Encoding and Decoding with Deep Learning and MRI data

Thierry Artières

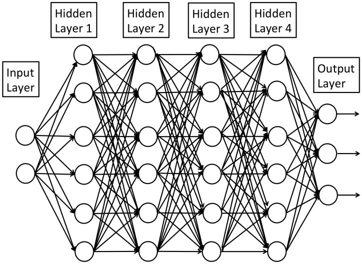
June 10, 2024

Outline

Outline

Sequence of representation spaces implemented by successive hidden layers

A series of hidden layers



Task-oriented vs. unsupervised models

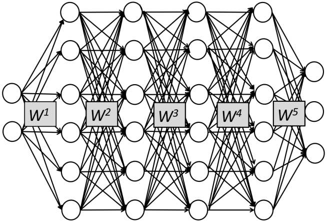
- Task-oriented: classification, regression
- Unsupervised: autoencoders

Deep Learning = Representation Learning

Sequence of representation spaces implemented by successive hidden layers

Computes a complex function of the input

$$y = g(W^k \times g(W^{k-1} \times g(...g(W^1 \times x))))$$



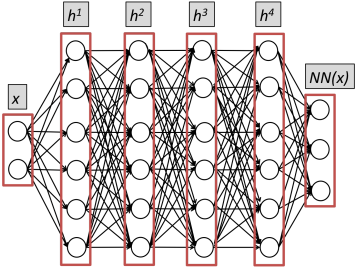
Task-oriented vs. unsupervised models

- Task-oriented: classification, regression
- Unsupervised: autoencoders

Deep Learning = Representation Learning

Sequence of representation spaces implemented by successive hidden layers

Computes new representations of the input



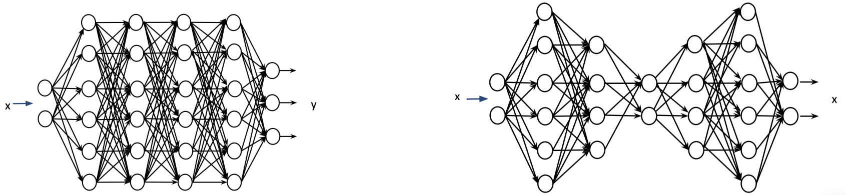
Task-oriented vs. unsupervised models

- Task-oriented: classification, regression
- Unsupervised: autoencoders

Sequence of representation spaces implemented by successive hidden layers

Task-oriented vs. unsupervised models

- Task-oriented: classification, regression
- Unsupervised: autoencoders



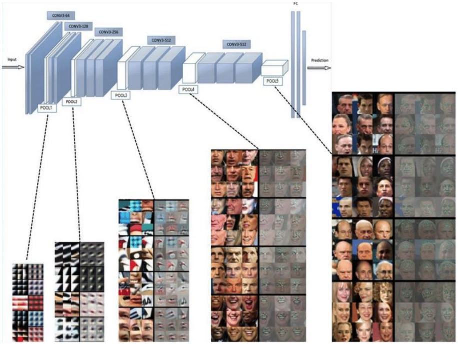
Deep Learning = Representation Learning

What are good features of a representation space?

- Naturally induced by the learning objective
  - Contain sufficient and useful information (for the targeted task)
  - Noise free
- Expected, and partially observed, but not so easy to favor
  - High (semantic) level
- Not observed without specific effort but one may try to favor
  - Disentangled vs. distributed
  - Interpretable vs. black box
  - Compositionality

Which ones?

- Very popular credo: A learned NN implements a hierarchy of (more and more semantic) features
  - Induced by CNN's increasing receptive field size in CNNs
  - It might rather be a refinement of representations (in a context) in transformers and ResNets-based architectures



Which ones?

- Very popular credo: A learned NN implements a hierarchy of (more and more semantic) features
  - Induced by CNN's increasing receptive field size in CNNs
  - It might rather be a refinement of representations (in a context) in transformers and ResNets-based architectures
- Actually not so easy to know what a NN uses as information... For instance: what do CNN actually uses in an image? **shapes**? or textures?

(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

Figure 1: Classification of a standard ResNet-50 of **(a)** a texture image (elephant skin: only texture cues); **(b)** a normal image of a cat (with both shape and texture cues), and **(c)** an image with a texture-shape cue conflict, generated by style transfer between the first two images.

(Geirhos et al. [Gei+19])

How lo learn good representations?

Many ways dedicated to various goals

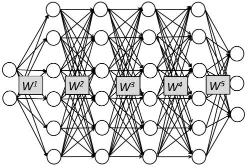
- Standard: Iterative refinement in successive hidden layers *(hierarchy of representations)*
- Structural bias *(convolution, recurrence...)*
- Adding constraints on the hidden representation space on a sample per sample basis *(sparsity, norm, etc)*
- Adding constraints on the hidden representation space on a distributional basis (Adversarial loss)
- Using the context

Adding a regularization loss

- Use a combined loss

$$C(w) = \underbrace{\sum_i loss_i(w)}_{\text{Data Fit term}} + \Omega(w)$$

- Many possibilities for Ω
  - Standard L1 and L2 regularization strategies: $\|w\|$ or $\|w\|^2$ for the full NN or for a single layer
  - Sparsifying activations in a layer l: $\sum_i \|h_i^l\|^2$
  - Limiting sensitivity to input features : $\|J_l(x)\|^2 = \sum_{p,k} \left(\frac{\partial h_k^l(x)}{\partial x_p}\right)^2$

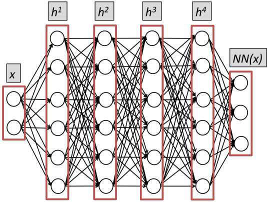


Adding a regularization loss

- Use a combined loss

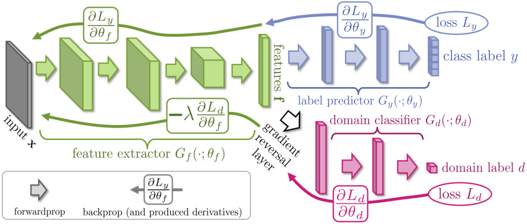
$$C(w) = \underbrace{\sum_i loss_i(w)}_{\text{Data Fit term}} + \Omega(w)$$

- Many possibilities for Ω
  - Standard L1 and L2 regularization strategies: $\|w\|$ or $\|w\|^2$ for the full NN or for a single layer
  - Sparsifying activations in a layer l: $\sum_i \|h_i^l\|^2$
  - Limiting sensitivity to input features : $\|J_l(x)\|^2 = \sum_{p,k} \left(\frac{\partial h_k^l(x)}{\partial x_p}\right)^2$



Removing information from a representation space

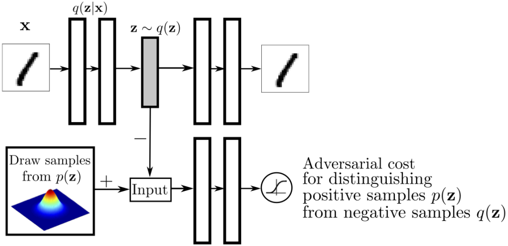
- Learn a predictor (classifier) that predicts some specific information from a hidden layer's output
- Learn the feature extractor (i.e. NN below the hidden layer) so that the classifier cannot recover the specific information



(Ganin and Lempitsky [GL15])

Adding a distributional constraint

- Enforcing the distribution in a hidden layer to obey a predefined distribution with an adversarial discriminator

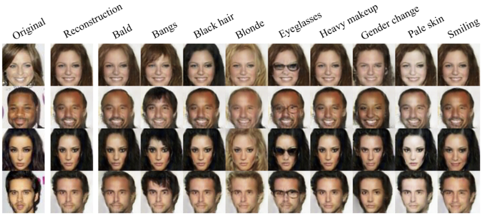


(Makhzani [Mak18])

Outline

Main idea

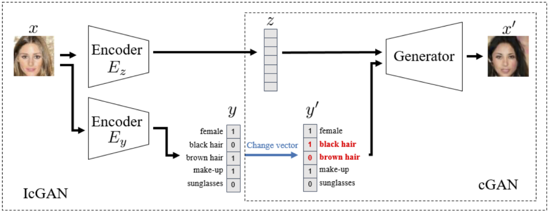
- Identify main factors of variation of the data (e.g. hair colour, w/wo glasses in face images)
- Encode the factors in different components of the latent space
- Allows easier (semantic) edition of data
- Better with some supervision (Cf. impossibility theorem in unsupervised mode by (Locatello et al. [Loc+18]))



(Perarnau et al. [Per+16])

Main idea

- Identify main factors of variation of the data (e.g. hair colour, w/wo glasses in face images)
- Encode the factors in different components of the latent space
- Allows easier (semantic) edition of data
- Better with some supervision (Cf. impossibility theorem in unsupervised mode by (Locatello et al. [Loc+18]))



(Perarnau et al. [Per+16])

Outline

1. Learning representations
   - Basics
   - Disentangled representations
   - Compositionality
   - Editing in the latent space

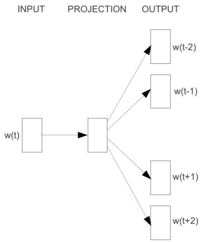
Pioneer results in Natual Language Processing (Skipgram architecture (Le and Mikolov [LM14]))

- Unsupervised learned word representations (called embeddings, $e(word)$) exhibit a compositionality feature:

$$e(uncle) + (e(woman) - e(man)) \approx e(aunt)$$

$$e(King) + (e(plural) - e(singular)) \approx e(Kings)$$

$$e(France) + (e(capital) - e(country)) \approx e(Paris)$$



Skip-gram

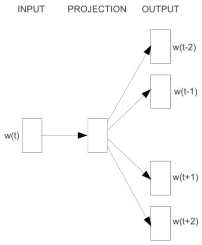
Pioneer results in Natual Language Processing (Skipgram architecture (Le and Mikolov [LM14]))

- Unsupervised learned word representations (called embeddings, $e(word)$) exhibit a compositionality feature:

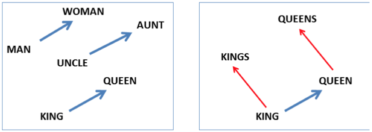
$$e(uncle) + (e(woman) - e(man)) \approx e(aunt)$$

$$e(King) + (e(plural) - e(singular)) \approx e(Kings)$$

$$e(France) + (e(capital) - e(country)) \approx e(Paris)$$



Skip-gram



Compositionality in latent representations

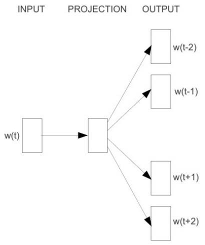
Pioneer results in Natual Language Processing (Skipgram architecture (Le and Mikolov [LM14]))

- Unsupervised learned word representations (called embeddings, $e(word)$) exhibit a compositionality feature:

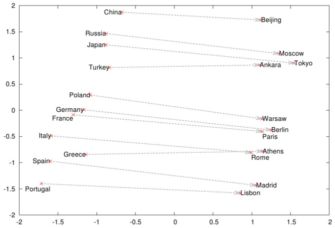
$$e(uncle) + (e(woman) - e(man)) \approx e(aunt)$$

$$e(King) + (e(plural) - e(singular)) \approx e(Kings)$$

$$e(France) + (e(capital) - e(country)) \approx e(Paris)$$



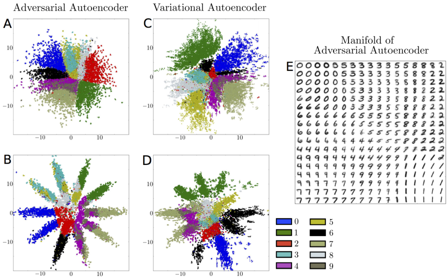
Skip-gram



Outline

Traversals

- One may compute interpolated representations of samples (more relevant in autoencoder-like architectures since details might be removed in classifiers)
- Example
  - Consider two samples x_1 and x_2
  - Compute their latent representations with encoder E: $h_i = E(x_i)$
  - Interpolate $h_\alpha = \alpha h_1 + (1 - \alpha) h_2$ for some $\alpha \in [0, 1]$
  - Decode $x' = D(h_\alpha)$



Exploring the latent space

Edition in the latent rep

- One may *edit* data in a hidden layer representation space (*latent space*)
- Edition process
  - Compute the latent representation h of input x with encoder E: $h = E(x)$
  - Add some vector z to h
  - Decode $x' = D(E(x) + z)$
- If the latent space "is semantic" then one may get a transformed version of x by using an appropriate semantic translation vector z
  - For instance z is an

When should it work?

- The latent space should be "semantic"
- The latent space should be well occupied by training data: whatever x, $E(x) + z$ should have been encountered by the decoder D in the training stage or it should be able to generalize.

Exploring the latent space

Editing with statistics in the latent space

- One may compute means of samples' representations in a hidden layer representation space (*latent space*).
- If the latent space "is semantic" then one may get some latent space representation of a specific label in the latent space
  - Mean of latent representation of faces of men
  - Mean of latent representation of faces of men minus that of women

Outline

Outline

Goal

- Given data $\{x_1, ..., x_N\}$ over a data space $\mathcal{X} = \mathbb{R}^d$ learn the underlying distribution p^*
- Learn a model of the density of data / able to sample with this density
  - Postulate a parametric model / family $\mathcal{P}_\theta = \{p_\theta, \theta \in \Theta\}$
  - Learning = select the best θ^*
    - Requires a performance measure : distance/loss between p_{θ^*} and p^*: $L(p_{\theta^*}, p^*)$

What we may expect

- p_θ assigns high density to samples taken from the true p^*

$$x \sim p^*(x) \Rightarrow p_\theta(x) \text{ is "high"}$$

- Samples taken from p_θ behave similarly to real samples from p^*

$$x \sim p_\theta(x) \Rightarrow p^*(x) \text{ is "high"}$$

- Of course both properties are related by the normalization feature of densities

Strategies

- Dealing with first or second expectation yields different choices for the loss $\mathcal{L}$ and different behaviours for p_θ
- Focusing with first property (p_θ assigns high density to samples taken from the true p^*)
  - "Coverage driven" strategy
  - Easier since it requires only samples from $p^* \Rightarrow$ MLE approaches, Normalizing flows, Variational autoencoders (Kingma and Welling [KW14])...
- Focusing with second property (Samples taken from p_θ behave similarly to real samples from p^*)
  - "Quality driven" strategy
  - Less convenient as a right implementation would require access to $p^* \Rightarrow$ GANs (Goodfellow et al. [Goo+14])...

Outline

2. Generative models
   - Basics
   - Generative Adversarial Networks

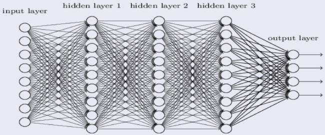
Adversarial learning principle

Principle

- Use a two player game
  - Learn both a generator of artificial samples AND a discriminator that learns to distinguishes between true and fake samples.
  - The generator wants to flue the discriminator
- If an equilibrium is reached the generator produces samples with the true density

Adversarial Learning: Generator

Determinitic NN as a generative model



Using a deterministic NN as a generative model

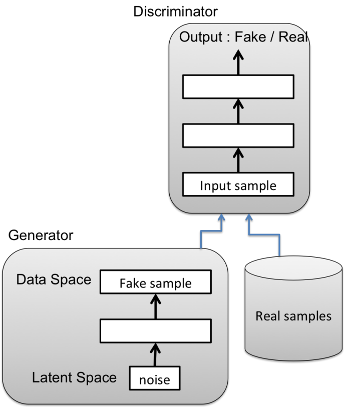
- Let note the function implemented by the model as G
- Let note the input $z \rightarrow$ The NN computes $G(z)$
- Assume z obeys a prior (noise) distribution, p_z, e.g. Gaussian distribution
- then the output x of the NN follows a distribution

$$\Rightarrow p_G(x) = \int_{z \text{ s.t. } G(z)=x} p_z(z) dz$$

Generative Adversarial Networks [Goodfellow and al., 2014]

Principle

- Two players game: Generator G and Discriminator D
  - D aims at distinguishing true samples from fake samples
  - G aims at fooling D



Adversarial Learning criterion

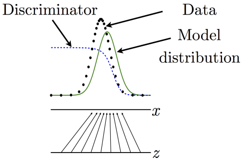
Criterion from (Goodfellow et al. [Goo+14])

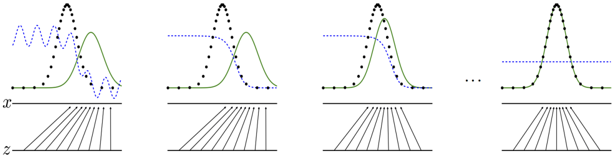
- Generator G and Discriminator D are two NNs
  - Whose parameters are noted θ_g and θ_d
- Distributions
  - p_{data} stands for the empirical distribution of the data from the training set
  - p_z is a prior noise distribution, e.g. a Gaussian distribution
  - On convergence we want $p_g = p_{data}$
- Learning criterion:

$$min_g max_d v(\theta_g, \theta_d) = \mathbf{E}_{x \sim p_{data}} [logD(x)] + \mathbf{E}_{z \sim p_z} [log(1 - D(G(z)))]$$

  - Assume G is fixed: D is trained to distinguish between fake and true samples
  - Assume D is fixed : G is trained to generate samples as realistic as possible

Adversarial Learning theory: What happens during Learning



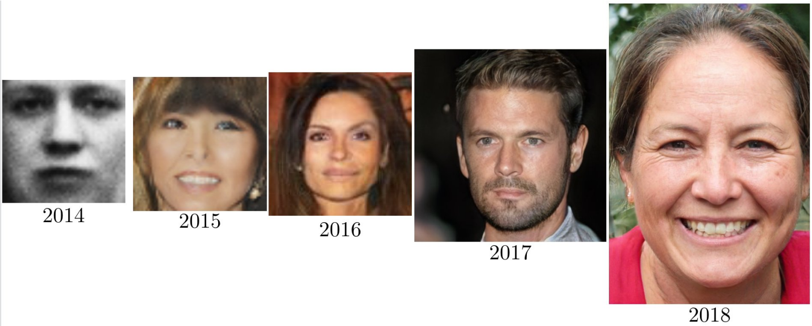


Interpolating with GANs (Goodfellow et al. [Goo+14])

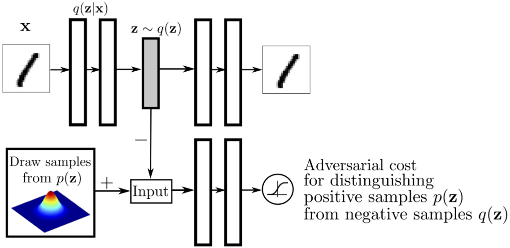
Idea

- The latent code space is fully occupied
- Any sample drawn by sampling with the generator should be realistic
- One may interpolate between two latent codes and see

Figure 3: Digits obtained by linearly interpolating between coordinates in z space of the full model.


2014
2015
2016
2017
2018

Adversarial AE (Makhzani et al. [Mak+15])

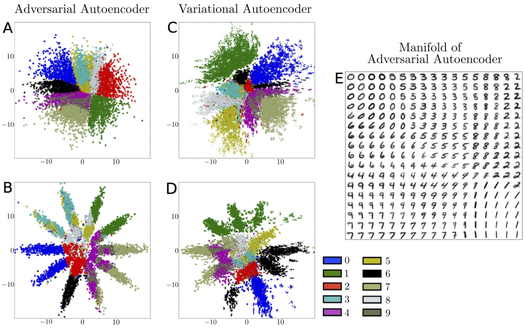


Learning criterion

- Few definitions for $q(z|x)$: simplest = deterministic
- Learning criterion:

$$min_g max_d v(\theta_g, \theta_d) = \mathbf{E}_{x \sim p_{data}} \left[\|D_c(E_c(x)) - x\|^2 \right] + \mathbf{E}_{z \sim p_z} [logD(z)]$$
$$+ \mathbf{E}_{x \sim p_{data}} [log(1 - D(q(z|x)))]$$

Adversarial AE (Makhzani et al. [Mak+15])



Learning criterion

- Few definitions for $q(z|x)$: simplest = deterministic
- Learning criterion:

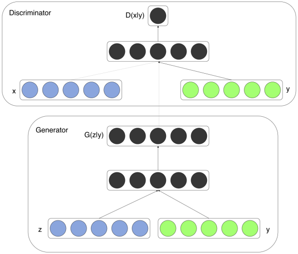
$$min_g max_d v(\theta_g, \theta_d) = \mathbf{E}_{x\sim p_{data}}\left[\|D_c(E_c(x)) - x\|^2\right] + \mathbf{E}_{z\sim p_z}[logD(z)] \\ + \mathbf{E}_{x\sim p_{data}}[log(1 - D(q(z|x)))]$$

Conditional GANs (Mirza and Osindero [MO14])

Learning criterion

- Citerion

$$min_g max_d v(\theta_g, \theta_d) = \mathbf{E}_{x,y \ p_{data}} [logD(x,y)] + \mathbf{E}_{z \ p_z, y' \ p_y} \left[log(1 - D(G(z,y'),y')) \right]$$



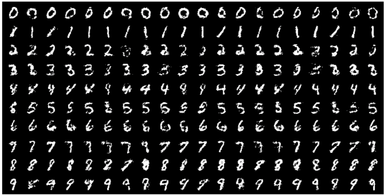


Figure 2: Generated MNIST digits, each row conditioned on one label

Outline

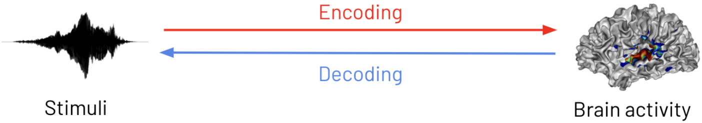
Outline

3. Few Deep Learning studies for MRI data
   - Objectives and means
   - Computational vs. neural representations
   - Comparing computational and neural representations
   - Encoding and decoding via intermediate representation space
   - Constraining the computational representation from neural representation

Usual tasks

Encoding and decoding

- Encoding: Predict the brain activity from the stimuli
- Decoding: Predict the stimuli (or its class) from brain activity

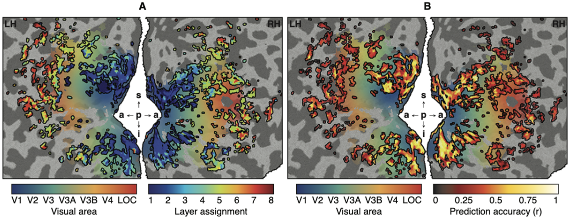


Encoding and decoding

- Encoding: Predict the brain activity from the stimuli
- Decoding: Predict the stimuli (or its class) from brain activity

Brain mapping

- Explain / understand the (level of) processing pefomed in areas of the brain



(Güçlü and Gerven [GG15])

Usual tasks

Encoding and decoding

- Encoding: Predict the brain activity from the stimuli
- Decoding: Predict the stimuli (or its class) from brain activity

Brain mapping

- Explain / understand the (level of) processing pefomed in areas of the brain

Main problems

- Encoding and decoding: both supervised tasks but limited training data
- Encoding and decoding are likely complex nonlinear mappings
- Noisy data (MRI)
- Small size dataset
- Large inter-subject variability

How to infer knowledge on the brain?

Encoding, decoding, brain mapping

- Using standard prediction tools
  - Decoding: Predicting some information about the stimuli from the whole brain activity and use explainability strategies to find where some information is encoded
  - Decoding: Predicting some information about the stimuli from the brain activity from a specific area
  - Encoding: Predicting voxel activity from specific features (spectrogram vs semantic) of a stimuli (speech)
  - ...
- Using Representation Similarity Analysis (RSA)
  - Compare neural and computational representation spaces

Examples of outcomes

- Where is encoded some feature of a stimuli presented to a subject (eg. gender/age/emotion of a speaker etc)
- Identify areas of low-level vs. high-level processing of stimuli

Interpreting encoding and decoding models: caveats (Kriegeskorte and Douglas [KD19])

Encoding

- "Brain regions do not in general form chains of processing stages without skipping connections or recurrent signaling"
- "The primate visual hierarchy is a case in point, where cortical areas interact in a network with about a third of all possible pairwise inter-area connections"

Decoding

- "Decoding reveals the products, not the process of brain computation. However, it is a useful tool for testing whether a brain region contains a particular kind of information in a particular format."
- "Linear decodability indicates "explicit" information"

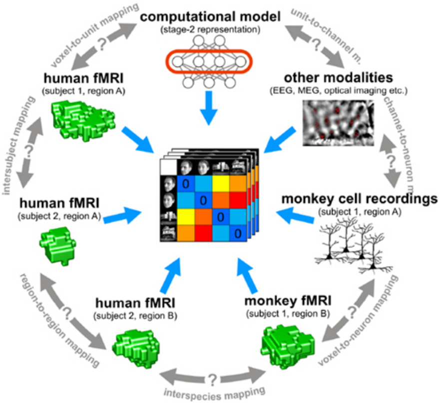
Outline

3. Few Deep Learning studies for MRI data
   - Objectives and means
   - **Computational vs. neural representations**
   - Comparing computational and neural representations
   - Encoding and decoding via intermediate representation space
   - Constraining the computational representation from neural representation

Representation Similarity Analysis (RSA) (McClure and Kriegeskorte [MK16])

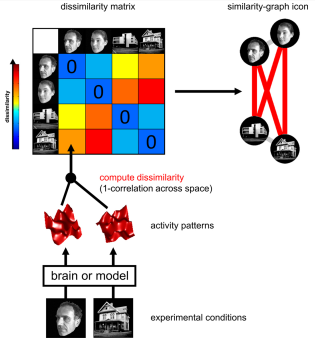
Motivation (Credit images)

- Needs to compare very different representation spaces
- Offers a simple way to compare and get insight on the similarity of two representation spaces without learning/tuning a predictor



Comparing 2 representation spaces (RS)

- Consider N "objects" that were observed in the two RS
- Compute a $N \times N$ dissimilarity matrix (RDM) for each modality
  - Euclidean, Cosine distance...
- Compute a similarity between RDM_1 and RDM_2
  - Pearson correlation coefficient, Rank correlation...

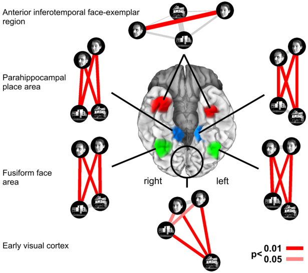


(Kriegeskorte, Mur, and Bandettini [KMB08])

Representation Similarity Analysis (RSA)
(McClure and Kriegeskorte [MK16])

Comparing 2 representation spaces (RS)

- Consider N "objects" that were observed in the two RS
- Compute a $N \times N$ dissimilarity matrix (RDM) for each modality
  - Euclidean, Cosine distance...
- Compute a similarity between RDM_1 and RDM_2
  - Pearson correlation coefficient, Rank correlation...

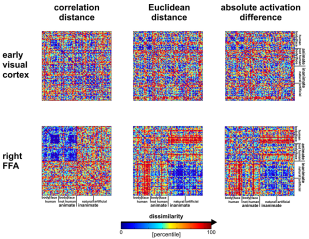


(Kriegeskorte, Mur, and Bandettini [KMB08])

Representation Similarity Analysis (RSA)
(McClure and Kriegeskorte [MK16])

Comparing 2 representation spaces (RS)

- Consider N "objects" that were observed in the two RS
- Compute a $N \times N$ dissimilarity matrix (RDM) for each modality
  - Euclidean, Cosine distance...
- Compute a similarity between RDM_1 and RDM_2
  - Pearson correlation coefficient, Rank correlation...



(Kriegeskorte, Mur, and Bandettini [KMB08])

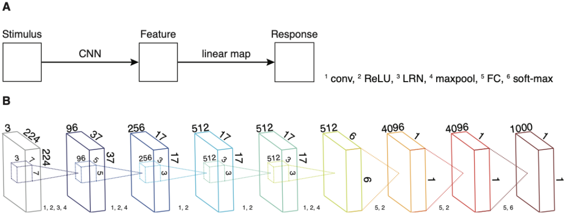
Outline

3. Few Deep Learning studies for MRI data
   - Objectives and means
   - Computational vs. neural representations
   - Comparing computational and neural representations
   - Encoding and decoding via intermediate representation space
   - Constraining the computational representation from neural representation

Characterizing the processing level of stimulus in the brain (Güçlü and Gerven [GG15])

Principle

- Use a pretrained NN (for image classification)
  - Assuming that successive layers implement RS of increasing level of processing of the input
  - Identify the layer whose RS best matches the RS of each area in the brain (actually an area is centred on a voxel, spotlight approach)
- A first approach Güçlü and Gerven [GG15]
  - Learn a linear model to predict each voxel activity from the hidden layer representation of a stimulus (training set = set of stimulus)
  - Label each voxel according to the depth of the hidden layer yielding most accurate prediction

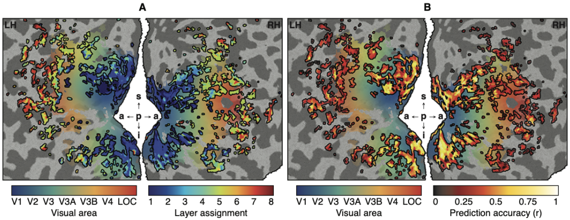


[GG15]

Characterizing the processing level of stimulus in the brain (Güçlü and Gerven [GG15])

Principle

- Use a pretrained NN (for image classification)
  - Assuming that successive layers implement RS of increasing level of processing of the input
  - Identify the layer whose RS best matches the RS of each area in the brain (actually an area is centred on a voxel, spotlight approach)
- A first approach Güçlü and Gerven [GG15]
  - Learn a linear model to predict each voxel activity from the hidden layer representation of a stimulus (training set = set of stimulus)
  - Label each voxel according to the depth of the hidden layer yielding most accurate prediction

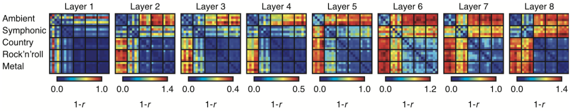


[GG15]

Characterizing the processing level of stimulus in the brain using RSA (Güçlü et al. [Güç+16])

Same principle as before

- Pretrained models for music tag prediction
  - Either time or frequency representations (96000 dimensional), or both, of 6s long audio signals
- RSA used to label the voxels in the STG

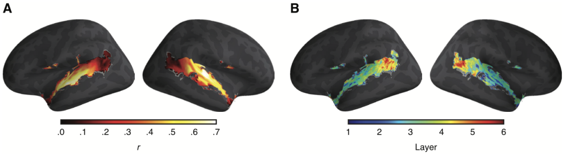


[GG15]

Characterizing the processing level of stimulus in the brain using RSA (Güçlü et al. [Güç+16])

Same principle as before

- Pretrained models for music tag prediction
  - Either time or frequency representations (96000 dimensional), or both, of 6s long audio signals
- RSA used to label the voxels in the STG



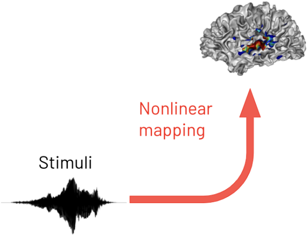
[Güç+16]

Outline

3. Few Deep Learning studies for MRI data
   - Objectives and means
   - Computational vs. neural representations
   - Comparing computational and neural representations
   - Encoding and decoding via intermediate representation space
   - Constraining the computational representation from neural representation

Main assumption

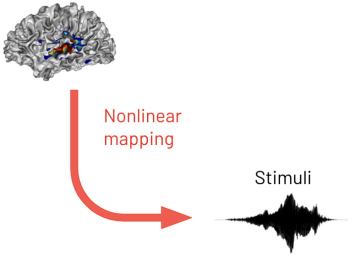
- One can efficiently learn higher-level representation of stimuli using large datasets, either in unsupervised or supervised way
- One may assume that the learned representation is a non-linear function of the input
- One may expect that the mapping between the intermediate representation space and the brain activity space is a simpler (hopefully linear) function than the original mapping



Relying on an intermediate representation space

Main assumption

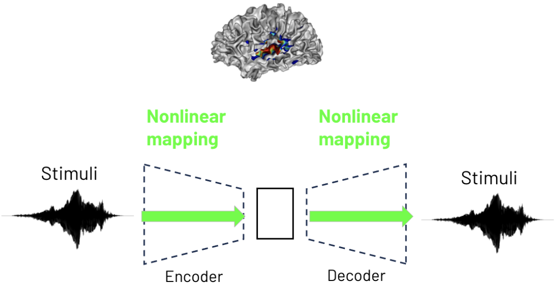
- One can efficiently learn higher-level representation of stimuli using large datasets, either in unsupervised or supervised way
- One may assume that the learned representation is a non-linear function of the input
- One may expect that the mapping between the intermediate representation space and the brain activity space is a simpler (hopefully linear) function than the original mapping



Relying on an intermediate representation space

Main assumption

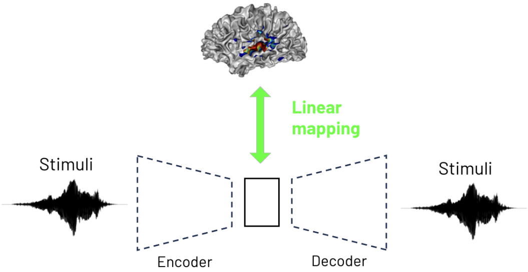
- One can efficiently learn higher-level representation of stimuli using large datasets, either in unsupervised or supervised way
- One may assume that the learned representation is a non-linear function of the input
- One may expect that the mapping between the intermediate representation space and the brain activity space is a simpler (hopefully linear) function than the original mapping



Relying on an intermediate representation space

Main assumption

- One can efficiently learn higher-level representation of stimuli using large datasets, either in unsupervised or supervised way
- One may assume that the learned representation is a non-linear function of the input
- One may expect that the mapping between the intermediate representation space and the brain activity space is a simpler (hopefully linear) function than the original mapping



Relying on an intermediate representation space

Main assumption

- One can efficiently learn higher-level representation of stimuli using large datasets, either in unsupervised or supervised way
- One may assume that the learned representation is a non-linear function of the input
- One may expect that the mapping between the intermediate representation space and the brain activity space is a simpler (hopefully linear) function than the original mapping

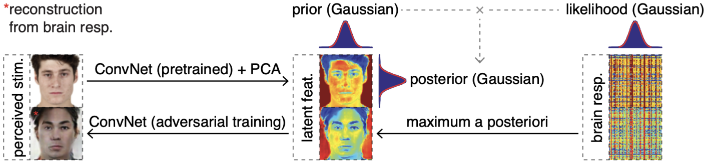
Method

- Decouple the learning task into
    - A non-linear mapping learned with a large dataset
    - A simpler (linear) mapping for learning the mapping with MRI data (with fewer training samples)
- Relies in the hypothesis that the mapping will be easier to learn
- Does depend on the nonlinear mapping.

Method

- Use a pretrained encoder model ϕ (extractor part of a convnet, VGG-Face) as in (Güçlü and Gerven [GG15])
- Learn a linear predictor from latent space to brain space
- Invert it
- Learn a decoder (ϕ^{-1}) with adversarial learning using a discriminator ψ

$$-\lambda_{adv}\mathbb{E}[log(\psi(\phi^{-1}(z)))] + \lambda_{feature}\mathbb{E}[||\xi(x) - \xi(\phi^{-1}(z)||^2] + \lambda_{sti}\mathbb{E}[||x - \phi^{-1}(z)||^2]$$

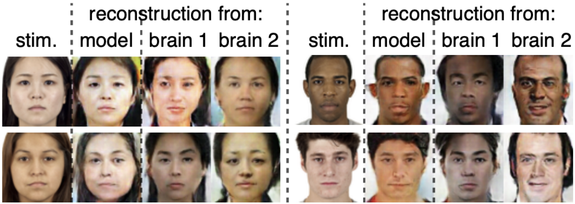


Combining probabilistic inference with deep learning [Güç+17]

Method

- Use a pretrained encoder model ϕ (extractor part of a convnet, VGG-Face) as in (Güçlü and Gerven [GG15])
- Learn a linear predictor from latent space to brain space
- Invert it
- Learn a decoder (ϕ^{-1}) with adversarial learning using a discriminator ψ

$$-\lambda_{adv}\mathbb{E}[log(\psi(\phi^{-1}(z)))] + \lambda_{feature}\mathbb{E}[||\xi(x) - \xi(\phi^{-1}(z)||^2] + \lambda_{sti}\mathbb{E}[||x - \phi^{-1}(z)||^2]$$



Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

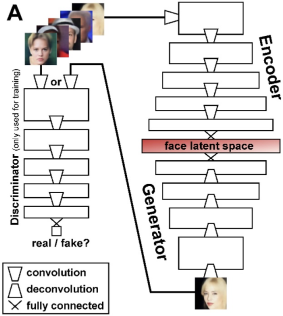
- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space

$$y = W \times z$$

  - ...and inverse it for decoding

$$\hat{z} = (W^T W)^{-1} W^T \times y$$

$$\Rightarrow \hat{x} = D(\hat{z})$$

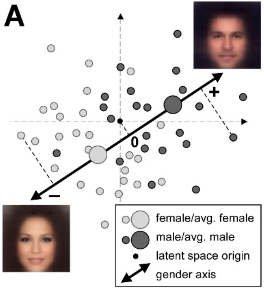


Model used

Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space
  $$y = W \times z$$
  - ...and inverse it for decoding
  $$\hat{z} = (W^T W)^{-1} W^T \times y$$
  $$\Rightarrow \hat{x} = D(\hat{z})$$

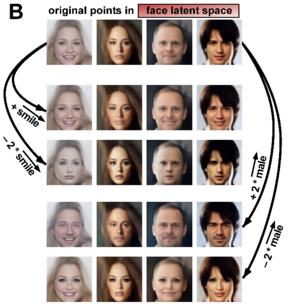


The latent space

Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space
    $$y = W \times z$$
  - ...and inverse it for decoding
    $$\hat{z} = (W^T W)^{-1} W^T \times y$$
    $$\Rightarrow \hat{x} = D(\hat{z})$$

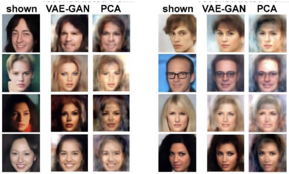


Editing in the latent space

Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space
    $$y = W \times z$$
  - ...and inverse it for decoding
    $$\hat{z} = (W^T W)^{-1} W^T \times y$$
    $$\Rightarrow \hat{x} = D(\hat{z})$$



Reconstruction from subjects 1 and 2

Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

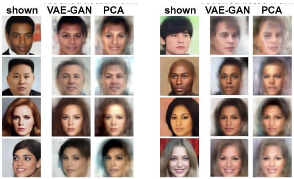
- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space

$$y = W \times z$$

  - ...and inverse it for decoding

$$\hat{z} = (W^T W)^{-1} W^T \times y$$

$$\Rightarrow \hat{x} = D(\hat{z})$$



Reconstruction from subjects 3 and 4

Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

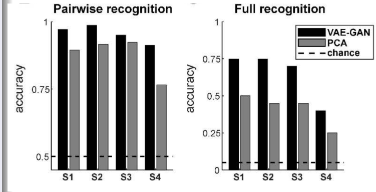
- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space

$$y = W \times z$$

  - ...and inverse it for decoding

$$\hat{z} = (W^T W)^{-1} W^T \times y$$

$$\Rightarrow \hat{x} = D(\hat{z})$$



Recognition accuracy

Decoding with a deep Adversarial Autoencoder (VanRullen and Reddy [VR19])

Method

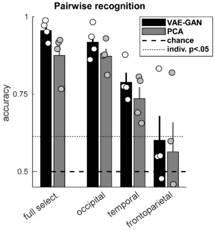
- Learn a Gan-VAE
  - i.e.: an autoencoder ($E + D$), whose reconstruction are driven to be more realistic using an adversarial discriminator
  - Input: $x \rightarrow$ hidden representation: $z = E(x)$ $\rightarrow$ reconstruction: $D(z) = D(E(x))$
- Then:
  - Learn a linear predictor from latent space to neural space

$$y = W \times z$$

  - ...and inverse it for decoding

$$\hat{z} = (W^T W)^{-1} W^T \times y$$

$$\Rightarrow \hat{x} = D(\hat{z})$$



Reconstruction accuracy per brain area

Outline

3. Few Deep Learning studies for MRI data
   - Objectives and means
   - Computational vs. neural representations
   - Comparing computational and neural representations
   - Encoding and decoding via intermediate representation space
   - Constraining the computational representation from neural representation

Enforcing Learning relevant representation for brain mapping

Drawback of previous approaches

- One expects the learned mapping by a pretrained or independently trained NN to enable a linear mapping between the latent space and the brain space
- This likely hides the selection of a "relevant" NN which exhibits such a behaviour, amongst a number of trained models.

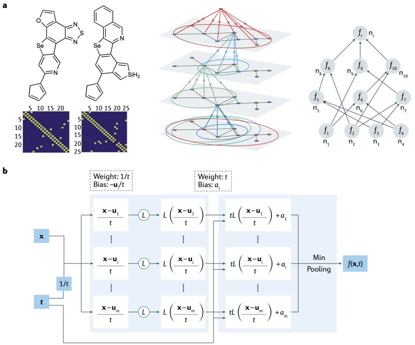
Two main answers

- Hard constraint / Inductive bias
- Soft constraint / Adding a loss term

Inductive bias in DL models

Extensively used

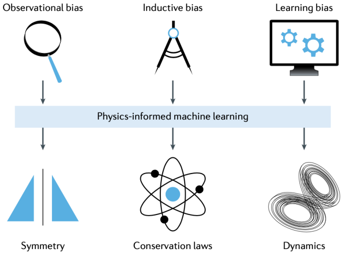
- A standard strategy
  - Convolution layers for images
  - Recurrent layers for times series
  - More recently in physics informed ML/DL



(Karniadakis et al. [Kar+21])

Extensively used

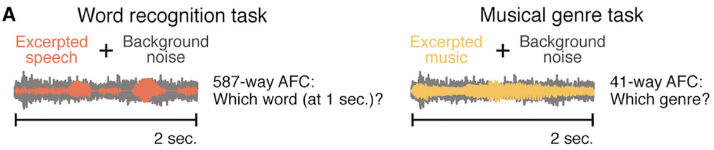
- A standard strategy
  - Convolution layers for images
  - Recurrent layers for times series
  - More recently in physics informed ML/DL



(Karniadakis et al. [Kar+21])

A Task-Optimized Neural Network Replicates Human Auditory Behavior...

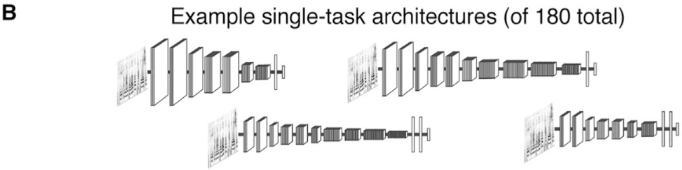
- Question: is auditory cortical computation hierarchical, potentially corresponding to cortical regions?
- Study: optimization (selection amongst $\approx$ 200 architectures) of the best architecture trained for two tasks: word recognition and musical genre identification
  - Best model contain separate music and speech path-ways following early shared processing, *potentially replicating human cortical organization*
- Results
  - Human-like errors
  - Hierarchical organization



Implicit bias for audio processing (Kell et al. [Kel+18])

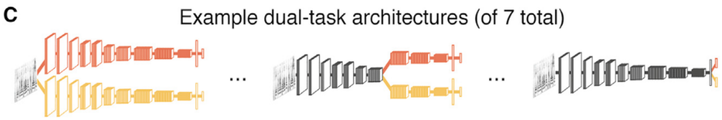
A Task-Optimized Neural Network Replicates Human Auditory Behavior...

- Question: is auditory cortical computation hierarchical, potentially corresponding to cortical regions?
- Study: optimization (selection amongst $\approx$ 200 architectures) of the best architecture trained for two tasks: word recognition and musical genre identification
  - Best model contain separate music and speech path-ways following early shared processing, *potentially replicating human cortical organization*
- Results
  - Human-like errors
  - Hierarchical organization



A Task-Optimized Neural Network Replicates Human Auditory Behavior...

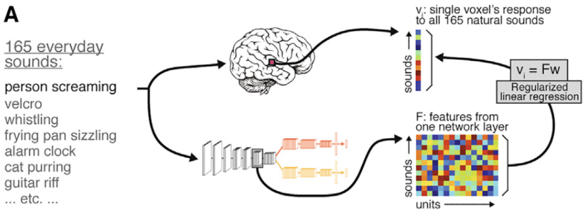
- Question: is auditory cortical computation hierarchical, potentially corresponding to cortical regions?
- Study: optimization (selection amongst ≈ 200 architectures) of the best architecture trained for two tasks: word recognition and musical genre identification
  - Best model contain separate music and speech path-ways following early shared processing, *potentially replicating human cortical organization*
- Results
  - Human-like errors
  - Hierarchical organization



Implicit bias for audio processing (Kell et al. [Kel+18])

A Task-Optimized Neural Network Replicates Human Auditory Behavior...

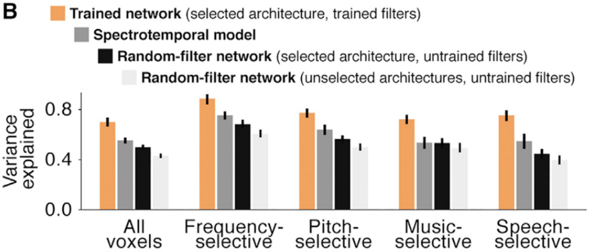
- Question: is auditory cortical computation hierarchical, potentially corresponding to cortical regions?
- Study: optimization (selection amongst ≈ 200 architectures) of the best architecture trained for two tasks: word recognition and musical genre identification
  - Best model contain separate music and speech path-ways following early shared processing, *potentially replicating human cortical organization*
- Results
  - Human-like errors
  - Hierarchical organization



Implicit bias for audio processing (Kell et al. [Kel+18])

A Task-Optimized Neural Network Replicates Human Auditory Behavior...

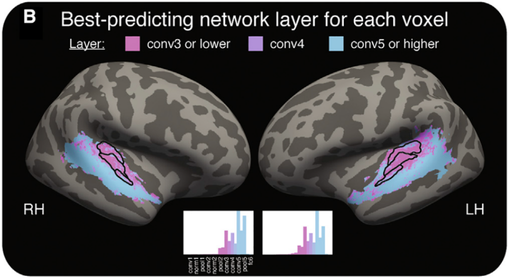
- Question: is auditory cortical computation hierarchical, potentially corresponding to cortical regions?
- Study: optimization (selection amongst ≈ 200 architectures) of the best architecture trained for two tasks: word recognition and musical genre identification
  - Best model contain separate music and speech path-ways following early shared processing, *potentially replicating human cortical organization*
- Results
  - Human-like errors
  - Hierarchical organization



Implicit bias for audio processing (Kell et al. [Kel+18])

A Task-Optimized Neural Network Replicates Human Auditory Behavior...

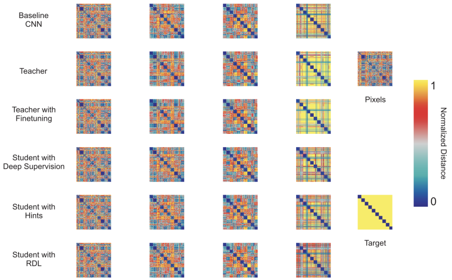
- Question: is auditory cortical computation hierarchical, potentially corresponding to cortical regions?
- Study: optimization (selection amongst $\approx$ 200 architectures) of the best architecture trained for two tasks: word recognition and musical genre identification
    - Best model contain separate music and speech path-ways following early shared processing, *potentially replicating human cortical organization*
- Results
    - Human-like errors
    - Hierarchical organization



Constraining the representation space to match neural RDMs

First attempt (McClure and Kriegeskorte [MK16])

- Motivation: Studies have shown that DNNs trained for object recognition learn similar representations to those found in the human ventral stream
- How to encourage this
    - Add prediction from intermediate layers (comes next)
    - Enforce the hidden representation spaces to replicate RDMs from brain responses
- First tries on MNIST and CIFAR data for learning a student model from a teacher model



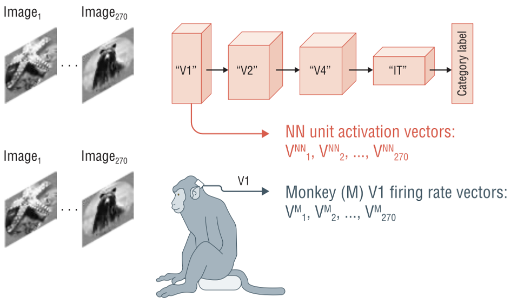
Constraining the representation space to match neural RDMs

Learn a model to match RDM of stimuli in the brain (Federer et al. [Fed+20])

- Applied with Macaques (actually not MRI data but neural firing rates recordings)
  - Cost function using RDM constraint on layer l

$$\lambda \sum_{i,j} ||RDM_{i,j}^{macaque} - RDM_{i,j}^{NN(l)}||^2 + Loss_{Classif}$$

  - λ updated so that the ratio of the two loss term remains equal to a constant r
  - Use the combined loss for a few epochs then use the classification loss only
- The DNN achieves better image classification results on CIFAR 100

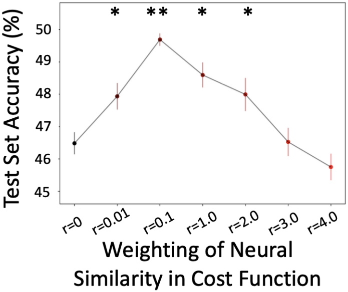


Learn a model to match RDM of stimuli in the brain (Federer et al. [Fed+20])

- Applied with Macaques (actually not MRI data but neural firing rates recordings)
  - Cost function using RDM constraint on layer l

$$\lambda \sum_{i,j} ||RDM_{i,j}^{macaque} - RDM_{i,j}^{NN(l)}||^2 + Loss_{Classif}$$

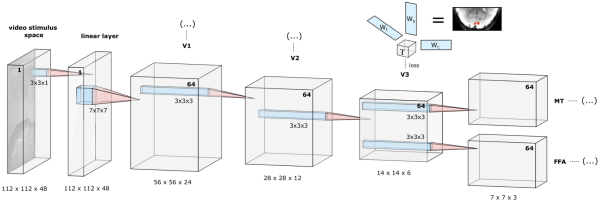
  - λ updated so that the ratio of the two loss term remains equal to a constant r
  - Use the combined loss for a few epochs then use the classification loss only
- The DNN achieves better image classification results on CIFAR 100



A Model of Neural computation (Seeliger et al. [See+21])

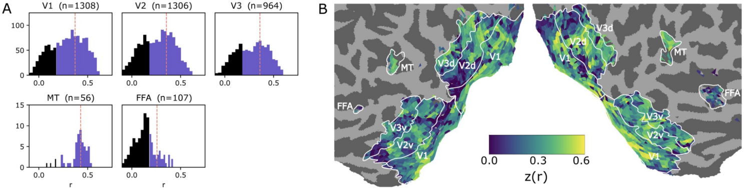
Neural Information Flow

- Data driven approach that represent neural information processing between different cortical areas
- No computational task / End to end learning
- Method: Add predictors from hidden layers to predefined brain regions
  - Introduce simplifying assumption in the predictors to enable learning from small data



Neural Information Flow

- Data driven approach that represent neural information processing between different cortical areas
- No computational task / End to end learning
- Method: Add predictors from hidden layers to predefined brain regions
    - Introduce simplifying assumption in the predictors to enable learning from small data



Outline

Outline

4. A focus on speaker decoding
   - Basics
   - Approach
   - Results

A focus on decoding vocal (identity) from the brain (Lamothe et al. [Lam+24])

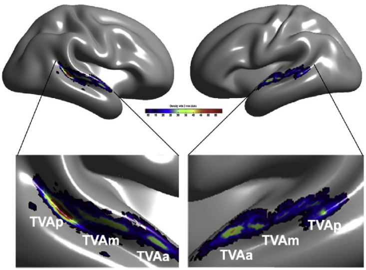
Joint work with

- Charly Lamothe (His PhD's work)
- INT people: Pascal Belin, Bruno Giordano, Etienne Thoret, Sylvain Takerkard

A focus on decoding vocal (identity) from the brain (Lamothe et al. [Lam+24])

The vocal brain

- The brain areas that process the audio vocal signal "before linguistics"
- How voice information is represented in neuronal populations ?
- More particularly how speaker identity (including gender, age etc) is encoded ?



Extracted from Pernet et al., 2015

TVAs / voice areas

- The cerebral processing of voice information involves a set of temporal areas (TVAs) in second auditory cortical regions
- The TVAs respond more strongly to sounds of voice but the nature of the information encoded at these stages (especially related to speaker identity) remains largely unknown

Questions

Problems

- Still poorly understood
- Much less studied/known than the neural bases of speech processing and of visual processing
- Not so clear existence of a hierarchy of representations such as in vision areas

In this study

- Q1: How does the VLS (Voice Latent Space) account for the brain responses to speaker identities in A1 and the TVAs? How does it compare to a linear latent space?
- Q2: How does the geometry of the VLS account for the representational geometry for voice identities in the auditory cortex?
- Q3: How well can we reconstruct a stimulus from brain activity?

Voice Data (Ardila et al. [Ard+19])

Preprocessing

- **Short** sample signals (250ms) Example
  - To emphasize speaker identity over linguistic information
  - Allows presentation of many more stimuli
- Features input to DNNs
  - Amplitude spectrograms: (21 time steps $\times$ 401 frequency bins)

DNN training data

- About 182k voice samples (250ms long)
- 405 speakers / 8 languages

MRI Data

MRI data (gathered at INT/INS in Marseille)

- 3 healthy volunteers...
- ... were scanned over 10+ hours...
- ...in response to $\approx$ 12000 voice samples (called BrainVoice dataset hereafter, different from the training set of DNNs)
- Different sets of stimuli were used for each participant: samples from 119 speakers in 8 languages.

Comments

- Too few subjects to generalize

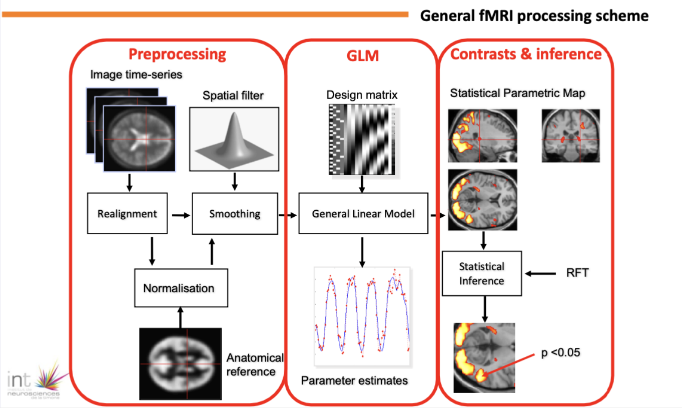
MRI Pre-processing steps

We are far from csv-like data ML practitioners love so much

- Many sources of noise and variability
  - Pure noise
  - Inter-subject variability (brain areas, shape etc)
  - Distracting activity while scanning (motion etc)
  - Hemodynamic response
- The engineering of presenting audio in the MRI device...
- Few steps
  - Aligning: Account for brain diversity
  - Denoising: Many distracting information in the brain
  - Gathering relevant information with GLM models

MRI data: A complex pipeline

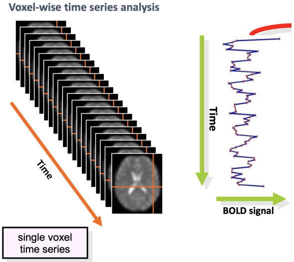
Borrowed from Pascal Belin (INT, Marseille)



Design Matrix

Images borrowed from a tutorial by P. Belin (INT, Marseille)

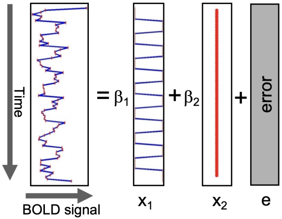
- 1 time series / voxel
- The time series for all voxels y is the product of X, the design matrix, and β
- The design matrix is built from indicators (0/1 signals from one-hot-encodings) and additional regressors
  - 1 if speaker i is speaking, 0 otherwise
  - 1 if stimuli i is played, 0 otherwise
- The design matrix (indicators only) is convolved with a HRF (Hemodynamic Response)



Design Matrix

Images borrowed from a tutorial by P. Belin (INT, Marseille)

- 1 time series / voxel
- The time series for all voxels y is the product of X, the design matrix, and β
- The design matrix is built from indicators (0/1 signals from one-hot-encodings) and additional regressors
  - 1 if speaker i is speaking, 0 otherwise
  - 1 if stimuli i is played, 0 otherwise
- The design matrix (indicators only) is convolved with a HRF (Hemodynamic Response)

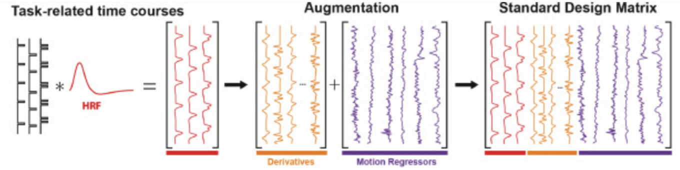


$$y = x_1\beta_1 + x_2\beta_2 + e$$

Design Matrix

Images borrowed from a tutorial by P. Belin (INT, Marseille)

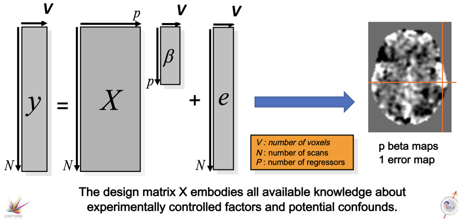
- 1 time series / voxel
- The time series for all voxels y is the product of X, the design matrix, and β
- The design matrix is built from indicators (0/1 signals from one-hot-encodings) and additional regressors
  - 1 if speaker i is speaking, 0 otherwise
  - 1 if stimuli i is played, 0 otherwise
- The design matrix (indicators only) is convolved with a HRF (Hemodynamic Response)



Design Matrix

Images borrowed from a tutorial by P. Belin (INT, Marseille)

- 1 time series / voxel
- The time series for all voxels y is the product of X, the design matrix, and β
- The design matrix is built from indicators (0/1 signals from one-hot-encodings) and additional regressors
  - 1 if speaker i is speaking, 0 otherwise
  - 1 if stimuli i is played, 0 otherwise
- The design matrix (indicators only) is convolved with a HRF (Hemodynamic Response)



A GLM model

Design matrix and regressors

- Bold signal $Y \in \mathbb{R}^{S\times ?}$
- Considered regressors $X \in \mathbb{R}^{T\times V}$
- Assumption: $Y = X \times \beta$ with $\beta \in \mathbb{R}^{V\times p}$
- Least square approximation:

$$\hat{\beta} = \arg\max_{\beta} \|Y - X \times \beta\|^2 + \Omega(\beta)$$

- Usage
  - X includes regressors for information one wants to remove $\Rightarrow$ denoise with $Y_d = Y - X \times \hat{\beta}$
  - X includes regressors that we are interested in $\Rightarrow$ $\hat{\beta}$ becomes the quantity of interest

First GLM: Denoising

- Desig matrix X: 36 regressors motion and head and ...
- Convolve X with hemodynamic response (still noted X)
- Precict Y from regressors:

$$\beta_d = \arg\max_{\beta} \|Y - X \times \beta\|^2 + \Omega(\beta)$$

- Remove noise predicted from distracting and irrelevant regressors

$$Y_d = Y - X \times \beta_d$$

Second GLM: Stimuli representation

- Design matrix $X \in \mathbb{R}^{S\times(N+1)}$ (with $N = 6000$): Stimuli regressors
- Convolve X with hemodynamic response (still noted X)
- Predict Y from regressors:

$$\beta_s = \arg\max_{\beta} \|Y - X \times \beta\|^2 + \Omega(\beta)$$

- Model the silence through one (last) regressor, removed by substraction
- $\beta_s[i,:]$ are stimuli representations

representation

- Design matrix $X \in \mathbb{R}^{S \times (N_i+1)}$ with $N_i = 415$: Identity regressors
- Convolve X with hemodynamic response (still noted X)
- Predict Y from regressors:

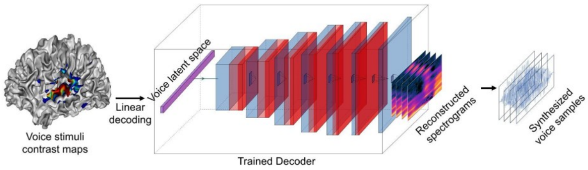
$$\beta_i = \arg\max_{\beta} \| Y - X \times \beta \|^2 + \Omega(\beta)$$

- Model the silence through one (last) regressor, removed by substraction
- $\beta_i[s,:] \in \mathbb{R}^V$ is speaker s's representation

Outline

Rather simple

- Very similar to (VanRullen and Reddy [VR19])
- Pretrained VAE ⇒ unsupervised learning rather than task-oriented DNN
  - Comparison with a linear model
- Learn a linear predictor from neural representation to latent space
- Use the decoder to reconstruct a spectrogram from inferred latent representation
- Few attempts to improve the baseline
  - Use a reconstruction loss defined on mfcc rather than on spectrograms
  - Learn an adversarial discriminator to "beautify" the reconstruded spectrograms
  - Joint learning of the autoendoder and of the linear mapping from brain space to latent space
  - Add a RDM constraint on the latent space



Outline

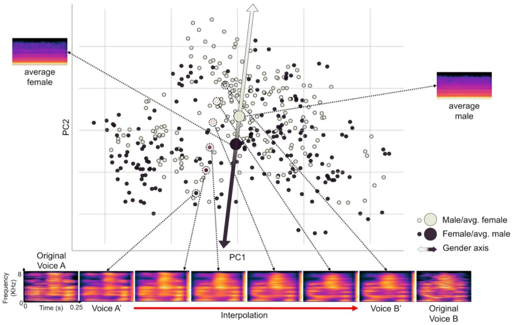
4. A focus on speaker decoding
   - Basics
   - Approach
   - Results

Traversal between 2 samples

- Voice A to Voice B
- A – – – – B

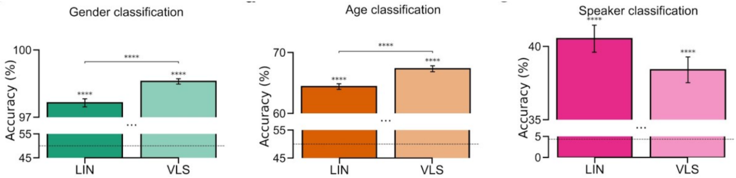
Feminizing a sample

- Changing sample A by decoding $z_A + z_{female} - z_{male}$
- Original - Feminized



Assessing speaker identity encoding

- Compute a latent vector per speaker by averaging latent vectors of all his stimulus
- Probe the informational content by learning a linear classifier to predict gender / age / identity
- All results significantly above chance (student test)



Questions

In this study

- *Q1: How does the VLS account for the brain responses to speaker identities in A1 and the TVAs? How does it compare to a linear latent space?* $\Rightarrow$ Encoding speaker identity study
- Q2: How does the geometry of the VLS account for the representational geometry for voice identities in the auditory cortex?
- Q3: How well can we reconstruct a stimulus from brain activity?

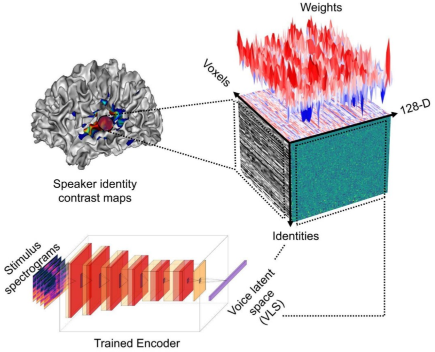
Encoding speaker identity

Method

- Compute β_i's (speaker sensitivity maps)
- Learn a linear regression model to predict β_i from the latent of speaker i
- Perform the study for each TVA
- Assess performance in a cross Validation setting

Results

- Significativity analysis using ANOVA and Student t-test
  - ANOVA shows strong effect of feature (LIN vs VLS) and ROI
- Models are complementary. No significant advantage of one over the other
- Note the level of (Pearson) correlations (distribution over the voxels)



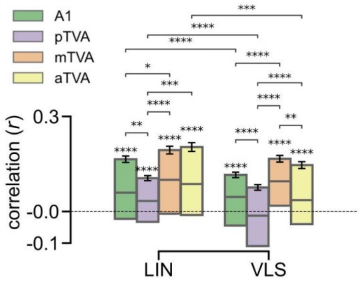
Encoding speaker identity

Method

- Compute β_i's (speaker sensitivity maps)
- Learn a linear regression model to predict β_i from the latent of speaker i
- Perform the study for each TVA
- Assess performance in a cross Validation setting

Results

- Significativity analysis using ANOVA and Student t-test
  - ANOVA shows strong effect of feature (LIN vs VLS) and ROI
- Models are complementary. No significant advantage of one over the other
- Note the level of (Pearson) correlations (distribution over the voxels)



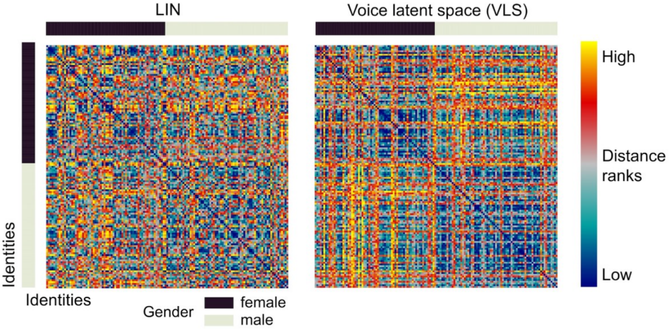
Questions

In this study

- Q1: How does the VLS account for the brain responses to speaker identities in A1 and the TVAs? How does it compare to a linear latent space?
- *Q2: How does the geometry of the VLS account for the representational geometry for voice identities in the auditory cortex?* $\Rightarrow$ RSA study at the speaker level
- Q3: How well can we reconstruct a stimulus from brain activity?

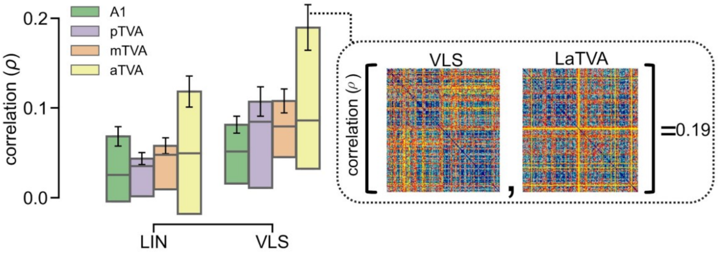
The geometry of the VLS space better matches that of TVAs

- Method
  - One RDMs for each ROI (A1, aTVA, mTVA, pTVA) and hemisphere (dissimilarity using Pearson's correlation)
  - One RDM per model (dissimilarity using cosine distance)
  - Similarity between two RDMS (two representations spaces) is computed as Spearman correlation coefficient
  - Statistical test to compare to null correlation using random permutations of the model's RDM columns



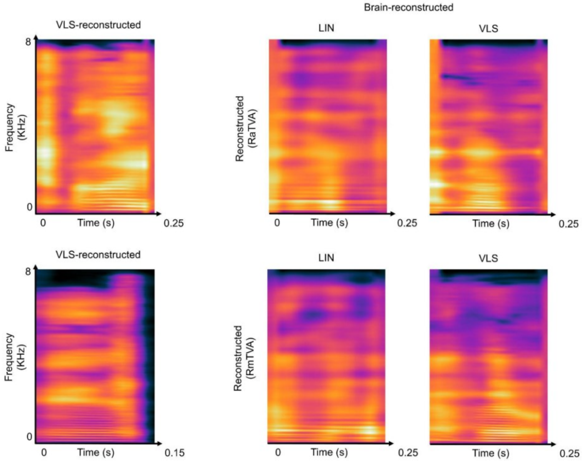
The geometry of the VLS space better matches that of TVAs

- Method
  - One RDMs for each ROI (A1, aTVA, mTVA, pTVA) and hemisphere (dissimilarity using Pearson's correlation)
  - One RDM per model (dissimilarity using cosine distance)
  - Similarity between two RDMS (two representations spaces) is computed as Spearman correlation coefficient
  - Statistical test to compare to null correlation using random permutations of the model's RDM columns



Decoding examples

- Example 1 (top): VLS reconstructed - Brain Lin - Brain NLin
- Example 2 (bottom): VLS reconstructed - Brain Lin - Brain NLin



Objective and subjective results

Main results

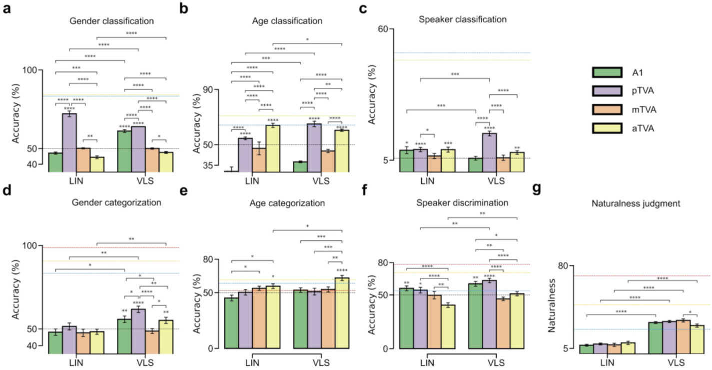
- NLin outperforms Lin to preserve genre, age and identity in at least one TVA
- pTVA outperforms other ROIs in gender, age and identity
- 13 human participants judged naturalness, gender, age, and speaker categorization. Better results for the NLin model in specific cases (naturalness for A1 and TVAs etc)

Performance measures

- Objective measures (top): linear classifiers for gender, age and identity
- Subjective measures : Listener performance at categorizing gender, age, and identity

Main results

- NLin outperforms Lin to preserve genre, age and identity in at least one TVA
- pTVA outperforms other ROIs in gender, age and identity
- 13 human participants judged naturalness, gender, age, and speaker categorization. Better results for the NLin model in specific cases (naturalness for A1 and TVAs etc)



Outline

Comments rather than a conclusion

Somehow different from traditional ML studies

- A neuroscience paper answers a neuroscientific question
  - Specific dataset and preprocessing
  - While benchmark datasets have been a motor for huge progress in ML
- Noisy data / hard tasks / etc
  - Often only weak conclusions (e.g. significantly different from random)
  - Sometimes (too) strong conclusions on the brain
- Risk of biased results?
  - Results might be biased towards expected results and/or prior knowledge

A challenging field for ML practitioners

- Understanding the brain is a fascinating field
- And a very difficult one
- Lots of interesting questions to answer which require innovation in ML and DL

[KMB08] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. "Representational similarity analysis - connecting the branches of systems neuroscience". In: *Frontiers in Systems Neuroscience* 2 (2008). ISSN: 1662-5137. DOI: 10.3389/neuro.06.004.2008. URL: https://www.frontiersin.org/articles/10.3389/neuro.06.004.2008.

[Goo+14] Ian J. Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 2672–2680. URL: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

[KW14] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6114.

[LM14] Quoc V. Le and Tomás Mikolov. "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 1188–1196. URL: http://proceedings.mlr.press/v32/le14.html.

[MO14] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

[GL15] Yaroslav Ganin and Victor S. Lempitsky. "Unsupervised Domain Adaptation by Backpropagation". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by Francis R. Bach and David M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1180–1189. URL: http://proceedings.mlr.press/v37/ganin15.html.

[GG15] Umut Güçlü and Marcel A. J. van Gerven. "Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream". In: *J. Neurosci.* 35.27 (July 2015), p. 10005. DOI: 10.1523/JNEUROSCI.5023-14.2015. URL: http://www.jneurosci.org/content/35/27/10005.abstract.

[Mak+15] Alireza Makhzani et al. "Adversarial Autoencoders". In: *CoRR* abs/1511.05644 (2015). arXiv: 1511.05644. URL: http://arxiv.org/abs/1511.05644.

[Güç+16] Umut Güçlü et al. "Brains on Beats". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al. 2016, pp. 2101–2109. URL: https://proceedings.neurips.cc/paper/2016/hash/b9d487a30398d42ecff55c228ed5652b-Abstract.html.

[MK16] Patrick McClure and Nikolaus Kriegeskorte. "Representational Distance Learning for Deep Neural Networks". In: *Frontiers in Computational Neuroscience* 10 (2016). ISSN: 1662-5188. DOI: 10.3389/fncom.2016.00131. URL: https://www.frontiersin.org/articles/10.3389/fncom.2016.00131.

[Per+16] Guim Perarnau et al. "Invertible Conditional GANs for image editing". In: *CoRR* abs/1611.06355 (2016). arXiv: 1611.06355. URL: http://arxiv.org/abs/1611.06355.

[Güç+17] Yagmur Güçlütürk et al. "Reconstructing perceived faces from brain activations with deep adversarial neural decoding". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 4246–4257. URL: https://proceedings.neurips.cc/paper/2017/hash/efdf562ce2fb0ad460fd8e9d33e57f57-Abstract.html.

[Kel+18] Alexander J. E. Kell et al. "A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy". In: *Neuron* 98 (2018), 630–644.e16. URL: https://api.semanticscholar.org/CorpusID:5084719.

[Loc+18] Francesco Locatello et al. "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *CoRR* abs/1811.12359 (2018). arXiv: 1811.12359. URL: http://arxiv.org/abs/1811.12359.

[Mak18] Alireza Makhzani. "Implicit Autoencoders". In: *CoRR* abs/1805.09804 (2018). arXiv: 1805.09804. URL: http://arxiv.org/abs/1805.09804.

Bibliography V

[Ard+19] Rosana Ardila et al. "Common Voice: A Massively-Multilingual Speech Corpus". In: *CoRR* abs/1912.06670 (2019). arXiv: 1912.06670. URL: http://arxiv.org/abs/1912.06670.

[Gei+19] Robert Geirhos et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=Bygh9j09KX.

[KD19] Nikolaus Kriegeskorte and Pamela K Douglas. "Interpreting encoding and decoding models". In: *Current Opinion in Neurobiology* 55 (2019). Machine Learning, Big Data, and Neuroscience, pp. 167–179. ISSN: 0959-4388. DOI: https://doi.org/10.1016/j.conb.2019.04.002. URL: https://www.sciencedirect.com/science/article/pii/S0959438818301004.

[VR19] Rufin VanRullen and Leila Reddy. "Reconstructing faces from fMRI patterns using deep generative neural networks". In: *Communications Biology* 2.1 (May 2019), p. 193. ISSN: 2399-3642. DOI: 10.1038/s42003-019-0438-y. URL: https://doi.org/10.1038/s42003-019-0438-y.

[Fed+20] Callie Federer et al. "Improved object recognition using neural networks trained to mimic the brain's statistical properties". In: *Neural Networks* 131 (2020), pp. 103–114. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2020.07.013. URL: https://www.sciencedirect.com/science/article/pii/S0893608020302549.

[Kar+21] George Em Karniadakis et al. "Physics-informed machine learning". English (US). In: *Nature Reviews Physics* 3.6 (June 2021). Publisher Copyright: © 2021, Springer Nature Limited., pp. 422–440. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00314-5.

[See+21] Katja Seeliger et al. "End-to-end neural system identification with neural information flow". In: *PLoS Comput. Biol.* 17.2 (2021). DOI: 10.1371/JOURNAL.PCBI.1008558. URL: https://doi.org/10.1371/journal.pcbi.1008558.

[Lam+24] Charly Lamothe et al. "Reconstructing Voice Identity from Noninvasive Auditory Cortex Recordings". In: *bioRxiv* (2024). DOI: 10.1101/2024.02.27.582302. eprint: https://www.biorxiv.org/content/early/2024/03/19/2024.02.27.582302.full.pdf. URL: https://www.biorxiv.org/content/early/2024/03/19/2024.02.27.582302.