Probing machine-learning classifiers using noise, bubbles, and reverse correlation

Etienne Thoret, Thomas Andrillon, Damien Léger, Daniel Pressnitzer

Probing machine-learning classifiers using noise, bubbles, and reverse correlation

Etienne Thoret[*1,4], Thomas Andrillon[3], Damien Léger[2], Daniel Pressnitzer[1]

[1] Laboratoire des systèmes perceptifs, Département d'études cognitives, École normale supérieure, PSL University, CNRS, 75005 Paris, France.

[2] Université de Paris, APHP, Hotel Dieu, Centre du Sommeil et de la Vigilance & EA 7330 VIFASOM, Paris 75006, France.

[3] Turner Institute for Brain & Mental Health and School of Psychological Sciences, Monash University, Melbourne 3168, Australia.

[4] Aix Marseille Univ, CNRS, PRISM, LIS, ILCB, Marseille, France

[*]corresponding author: etiennethoret@gmail.com

Abstract

Many scientific fields now use machine-learning tools to assist with complex classification tasks. In neuroscience, automatic classifiers may be useful to diagnose medical images, monitor electrophysiological signals, or decode perceptual and cognitive states from neural signals. Tools such as deep neural networks regularly outperform humans with such large and high-dimensional datasets. However, such tools often remain black-boxes: they lack interpretability. A lack of interpretability has obvious ethical implications for clinical applications, but it also limits the usefulness of machine-learning tools to formulate new theoretical hypotheses. Here, we propose a simple and versatile method to help characterize and understand the information used by a classifier to perform its task. The method is inspired by the reverse correlation framework familiar to neuroscientists. Specifically, noisy versions of training samples or, when the training set is unavailable, custom-generated noisy samples are fed to the classifier. Variants of the method using uniform noise and noise focused on subspaces of the input representations, so-called “bubbles”, are presented. Reverse correlation techniques are then adapted to extract both the discriminative information used by the classifier and the canonical information for each class. We provide illustrations of the method for the classification of written numbers by a convolutional deep neural network and for the classification of speech versus music by a support vector machine. The method itself is generic and can be applied to any kind of classifier and any kind of input data. Compared to other, more specialized approaches, we argue that the noise-probing method could provide a generic and intuitive interface between machine-learning tools and neuroscientists.

Keywords: Data analysis – Interpretability – Deep neural networks – Automatic classifiers – Reverse correlation – Auditory models

Introduction

Applications of machine-learning techniques permeate more and more scientific fields, with rapid and sometimes unexpected success (LeCun et al., 2015; Jordan & Mitschell, 2015; Krigeskorte & Douglas, 2018; Richards et al., 2019). At the same time, it is becoming a widely-acknowledged issue that many of these tools are often used as black boxes, and need to be interpreted (Molnar, 2020; Doshi-Velez & Kim, 2017). For instance, if a Deep Neural Network (DNN) was used to make life-changing decisions such as deciding on an intervention based on medical imagery, both the clinicians and patients would have a clear desire to know the rationale that motivated the decision. Also, the power of classifiers to detect useful information in large datasets holds many promises to improve theoretical models, but then, understanding at least to some extent the classifier's operation is crucial (Zihni et al., 2020).

Understanding what a complex classifier does after being trained on possibly millions or billions of samples is usually hard. It is hard for a reason: if the task that the classifier solves had a known explicit solution, then there probably would not have been any incentive to develop the classifier in the first place. In addition, modern techniques involve artificial network architectures with interconnected layers, each including highly non-linear operations (Sejnowski, Kienker, & Hinton, 1986). A lot of the computational power of such algorithms lies in such cascades of feed-forward and feed-back non-linear operations. Unfortunately, human reasoning seems most at ease to generate intuitions with linear processes, and not for complex combinations of non-linear ones.
As a consequence, designing methods to interpret machine learning tools is a fast-growing field of research of its own right, often designated under the term *Explainable AI* (Guidotti et al., 2018). It has dedicated journals within the machine learning community (e.g. *Distill*) and an associated DARPA challenge (*XAI*). Recent reviews covering the types of methods exist (Molnar, 2020), also covering more specifically the feature visualization approach taken here (Olah et al., 2017). Within this context, our aim is not to outperform the state-of-the art specialized interpretability methods, but rather to provide a general tool that will hopefully be intuitive to neuroscientists, as it is based on familiar methods for this community. The manuscript describes the method, provides an open

software library to use it, and shows examples of application, demonstrating how it can achieve useful results.

The gist of the method is to try and reveal the input features used by an automatic classifier, a black-box, to achieve its task *without any knowledge* about what is inside the black-box. As such, it is what is termed an "agnostic" method of explanation: it does not attempt to describe mechanistically the operation of a specific classifier, which it considers unknown (even if the classifier's details are available, as they may be too complex to understand intuitively). Rather, the aim is to relate features of the input space to the classifier's decisions. Such a problem is closely related to issues that neuroscientists and experimental psychologists have been addressing for years: providing useful insights for theoretical models of, for instance, human perception, without a full knowledge of the highly complex and non-linear underlying information processing performed by the brain.

In particular, the method we propose is directly inspired from the reverse correlation techniques developed for studying human vision (Ahumada et al., 1971; Neri et al., 1999; Gosselin & Shyns, 2001, 2003). Reverse correlation is based on linear systems analysis (Wiener, 1966). It uses stochastic perturbations of a system to observe its output. If the system were linear, an average of the inputs weighed by the observed outputs would be able to fully characterize the system. However, even for the highly non-linear systems studied by neuroscience, reverse correlation has a track-record of useful applications. For neurophysiology, averaging input stimuli according to neural firing rates has been used to describe neural selectivity (Ringach & Shapeley, 2004 for a review). For psychophysics, averaging input stimuli according to participant's decisions has revealed stimulus features on which such decisions are made for detection or discrimination tasks (Ahumada et al., 1971; Gosselin & Shyns, 2001, 2002). In this spirit, it seems appropriate to add reverse correlation to the toolbox of techniques to probe automatic classifiers, as its advantages and limitations are already well understood for non-linear systems.

One important benefit of using the reverse correlation framework is its complete independence from the underlying classifier's architecture. Unlike efficient but specific methods tuned to a classifier's architecture (see Guidotti et al., 2018 for a review), the reverse correlation can be used to probe any algorithm that separates the input data into distinct classes. Even for the

currently popular agnostic interpretability methods, this is not always the case: Class Activation Maps (Zhou et al., 2016) are specific to convolutional networks; LIME (Ribeiro et al., 2016) and RISE (Petsiuk et al., 2018) highlight features of specific examples which may or may not be representative of the classification task in general. Also, the method operates in the same representation space used as an input to the classifier, and can be applied to any type of representation (2D images, 1D time series such as audio, higher-dimensional brain imaging data, for instance).

The outline of the method is as follows. First, a set of stochastic inputs are generated, by introducing noise on the training dataset when available, or, when unavailable, by generating broad-band noise to cover systematically the input space. The noise takes two forms: additive noise, as is classically the case, but also multiplicative low-pass noise known as “bubbles” (Gosselin & Shyns, 2001, 2002) to focus the exploration on sub-spaces of the input representation. Second, the inputs are sorted according to the classification results. Third, inputs belonging to the same class are grouped together, with some refinements of the standard reverse correlation methods inspired by signal detection theory (Green & Wets, 1966) to weigh the results with the variability observed after classification. Two variants are described, aiming to probe two kinds of possibly overlapping but not necessarily identical input features: (1) the *discriminative* features, which correspond to the part of the input representation that is the most useful to ascribe a category (2) the *canonical* features, which correspond to the input features most representative of each category. In the machine-learning literature, these would loosely correspond to the “attribution” versus “feature visualization” problems (Ohla et al., 2017). In psychophysics, the distinction overlaps with the “potent information” (Gosselin et al., 2001) versus “prototypical information” (Rosch, 1983).

1. Material and Methods

1.1. Probing discriminative features

We term “discriminative features” the subspaces of the input space that are the most potent in the decision taken by the classifier (Gosselin & Shyns, 2002). The aim of this first method is to

visualize such subspaces in the input space. In the following, we assume that the classifier has been trained and is available to the probing method.

1.1.1. Procedure

To identify discriminative features, the input space is pseudo-randomly sampled with multiplicative low-pass filtered noise. The subspace enabling the highest classification performance is then identified by a reverse correlation analysis of all classified samples. The algorithm is directly inspired by the "bubbles" method (Gosselin & Shyns, 2001), originally designed to characterize the visual features underlying human behavioral performance for image classification tasks.

We present two sub-variants of the method, to account for the availability or not of the training set: a) multiplicative lowpass noise is applied to the training set; 1b) multiplicative lowpass noise is applied to broadband noise generated in the input stimulus space. We now describe the algorithm, jointly for a) and b). A textual description is provided as well as a software repository written in Python programming language (https://github.com/EtienneTho/proise) and a schematic illustration (Figure 1).

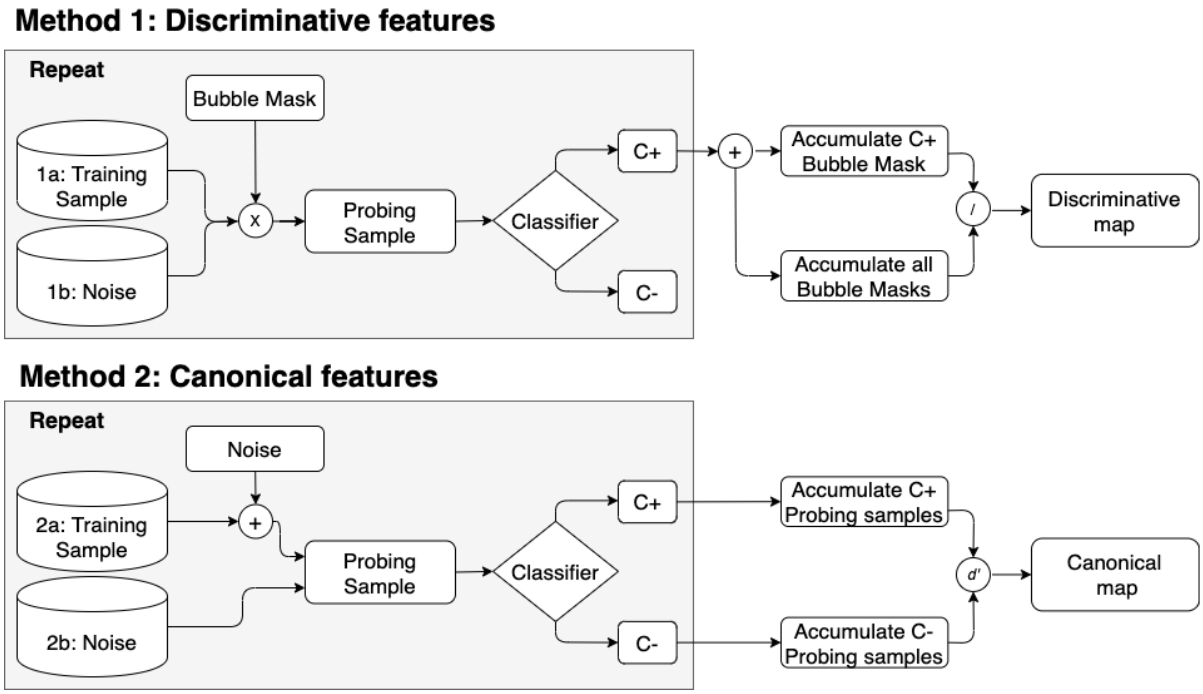


Fig. 1. *Summary of the two probing methods. Both method have 2 variants depending on the availability of the training set.* ***Method 1:*** *Training samples (1a) or noise (1b) are multiplied by bubble masks and then fed to the trained classifier. The bubble masks are then sorted according to*

the output of the classifier to compute the discriminative map. ***Method 2:*** *Training samples with additive noise (2a) or noise (2b) are fed to the trained classifier. The probing samples are then sorted according to the output of the classifier to compute the canonical map. For methods 1b and 2b, the noise can be either a gaussian noise or pseudo-random samples.*

For each pass (gray box in Fig. 1):

1. A bubble mask is generated. This consists of a mask in the input space, of dimension N, consisting of randomly positioned N-dimensional Gaussian windows (see Figure 2). The number of bubbles, *nbBubbles,* as well as the size of the bubbles in terms of the Gaussian standard deviations can be arbitrarily chosen and are parameters of the algorithm. In practice, an input array of dimension N populated by zeroes except for *nbBubbles* unit values is convolved with N-dimensional Gaussian windows. The resulting mask is denoted *BubbleMask*.
2. The probing data is generated. For variant a), the probing data is one exemplar of the training dataset, randomly chosen. For variant b), the probing data is an N-dimensional activation noise (see section 2.1.2 for details). The probing data is denoted *ProbingData*.
3. The probing sample is obtained by multiplying the bubble mask with the probing noise: *ProbingSample = BubbleMask * ProbingData.*
4. The probing sample is fed to the classifier and the output class is recorded. The probing sample is labeled C+ if it classified in the target class, C- otherwise.

Analysis:

For each point, *i*, in the stimulus space, the discriminative map for the class C+, $D_{i,C+}$, is computed as the sum of all C+ bubble masks divided by the sum of all masks C+ and C-: $D_{i,C+} = \frac{\sum BubbleMask_{C+,i}}{\sum BubbleMask_i}$ (Eq. 1). It should be noted that the analysis is performed on the bubble masks, and not on the probing samples.

1.1.2. Generation of noise activations when the training set is unavailable

As mentioned above, when the training set of the classifier is unavailable, the probing samples are generated from noise in the input space. The choice of the noise distributions is a free parameter of the method. The simplest choice is to draw samples from a uniform distribution at each point of the input space, covering the full range of valid input values. However, this sometimes leads to uneven coverage of the output categories, for instance if the classifier's boundaries are especially complex or if the decision algorithm is highly non-linear. In this case, we suggest to generate pseudo-random probing samples by first whitening the input dimensions, using a Principal Component Analysis (PCA). As the training set is unavailable, the PCA can be done on a representative set of inputs relative to the classifier's task. After the PCA, uniform noise can be generated in the low dimension space – which can be seen as the latent PCA's space – and inverted to obtain noise in the input space. The qualitative goal during the choice of the noise distribution is a balanced coverage of all output categories, and iterative choices may be a part of understanding the classifier's features.

1.1.3. Statistics

The discriminative maps show, in the input space, the features used by the classifier to assign samples to a given category. Visual inspection may be sufficient to get a qualitative understanding of the classifier's operation. However, in some cases, it is desirable to assess statistically the relevance of each part of the discriminative map.

There are many options to assess significance of such data, from which we outline one possible methodological choice. First, the maps can be shuffled by running the algorithm described above many times while randomly assigning output categories to each sample. For each point in the actual map, a *t*-test (or a non-parametric equivalent) is applied to compare the map value with the mean of the shuffled data. Maps are usually high-dimensional so a correction for multiple comparisons is needed. Again, several choices exist which are not specific to the methodology presented here, including Bonferroni correction, cluster permutation (Maris & Oostenveld, 2007), or False Discovery Rate (FDR) (Benjamini & Hochberg, 1995). In the illustrative examples below we only provide the raw discriminative and canonical maps, without statistics.

1.2. Probing canonical features

We define "canonical features" as the representation, in the input space, that would best match the different items of a given class. As an analogy, the canonical information may be viewed as the centroid of a category in the input space.

1.2.1. Procedure

To build the canonical map, the whole input space is randomly perturbed, without bubbles, so the search is not focused on any subspace. The aim is to probe the whole feature space. Then, all probes classified as members of the same category are averaged, in a direct adaptation of the classic reverse correlation method. However, we introduce here two differences, though. First, for generality, we do not separate correct classifications from false positives or false negatives. This would require to know the training dataset or to have a large labeled testing dataset. Second, a normalization of the feature map is introduced, using standard deviations estimates at each point of the map. This facultative step serves to display units similar to *z*-scores and not arbitrary input values. Again, we propose two sub-variants of the algorithm depending on the availability or not of the training dataset: a) broadband noise in the input space is added to the training set; b) broadband noise is generated in the input space. We now describe the algorithm, jointly for a) and b). A textual description is provided as well as the scripts (https://github.com/EtienneTho/proise) and a schematic description (Figure 1).

For each pass:

1. The probing sample is generated. For a), the probing sample is one randomly chosen exemplar of the training dataset, with noise added. The goal is to perturb the input to introduce variability, so that only the most salient information (to the classifier) remains in the reverse correlation average. For b), the probing sample is an N-dimensional activation noise. The probing sample is denoted *ProbingSample*.
2. The probing sample is fed to the classifier and the output class is recorded. The probing sample is labeled C+ if it classified in the target class, C- otherwise.

Reverse correlation analysis:

- For each point, *i*, in the stimulus space, the discriminative information is computed as the mean of all C+ probing samples minus the mean of all C- probing samples, normalized by the standard deviation of all probing samples at this point in the input space:

$$P_i = \frac{mean(ProcingSample_{i,C+}) - mean(ProbingSample_{i,C-})}{std(ProbingSample)} \quad \text{(Eq. 2)}$$

This reverse correlation definition adds a normalization factor to the simple average, using the standard deviation observed over all probing samples. This normalization is inspired from the discriminability index *d'* of signal detection theory (Green & Swets, 1966). It aims to visually emphasize reliably high values in the canonical map, by transforming the input units to *z*-score units. Note also that in a binary classification task, P_i is symmetric for the two classes.

1.2.2. Dimensionality reduction

Depending on the architecture of the classifying pipeline and input space, the estimation of the canonical map with reverse correlation can be more or less efficient. In particular, a standard technique to improve efficiency when training a classifier is to reduce the number of dimensions of the input space, for instance by using PCA. (e.g., Patil et al., 2012). Here as well, the probing and reverse correlation analysis can be performed in the space with reduced dimensionality before inverting back to the original input space.

For the generation of probing noise in variant b), the same remarks made in section 2.1.2 apply, with the same use of PCA to shape the noise for a balanced coverage of all output classes.

1.2.3. Statistics

The statistical analysis of canonical maps can be done with the same tools as for discriminative maps, described in section 2.1.3.

2. Results

To illustrate the methods introduced above and their generality, we present two different use cases: interpreting the classification of handwritten digits, a visual task (2-D input space) performed with a deep neural network; interpreting the classification of speech versus music, an audio task (1-D time series converted to a 4-D auditory model) performed by a support vector machine. These two cases also cover binary versus multiclass decisions. Although voluntarily simple, these examples should cover most of the ingredients needed for use cases relevant to neuroscience, such as vocal

classification (Paquette et al., 2018), biomedical images classification (Wang et al., 2016), EEG decoding (King & Dehaene, 2015), Multi-Voxel Pattern Analysis (Formisano et al., 2008).

2.1. Digits classification

In this first example, we classify visual samples of handwritten digits from the *MNIST* database (Deng, 2012). This is a standard database for evaluating image classification algorithms in the machine-learning community. It is composed of handwritten digits, from 0 to 9, with 60000 samples in the training set and 10000 samples in the test set. Each sample is a two-dimensional greyscale image with pixels values between 0 and 1.

Many algorithms can now successfully perform this classification task. Here we trained a Convolutional Neural Network (CNN) to discriminate between digits, with the following architecture: 2D-convolutional layer (3, 3), Max Polling layer (2, 2), 2D-convolution layer (3, 3), flattening layer, dense layer with 10 outputs and a softmax activation. Three epochs were run and, as expected, a high classification accuracy of 97% was obtained on the test set.

The CNN was probed to visualize the output of our algorithms for discriminative and canonical features. The two variants, with the training set available and without the training set, were compared. Figure 2 visually illustrates the method on these 2-D examples, for the bubbles variant.

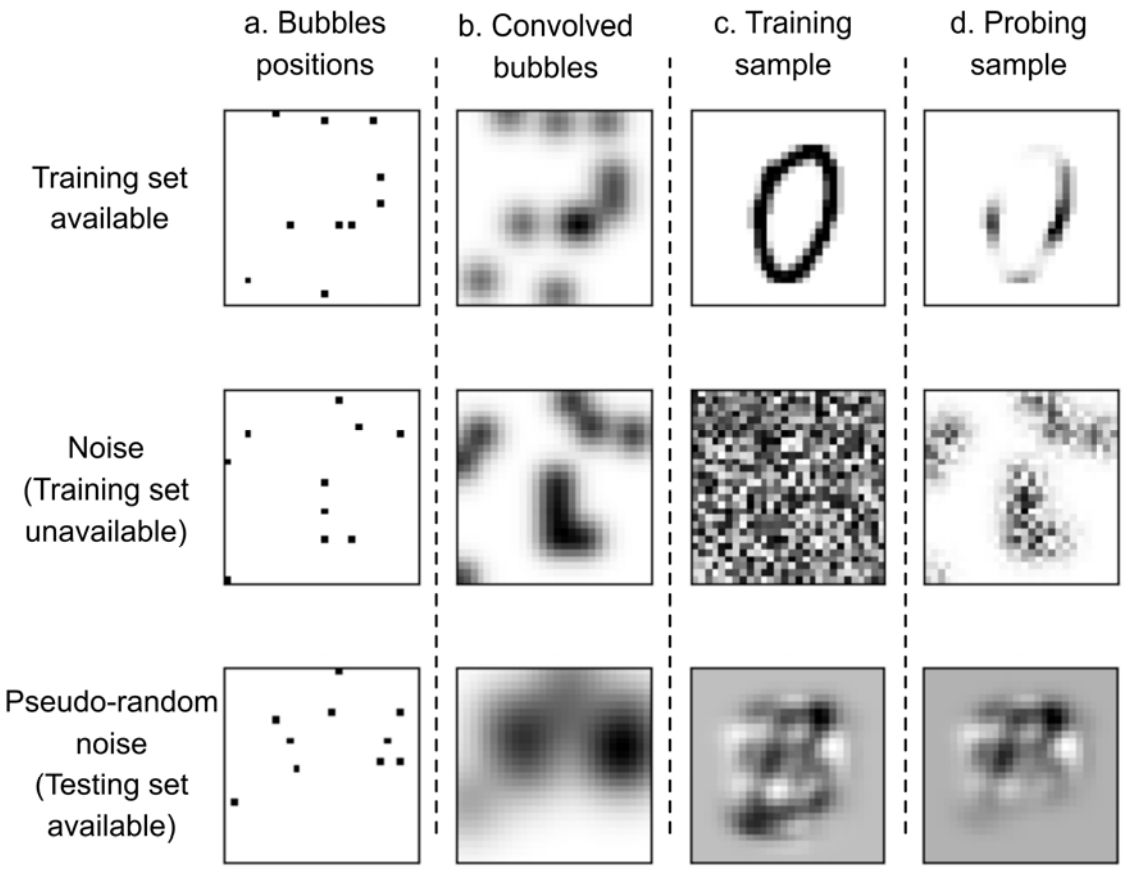


Fig. 2. Illustration of the probing method with bubbles on 2-D images. Top row: construction of the probing samples for discriminative features with the training set available. a) *Random placement of bubbles* ***b)*** *Convolution with 2-D Gaussian distributions to obtain the bubble mask* ***c)*** *A training sample* ***d)*** *Bubble mask applied to the training sample to obtain the probing sample.* ***Middle row: discriminative features with training set unavailable, uniform noise.*** *Columns as above. Using such probing samples do not cover the output categories efficiently (see text), likely because of their qualitative differences with digits.* ***Bottom row: discriminative features with training set unavailable, dimension-reduced noise.*** *Here noise is generated in a reduced dimension space extracted from PCA over the test set. The resulting probing sample is not a recognizable digit but shares visual features with actual digits.*

Figure 3 shows the discriminative features obtained with Methods 1a) and 1b), expressed as the Discriminative Index of Eq. 1 visualized in the 2-D input space (the visual image). The resulting

classification, that is, the position of the bubbles most useful for identifying each class. These do not need to correspond to the actual shape of a digit (which will be targeted by canonical features later on). For example, for the digit "1", the most useful regions are *around* the digit: knowing that there are no active pixels in such surrounding regions is most efficient for deciding that the narrow-shape of "1" was the input. For the digit "7", the discriminative map highlights the top-right corner, which corresponds to the position of a sharp angle unique to "7". In summary, while these maps may not make immediate intuitive sense on their own, they do orient the analysis of the input set towards regions of interest. Moreover, if the task was now to classify "7" versus all other digits, the input space could be weighed to emphasize the top-right corner to simplify the new classifier.

The availability of training data is expected to provide faster and more robust convergence towards the features of interest. For each case, 60000 probing samples and 10 bubbles with standard deviation of 4 pixels were used. In the case of Method 1b), a uniform random noise was first tested but only lead to categorization in 5 digits categories, so a pseudo random noise obtained from the inversion of a PCA was instead used generated to probe the CNN. This new noise led to decisions covering the 10 categories. The discriminative information obtained in the two cases correlate strongly (r = .92 (SD = .01), df = 783, $p < 10^{-3}$), showing that the methods' sub-variants with or without the training dataset converge toward the same masks.

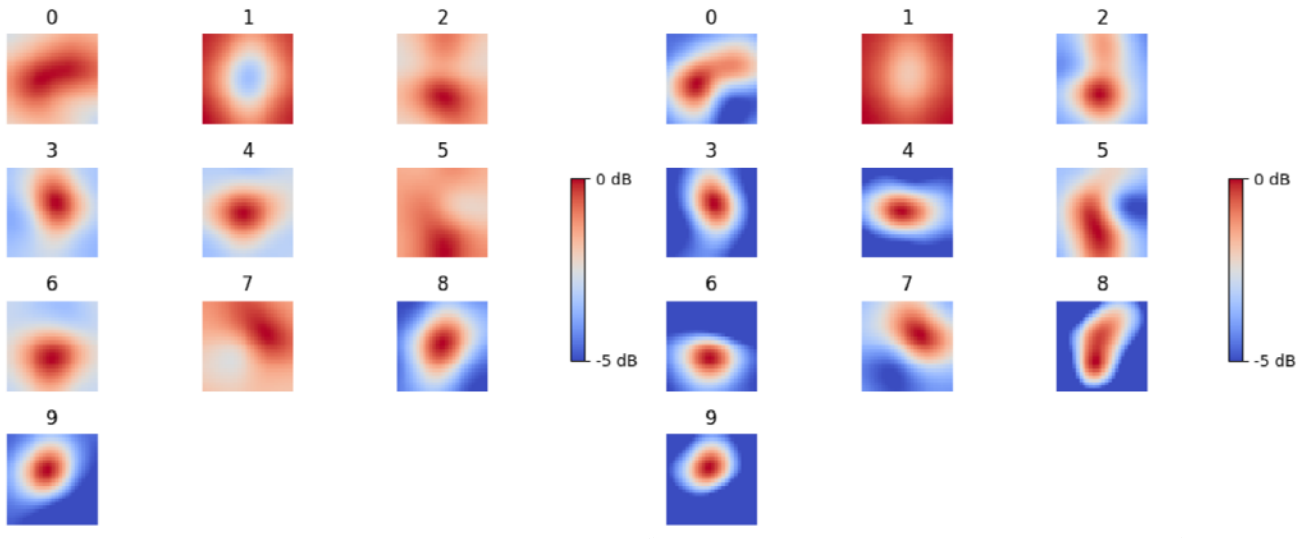


Fig. 3. Discriminative features maps for a CNN classifying handwritten digits. *The maps show the discriminative maps, in dB, obtained for each digit with Method 1a) with the training set available (left) and Method 1b) with the training set unavailable (right). Regions in red correspond to sub-spaces of the input most important for a correct classification of each digit. The maps are here normalized for each digit and presented in dB (20 $\log_{10} P_i/P_{max}$) for a sake of comparability between digits.*

Figure 4 shows the canonical features obtained with Methods 2a) and 2b), with or without the training set available. These canonical maps look different from the discriminative ones. Here, the maps are weighted averages of probing samples themselves, and not low-pass bubble masks, so finer details are available. As a result, and as intended with a reverse correlation approach, the canonical maps are readily identifiable and visually resemble the average written digits' representation. Such insight is perhaps not very surprising with simple digits, except perhaps for the 'negative' regions in blue that further specify which features are canonically absent from a given digit. Again, Methods 2a) and 2b) provide strongly correlated maps (r = .67 (SD = .18), df = 783, $p < 10^{-3}$). It can nevertheless be noted that Method 2b) tends to focus on the center of the input space. In particular, some border pixels were never associated with one or the other classification decision, leading to missing values when computing *d'*.

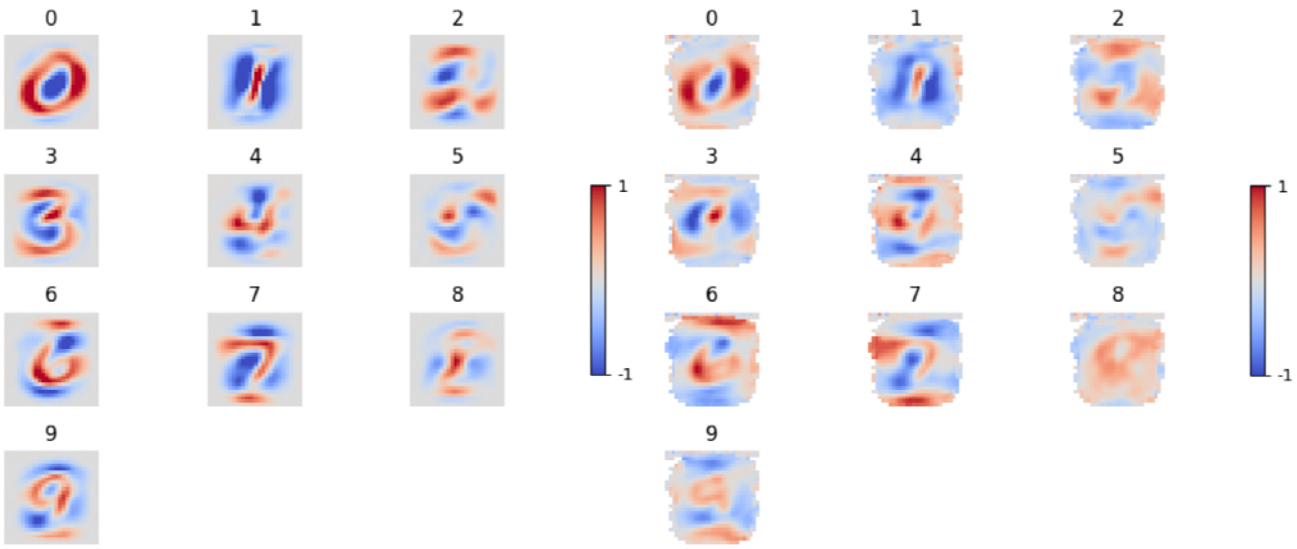


Fig. 4. Canonical features maps for a CNN classifying handwritten digits. *The maps show the d' sensitivity index for each point of the input space, obtained with Method 2a) (left) and Method 2b) (right). The red portions of the maps indicate the input features most associated with a given class. They visually resemble each digit, more or less blurred. The blue portions of the maps indicate the input features that are most reliably not present for a given class.*

2.2. Speech vs. music

In this second example, we classified audio samples in a speech versus music task. We used the GTZAN database composed of 132 excerpts of speech and music (Tzanetakis & Cook, 2002). The database was preprocessed to create samples with a fixed duration of 5 seconds, leading to a dataset of 768 samples. Those samples were randomly separated into a training set (691 excerpts) and a test set (77 excerpts, 10% of the dataset).

Following Patil et al. (2012), who performed an automatic classification of the musical timbre of short audio samples, sounds were first processed by an auditory model (Chi et al., 2005). The idea is to cast the input space into a representation that is interpretable in terms of auditory processing, unlike the raw waveform representation. Briefly, a filterbank corresponding to cochlear tonotopy is initially applied, followed by a 2-D Fourier analysis of the resulting time-frequency representation. The model output thus represents temporal modulations and spectral modulations contained in the

input sound (Chi et al., 2005, Elliot & Theunissen, 2009). The 4-D resulting arrays, with dimensions of time, frequency, scale of spectral modulations, and rate of temporal modulation, are termed here Spectro-Temporal Modulation representations (STM). We averaged the time dimension over the 5s of each sample. Next, we applied a PCA to reduce dimensionality (30976 dimensions in our implementation: 128 frequency channels x 11 scales x 22 rates, reduced to150 dimensions to preserve 98% of the variance).

For classification, the output of the reduced PCA was fed to a Support Vector Machine (SVM) with a Radial Basis Function (RBF). All of these steps are identical to Patil et al. (2012), to which the reader is referred to for further details, as the specifics of the classifier are not critical to illustrate the probing method. Briefly, a grid search on the RBF was performed to determine the best set of parameters and the classifier accuracy was tested with a 10-fold cross-validation. We obtained an average classification accuracy, i.e. whether the classifier is classifying the STM of a sound to the correct music or speech class, of 94% (SD = 6%) with the 10-fold cross-validation and 98% on the test set.

Figure 5 shows the discriminative feature maps for the speech versus music classification task. For each case, we used 691 probing samples and 30 bubbles with standard deviation of 10 Hz in the frequency dimension, 6 Hz in the rate dimension, and 3 cycles/octave in the scale dimension. As the task is a binary classification, the maps for speech and music are simply mirror images of each other. The discriminative regions of the auditory model STM representation appear to be mostly visible in the frequency dimension: speech can be best classified by looking at the input in a broad frequency range around 500 Hz, corresponding roughly to the position of the first formant in speech (Peterson & Barney, 1952). For the other dimensions, the classification depends on slow positive rates and low scales. In other words, the difference between speech and music was in the presence of slow modulations and broad spectral shapes for speech. Again, this matches prosodic and syllabic features of speech, together with the broad spectral shape of formants. By construction the two maps are complementary, but for music, a richness in spectrum, including high and low frequency regions, associated with fine spectral details (high scales) is characteristic of musical instruments (Elhilali, 2019), which have been designed to go beyond the physical constraints imposed by voice production.

In the case of the Method 1b), with the training set unavailable, a first probing was attempted with a uniform white noise but failed to provide classification decisions sampling the two categories: all noises were classified as music, a perhaps amusing finding which we will not develop here. The uniform white noise was thus replaced by a pseudo random noise generated in a PCA-reduced representation obtained with the testing set. A whitened PCA was first applied to the testing set to reduce it to 40 dimensions and a uniform gaussian white noise was generated on the 40 dimensions to generate samples in the reduced space. Each random reduced sample was then transformed into the original input space by applying the inverse PCA transformation. This procedure allowed to generate noisy samples with distribution relevant regarding the representative set of data relevant to the classification task. The information obtained in the two cases then strongly correlate (r = .84, df = 30974, $p < 10^{-5}$).

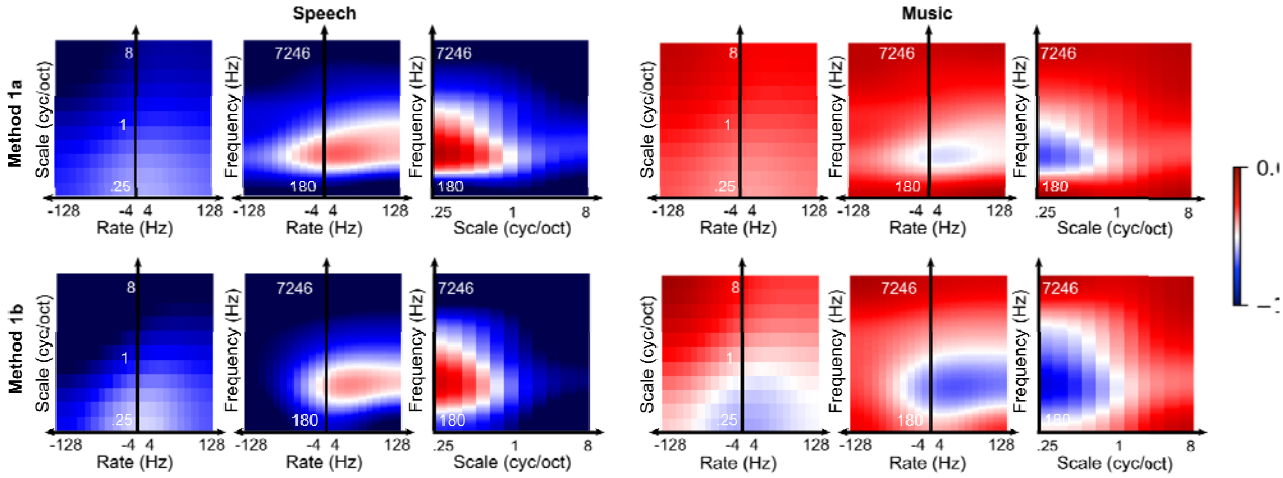
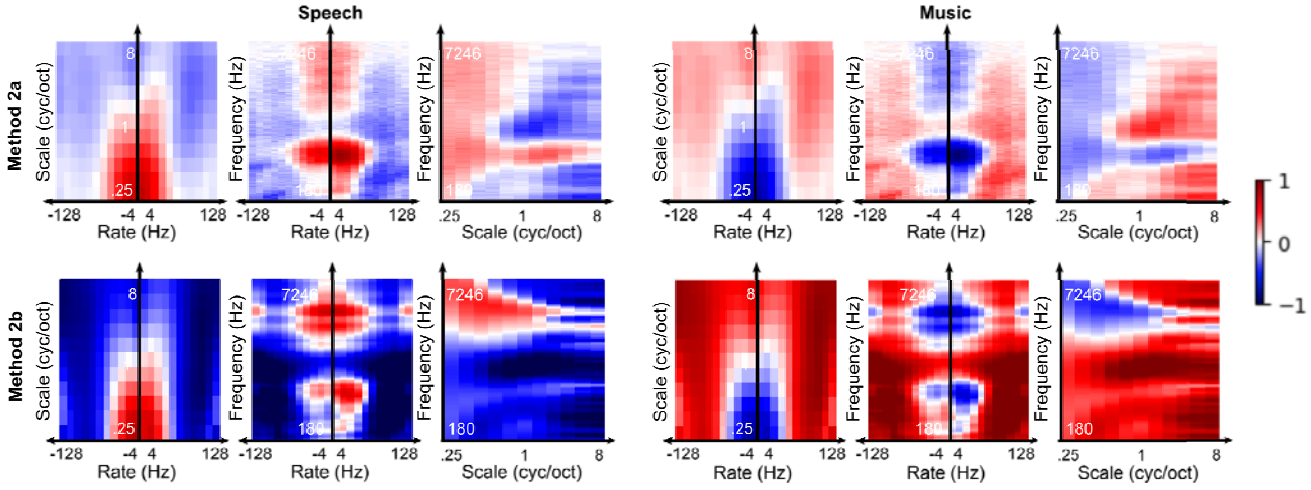


Fig. 5. Discriminative maps for speech and music in the STM representation. *The 4-D STM representations are projected in the three dimensions (frequency, scale, rates), and expressed in dB. Method 1a) (left) and method 1b) (right). The complete STM matrices are available in supplementary Figure S1. Method 1a) uses the training set while Method 1b) uses pseudo random noise. The red regions of each map correspond to the features necessary to categorize an audio sample in the given class. The blue regions correspond to less important features. As this is a binary classification task, the speech and music masks are simply opposite versions of each other.*

Figure 6 shows the canonical feature maps for the speech versus music classification task. Compared

389 the STM representation, or, similarly, they indicate the "average" speech and music sounds learnt by
390 the SVM. For speech, some formantic structure is visible on the frequency dimension, associated
391 with low rates typical of prosodic modulations (middle panels). These formantic regions extend to
392 higher scales (right panels), perhaps because formants are superimposed on a harmonic structure
393 during vowel sounds. Conversely, musical sounds more typically contain high modulation rates and
394 spectral scales. These observations are consistent with previous analyses of STM representations
395 (Elliott & Theunissen, 2009; Chi et al., 2005). These observations are consistent with previous
396 analyses of STM representations (Elliott & Theunissen, 2009; Chi et al., 2005). Again, the canonical
397 features observed for speech and music are complementary by construction with our method. It
398 should be noted that, as intuitively expected, canonical features depend on the acoustic characteristics
399 of speech and music, but they also depend on the task of the classifier. Probing a classifier trained to
400 discriminate speech from e.g. environmental sounds would likely provide different canonical features
401 for speech. This result may seem like a limitation of the method, but it also highlights the way an
402 automatic classifier performs a binary task. This may be an important difference to keep in mind
403 when comparing classifiers with human perception, which has to perform many concurrent tasks in
404 parallel. Yet, "opportunistic features" that depend both on sensory information and the task at hand
405 have been suggested for auditory timbre recognition, a task not unlike the one probed here (Agus et
406 al., 2019), a task not unlike the one probed here.
407



408

Fig. 6. Canonical STM representations for speech and music. *The d' sensitivity index is displayed for projections of the 4-D representation. The complete STM matrices are illustrated in supplementary Figure S2. The red regions of the maps indicate features most often encountered within each category, whereas the blue regions indicate features most often not encountered within each category.*

3. Discussion

4.1 Summary

The method presented in this paper used a reverse correlation framework to visualize the input features discovered by an automatic classifier to reach its decisions. When the classifier is successful, such features should provide insights about the structure of the input dataset. Over two examples using different kinds of classifiers (a CNN and an SVM with RBF) and using different kinds of input representations (2-D visual images and 1-D audio samples converted to a 4-D auditory model), we illustrated how the method could highlight relevant aspects of a classifier's operation. Moreover, by combining standard noise perturbation techniques with so-called bubbles (Gosselin & Shyns, 2002), we showed that the probing method can be focused either on discriminative features, related to the decision strategy of the classifier, or on canonical features, related to the output classes' main characteristics.

4.1 Benefits

In the context of neuroscience and experimental psychology, there are benefits in using a reverse correlation framework to interpret classifiers, as a way to complement other more specialized machine-learning interpretation techniques (Zhou et al., 2016; Ribeiro et al., 2016; Petsiuk et al., 2018; Borji & Lin, 2019; Xu et al., 2018).

First and foremost, reverse correlation is a familiar tool in the field of neuroscience and experimental psychology. It has proved useful to gain insights about stimulus features relevant to neural activity, at the single neuron (Eggermont et al., 1983; Neri & Levi, 2006) or network level (Arnal et al., 2015; Adolphs et al., 2005; Ringach & Shapley, 2010), and to understand human perceptual decisions (Gosselin & Shyns, 2001; Venezia et al., 2016). Applying it to interpret

classifiers amounts to translating and applying a familiar toolbox to another, conceptually similar problem of characterizing a black-box system.

Second, the method is by design fully agnostic by design. It operates on the input space of the classifier, whatever this space might be. It does not make assumptions on the classifier's architecture or inner operations. Focusing on the input space rather than the classifier's architecture is especially desirable in situations where the classifier is not the main interest of study, but rather, the structure of the input dataset is.

Third, it can be applied to classifiers that have not been designed by the user, as it does not even require the availability of the training dataset. Access to labeled input data is helpful in improving the efficiency of the method, for instance by allowing to shape the perturbation noise, but this is a mild constraint: there are no interesting situations we can think of for which both the classifier and the type of data to classify would be unknown.

Finally, the output of the method is a visualization (with statistical evaluation if required) in the input space. Such a representation should make intuitive sense to the user of the method, and the features discovered can be interpreted *a posteriori* in terms of attributes of the stimuli. If the representation does not make intuitive sense, then one possible benefit of the method is to help re-cast the input space into a more meaningful representation, as was done here in the audio example for which the waveform samples were pre-processed with an auditory model. This idea is further detailed in the "Perspectives" subsection.

4.2 Limitations

There are also limitations associated to the use of a reverse correlation approach to interpret automatic classifiers. Broadly speaking, these limitations follow those already described for reverse correlation in neuroscience.

First, reverse correlation is inspired from the analysis of linear systems, whereas machine-learning classifiers often rely on a cascade of non-linear operations to achieve computational power. The issue of non-linearity is well-described already in the reverse correlation literature, and its consequences have been clearly described (Theunissen et al., 2000). There are extensions to the reverse correlation technique to describe lower-order non-linear interactions in the input space (Neri

& Heeger, 2002). Such extensions could be applied to the interpretation of classifier's features. Interestingly, the reverse correlation approach bears some similarities with the "distillation" method from the machine learning literature (Hinton et al., 2015). Distillation consists in mimicking the behavior of a black-box classifier with an easily-interpretable classifier, such as a linear one (linear SVM, etc.). Both techniques can thus be viewed as attempting to find linear approximations of a classifier's operations, but their precise relationship remain to be investigated.

Second, the method has a number of parameters the number and size of bubbles, the space to generate the probing noise with reduction dimension methods such as PCA when the training set is available, which are not algorithmically constrained. In the examples above, the parameter space was explored heuristically. One suggested heuristic was to try and cover the output classes in a balanced manner with the probing set. However, even though statistical tests of the resulting features are available, we do not provide any fitness criterion, i.e. a way to quantify the efficiency of the method for a given set of parameters, for the features obtained with the method. Rather, we would argue that the iterative process for parameter tuning can be part of the interpretation process since finding the right probing structure provides some information on the structure of the dataset. Also, assessing whether the discovered features make intuitive sense relies mostly on the knowledge and goals of the user. Thus, it may not be easily formalized into a fitness criterion. If more formally defined methods are needed, either from the outset or after a first exploration of the classifier with reverse correlation, other classifier-specific tools exist (e.g., Zhou et al., 2016; Ribeiro et al., 2016; Petsiuk et al., 2018).

Third, the method implicitly assumes that there are no invariances by translation or otherwise in the classifier's algorithm. With reverse correlation, each point of the input space is treated independently of all others, so a feature discovered in one sub-part of the input space will not impact other, perhaps similar features in other sub-parts. This assumption is obviously falsified by CNN architectures, which are purposely designed to incorporate such invariances. In the CNN example illustrated here with digits recognition, this limitation was circumvented by the fact that all digits in the probing set were roughly spatially aligned. For the SVM on audio data, a time-averaging over the time dimension achieved a similar effect. Thus, a mitigation strategy is available: a rough alignment of the probing data (spatially or temporally) should be sufficient for the reverse correlation to

produce meaningful results. Another possible direction to address these invariance issues is to generate the probing noise in an appropriate space. Using a PCA partly achieves this. Finally, using another representation space with built-in invariances, e.g. by using wavelets transforms, can be considered.

4.3 Perspectives

The probing method is technically applicable to any classifier's architecture with any kind of input data. It is thus beyond the scope of this final section to list all possible use cases in the context of neuroscience. We will simply provide a few suggestions, to illustrate the kind of problems that could benefit from the probing method.

When studying perceptual decisions, one possible insight gained from interpreting a classifier is the exploration of the input representation fed to the classifier. The hypothesis is that, the more appropriate the representation, the more explainable the classifier should be. For instance, one could assume that the massively non-linear transformations of auditory and visual information that characterize perceptual systems serve to build a stimulus manifold within which perceptual boundaries are approximately linear (Georgopoulos et al., 1986; Jazayeri & Movshon, 2006; Kell et al., 2018). So, with the correct representation, a classifier modeling a perceptual decision process should be easily interpretable, or at least more easily interpretable than if the input representation was not reflecting perceptual processing. It is with this hypothesis in mind that the audio samples of the example illustrated above were first processed with an auditory model. Even though there are successful deep learning models operating on the raw audio waveforms (e.g. Wavenet, Oord et al., 2006), it is not expected that interpreting them in terms of waveform features will be meaningful. For instance, inaudible phase shifts between frequency components in the input would impact the waveform representation, but should not change the classifier's decision. An auditory model, in contrast, incorporates transforms inspired by the neurophysiology of the hearing system. If the features extracted resemble those available to a human observer, then they should be revealed when probing a classifier. In fact, the ease of interpreting a classifier feature could be a proxy to evaluate an input representation's adequation to a perceptual task.

Another possible application is when building "ideal observer" models (Geisler, 2004). The idea of an ideal observer model is to compute the best theoretical performance on a task, given a set of assumptions (classically, endowing the ideal observer with unbiased decision criteria, perfect and unlimited memory, and so on). This upper performance boundary is then compared to the observed performance with human participants or neural recordings. When considering classification or discrimination tasks, and when a formal model of the ideal observer is unavailable, it can be of interest to build pseudo-ideal observer models with machine learning classifiers. The advantage of our probing method is then that the classifier's strategy can be directly compared to a reverse correlation analysis of neural or psychophysical data, to ask whether the classifier and the experimental observer used the same decision features.

Finally, the general benefits of interpreting classifiers also apply to the field of neuroscience. In a broad sense, probing is intended to help an expert making sense of a classifier's strategy. If the features discovered through probing fit a theoretical model, this would reassure the expert that the performance relies on reasonable principles, which is especially important in clinical applications. In return, the expert's intuition may also help improve the classifier, for instance by simplifying its input representation through pre-processing, and so hopefully making it less brittle to irrelevant variations in input that may have been picked up by overfitting during training (Goodfellow et al., 2015). The discriminative features could be particularly useful to reduce the complexity of a classifier. Based on the discriminative features map, it may be possible to select a subset of important and intelligible features, which can then be used to build a more computationally efficient classifier, for very large dataset and/or for real-time processing.

4. Conclusions

We presented a novel method to interpret machine-learning classifiers, with the aim that the method should be agnostic and well-suited to applications in the neuroscience domain. Based on the reverse correlation framework, the method uses stochastic perturbation of inputs to observe the classifier's output. It then visualizes, in the input space, the discriminative and canonical features discovered by the classifier for each category. In theory the method can be applied to any kind of classifier, including deep neural networks, support vector machines, etc. It displays the same well-established

benefits and limitations as reverse correlation when applied to psychophysical or neural data. Our hope is that such a method can provide a simple and generic interface between neuroscientists and machine-learning tools.

5. References

Ahumada Jr, A., & Lovell, J. (1971). Stimulus features in signal detection. The Journal of the Acoustical Society of America, 49(6B), 1751-1756. https://doi.org/10.1121/1.1912577

Adolphs, R., Gosselin, F., Buchanan, T. W., Tranel, D., Schyns, P., & Damasio, A. R. (2005). A mechanism for impaired fear recognition after amygdala damage. Nature, 433(7021), 68-72. https://doi.org/10.1038/nature03086

Arnal, L. H., Flinker, A., Kleinschmidt, A., Giraud, A. L., & Poeppel, D. (2015). Human screams occupy a privileged niche in the communication soundscape. Current Biology, 25(15), 2051-2056.https://doi.org/10.1016/j.cub.2015.06.043

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal statistical society: series B (Methodological), 57(1), 289-300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Borji, A., & Lin, S. (2019). White Noise Analysis of Neural Networks. arXiv preprint arXiv:1912.12106.

Chi, T., Ru, P., & Shamma, S. A. (2005). Multiresolution spectrotemporal analysis of complex sounds. The Journal of the Acoustical Society of America, 118(2), 887-906. https://doi.org/10.1121/1.1945807

Elhilali, M. (2019). Modulation Representations for Speech and Music. In Timbre: Acoustics, Perception, and Cognition (pp. 335-359). Springer. https://doi.org/10.1007/978-3-030-14832-4_12

Elliott, T. M., & Theunissen, F. E. (2009). The modulation transfer function for speech intelligibility. PLoS computational biology, 5(3). https://doi.org/10.1371/journal.pcbi.1000302

Eggermont, J. J., Johannesma, P. I. M., & Aertsen, A. M. H. J. (1983). Reverse-correlation methods in auditory research. Quarterly reviews of biophysics, 16(3), 341-414. https://doi.org/10.1017/s0033583500005126

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6), 141-142.https://doi.org/10.1109/msp.2012.2211477

579 Geisler, W. S. (2004). “Ideal Observer analysis,” in Visual Neurosciences, eds L. Chalupa and J.
580 Werner (Boston, MA: MIT press), 825–837.
581 Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1986). Neuronal population coding of
582 movement direction. Science, 233(4771), 1416-1419. https://doi.org/10.1126/science.3749885
583 Gosselin, F., & Schyns, P. G. (2002). RAP: A new framework for visual categorization. Trends in
584 Cognitive Sciences, 6(2), 70-77. https://doi.org/10.1016/s1364-6613(00)01838-6
585 Gosselin, F., & Schyns, P. G. (2001). Bubbles: a technique to reveal the use of information in
586 recognition tasks. Vision research, 41(17), 2261-2271. https://doi.org/10.1016/s0042-
587 6989(01)00097-9
588 Gosselin, F., & Schyns, P. G. (2003). Superstitious perceptions reveal properties of internal
589 representations. Psychological science, 14(5), 505-509. https://doi.org/10.1111/1467-
590 9280.03452
591 Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples.
592 arXiv preprint arXiv:1412.6572. https://doi.org/10.5220/0006123702260234
593 Green, D. M., & Swets, J. A. (1966). Signal detection theory and psychophysics (Vol. 1). New York:
594 Wiley.
595 Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey
596 of methods for explaining black box models. ACM computing surveys (CSUR), 51(5), 1-42.
597 https://doi.org/10.1145/3236009
598 Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv
599 preprint arXiv:1503.02531.
600 Jazayeri, M., & Movshon, J. A. (2006). Optimal representation of sensory information by neural
601 populations. Nature neuroscience, 9(5), 690-696. https://doi.org/10.1038/nn1691
602 Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects.
603 Science, 349(6245), 255-260. https://doi.org/10.1126/science.aaa8415
604 Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., & McDermott, J. H. (2018). A
605 task-optimized neural network replicates human auditory behavior, predicts brain responses,

606 and reveals a cortical processing hierarchy. Neuron, 98(3), 630-644.
607 https://doi.org/10.1016/j.neuron.2018.03.044
608 King, J. R., & Dehaene, S. (2014). Characterizing the dynamics of mental representations: the
609 temporal generalization method. Trends in cognitive sciences, 18(4), 203-210.
610 https://doi.org/10.1016/j.tics.2014.01.002
611 Kriegeskorte, N., & Douglas, P. K. (2018). Cognitive computational neuroscience. Nature
612 neuroscience, 21(9), 1148-1160. https://doi.org/10.1038/s41593-018-0210-5
613 LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
614 https://doi.org/10.1038/nature14539
615 Lillicrap, T. P., & Kording, K. P. (2019). What does it mean to understand a neural network?. arXiv
616 preprint arXiv:1907.06374.
617 Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG-and MEG-data. Journal
618 of neuroscience methods, 164(1), 177-190. https://doi.org/10.1016/j.jneumeth.2007.03.024
619 Neri, P., Parker, A. J., & Blakemore, C. (1999). Probing the human stereoscopic system with reverse
620 correlation. Nature, 401(6754), 695-698. https://doi.org/10.1038/44409
621 Neri, P., & Heeger, D. J. (2002). Spatiotemporal mechanisms for detecting and identifying image
622 features in human vision. Nature neuroscience, 5(8), 812-816. https://doi.org/10.1038/nn886
623 Neri, P., & Levi, D. M. (2006). Receptive versus perceptive fields from the reverse-correlation
624 viewpoint. Vision research, 46(16), 2465-2474. https://doi.org/10.1016/j.visres.2006.02.002
625 Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K.
626 (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.
627 Paquette, S., Takerkart, S., Saget, S., Peretz, I., & Belin, P. (2018). Cross-classification of musical
628 and vocal emotions in the auditory cortex. Ann. NY Acad. Sci, 1423, 329-337.
629 https://doi.org/10.1111/nyas.13666
630 Patil, K., Pressnitzer, D., Shamma, S., & Elhilali, M. (2012). Music in our ears: the biological bases
631 of musical timbre perception. PLoS computational biology, 8(11).
632 https://doi.org/10.1371/journal.pcbi.1002759

633 Peterson, G. E., & Barney, H. L. (1952). Control methods used in a study of the vowels. The Journal
634 of the acoustical society of America, 24(2), 175-184. https://doi.org/10.1121/1.1906875
635 Petsiuk, V., Das, A., & Saenko, K. (2018). Rise: Randomized input sampling for explanation of
636 black-box models. arXiv preprint arXiv:1806.07421.
637 Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., ... & Gillon,
638 C. J. (2019). A deep learning framework for neuroscience. Nature neuroscience, 22(11), 1761-
639 1770. https://doi.org/10.1038/s41593-019-0520-2
640 Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" Explaining the
641 predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international
642 conference on knowledge discovery and data mining (pp. 1135-1144).
643 https://doi.org/10.1145/2939672.2939778
644 Ringach, D., and Shapley, R. (2004). Reverse correlation in neurophysiology. Cogn. Sci. 28, 147–
645 166. doi: 10.1207/s15516709cog2802_2
646 Rosch, E. (1983). Prototype classification and logical classification: The two systems. New trends in
647 conceptual representation: Challenges to Piaget's theory, 73-86.
648 Theunissen, F. E., Sen, K., & Doupe, A. J. (2000). Spectral-temporal receptive fields of nonlinear
649 auditory neurons obtained using natural sounds. Journal of Neuroscience, 20(6), 2315-2331.
650 https://doi.org/10.1523/jneurosci.20-06-02315.2000
651 Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. IEEE Transactions
652 on speech and audio processing, 10(5), 293-302. https://doi.org/10.1109/tsa.2002.800560
653 Venezia, J. H., Hickok, G., & Richards, V. M. (2016). Auditory "bubbles": Efficient classification of
654 the spectrotemporal modulations essential for speech intelligibility. The Journal of the
655 Acoustical Society of America, 140(2), 1072-1088. https://doi.org/10.1121/1.4960544
656 Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016). Deep learning for identifying
657 metastatic breast cancer. arXiv preprint arXiv:1606.05718.
658 Wiener, N. (1966). Nonlinear problems in random theory. Nonlinear Problems in Random Theory,
659 by Norbert Wiener, pp. 142. ISBN 0-262-73012-X. Cambridge, Massachusetts, USA: The MIT
660 Press, August 1966.(Paper), 142. https://doi.org/10.1063/1.3060939

Xu, T., Garrod, O., Scholte, S. H., Ince, R., & Schyns, P. G. (2018). Using psychophysical methods to understand mechanisms of face identification in a deep neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 1976-1984). https://doi.org/10.1109/cvprw.2018.00266

Zihni, E., Madai, V. I., Livne, M., Galinovic, I., Khalil, A. A., Fiebach, J. B., & Frey, D. (2020). Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome. Plos one, 15(4), e0231166.https://doi.org/10.1371/journal.pone.0231166

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2921-2929). https://doi.org/10.1109/cvpr.2016.319

Acknowledgements: *Research supported by grants ANR-16-CONV-0002 (ILCB), ANR-11-LABX-0036 (BLRI) and the Excellence Initiative of Aix-Marseille University (A*MIDEX) (ET), ANR-10-LABX-0087 IEC and ANR-10-IDEX-0001-02 PSL (DP). TA was supported by the Human Frontier Science Program (LT000362/2018-L).*

Supplementary Materials

Figure S1. Complete Spectro-Temporal Modulation representations discriminative maps, in dB, for speech vs. music classification task.

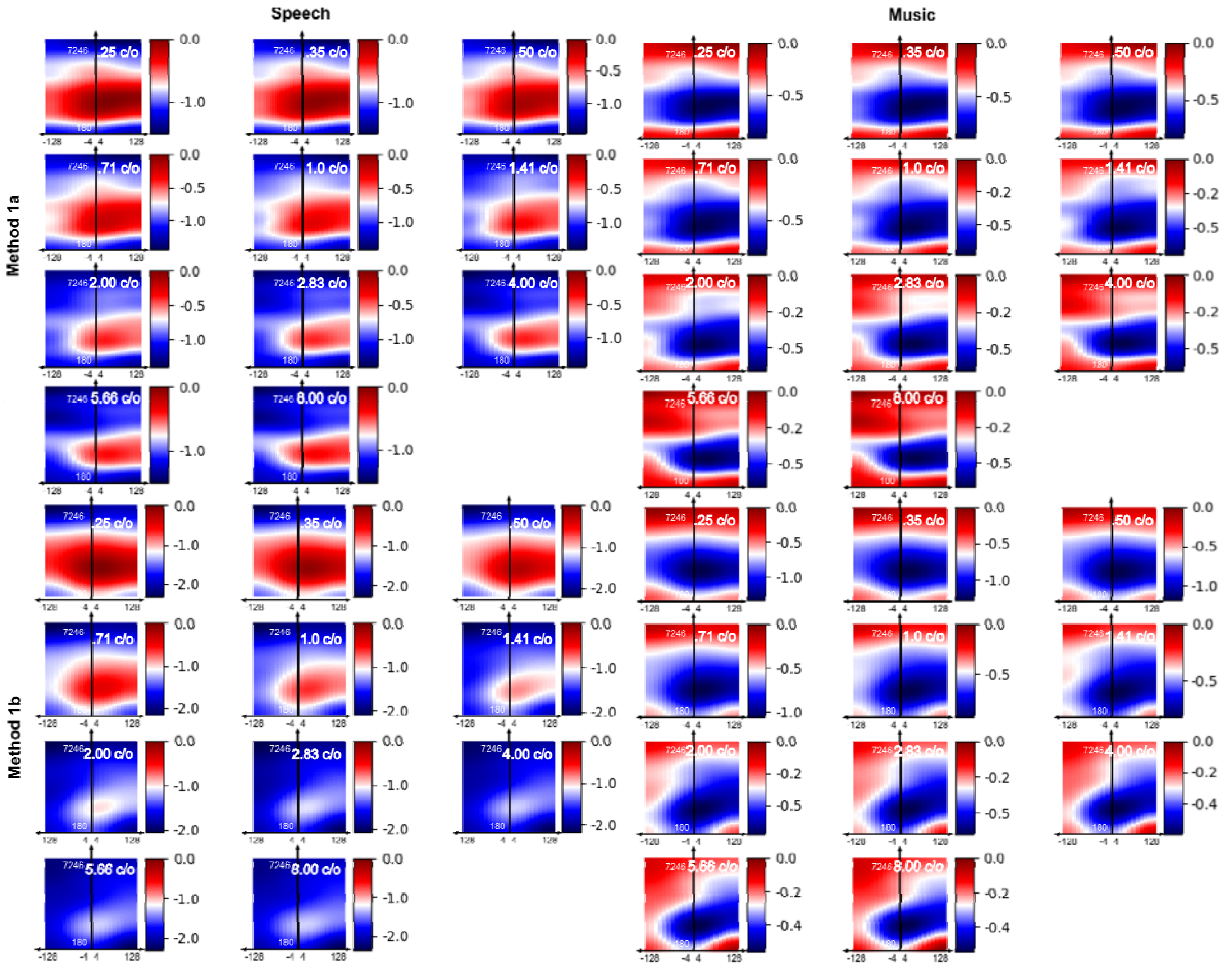


Figure S2. Complete Spectro-Temporal Modulation representations canonical maps (*d'*), for speech vs. music classification task.