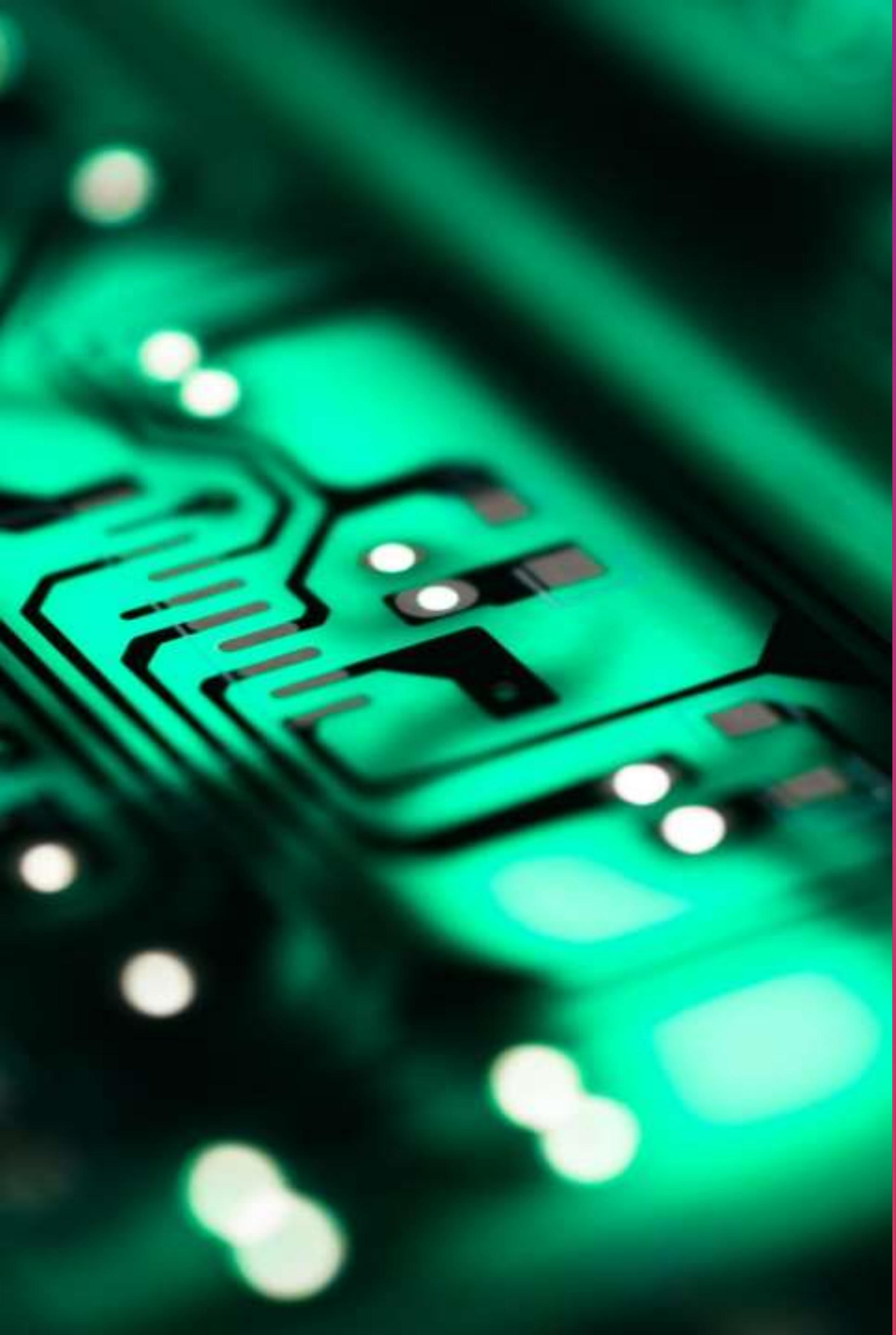


METHODS FOR COMPARING REPRESENTATIONS OF ARTIFICIAL NEURAL NETWORKS

MARCO ZULLICH, PHD STUDENT @ DEPT. OF ENGINEERING AND ARCHITECTURE

UNIVERSITY OF TRIESTE

WHY COMPARING REPRESENTATIONS OF ARTIFICIAL NEURAL NETWORKS?

- The dynamics behind the training process of SGD are still vaguely understood, as are the generalization capability and some properties of hidden representations
- It may be interesting, for instance, to compare layers during training
- On the other hand, given two identically structured ANNs trained on the same dataset with two different random seeds, do they learn similar representations or the representations are different even if the final performance is the same?
- Can we establish connections between the representations learned by ANNs and by their biological counterpart?

COMPARING LAYERS OF ARTIFICIAL NEURAL NETWORKS (ANNs)

- What defines two ANNs as «similar»?
  - Their parameters (weights)
  - Their outputs
  - Their neurons

  …*other ideas?*

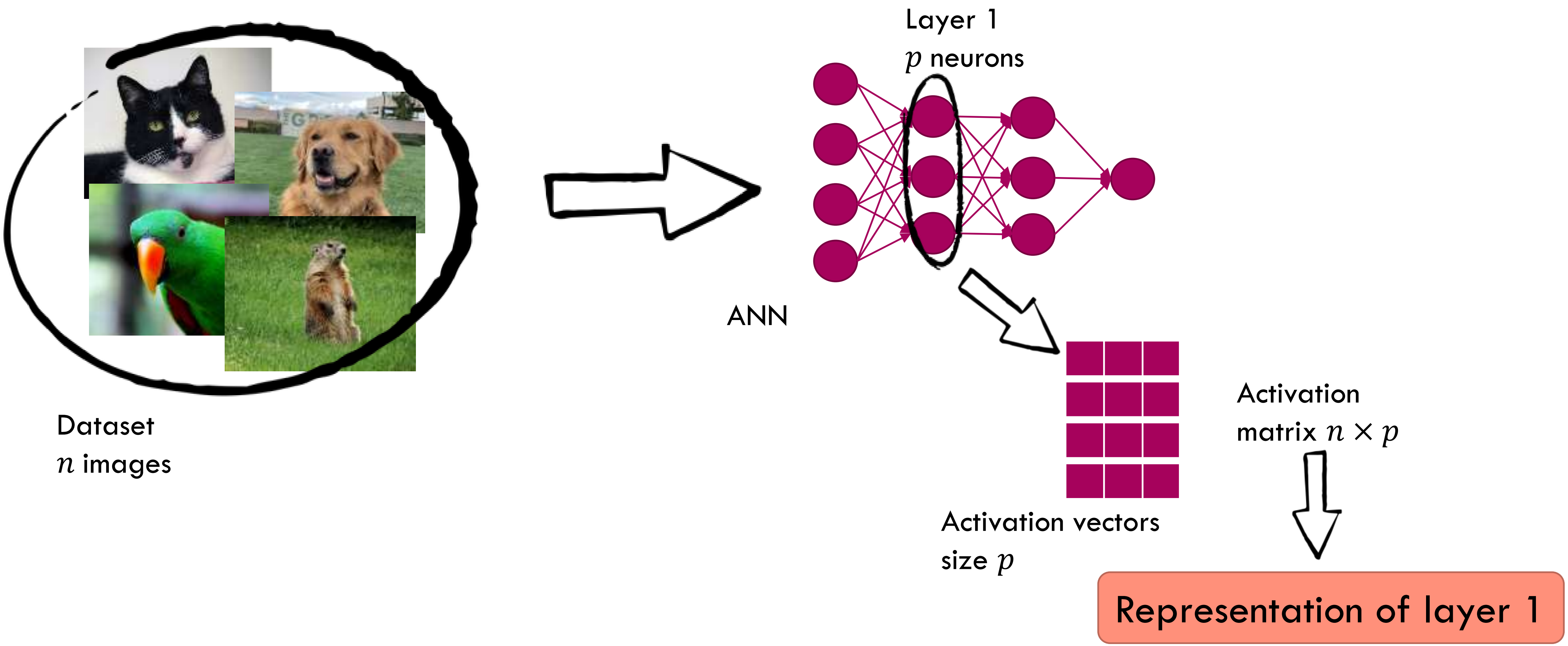
COMPARING ANNs BY THEIR NEURONS

- What are their values given the input?
- How can we approximate their probability density (given the data manifold) over $\mathbb{R}$?
  - $p(a_{ij}|x)$
- «How does the neuron respond to the data manifold»?

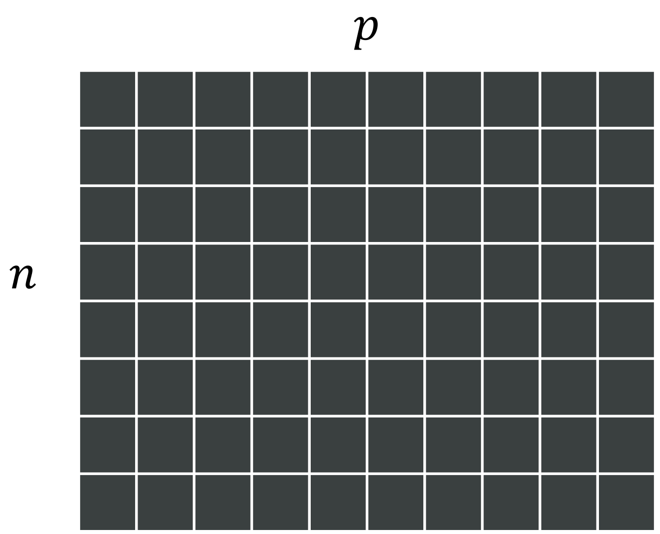
$\rightarrow$ evaluate the network over a limited, yet «large enough» dataset of points and collect the neurons activations.

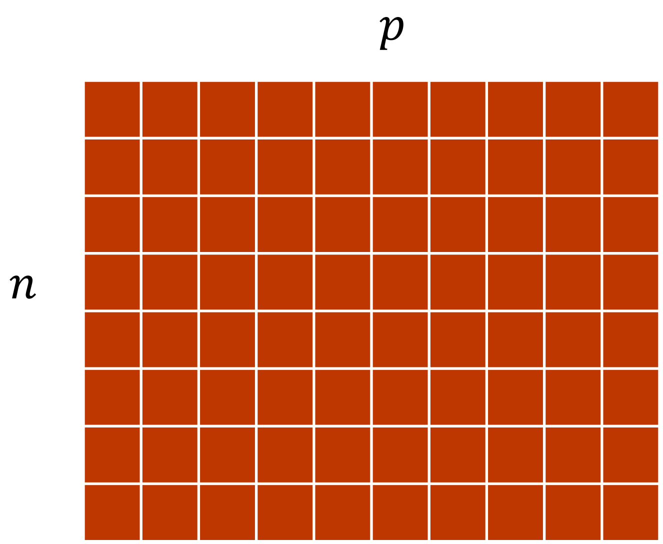
«Monte-Carlo approximation»

A BIT MORE FORMALLY



COMPARING TWO SINGLE-DIMENSIONAL LAYERS





What techinques do you know for comparing two matrices of the same size?

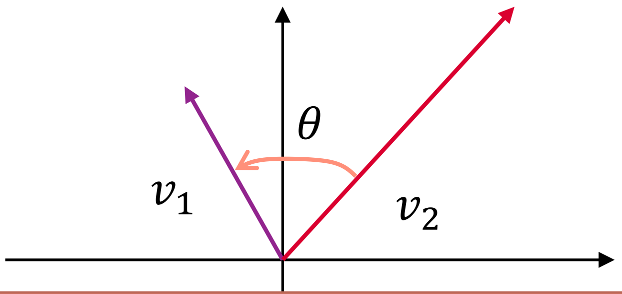
COMPARING MATRICES WITH COSINE SIMILARITY

- The cosine similarity is a **similarity metric** between two vectors lying in the same space
- Is equivalent to the **cosine of the angle** between the two

$$v_1^T v_2 = \|v_1\| \cdot \|v_2\| \cdot \cos(\theta)$$

hence

$$\cos(\theta) = \frac{v_1^T v_2}{\|v_1\| \cdot \|v_2\|} = \frac{\langle v_1, v_2 \rangle}{\|v_1\| \cdot \|v_2\|}$$



What is wrong with cosine similarity when comparing two flattened layers?

THE REQUISITE OF ROTATION INVARIANCE

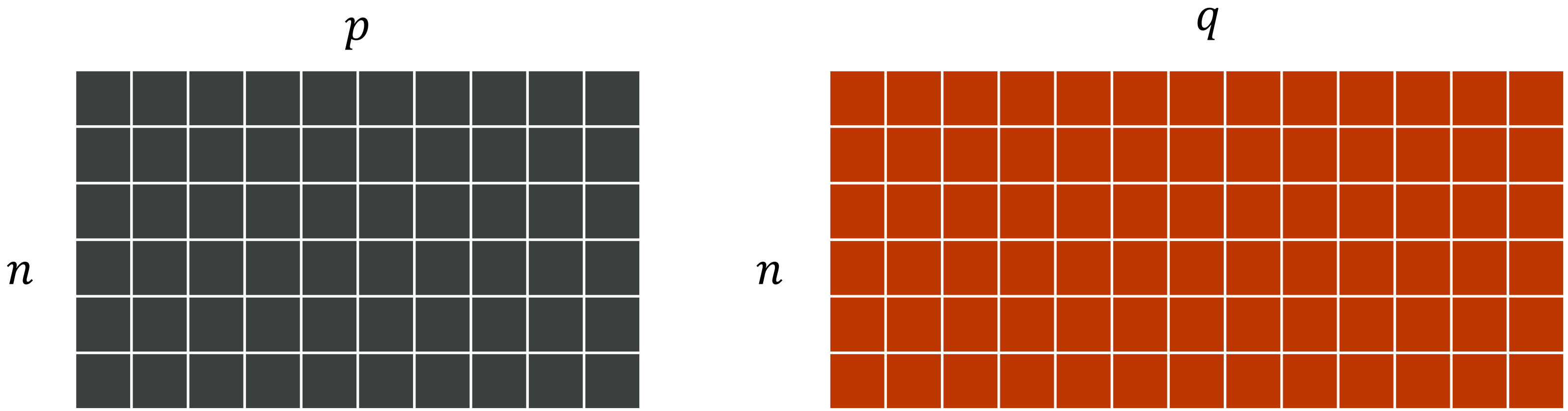
- One of the properties of the neural network is that, by carefully changing the order of the neurons (and their corresponding weights) in the hidden layers, we can obtain two networks behaving identically
- Ideally we would like a similarity metric to be invariant to that property
- If we view the **neurons as directions** within the **space of representations**

  the permutation can be seen as a **rotation** within this space

  and the requirement becomes **rotational invariance.**

…AN ADDITIONAL REQUISITE

- In addition to the rotational invariance requisite, we would also like our metric to be more flexible and adapt to generic situations in which the two layers have different sizes
- In pratical terms…



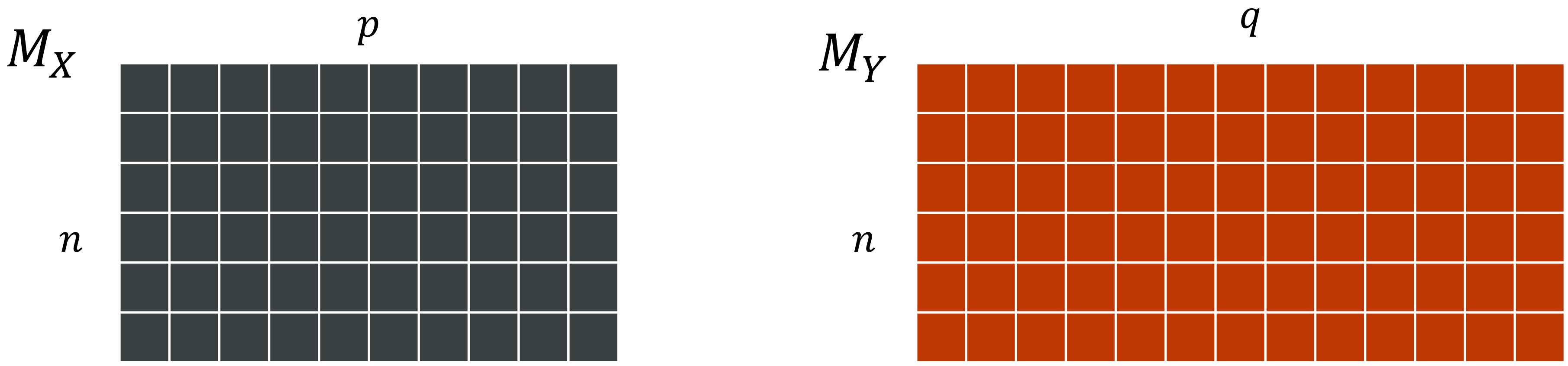
$$p \lesseqqgtr q$$

INTRODUCING CANONICAL CORRELATION ANALYSIS (CCA)

- CCA is a technique introduced in the ‘30s by the statistician Harold Hotelling
- Is a technique for **establishing connections between two generic sets of continuous variables** called **phenomena**
- $X = (X_1, \dots, X_p); \quad Y = (Y_1, \dots, Y_q), \quad p \gtreqless q$
- We operate by constructing a so-called **view** of these two **phenomena**
    - We consider n statistical units (*individuals*)
    - We evaluate these individuals over our phenomena

CCA {2}

- Now, we may store these views in two matrices, M_X, M_Y



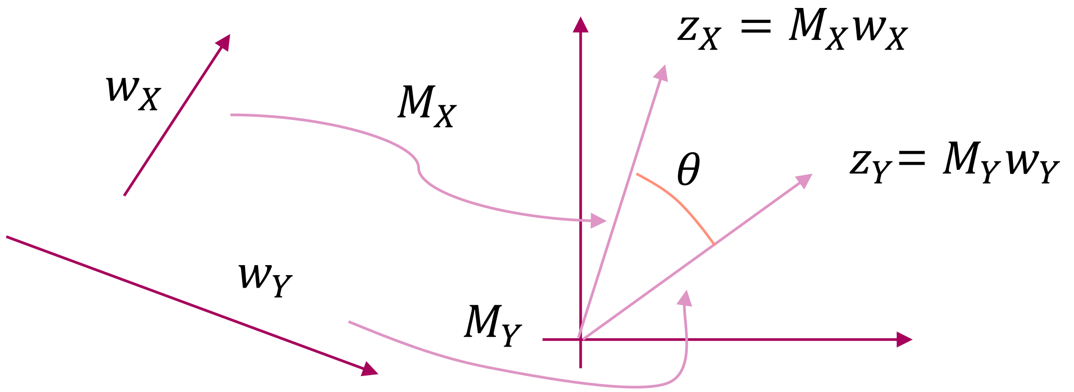
CCA {3}

- CCA acts by applying a **linear transformation** to two **unknown vectors** $w_X \in \mathbb{R}^p, w_y \in \mathbb{R}^q$
- The linear transformation is the one **implied** by M_X and M_Y
- $M_X w_X = z_X \in \mathbb{R}^n; \quad M_Y w_Y = z_Y \in \mathbb{R}^n$

The constraint over w_X, w_y is that the corresponding z_X, z_y

1) are unit vectors
2) have maximum Pearson correlation.

Geometrically, Pearson correaltion = cosine of enclosing angle θ (*)



Does this ring a bell?

(*) for 0-mean random variables/representations

CCA {4}

- Call $\rho \triangleq \cos\theta \rightarrow$ CANONICAL CORRELATION (CC)
- We now wish to obtain another set $w_X^{(2)}, w_Y^{(2)}$
- Such that the corresponding $z_X^{(2)}, z_Y^{(2)}$ respect all previous properties
- and they're orthogonal to z_X, z_y respectively
- We will have the corresponding $\rho^{(2)} \leq \rho$
- We can find **an iterative method** which produces a **decreasing sequence of CCs** $\left(\rho^{(1)}, \ldots, \rho^{(\min(p,q))}\right)$

CCA {5}

- We can summarize all this in matrix notation
- $W_X \in \mathbb{R}^{p \times \min(p,q)}, Z_X \in \mathbb{R}^{n \times \min(p,q)}$ (analogous for W_Y, Z_Y)
- $M_X W_X = Z_X$ (analogous for W_Y, Z_Y, M_Y)
- The rows of Z_X hold the s.c. **canonical variables**
- Z_X, Z_Y are **orthonormal bases of the space** $\mathbb{R}^n$
- Pearson correlation between $z_X^{(i)}, z_Y^{(i)}$ is maximum
- $\mathrm{P} = \left(\rho^{(i)}\right)_i$ is trivially obtained as the row-wise cosine similarity between Z_X, Z_Y

CCA {6}

But how can we obtain W_X, W_Y?

Σ_{XX}	$\mathrm{VAR}(M_X)$
Σ_{YY}	$\mathrm{VAR}(M_Y)$
Σ_{XY}	$COV(M_X, M_Y)$

Singular Value Decomposition (SVD)

$$\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2} = USV$$

Left singular vectors

$\Sigma_{XX}^{1/2}u_i$ yields the corresponding $w_X^{(i)}$

Right singular vectors

$\Sigma_{YY}^{1/2}v_i$ yields the corresponding $w_Y^{(i)}$

Singular values

s_i corresponds to the canonical correlation $\rho^{(i)}$

MEAN CCA SIMILARITY

$\boldsymbol{\rho}^{(1)}$	$\boldsymbol{\rho}^{(2)}$	$\boldsymbol{\rho}^{(3)}$	…	$\boldsymbol{\rho}^{(r)}$

$r = \min(p, q)$

These coefficient convey an information on **relatedness** between M_X and M_Y.

$$\mathrm{CCA}_{\mathrm{sim}}(M_X, M_Y) \triangleq \frac{\sum_{j=1}^{r} \rho^{(j)}}{r}$$

Mean CCA similarity

RECAP ON CCA

X, Y are our **layers**

M_X, M_Y are our **representations**

Study connections between two phenomena X, Y of **possibly different sizes**

Build views by means of n observations over $X, Y \rightarrow M_X, M_Y$

Linearly project the columns of M_X, M_Y in orthonormal bases of size $\mathbb{R}^n \rightarrow Z_X, Z_Y$

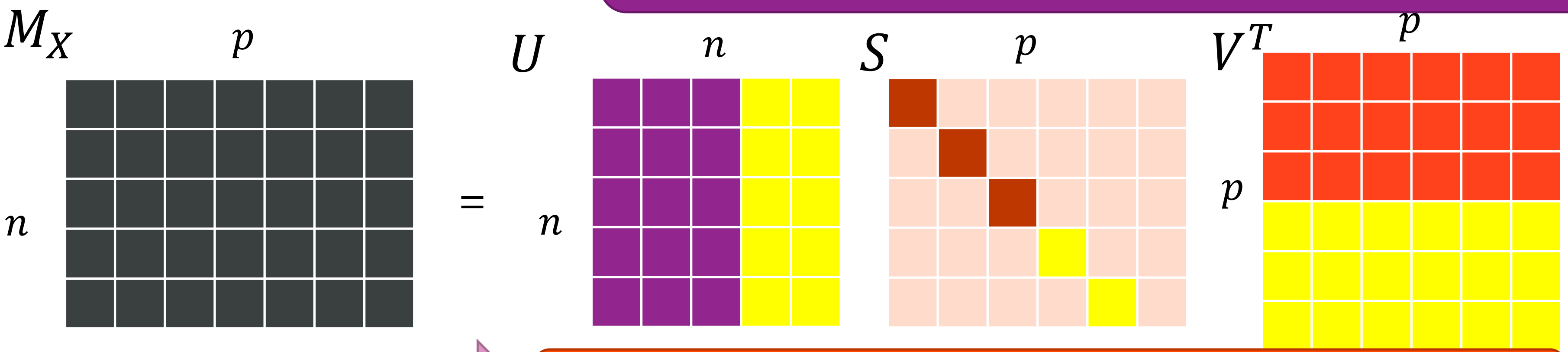
The projection maximizes correlation ρ between rows of these two bases

ρ's can be easily obtained *one-shot* via SVD applied on variance-covariance matrices of M_X, M_Y

SVCCA [1]

- A technique thought explicitly for applying CCA to compare ANN layers
- Assumption: *most of the neurons within a layer contribute close to nothing to the variance of its representation*

Operate SVD for dimensionality reduction



$$s_{jj}^2 = \mathrm{VAR}(M_X)_j$$
$$s_{jj} \geq s_{kk}, j > k$$

Operate CCA and obtain similarity

SIMILARITY OF REPRESENTATIONS USING KERNELS

Linear kernel over a representation M_X

$$L_X = M_X M_X^T \in \mathbb{R}^{n \times n}$$
$$L_Y = M_Y M_Y^T \in \mathbb{R}^{n \times n}$$

To *measure* the dissimilarity between these two matrices, we can use the inner product of their vectorized version

$$\langle \mathrm{vec}(L_X), \mathrm{vec}(L_Y) \rangle = \mathrm{tr}(L_X L_Y)$$
$$= \left\| M_Y^T M_X \right\|_{\mathrm{Fr}}^2$$
$$= (n-1)^2 \left\| \mathrm{COV}(M_X^T, M_Y^T) \right\|^2$$

Note: $\mathrm{COV}(A,B) = \frac{AB^T}{n-1} \Rightarrow \|\mathrm{COV}(A,B)\|^2 = \frac{\left\|AB^T\right\|^2}{(n-1)^2}$

$$\left\| \mathrm{COV}(M_X^T, M_Y^T) \right\|^2 = \frac{\mathrm{tr}(L_X L_Y)}{(n-1)^2}$$

HILBERT SCHMIDT INDEPENDENCE CRITERION (HSIC)

$$\left\|\mathrm{cov}\left(M_X^T, M_Y^T\right)\right\|^2 = \frac{\mathrm{tr}(L_X L_Y)}{(n-1)^2}$$

→ HSIC for linear kernels

$$\mathrm{HSIC}\left(\widetilde{K_X}, \widetilde{K_Y}\right) = \frac{\mathrm{tr}(\widetilde{K_X}\widetilde{K_Y})}{(n-1)^2}$$

$\widetilde{K_X} = K_X(I - n^{-1}\mathbf{1}\mathbf{1}^T)$ kernels centered w.r.t. row and column means

HSIC is a statistic for determining whether two generic sets of variables are independent

$\mathrm{HSIC} \rightarrow 0$ stochastic independence

$\mathrm{HSIC} \rightarrow 1$ stochastic dependence

CENTERED KERNEL ALIGNMENT (CKA)

- Normalization of HSIC

$$\mathrm{CKA}\left(\widetilde{K_X}, \widetilde{K_Y}\right) = \frac{\mathrm{HSIC}\left(\widetilde{K_X}, \widetilde{K_Y}\right)}{\sqrt{\mathrm{HSIC}\left(\widetilde{K_X}, \widetilde{K_X}\right)\mathrm{HSIC}\left(\widetilde{K_Y}, \widetilde{K_Y}\right)}} = \frac{\left\langle \widetilde{K_X}, \widetilde{K_Y} \right\rangle}{\left\|\widetilde{K_X}\right\|\left\|\widetilde{K_Y}\right\|} \in [0,1]$$

- Linear CKA

$$\mathrm{CKA}_{\mathrm{lin}}(M_X, M_Y) = \frac{\left\|M_Y^T M_X\right\|^2}{\left\|M_X^T M_X\right\|\left\|M_Y^T M_Y\right\|} \in [0,1]$$

CHOICE OF KERNELS

- **Radial Basis Function** (RBF) kernel:
- $\kappa(x, y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right)$
- [2] cites no substantial difference between using RBF kernel with $\sigma \in [0.2, 0.6]$ w.r.t. a linear kernel
- On the other hand, [3] refers that CKA with RBF with *very small* sigma may be a better choice for a more accurate similarity metric, but more on that on the next keynote.

ON THE *DESIRABLE* AND *INDESIRABLE* INVARIANCES OF METRICS

- Invariance to orthogonal transformations was already discussed before **Good**
- **Invariance to isotropic scaling** (arbitrary scaling of the features)
  - $\mathrm{SIM}(M_X, M_Y) = \mathrm{SIM}(\alpha M_X, \beta M_y), \quad \alpha, \beta \in \mathbb{R}^+$ **Good**

Invariance to invertible linear transformations **Bad**

This invariance poses problems when $n < p$, as for full-rank matrices A, B, the similarity $SIM(A, C) = SIM(B, C)$ [2]

The training process is sensible w.r.t. invertible linear transforms. Just think of BATCH NORMALIZATION

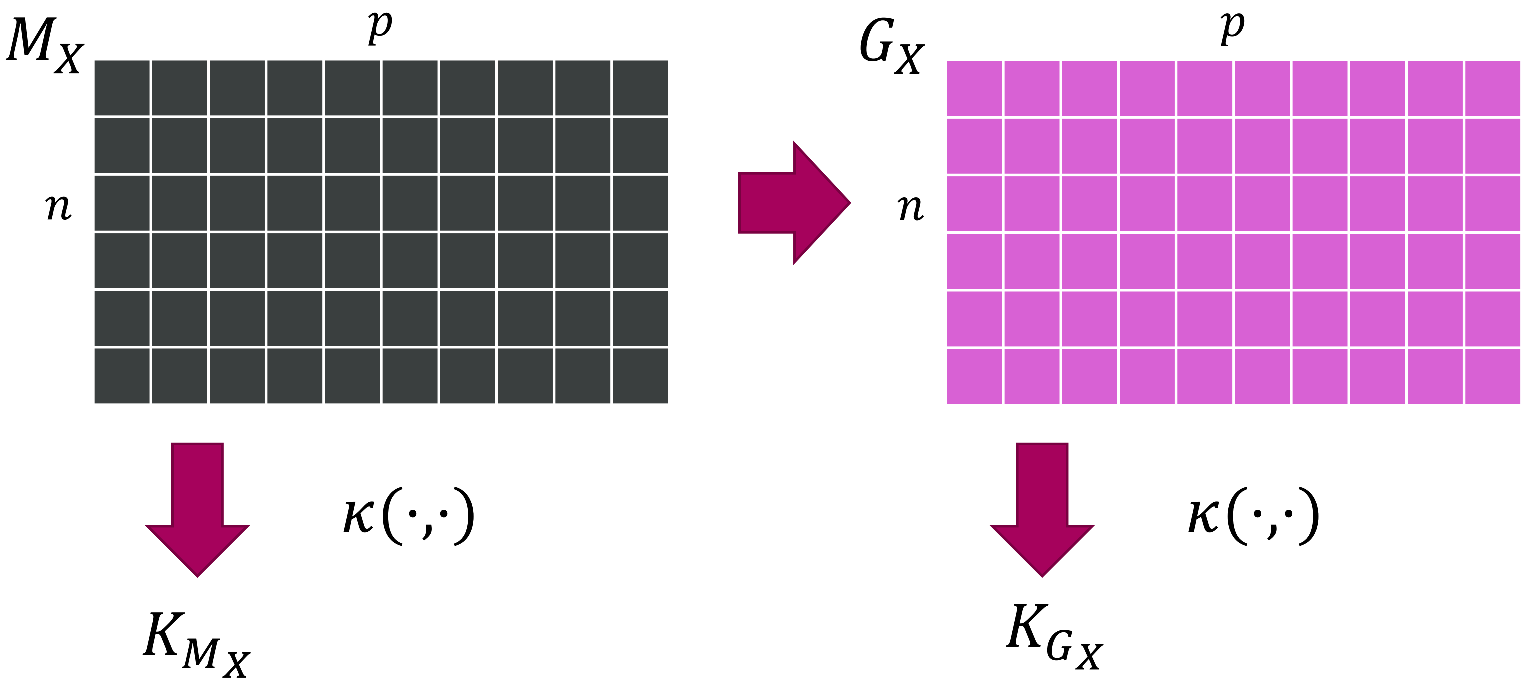
SUMMARY OF METRICS INVARIANCES

Similarity Index	Formula	Invariant to: Invertible Linear Transform	Invariant to: Orthogonal Transform	Invariant to: Isotropic Scaling
Linear Reg. (R^2_{LR})	$\|\|Q_Y^{\text{T}} X\|\|_{\text{F}}^2 / \|\|X\|\|_{\text{F}}^2$	Y only	✓	✓
CCA (R^2_{CCA})	$\|\|Q_Y^{\text{T}} Q_X\|\|_{\text{F}}^2 / p_1$	✓	✓	✓
CCA ($\bar{\rho}_{\text{CCA}}$)	$\|\|Q_Y^{\text{T}} Q_X\|\|_* / p_1$	✓	✓	✓
SVCCA (R^2_{SVCCA})	$\|\|(U_Y T_Y)^{\text{T}} U_X T_X\|\|_{\text{F}}^2 / \min(\|\|T_X\|\|_{\text{F}}^2, \|\|T_Y\|\|_{\text{F}}^2)$	If same subspace kept	✓	✓
SVCCA ($\bar{\rho}_{\text{SVCCA}}$)	$\|\|(U_Y T_Y)^{\text{T}} U_X T_X\|\|_* / \min(\|\|T_X\|\|_{\text{F}}^2, \|\|T_Y\|\|_{\text{F}}^2)$	If same subspace kept	✓	✓
PWCCA	$\sum_{i=1}^{p_1} \alpha_i \rho_i / \|\|\alpha\|\|_1,\ \alpha_i = \sum_j \|\langle \mathbf{h}_i, \mathbf{x}_j \rangle\|$	✗	✗	✓
Linear HSIC	$\|\|Y^{\text{T}} X\|\|_{\text{F}}^2 / (n-1)^2$	✗	✓	✗
Linear CKA	$\|\|Y^{\text{T}} X\|\|_{\text{F}}^2 / (\|\|X^{\text{T}} X\|\|_{\text{F}} \|\|Y^{\text{T}} Y\|\|_{\text{F}})$	✗	✓	✓
RBF CKA	$\text{tr}(KHLH) / \sqrt{\text{tr}(KHKH)\text{tr}(LHLH)}$	✗	✓	✓*

Table from [2].

AUGMENTING CKA WITH INFORMATION ON GRADIENTS

[4] proposes an augmentation of CKA by incorporating information on gradients for the given layer(s)



$$K_X = K_{M\,X} \odot K_{G\,X}$$

$$K_Y = K_{M\,Y} \odot K_{G\,Y}$$

$$\mathrm{CKA}_{Gr.}\left(\widetilde{K_X}, \widetilde{K_Y}\right) = \frac{\langle \widetilde{K_X}, \widetilde{K_Y} \rangle}{\|\widetilde{K_X}\| \|\widetilde{K_Y}\|}$$

WHY CKA AND CCA MIGHT BE WRONG

Classical statistics:
FIXED features (*variables*)
VARIABLE datapoints (*observations*)
p *fixed;* $n \to \infty$

Most of the results we know and use everyday in research adhere to this paradigm.
e.g. law of large numbers

High dimensional statistics [7]:
VARIABLE features (*variables*)
VARIABLE datapoints (*observations*)
$p \to \infty;\ n \to \infty$

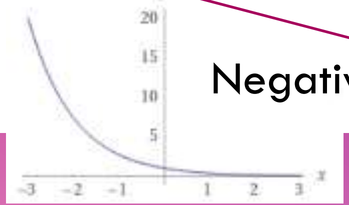
In high dimensional statistics, some results from classical statistics do not hold or may present abnormally large errors

In Deep Learning, the focus is either on depth and width. In wide ANNs, we're essentially increasing *p* which may be way larger than *n*. Consider moreover that both CCA and CKA are essentially based on the concept of COVARIANCE. Maybe, we might want to detach from a classical statistical view and go to more "*non-parametric*" techniques to obtain metrics.

$$P(\|\Sigma - \hat{\Sigma}\| \geq \delta) \leq 2p \exp\left(\frac{-n\delta^2}{2b(\|\Sigma\| + \delta)}\right), b \in \mathbb{R}, \delta \to 0^+$$

What is this?

Linear increase



Negative exponential decay

IMD

CRAZY STUFF!

- IMD is a metric recently proposed at ICLR 2020 [6]
- Compares generic data manifolds, unaligned and different in dimension
- Underlying theory is absurdly difficult
- If you really want to have a go at it
  - https://github.com/xgfs/imd
- Focus is on generative models and language models, but it should work fine on simple MLPs as well

RECAPPING

Comparing hidden representations produced by MLPs is a difficult task

Unaligned representations: Neuron i in ANN 1 might not be the same as neuron i in ANN2

Different dimensionalities: Representations may not be composed by the same number of neurons

Metrics such as CKA and (SV)CCA overcome these hurdles

Curse of dimensionality: Human intuition fails when the number of variables (*neurons*) in the representation is very high

High dimensional statistics: Regular statistical results may fail when $p \longrightarrow$ *inf*. It is shown that covariance is tricky in that scenario

REFERENCES

1. Raghu, Maithra, et al. "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability." *NeurIPS, 2018*.
2. Kornblith, Simon, et al. "Similarity of neural network representations revisited." *International Conference on Machine Learning*. PMLR, 2019.
3. Doimo, Diego, et al. "Hierarchical nucleation in deep neural networks." *NeurIPS, 2020*.
4. Tang, Shuai, et al. "Similarity of neural networks with gradients." *arXiv preprint arXiv:2003.11498* (2020).
5. Zullich, Marco, et al. "Investigating Similarity Metrics for Convolutional Neural Networks in the Case of Unstructured Pruning". ICPRAM, 2020.
6. Tsitsulin, Anton, et al. "The Shape of Data: Intrinsic Distance for Data Distributions." ICLR, 2020.
7. Wainwright, Martin J. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press, 2019.