CARMINE-EMANUELE CELLA

NEURAL NETWORKS FOR MUSIC APPLICATIONS (I)

MUSIC 159

Geometric approach

OUTLINE

- signals can be represented in vector spaces with inner product;
- projections compute similarity, reconstruction from projections is possible with bases;
- Fourier: a-priori representation for frequency, invariant to time (independent), instable to frequency (dependent variable);
- from Fourier to Wavelets: a-priori representation for time-frequency, invariant to time, stable to frequency, information is lost (not unique), direct reconstruction is not possible;
- scattering: a-priori representation for time-frequency, invariant to time, stable to frequency, information is recovered from upper layers, direct reconstruction is not possible, first layer similar to MFCC;
- joint scattering: as scattering but able to connect different variables; e.g. spiral scattering acts on a third (dependent) variable;
- clustering: unsupervised representation based on geometric proximity;
- MLP: supervised representation that maximises projectors for the specific problem, can act on many variables non joinlty but it is difficult to know what are the variables;
- CNN: as MLP but can act jointly on variables;

PROJECTIONS

2.2. **Projection.** An important application of inner product is to project one vector over another. The projection of x on y is defined as:

$$\mathfrak{P}_x(y) = \frac{\langle x, y \rangle}{\|x\|^2} \cdot x \tag{3}$$

where the ratio between the inner product and the squared norm of x is called *coefficient of projection.*

•: **KEYPONT: projections compute the *similarity* (covariance) of the two vectors**.

2.3. **Reconstruction from projections.** A vector can be reconstructed with a linear combination from its projections on another set of vectors if and only if the set used is a basis.

ANALYSIS AND SYNTHESIS

3.1. **Analysis.** The analysis is the representation ϕ_x of a signal given by the inner product of it by a basis in a vector space; it is therefore given by the projection

$$\phi_x = \sum_t x(t) * \overline{b_k} = \langle x, b_k \rangle \tag{4}$$

where b_k is a given basis and t is time.

3.2. **Synthesis.** The synthesis is the reconstruction of the original signal x by the summation of the products with the representation ϕ_x created by the analysis:

$$x(t) = \sum_k \phi_x b_k(t) = \sum_k \langle x, b_k \rangle b_k(t). \tag{5}$$

REPRESENTATIONS

Representations can be considered linear operators that need to be invariant to sources of unimportant variability, while being able to capture discriminative information from signals. As such, they must respect four basic properties; being x a signal and Φx its representation:

- *discriminability:* $\Phi x \neq \Phi y \implies x \neq y$;
- *stability:* $\|\Phi x - \Phi y\|_2 \leq C\|x - y\|_2$;
- *invariance (to group of transformations G):* $\forall g \in G, \Phi g.x = \Phi x$;
- *reconstruction:* $y = \Phi x \iff \tilde{x} = \Phi^{-1} y$.

Discriminability means that if the representations of two signals are different than the two signals must be different. Stability means that a small modification of a signal should be reflected in a small modification in the representation and vice-versa. Invariance to a group of transformation G, means that if a member of the group is applied to a signal, than the representation must not change; reconstruction, finally, is the possibility to go back to a signal that is *equivalent* to the original (in the sense of a group of transformations) from the representation. It is possible to divide representations in two major categories: *prior* and *learned*.

REPRESENTATIONS

NO LEARNING: FOURIER

4.1. **Fourier.** The Fourier representation is a specific case of analysis and synthesis, where the basis is given by a set of complex sinusoids: $b_k = e^i 2\pi k$ (where i is the imaginary unit).

The discrete Fourier analysis (DFT) will be therefore:

$$\boxed{\hat{x}(k) = \sum_t x(t) e^{\frac{-i2\pi kt}{T}}} \tag{6}$$

and, in the same way, the reconstruction (or inverse Fourier transform, IDFT) is given by:

$$\boxed{x(t) = \frac{1}{T} \sum_k \hat{x}(k) e^{\frac{i2\pi kt}{T}}}. \tag{7}$$

This can be thought of as a convolution with the Fourier basis

NO LEARNING: WAVELET/SCATTERING TRANSFROM

4.3. **Scattering.** The information lost in the wavelet representation is recovered by the following cascaded multi-layer representation:

$$S_1 x(t, \lambda_1) = |x \star \psi_{\lambda_1}| \star \phi(t). \tag{8}$$

These are called first-order scattering coefficients; second order scattering coefficients are defined as:

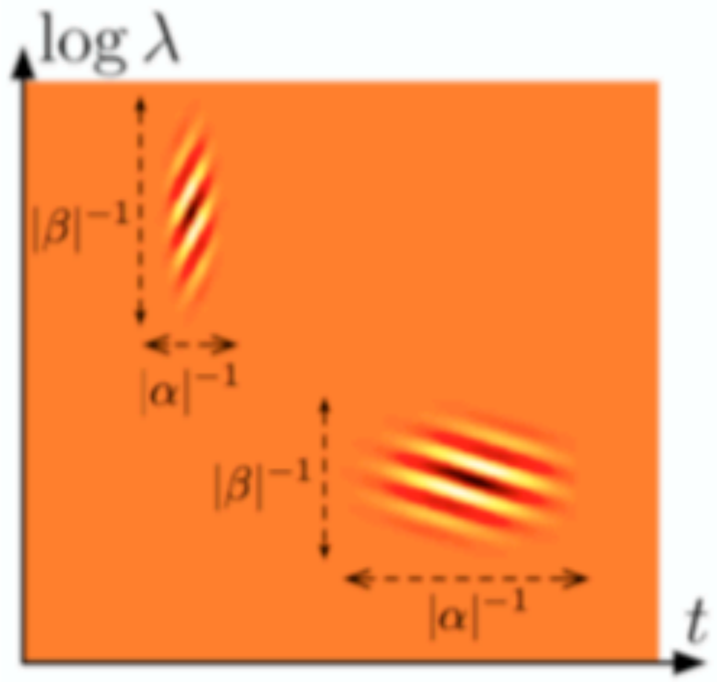
$$S_1 x(t, \lambda_1, \lambda_2) = ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi(t). \tag{9}$$

Iterating this process defines scattering coefficients at any order m:

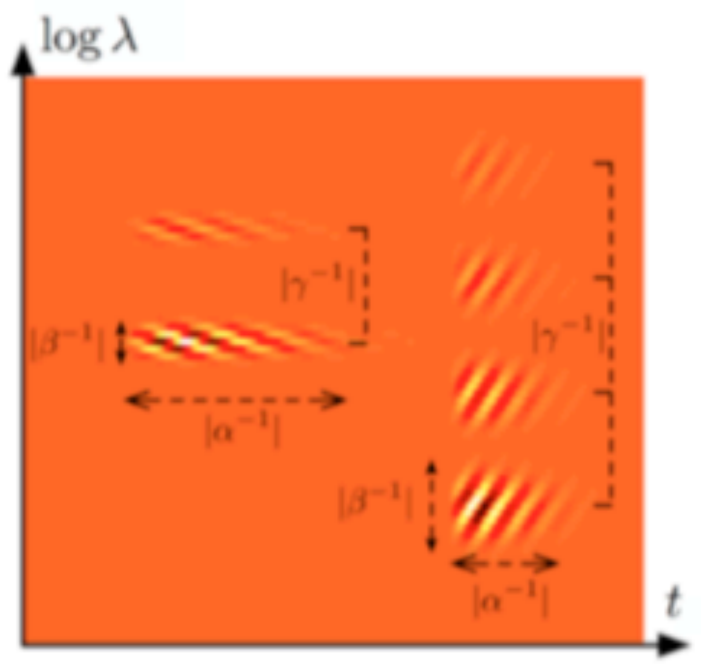
$$\boxed{S_m x(t, \lambda_1, \ldots, \lambda_m) = || \ldots |x \star \psi_{\lambda_1}| \star \ldots | \ldots \star \psi_{\lambda_m}| \star \phi(t)}. \tag{10}$$

NO LEARNING: JOINT SCATTERING TRANSFROM

- **joint-scattering**: $||x \overset{t}{\star} \psi_\lambda| \overset{t}{\star} \psi_\alpha \overset{log\lambda}{\star} \psi_\beta|$



- **spiral-scattering**: $||x \overset{t}{\star} \psi_\lambda| \overset{t}{\star} \psi_\alpha \overset{log\lambda}{\star} \psi_\beta \overset{oct}{\star} \psi_\gamma|$



SUPERVISED LEARNING: MULTI-LAYER PERCEPTRON

5.2. **Multi-layer perceptrons.** Set of linear transformations followed by non-linearities (like scattering) in which the projectors are learned with supervision with backpropagation [...]:

$$MLP_1 = \rho(Wx + b) \tag{11}$$

where:

- W is a a linear transformation made of weights found during learning;
- b is a translation vector;
- is a point-wise application of a non-linearity.

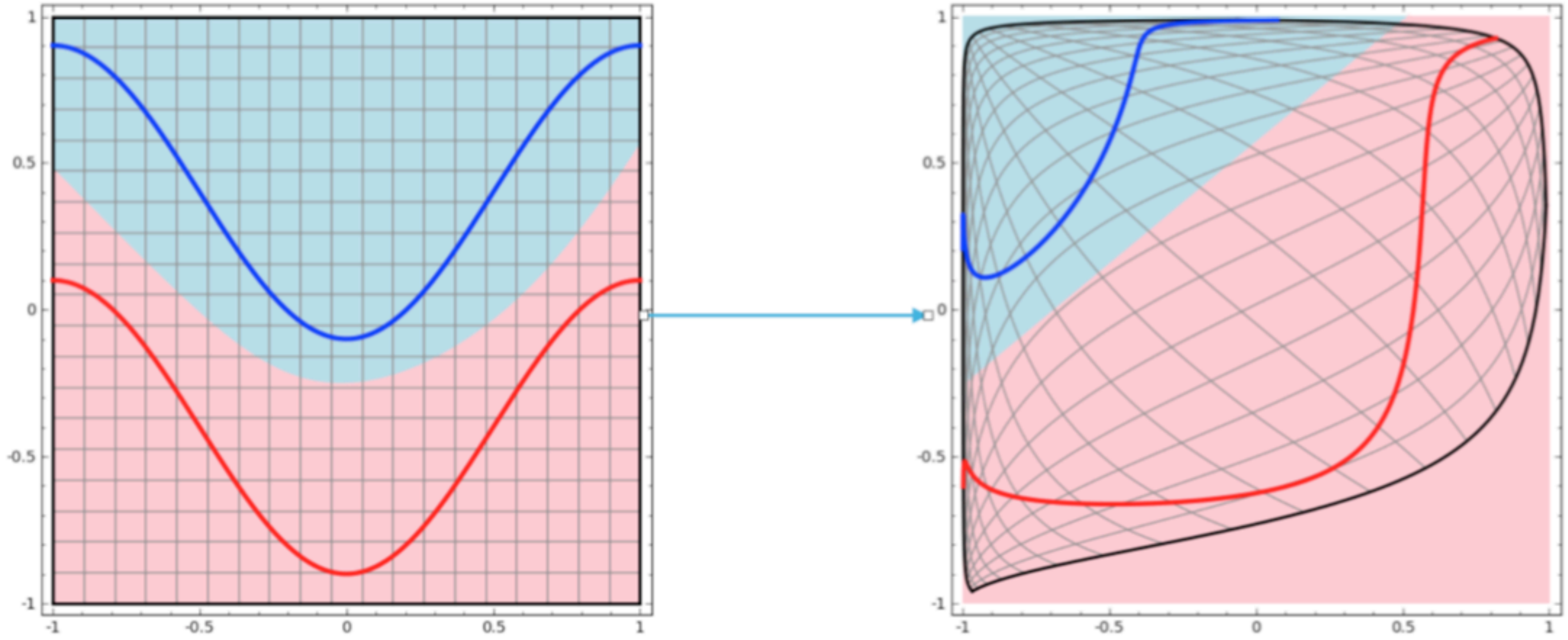
This structure can be repeated in a cascaded structure, creating invariances for different variables.

SUPERVISED LEARNING: CNN

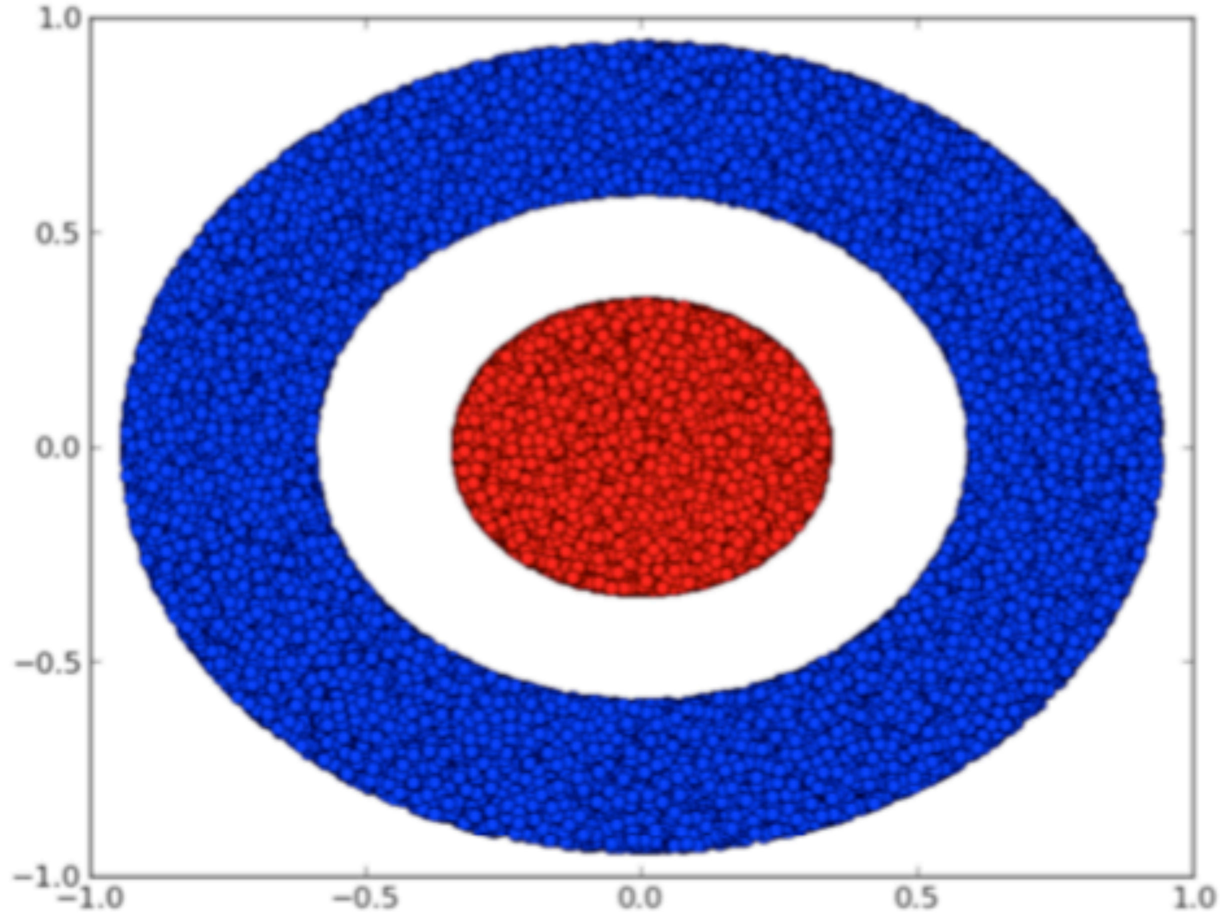
CNN are a supervised representations as MLP that can apply invariances and stabilize any number of variables and is also able to act *jointly* on the variable as joint scattering [...].

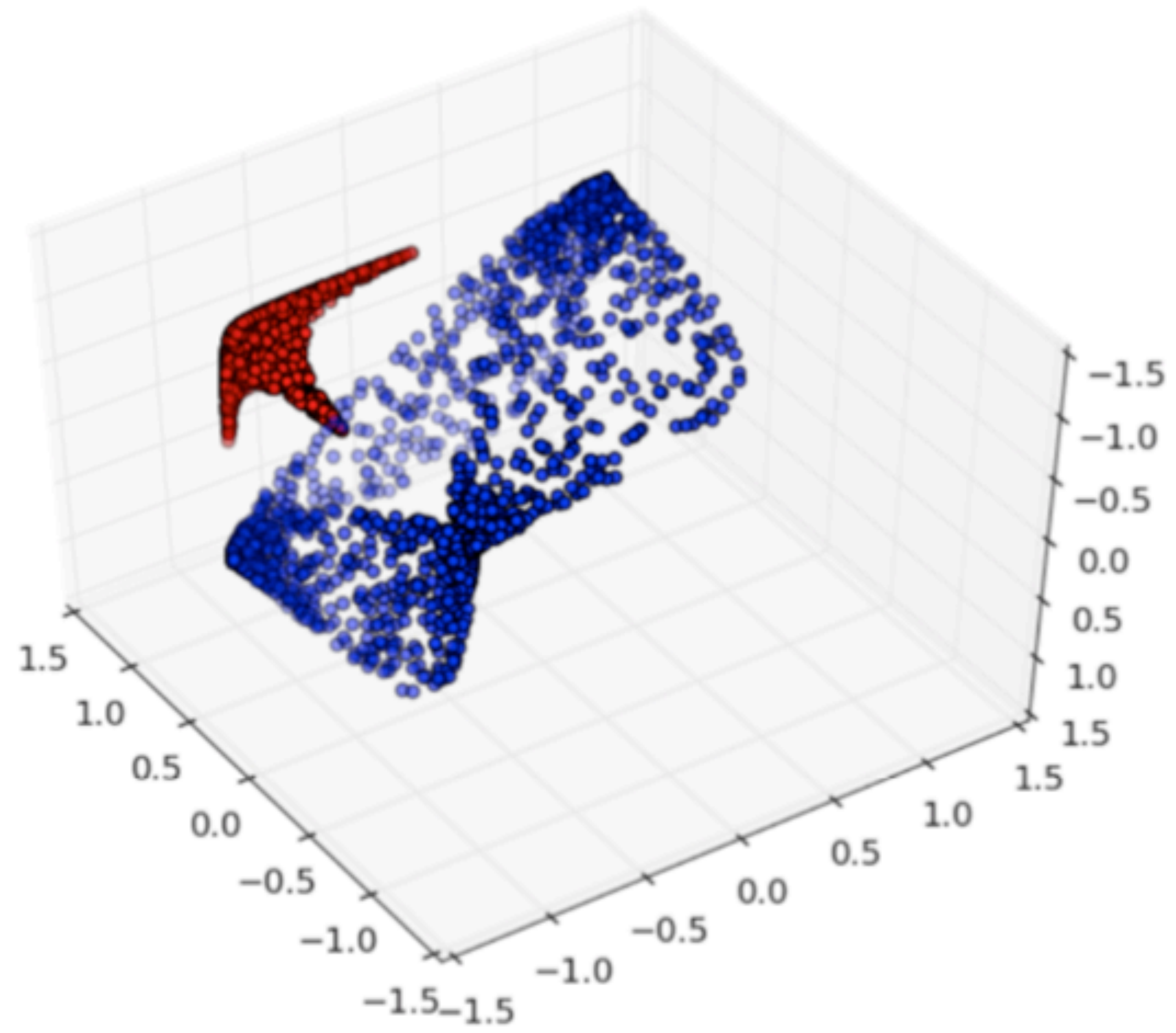
LINEARIZATION: 2D

5.3. **Linearization.** We can think that the role of projectors and non-linearities in MLP is to *linearize* the feature space [...]. See figures 5 and 6.

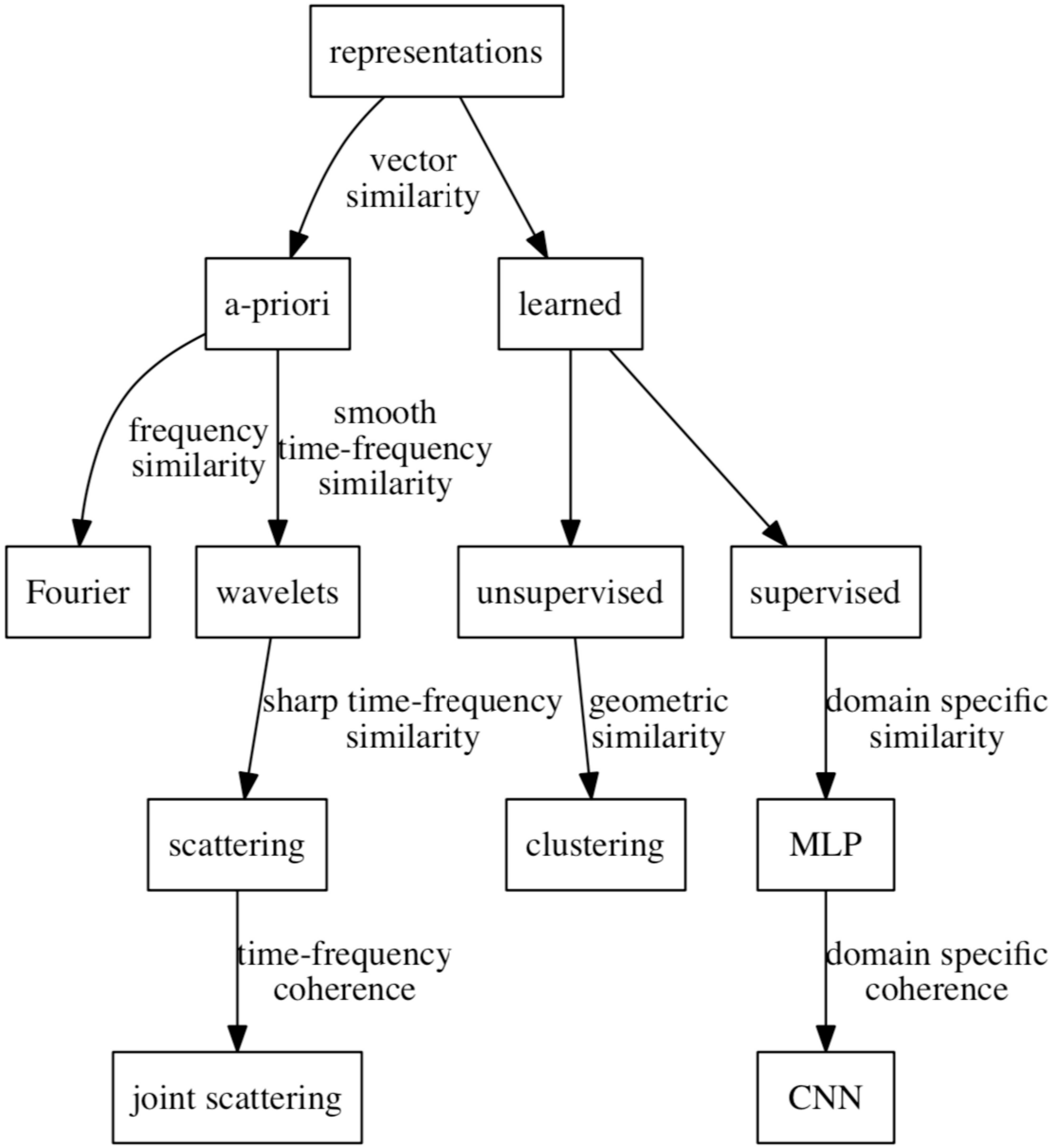


LINEARIZATION: 3D





UNIFIED OVERVIEW



THANK YOU!

Suggested exercise: review topics in linear algebra!