Deep Neural Collapse:[1]

4 empirical metrics that identify population risk minimizers[2]

J. Setpal

November 13, 2024

[1]Kothapalli. [TMLR 2023]
[2]E, Wojtowytsch. [PMLR 2022]

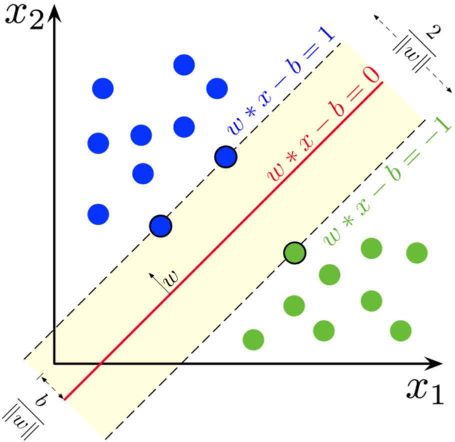
Outline

Outline

1. Background & Intuition
2. Conditions for Neural Collapse
3. Optimality of Neural Collapse

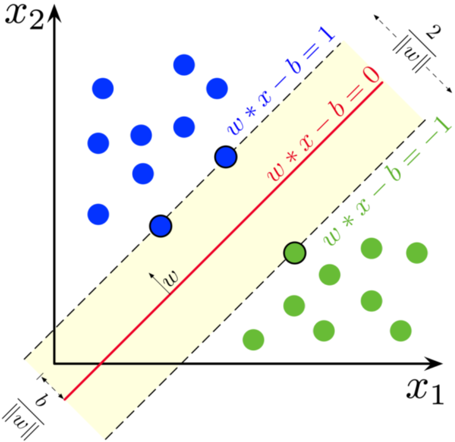
Primer – Support Vector Machines (SVMs)

We start with a linear SVM:

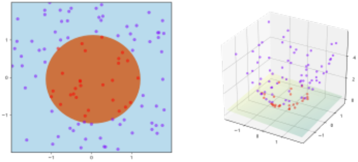


Primer – Support Vector Machines (SVMs)

We start with a linear SVM:

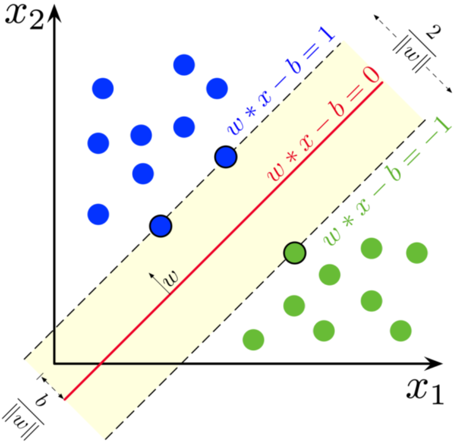


An approach to obtain a non-linear decision boundary is to learn a hyperplane in higher-dimensions:

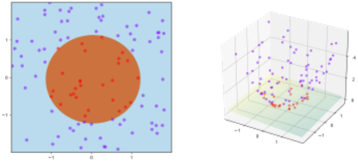


Primer – Support Vector Machines (SVMs)

We start with a linear SVM:



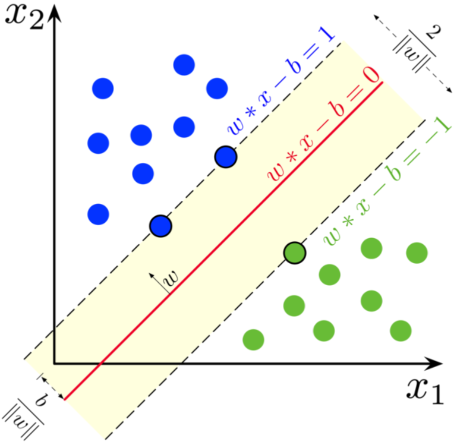
An approach to obtain a non-linear decision boundary is to learn a hyperplane in higher-dimensions:



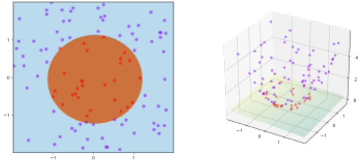
"Lazy" approaches to kernel choices include *polynomial* / *RBF* kernels.

Primer – Support Vector Machines (SVMs)

We start with a linear SVM:



An approach to obtain a non-linear decision boundary is to learn a hyperplane in higher-dimensions:



"Lazy" approaches to kernel choices include *polynomial* / *RBF* kernels.

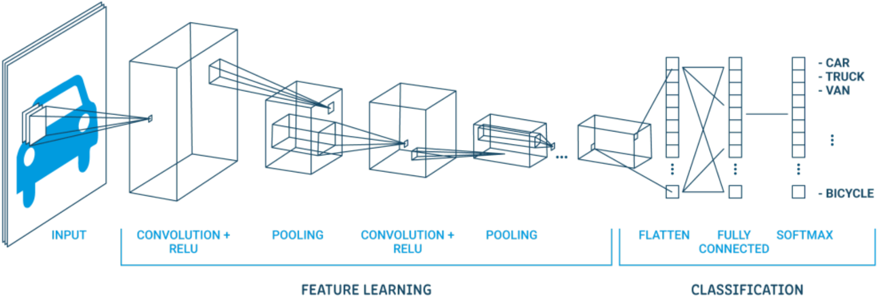
The "laziest" kernel of all is a **deep neural network**.

Neural Networks are Incredibly Overparameterized

Our study today is constrained to classifiers.

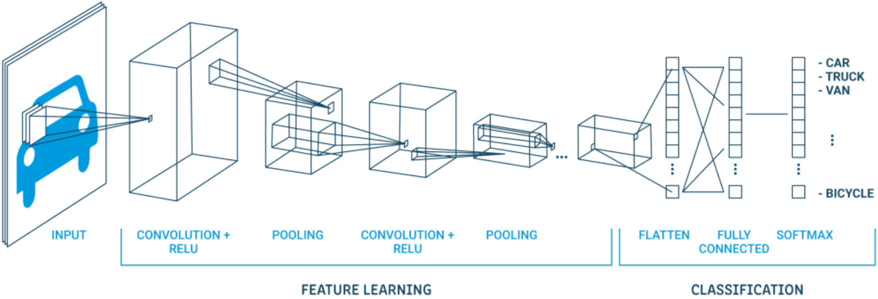
Neural Networks are Incredibly Overparameterized

Our study today is constrained to classifiers. WLOG, we can constrain our study to **image classifiers**.



Neural Networks are Incredibly Overparameterized

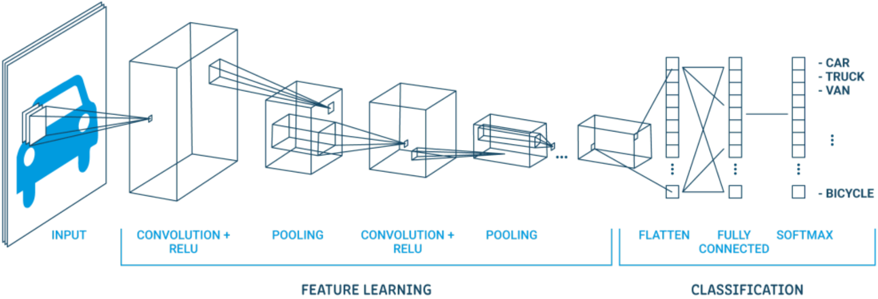
Our study today is constrained to classifiers. WLOG, we can constrain our study to **image classifiers**.



Traditional Learning: $n \geq d$; $W \in \mathbb{R}^d$, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$

Neural Networks are Incredibly Overparameterized

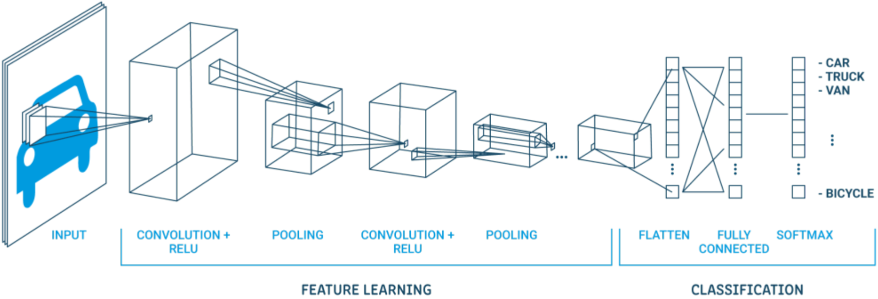
Our study today is constrained to classifiers. WLOG, we can constrain our study to **image classifiers**.



Traditional Learning: $n \geq d$; $W \in \mathbb{R}^d$, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
Overparameterized Learning: $d \geq n$

Neural Networks are Incredibly Overparameterized

Our study today is constrained to classifiers. WLOG, we can constrain our study to **image classifiers**.



Traditional Learning: $n \geq d$; $W \in \mathbb{R}^d$, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
Overparameterized Learning: $d \geq n$

Q: Why does overparameterized learning generalize?

Outline

What is Deep Neural Collapse (DNC)?

Deep Neural Collapse is a phenomenon describing *rigidity* in the feature representation(s) of the final layer(s) of *overtrained* Deep Neural Networks.

What is Deep Neural Collapse (DNC)?

Deep Neural Collapse is a phenomenon describing *rigidity* in the feature representation(s) of the final layer(s) of *overtrained* Deep Neural Networks.

$\mathbf{Q}_1$: What does *overtrained* mean?
$\mathbf{A}_1$: When a sufficiently expressive network h trained to minimize $\mathcal{L}(S_n)$ satisfies $h(x_i) = y_i \ \forall i$, it reaches the **Terminal Point of Training**. When trained beyond this point, the model is <u>overtrained</u>.

What is Deep Neural Collapse (DNC)?

Deep Neural Collapse is a phenomenon describing *rigidity* in the feature representation(s) of the final layer(s) of *overtrained* Deep Neural Networks.

$\mathbf{Q}_1$: What does *overtrained* mean?
$\mathbf{A}_1$: When a sufficiently expressive network h trained to minimize $\mathcal{L}(S_n)$ satisfies $h(x_i) = y_i \ \forall i$, it reaches the **Terminal Point of Training**. When trained beyond this point, the model is overtrained.

$\mathbf{Q}_2$: What does *rigidity* mean?
$\mathbf{A}_2$: We quantify *rigidity* by 4 key metrics, which iff satisifed, implies DNC.

What is Deep Neural Collapse (DNC)?

Deep Neural Collapse is a phenomenon describing *rigidity* in the feature representation(s) of the final layer(s) of *overtrained* Deep Neural Networks.

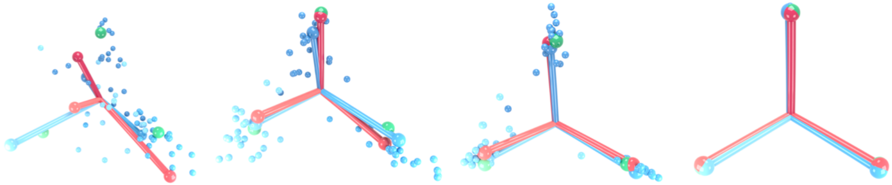
$\mathbf{Q}_1$: What does *overtrained* mean?
$\mathbf{A}_1$: When a sufficiently expressive network h trained to minimize $\mathcal{L}(S_n)$ satisfies $h(x_i) = y_i \ \forall i$, it reaches the **Terminal Point of Training**. When trained beyond this point, the model is overtrained.

$\mathbf{Q}_2$: What does *rigidity* mean?
$\mathbf{A}_2$: We quantify *rigidity* by 4 key metrics, which iff satisifed, implies DNC.

$\mathbf{Q}_{2_a}$: What are the 4 key metrics?
$\mathbf{A}_{2_a}$: We'll talk about this next.

$NC1$ – Collapse of Variability (1/2)

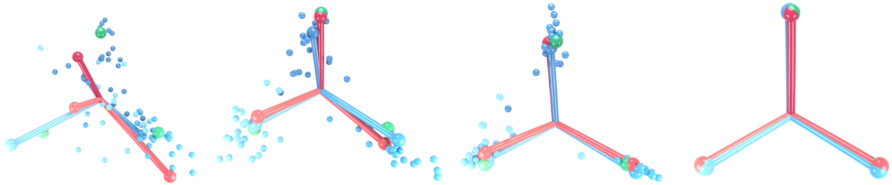
At a high level, the structure of the penultimate layer collapses towards:



Evolution of penultimate layer outputs on VGG13 trained on CIFAR10.

$NC1$ – Collapse of Variability (1/2)

At a high level, the structure of the penultimate layer collapses towards:

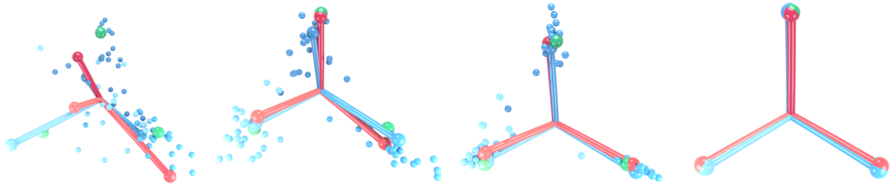


Evolution of penultimate layer outputs on VGG13 trained on CIFAR10.

For all classes $k \in [K]$, datapoints $i \in [n]$ within a class, & penultimate feature vector $f(k, i)$,

$NC1$ – Collapse of Variability (1/2)

At a high level, the structure of the penultimate layer collapses towards:



Evolution of penultimate layer outputs on VGG13 trained on CIFAR10.

For all classes $k \in [K]$, datapoints $i \in [n]$ within a class, & penultimate feature vector $f(k, i)$, we have class-specific & global means:

$$\boldsymbol{\mu}_k = \frac{1}{n} \sum_{i=1}^{n} f(k, i) \tag{1}$$

$$\boldsymbol{\mu}_G = \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{\mu}_k \tag{2}$$

$NC1$ – Collapse of Variability (2/2)

We can use them to calculate *intra* and *inter*-class differences:

$$\mathrm{Cov}_W = \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}((f(k,i)-\boldsymbol{\mu}_k)(f(k,i)-\boldsymbol{\mu}_k)^T) \in \mathbb{R}^{m\times m} \tag{3}$$

$$\mathrm{Cov}_B = \frac{1}{K}\sum_{k=1}^{K}((\boldsymbol{\mu}_k-\boldsymbol{\mu}_G)(\boldsymbol{\mu}_k-\boldsymbol{\mu}_G)^T) \in \mathbb{R}^{m\times m} \tag{4}$$

$NC1$ – Collapse of Variability (2/2)

We can use them to calculate *intra* and *inter*-class differences:

$$\mathrm{Cov}_W = \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}((f(k,i)-\boldsymbol{\mu}_k)(f(k,i)-\boldsymbol{\mu}_k)^T) \in \mathbb{R}^{m\times m} \tag{3}$$

$$\mathrm{Cov}_B = \frac{1}{K}\sum_{k=1}^{K}((\boldsymbol{\mu}_k-\boldsymbol{\mu}_G)(\boldsymbol{\mu}_k-\boldsymbol{\mu}_G)^T) \in \mathbb{R}^{m\times m} \tag{4}$$

Which we combine to measure overall **variability collapse**:

$$NC1 := \frac{1}{K}\,\mathrm{Tr}\left(\mathrm{Cov}_W\,\mathrm{Cov}_B^{\dagger}\right) \tag{5}$$

Aside: Psuedoinverses

The **inverse** of a matrix A is defined s.t. it satisfies the following condition:

$$A, B, I \in \mathbb{R}^{d \times d} \text{ s.t. } AB = BA = I_d; \; B := A^{-1}, \; A := B^{-1} \tag{6}$$

Aside: Psuedoinverses

The **inverse** of a matrix A is defined s.t. it satisfies the following condition:

$$A, B, I \in \mathbb{R}^{d \times d} \text{ s.t. } AB = BA = I_d; \; B := A^{-1}, \; A := B^{-1} \tag{6}$$

What about when $X \in \mathbb{R}^{n \times m}$?

Aside: Psuedoinverses

The **inverse** of a matrix A is defined s.t. it satisfies the following condition:

$$A, B, I \in \mathbb{R}^{d\times d} \text{ s.t. } AB = BA = I_d;\ B := A^{-1},\ A := B^{-1} \tag{6}$$

What about when $X \in \mathbb{R}^{n\times m}$? A **psuedoinverse** is a *generalized inverse*, which instead satisfies the following four conditions:

$$XX^{-1}X = X \tag{7}$$

$$X^{-1}XX^{-1} = X^{-1} \tag{8}$$

$$(XX^{-1})^* = XX^{-1} \tag{9}$$

$$X^{-1}X^* = X^{-1}X \tag{10}$$

Where X^* is the conjugate transpose of X.

Aside: Psuedoinverses

The **inverse** of a matrix A is defined s.t. it satisfies the following condition:

$$A, B, I \in \mathbb{R}^{d \times d} \text{ s.t. } AB = BA = I_d;\ B := A^{-1},\ A := B^{-1} \tag{6}$$

What about when $X \in \mathbb{R}^{n \times m}$? A **psuedoinverse** is a *generalized inverse*, which instead satisfies the following four conditions:

$$XX^{-1}X = X \tag{7}$$

$$X^{-1}XX^{-1} = X^{-1} \tag{8}$$

$$(XX^{-1})^* = XX^{-1} \tag{9}$$

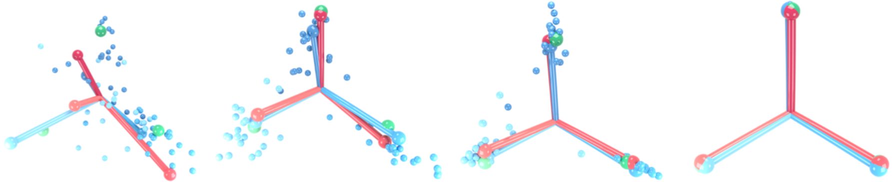
$$X^{-1}X^* = X^{-1}X \tag{10}$$

Where X^* is the conjugate transpose of X.

Implication: We can compute correlation b/w general matrix dimensions.

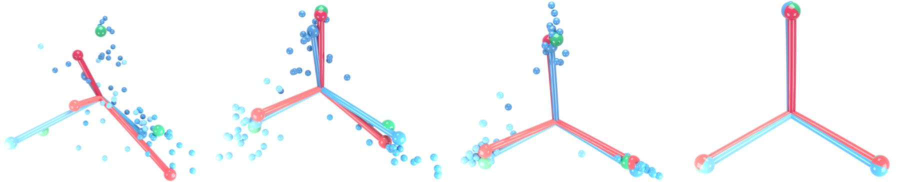
$NC2$ – The Simplex ETF (1/2)

This time, we can on focus the **structure** of the class means:



$NC2$ – The Simplex ETF (1/2)

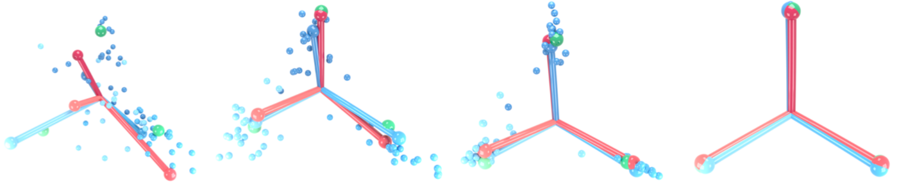
This time, we can on focus the **structure** of the class means:



A useful analogy is *VSEPR*[3] from Chemistry.

[3] I sincerely apologize for making this reference.

This time, we can on focus the **structure** of the class means:

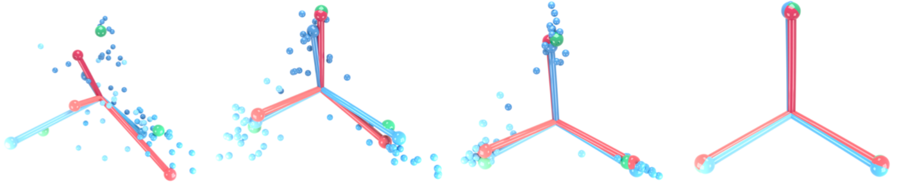


A useful analogy is *VSEPR*[3] from Chemistry. Each class (atom) repels the other creating a **simplex equiangular tight frame** (simplex ETF).

[3] I sincerely apologize for making this reference.

$NC2$ – The Simplex ETF (1/2)

This time, we can on focus the **structure** of the class means:



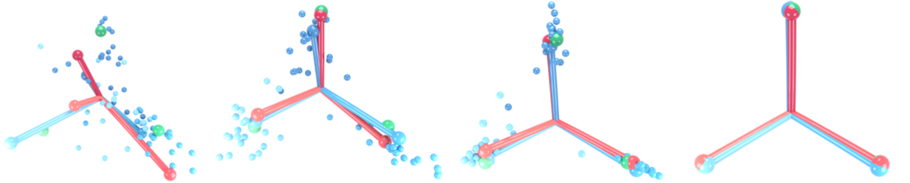
A useful analogy is *VSEPR*[3] from Chemistry. Each class (atom) repels the other creating a **simplex equiangular tight frame** (simplex ETF).

- **Simplex** is the simplest polytope (object with flat sides).

[3] I sincerely apologize for making this reference.

$NC2$ – The Simplex ETF (1/2)

This time, we can on focus the **structure** of the class means:



A useful analogy is *VSEPR*[3] from Chemistry. Each class (atom) repels the other creating a **simplex equiangular tight frame** (simplex ETF).

- **Simplex** is the simplest polytope (object with flat sides).
- **Equiangular Tight Frame** is a matrix $M \in \mathbb{R}^{K \times m}$ s.t.

$$|\langle \boldsymbol{m}_j, \boldsymbol{m}_k \rangle| = \alpha \; \exists \alpha \geq 0 \; \forall j, k \text{ s.t. } j \neq k \tag{11}$$

$$MM^T = \sqrt{\frac{C}{C-1}} \left(I_C - \frac{1}{C} \mathbb{1}_{C \times C} \right) \tag{12}$$

Satisfying equiangular and tight respectively.

[3] I sincerely apologize for making this reference.

$NC2$ – The Simplex ETF (2/2)

We can use this to define $NC2$. Given re-centered class means $\{\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\}_{k \in [K]}$, they are **equidistant** if:

$$\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\|_2 = \|\boldsymbol{\mu}_{k'} - \boldsymbol{\mu}_G\|_2 \ \forall k, k' \in [K] \tag{13}$$

$NC2$ – The Simplex ETF (2/2)

We can use this to define $NC2$. Given re-centered class means $\{\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\}_{k \in [K]}$, they are **equidistant** if:

$$\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\|_2 = \|\boldsymbol{\mu}_{k'} - \boldsymbol{\mu}_G\|_2 \;\forall k, k' \in [K] \tag{13}$$

We then normalize each feature vector to create our simplex ETF:

$$M = \text{Concat}\left(\left\{\frac{\boldsymbol{\mu}_k - \boldsymbol{\mu}_G}{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\|_2} \in \mathbb{R}^m\right\}^{[K]}\right) \in \mathbb{R}^{K \times m} \tag{14}$$

$NC2$ – The Simplex ETF (2/2)

We can use this to define $NC2$. Given re-centered class means $\{\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\}_{k\in[K]}$, they are **equidistant** if:

$$\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\|_2 = \|\boldsymbol{\mu}_{k'} - \boldsymbol{\mu}_G\|_2 \ \forall k, k' \in [K] \tag{13}$$

We then normalize each feature vector to create our simplex ETF:

$$M = \text{Concat}\left(\left\{\frac{\boldsymbol{\mu}_k - \boldsymbol{\mu}_G}{\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_G\|_2} \in \mathbb{R}^m\right\}^{[K]}\right) \in \mathbb{R}^{K\times m} \tag{14}$$

M is now compared to it's distance from the simplex ETF:

$$NC2 := \left\| \underbrace{\frac{MM^T}{\|MM^T\|_F}}_{\text{feature vector as a simplex}} - \underbrace{\frac{1}{\sqrt{K-1}}\left(I_K - \frac{\mathbb{1}_{K\times K}}{K}\right)}_{\text{canonical simplex}} \right\|_F \tag{15}$$

Setting up our second metric.

$NC3$ – Self-Dual Alignment

The final layer's weights $W \in \mathbb{R}^{K \times m}$ align with simplex ETF of M:

$$\frac{A}{\|A\|_F} \propto \frac{M}{\|M\|_F} \tag{16}$$

$NC3$ – Self-Dual Alignment

The final layer's weights $W \in \mathbb{R}^{K \times m}$ align with simplex ETF of M:

$$\frac{A}{\|A\|_F} \propto \frac{M}{\|M\|_F} \tag{16}$$

We can use this to setup the third metric:

$$NC3 := \left\| \underbrace{\frac{AM^T}{\|AM^T\|_F}}_{\equiv \text{ cosine similiarity}} - \underbrace{\frac{1}{\sqrt{K-1}} \left(I_K - \frac{\mathbb{1}_{K \times K}}{K} \right)}_{\text{canonical simplex}} \right\|_F \tag{17}$$

$NC4$ – Secretly k-NN

Finally, we observe that for x_{n+1}, the classification result $\equiv$ k-NN rule:

$$\arg\max \hat{y}_{n+1} = \arg\min_{k \in [K]} \|f(x_{n+1}) - \boldsymbol{\mu}_k\|_2 \tag{18}$$

$NC4$ – Secretly k-NN

Finally, we observe that for x_{n+1}, the classification result $\equiv$ k-NN rule:

$$\arg\max \hat{y}_{n+1} = \arg\min_{k \in [K]} \|f(x_{n+1}) - \boldsymbol{\mu}_k\|_2 \tag{18}$$

Which we can use to setup our final metric:

$$NC4 : \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathbb{1} \left[\arg\max \hat{y}_i \neq \arg\min_{k \in [K]} \|f(x_i) - \boldsymbol{\mu}_k\|_2 \right] \tag{19}$$

$NC4$ – Secretly k-NN

Finally, we observe that for x_{n+1}, the classification result $\equiv$ k-NN rule:

$$\arg\max \hat{y}_{n+1} = \arg\min_{k \in [K]} \|f(x_{n+1}) - \boldsymbol{\mu}_k\|_2 \tag{18}$$

Which we can use to setup our final metric:

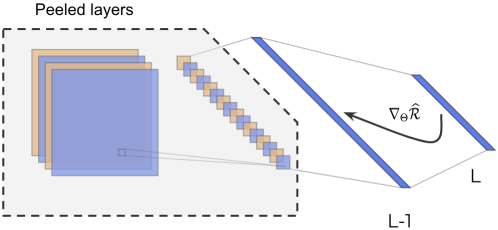
$$NC4 : \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathbb{1} \left[\arg\max \hat{y}_i \neq \arg\min_{k \in [K]} \|f(x_i) - \boldsymbol{\mu}_k\|_2 \right] \tag{19}$$

If each of the 4 previous metrics $\to 0$, the network is considered **collapsed**.

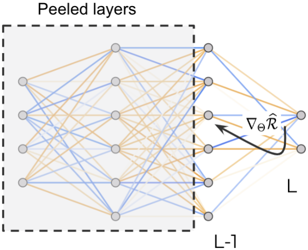
Outline

Modelling Neural Collapse

Unconstrained Features Model: To maintain the expressivity of $\mathcal{H}$, properties NC is studied by treating $f_i, i \in \{1, \ldots, L-1\}$ as *free optimization parameters*:



(a) CNN → UFM

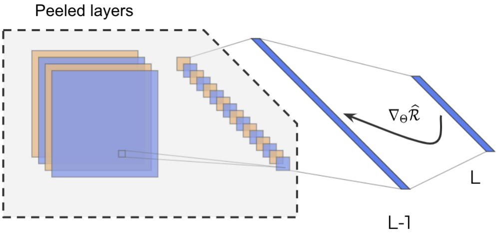


(b) MLP → UFM

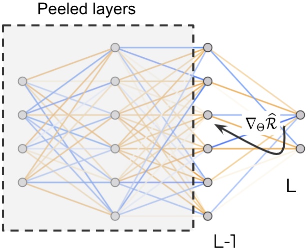
$$h_L(x) = A \underbrace{f_{1:L-1}(x)}_{NC} + b \qquad (20)$$

Modelling Neural Collapse

Unconstrained Features Model: To maintain the expressivity of $\mathcal{H}$, properties NC is studied by treating $f_i, i \in \{1, \ldots, L-1\}$ as *free optimization parameters*:



(a) CNN → UFM



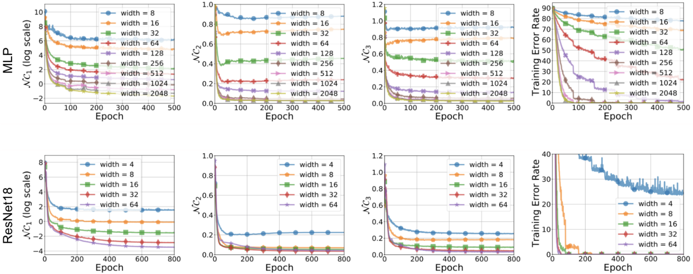
(b) MLP → UFM

$$h_L(x) = A \underbrace{f_{1:L-1}(x)}_{NC} + b \tag{20}$$

We can further discuss the ideal values of A, f, b and training dynamics (regularization, loss functions, normalization) that encourage it.

Do We Even Want This? – Data Independence

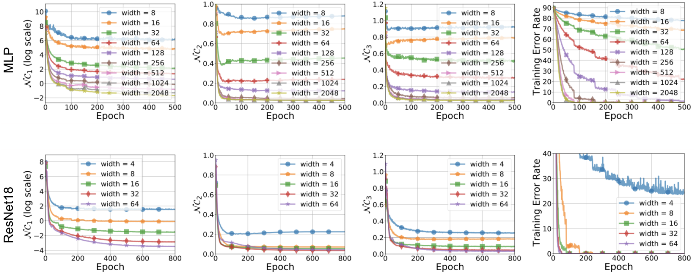
Here's what the metric convergence plots look like, with *random labels*.



(a) NC1 (b) NC2 (c) NC3 (d) Training error

Do We Even Want This? – Data Independence

Here's what the metric convergence plots look like, with *random labels*.

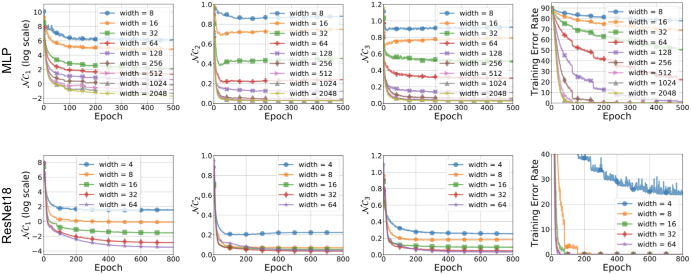


(a) NC1 (b) NC2 (c) NC3 (d) Training error

Q: Do we even want this?

Do We Even Want This? – Data Independence

Here's what the metric convergence plots look like, with *random labels*.



(a) NC1 (b) NC2 (c) NC3 (d) Training error

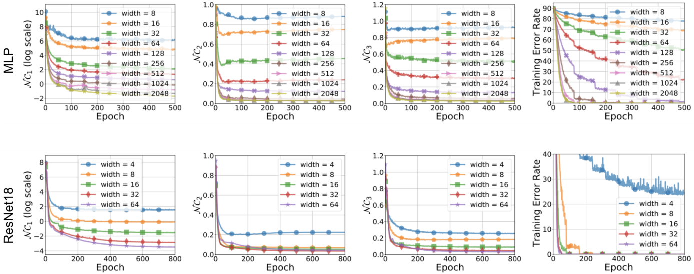
Q: Do we even want this?

A: Yes. Here's some reasons why:

1. **OOD**: $\{\text{NC1}, \text{NC2}, \text{NC3}\} \gg 0$ imply unconfident predictions.

Do We Even Want This? – Data Independence

Here's what the metric convergence plots look like, with *random labels*.



(a) NC1 (b) NC2 (c) NC3 (d) Training error

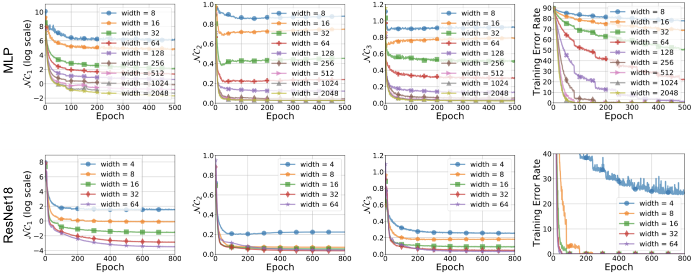
Q: Do we even want this?

A: Yes. Here's some reasons why:

1. **OOD**: $\{\text{NC1}, \text{NC2}, \text{NC3}\} \gg 0$ imply unconfident predictions.
2. **Forced ETF**: The final layer can be a fixed as a simplex.

Do We Even Want This? – Data Independence

Here's what the metric convergence plots look like, with *random labels*.



(a) NC1 (b) NC2 (c) NC3 (d) Training error

Q: Do we even want this?

A: Yes. Here's some reasons why:

1. **OOD**: {NC1, NC2, NC3} $\gg 0$ imply unconfident predictions.
2. **Forced ETF**: The final layer can be a fixed as a simplex.
3. **Data dependent explanation**: AGOP induces NC.

Optimality of NC (Softmax-CE Loss)

Softmax CE is defined element-wise as follows:

$$\Phi(z)_j = -\log \frac{\exp(z_j)}{\sum_{i=1}^{k} \exp(z_i)} = \log \sum_{i=1}^{k} \exp(z_i) + \log \exp(z_j) \tag{21}$$

is convex $\forall j \in \{1, \dots, k\}$.

Optimality of NC (Softmax-CE Loss)

Softmax CE is defined element-wise as follows:

$$\Phi(z)_j = -\log \frac{\exp(z_j)}{\sum_{i=1}^k \exp(z_i)} = \log \sum_{i=1}^k \exp(z_i) + \log \exp(z_j) \tag{21}$$

is convex $\forall j \in \{1, \ldots, k\}$.

$$z_k := \frac{1}{n} \int_{C_k} h(x) \mathbb{P}(dx) \tag{22}$$

$$\int_{C_k} \Phi_k(h(x)) \mathbb{P}(dx) \geq \int_{C_k} \Phi_k(z_k) \mathbb{P}(dx) \tag{23}$$

Optimality of NC (Softmax-CE Loss)

Softmax CE is defined element-wise as follows:

$$\Phi(z)_j = -\log \frac{\exp(z_j)}{\sum_{i=1}^{k} \exp(z_i)} = \log \sum_{i=1}^{k} \exp(z_i) + \log \exp(z_j) \tag{21}$$

is convex $\forall j \in \{1, \ldots, k\}$.

$$z_k := \frac{1}{n} \int_{C_k} h(x) \mathbb{P}(dx) \tag{22}$$

$$\int_{C_k} \Phi_k(h(x)) \mathbb{P}(dx) \geq \int_{C_k} \Phi_k(z_k) \mathbb{P}(dx) \tag{23}$$

Consequently, we have that:

$$\mathcal{R}(\bar{h}) \leq \min_{h \in \mathcal{H}} \mathcal{R}(h) \tag{24}$$

Establishing NC describing the optimal geometry within the final layer for *population* risk minimization.

Thank you!

Have an awesome rest of your day!

Slides: `https://cs.purdue.edu/homes/jsetpal/slides/dnc.pdf`