldsymbol{H} \in \mathscr{H}$*, gradient descent minimizing the cross-entropy loss, as a function of the classifier weights, tends to a limit. This limit is identical to the solution of the max-margin classifier problem:*

$$\min_{\{\boldsymbol{w}_c\}_{c=1}^{C}} \sum_{c=1}^{C} \|\boldsymbol{w}_c\|_2^2 \text{ s.t. } \forall i, c, c' \neq c : \langle \boldsymbol{w}_c - \boldsymbol{w}_{c'}, \boldsymbol{h}_{i,c} \rangle \ge 1. \qquad [9]$$

Theorem 2

Theorem 2 (Proposition 2 + $\overrightarrow{(\mathrm{NC1}-2)}$ Imply $\overrightarrow{(\mathrm{NC3}-4)}$). *Adopt the framework and assumptions of Proposition 2, as well as the end state implied by* (NC1) *and* (NC2) [*i.e.*, $\overrightarrow{(\mathrm{NC1})}$ *and* $\overrightarrow{(\mathrm{NC2})}$]. *The Soudry et al.* (29) *classifier* [**9**], *in this setting, has the additional properties* $\overrightarrow{(\mathrm{NC3})}$ *and* $\overrightarrow{(\mathrm{NC4})}$.

I am not going through the proof since it's long. But the sketch is that we do some linear optimization and algebra to show that the max margin classifier also has NC3 and NC4 properties.

This is similar to result of theorem 1, but now for Cross Entropy Loss

Deriving ETF Emergence

Abstract Feature Engineer chooses activations to minimize classification error

Assume we are given an observation $\boldsymbol{h} = \boldsymbol{\mu}_\gamma + \boldsymbol{z} \in \mathbb{R}^C$, where $\boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{I}\right)$ and $\gamma \sim \mathrm{Unif}\{1, \ldots, C\}$ is an unknown class index, distributed independently from $\boldsymbol{z}$. Our goal is to recover γ from $\boldsymbol{h}$, with as small an error rate as possible. We constrain ourselves to use a linear classifier, $\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}$, with weights $\boldsymbol{W} = [\boldsymbol{w}_c : c = 1, \ldots, c] \in \mathbb{R}^{C \times C}$ and biases $\boldsymbol{b} = (b_c) \in \mathbb{R}^C$; our decision rule is

$$\hat{\gamma}(\boldsymbol{h}) = \hat{\gamma}(\boldsymbol{h}; \boldsymbol{W}, \boldsymbol{b}) = \arg\max_c \langle \boldsymbol{w}_c, \boldsymbol{h} \rangle + b_c.$$

Our task is to design the classifier $\boldsymbol{W}$ and bias $\boldsymbol{b}$, as well as a matrix $\boldsymbol{M} = [\boldsymbol{\mu}_c : c = 1, \ldots, C] \in \mathbb{R}^{C \times C}$, subject to the norm constraints $\|\boldsymbol{\mu}_c\|_2 \leq 1$ for all c.

We can rewrite this as an optimal coding problem. We design a codebook and decoder that allows optimal retrieval of class identity from noisy information.

And we are interested in the large deviations error exponent

$$\beta(\boldsymbol{M}, \boldsymbol{W}, \boldsymbol{b}) = -\lim_{\sigma \to 0} \sigma^2 \log P_\sigma \{\hat{\gamma}(\boldsymbol{h}) \neq \gamma\}$$

Theorem 3 - Simplex ETF Emergence

We engineer collection of code works to optimize the vanishingly small theoretical misclassification, we obtain the standard simplex ETF or a rotation of it

Theorem 3. *Under the model assumptions just given in subsections A, B, and C, the optimal error exponent is*

$$\begin{aligned}\beta^\star &= \max_{\boldsymbol{M},\boldsymbol{W},\boldsymbol{b}} \beta(\boldsymbol{M},\boldsymbol{W},\boldsymbol{b}) \;\; \text{s.t.} \;\; \|\boldsymbol{\mu}_c\|_2 \le 1 \;\forall c \\ &= \tfrac{C}{C-1}\cdot\tfrac{1}{4},\end{aligned}$$

where the maximum is over $C \times C$ *matrices* $\boldsymbol{M}$ *with at most unit-norm columns, and over* $C \times C$ *matrices* $\boldsymbol{W}$ *and* $C \times 1$ *vectors* $\boldsymbol{b}$.

Moreover, denote $\boldsymbol{M}^\star = \sqrt{\frac{C}{C-1}}\left(\boldsymbol{I} - \frac{1}{C}11^\top\right)$ (i.e., $\boldsymbol{M}^\star$ is the standard simplex ETF). The optimal error exponent is precisely achieved by $\boldsymbol{M}^\star$:

$$\beta(\boldsymbol{M}^\star, \boldsymbol{M}^\star, \boldsymbol{0}) = \beta^\star.$$

Conclusion

- This is a study in TPT, in which a process called Neural Collapse emerges
- The four interconnected phenomena of Neural Collapse are distinct but related.
- Last layer classifier of a trained deepnet exhibits clear mathematical structure: ***INDUCTIVE BIAS***
- Last-layer features collapse to an ETF
- Last-layer classifier is equivalent to NCC decision rule
- Standard work flow for empirical deep learning:
  - Series of arbitrary steps that helped win prediction challenge contest
  - Careful analysis was never the point!
  - TPT benefits today standard deep learning training by leading to Neural Collapse
- Doors open to new formal insights