university of groningen / faculty of science and engineering

Extracting Concepts From Neural Networks Using Conceptor-based Clustering

Bachelor's Project Thesis

Joris Peters, s4001109, j.peters.13@student.rug.nl,
Supervisors: G. Pourcel & Prof Dr H. Jaeger

Abstract: Conceptors are versatile neuro-symbolic formalizations of concepts as they arise in neural networks, with promising results on supervised tasks. However, the use of conceptors in unsupervised settings remains largely unexplored. Meanwhile, previous brain science and AI research used clustering to extract concepts from neural representations. This study combines conceptor-based representations with clustering methods for the unsupervised extraction of human-meaningful and coherent concepts from the activations of neural networks. Concretely, experiments are conducted on the responses of an Echo State Network (ESN), a type of recurrent neural network, to phoneme utterances from the TIMIT Acoustic-Phonetic Continuous Speech Corpus. In preparation, conceptor-based classification was demonstrated, and ESN hyperparameters were tuned. Then, two clustering methods, generalized centroid-based hard clustering and hierarchical agglomerative clustering, are adapted to work with conceptors and extract concepts from the ESN's responses. The resulting concepts and concept hierarchies were significantly human-meaningful, resembling established phonetic categories, and coherent. Conceptor-based clustering, although in its infancy, represents a promising approach to unsupervised concept extraction and forming conceptors without supervision. Applications in neuro-symbolic computational creativity, brain sciences, time-series clustering, and neural network explainability are suggested.

1 Introduction

1.1 Background

Neural networks (NNs), both biological and artificial, can be framed as representational systems; their neural activity serves as a *representation* of inputs* like sound or images (Kriegeskorte and Kievit, 2013). Moreover, NNs form *concepts* – representations of sets, or categories, of inputs (Rips, Smith, and Medin, 2012, as cited in Jaeger, 2014b). For example, a NN may use the phoneme /t/ as a concept to represent a broad set of related speech sounds, like the aspirated [t^h] (as in "top") and the unaspirated [t] (as in "stop"), which are functionally equivalent in English. *Symbols* like the transcription "/t/" will be used as names for concepts. Concepts enable various human cognitive functions like classification, creativity, and language. Similarly, the study of concepts is relevant to artificial intelligence (AI), facilitating the development of these functions in artificial neural networks (ANNs) and the integration of currently disjoint paradigms (Hofstadter and Mitchell, 1994).

Cognitive sciences have often considered neural activity and concepts separately at *data-dynamical* or *conceptual-symbolic* levels of description, respectively (Jaeger, 2014b). The data-dynamical level considers high-dimensional, sub-symbolic, and neural phenomena of dynamical systems and NNs. In contrast, the conceptual-symbolic level operates on the concepts or symbols (in the broad sense) of, e.g., languages, logic, and Good old-fashioned AI. This dichotomy still trenches through various fields that deal with conceptual-symbolic phenomena emerging from data-dynamical events. Marrying these levels can enhance our understanding of cognition and its emulation, particularly in neuro-symbolic AI (Sarker, Zhou, Eberhart, and Hitzler, 2021).

The neuro-symbolic *conceptors* (Jaeger, 2014b)

*The terms *content* and *referent* are also commonly used by cognitive sciences to refer to the representee, but since I intend to focus on artificial NNs, the notion of *input* is used.

formalize the notion of concepts as they are encoded in neural activity. A conceptor is a positive semi-definite matrix that captures the geometry of a cloud of neural activation states (see Section 1.6.1). When computed from a collection of neural states corresponding to a specific concept (e.g., the neural response to a recorded speech sound), the conceptor represents the neural pattern characteristic for that concept.

Conceptors are particularly relevant and practical means of capturing concepts. Various types of neural activity patterns have been previously associated with concepts (attractors (Jaeger, 1999), positions (Mao, Gan, Kohli, Tenenbaum, and Wu, 2019; Bricman, Jaeger, and van Rij-Tange, 2022), regions (Jaeger, 2014b), directions (Graziani, Nguyen, O'Mahony, Müller, and Andrearczyk, 2023; Kim, Wattenberg, Gilmer, Cai, Wexler, Viegas, et al., 2018), and clusters (Jaeger, 1999) in state space, or more generally, in representational space (Borghesani and Piazza, 2017; Balkenius and Gärdenfors, 2016)). Conceptors stand out with mathematical properties and useful operators that play on their neuro-symbolic nature. For example, the conceptor formalism defines operators like abstraction ordering or logical conjunction and negation, which effectively treat neural patterns as conceptual-symbolic units.

Moreover, conceptors have led to promising results in classification and recognition tasks by linking neural responses with concepts. Both classification and recognition aim at linking a given input with the correct concept (cf. Jaeger, 1999); that is, activate the concept that represents the class of the input. In particular, *time series classification* aims to estimate the class of pre-segmented signals (one label per segment), a segment referring to a continuous portion of a time series isolated for analysis like a phoneme (unit of speech) or word, whereas *time series recognition* seeks to estimate the class present within unsegmented signals (multiple labels per signal over time) (Lopes and Perdigao, 2011). Much of conceptor-based time series classification research, including Experiment 1, has been inspired by Jaeger (2014b), which demonstrated this method in speaker recognition on the Japanese Vowels dataset to about 99.9% accuracy exceeding most previous methods on the task. In addition to its competitive performance, this method offers extensibility to new classes without retraining (incremental learning) and applicability across various ANN architectures. Moreover, it illustrates using conceptors to represent and relate class concepts. Later, the method was successfully adapted to classify brain data (Bartlett, Garcia, Thill, and Belpaeme, 2019) and non-stationary time series (Vlegels, 2022). Moreover, Chatterji (2022) adapted the method to time series recognition on the phonemes of the TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT) (Garofolo, Lamel, Fisher, Fiscus, Pallett, and Dahlgren, 1993), the dataset also used in the current study. Finally, most conceptor-based work, including the current one, has been demonstrated on Echo State Networks (ESNs), a type of recurrent neural network that, for properties, I elaborate on below, combine particularly well with conceptors. Thus, conceptors could successfully represent class concepts in supervised time series classification and recognition.

Previous research has also explored the unsupervised extraction of concepts from neural activity using *clustering*, an unsupervised method for identifying groups (clusters) among data points. On the biological side, studies in psychology and neuroscience have applied clustering to identify and relate the concepts represented in the brain (representational geometry analysis (Kriegeskorte, Mur, and Bandettini, 2008; Kriegeskorte and Kievit, 2013; Tucciarelli, Wurm, Baccolo, and Lingnau, 2019), categorical representation Chang, Rieger, Johnson, Berger, Barbaro, and Knight (2010); Beach, Ozernov-Palchik, May, Centanni, Gabrieli, and Pantazis (2021); Brouwer and Heeger (2013), clustering of existing categories (Mesgarani, Cheung, Johnson, and Chang, 2014; Huth, Nishimoto, Vu, and Gallant, 2012; Shepard, 1980), partitioning-of-activation-space theory (Laakso and Cottrell, 2000)). Typically, the procedure first defines some input (stimulus) space and a representational (neural response) space with functional correspondence. For example, the input space may contain a set of speech sounds presented to participants, while the representational space contains the neural responses to those sounds recorded from participants' superior temporal gyri (Mesgarani et al., 2014). Second, similarities of represented inputs are measured, in this case, using brain imaging, but behavioral data or cognitive models are also common (Kriegeskorte et al., 2008). Third, clustering algorithms like K-means (Chang et al., 2010; Brouwer and Heeger, 2013; Huth, De Heer, Griffiths, Theunissen, and

Gallant, 2016; Beach et al., 2021) and *hierarchical agglomerative clustering* (HAC) (Shepard, 1980; Tucciarelli et al., 2019) identify clusters among the represented inputs in representational space. Crucially, clusters in representational space are interpreted as concepts (Mok and Love, 2019) [†]. With this approach, Beach et al. (2021) found clusters of represented sub-phonemic speech corresponding to phonemes, and Mesgarani et al. (2014) found clusters among represented phonemes corresponding to super-phonemic categories like fricatives. In conclusion, this clustering method was used to extract concepts encoded by the brain.

Similarly, AI has applied clustering on the activations of ANNs to extract concepts, primarily to enhance explainability (Fel, Boutin, Béthune, Cadène, Moayeri, Andéol, Chalvidal, and Serre, 2024). ANN models can be difficult to explain on a data-dynamic level. By extracting concepts, studies have aimed to provide a condensed, more accessible representation of high-dimensional ANN activations. For example, like in some methods from brain sciences, Ghorbani, Wexler, Zou, and Kim (2019) used K-means on the Euclidean distances in state space to extract concepts, representing meaningful groups of pixels on input images. Liu and Arik (2020) used deep embedded clustering (DEC) on the activations at different layers of an ANN to extract concepts of varying abstractions. DEC maps layer activations to lower-dimensional latent spaces, then iteratively refines latent representations and cluster assignments until coherent clusters corresponding to visual concepts are reached. Similarly, Song, Liu, Huang, Wang, and Tan (2013) used used K-mean to identify concepts in the latent activations of an autoencoder. Alternative approaches guide the extracted concepts to align with human concepts for enhanced intelligibility (El Shawi, 2024), which, however, qualifies as supervised and may fail to reflect the model's representation (Kim and Chae, 2024).

While clustering of ANN activations has shown promise for unsupervised concept extraction, further advancements could be achieved by integration with conceptors. First, previous work took a passive, observational stance, extracting concepts for explainability without discussing their functional potential. Second, conceptors were not employed despite their aptitude in representing concepts, various practical operators, and theoretical relevance to neuro-symbolic integration. Thus, this exploratory study seeks to extract concepts by clustering with conceptor-based representations. Specifically, it investigates *whether conceptor-based clustering can extract human-meaningful and coherent concepts from ANN activations without supervision.* Ghorbani et al. (2019) inspired the desiderata of human-meaningfulness and coherency. An extracted concept is human-meaningful if its semantics (the set of represented inputs) align with that of a human concept, like a theoretically established category. An extracted concept is coherent if its instances are similar to each other.

1.2 Motivation

I will now elaborate on two motivations for exploring unsupervised concept extraction using conceptor-based clustering: enhancing computational creativity methods and extending the conceptor formalism. First, regarding creativity, many models use conceptual-symbolic mechanisms to operate on data-dynamical domains. For example, the movements of stick figures (represented as data-dynamical signals) can be morphed and newly recombined, provided they are discretized into classes like Walking and Jumping (Jaeger, 2014a). Similarly, the Omniglot challenge involves the generation of new handwritten characters, which is possible only through the currently manual identification of conceptual primitives among data-dynamical signals (Lake, Salakhutdinov, and Tenenbaum, 2019; Fabi, Otte, and Butz, 2021). The same reliance on pre-established concepts constrains traditional approaches like Hofstadter's models of analogical reasoning (Hofstadter, Mitchell, and French, 1987; Hofstadter and Mitchell, 1994). In these cases, unsupervised concept extraction methods may identify actionable concepts within the data-dynamical representations of ANNs. This addition could advance neuro-symbolic approaches to computational creativity, with possible implications for other cognitive functions (c.f., Sheth, Roy, and Gaur, 2023).

Second, I seek to extend the conceptor formalism to unsupervised settings. Conceptors were successfully applied to supervised designs where the concepts of interest had been given. For example, in

[†]A parallel between clusters and concepts lies in the similar desiderata of cohesion and coherency – members being close to each other – also reflected in the loss functions of many clustering algorithms and conceptors, respectively.

speech classification and recognition, concepts were provided as labels through the training data. However, the application of supervised conceptor-based methods falls short when labeled data is unavailable. Moreover, even when available, pre-established concepts may be suboptimal. For instance, the discretization of English speech into the phonemic classes provided by the transcriptions of TIMIT may be a suboptimal representation for speech recognition systems. In these situations, an unsupervised method to form conceptors corresponding to meaningful and coherent neural patterns may make conceptor-based methods more performant and broadly applicable. This is similar to children learning to distinguish phonemes without explicit instructions (Maye and Gerken, 2000).

However, the unsupervised formation of conceptors remains largely unexplored, with only one work combining conceptors and clustering; Mossakowski, Diaconescu, and Glauer (2019) fed speech recordings to an ANN and captured its responses using conceptors, one per speaker. Using HAC with a fuzzy generalization of the Löwner ordering as a (dis)similarity function, they organized the conceptors in an abstraction hierarchy. The authors discuss the potential of their method at enhancing conceptor-based classification, but a lack of testing left unclear whether the nodes of the resulting hierarchy coincided with any meaningful or coherent concepts.

1.3 Approach

1.3.1 ESNs

The present work focuses on ESNs for several reasons. These differ from other recurrent neural networks in their large size (number of internal neurons), high sparsity (the ratio of neurons to connections), and randomly initialized, untrained internal and input weights (Yildiz, Jaeger, and Kiebel, 2012) (see Section 1.5 for a formalization of ESNs). First, training is only required for the output weights, simplifying and isolating the developed methods. Second, due to their lack of training, ESNs rely on conceptors more than other types of ANNs as an external control mechanism. Third, ESNs facilitate the evaluation of the developed methods. They perform high-dimensional non-linear expansions, echos, of their inputs (cf. Lukoševičius, 2012). This expansion often retains much information, such that the known clusters or labels of the inputs can serve as ground truth to the extrinsic validation of clustering or classification of the responses. Thus, concepts are extracted from ESN states.

1.3.2 Phonemic Speech

Moreover, the methods are demonstrated on phonemic speech, a domain rich in conceptual-symbolic structure. Phonemes are the smallest, distinguishable units of speech of a given language that can change the meaning of a word. Sub-phonemically, each phoneme represents a set of related sound variations called phones, its allophones. For example, the phoneme /t/ of the English-proficient mind represents the phones [t^h], [t], and [ɾ][‡]. Superphonemically, various phonemic organizations have been proposed by phonetics – the branch of linguistics concerned with speech production and perception. For example, Figure 1.1 depicts a taxonomy of the phonemes present in the TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT)[§]. TIMIT provides a rich set of speech recordings with phonemic transcriptions and is the most common dataset in phoneme recognition and classification (Lopes and Perdigao, 2011). The depicted organization by the manner of phoneme production is standard across the classical (Rabiner, 1978) and more recent (Oh, Park, Kim, and Jang, 2021) literature and was provided in TIMIT's documentation.

These conceptual-symbolic structures make phonemic speech an ideal domain for demonstrating concept extraction by serving as ground truths, against which the extracted concepts can be evaluated (Shepard, 1980; Oh et al., 2021; Chang et al., 2010). ESNs have also shown an excellent capacity for processing speech (e.g., Jaeger, 2014b), partly due to their cyclic connections, which introduce memory into the system. Therefore, phonemic speech from TIMIT is used to demonstrate the pro-

[‡]Phonemes are broadly transcribed as indicated by slanted brackets. Phones are narrowly transcribed as indicated by square brackets. Phonetic transcriptions will be written using the International Phonetic Alphabet (International Phonetic Association, 1999).

[§]The TIMIT labels shown in Figure 1.1 and listed in Table A.1 follow the IPA. The translation keys to the Greek alphabet and the categories of the taxonomy come in the documentation file phoncode.doc of the corpus, available through the LDC website.

posed ESN-based methods.

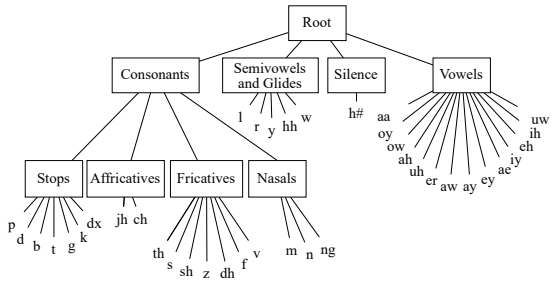

Figure 1.1: A taxonomy of the phonemes in TIMIT, created from the categories provided by the dataset's documentation, adding Root and Consonants nodes following Rabiner (1978) and Oh et al. (2021), and folding TIMIT's labels into phonemes according to Lee and Hon (1989) by the mapping shown in Table A.1.

.

1.3.3 Approach

Figure 1.2 illustrates the current approach to concept extraction. Two clustering algorithms are applied to group ESN responses to pre-processed phonemic speech into concepts. In Experiment 2, a *generalized centroid-based hard clustering* algorithm (GCHC, cf. Sarmiento, Fondón, Durán-Díaz, and Cruces, 2019) is adapted to extract a set of non-overlapping concepts from ESN responses to phonemic speech. In Experiment 3, HAC (Everitt, Landau, Leese, and Stahl, 2011) is adapted to extract a hierarchy of concepts from ESN responses to pre-grouped phonemic speech. Found concepts are then evaluated to determine whether they are coherent and correspond to human-meaningful phonetic categories.

In Experiment 1, phoneme classification was performed primarily to tune the ESN's hyperparameters in preparation for the following concept extraction experiments. Moreover, Experiment 1 has the positive side-effect of demonstrating the application of conceptors to the phoneme classification on TIMIT, building on the method of Jaeger (2014b). The currently highest accuracy of 78.4% for phoneme classification on TIMIT's test set was achieved using fixed-sized kernel logistic regression (Karsmakers, Pelckmans, Suykens, and hamme, 2007, as cited in Lopes and Perdigao, 2011).

1.4 Contributions and Outline

This thesis makes the following contributions:

1. It demonstrates conceptor-based phoneme classification on TIMIT.

2. It proposes an unsupervised method for extracting human-meaningful and coherent concepts from ANN activations, as demonstrated on ESN responses to phonemic speech.

3. It adapts two classical clustering algorithms to conceptors, providing an unsupervised method for forming relevant conceptors.

4. It suggests potential applications in neuro-symbolic computational creativity, brain sciences, time-series clustering, and ANN explainability to be explored in future research.

The remainder of the thesis is organized as follows. The next sections provide formal definitions of ESNs and conceptors, inspired by the detailed report Jaeger (2014b). The Methods and Results section covers data pre-processing, ESN response collection, and the conceptor-based classification and concept extraction by adaptations of GCHC and HAC. The results are given directly after the corresponding methods. In the Discussion, the methods, results, and their implications for, i.e., computational creativity and the conceptor formalism are discussed. The appendices contain additional information, experiments, proofs, and an index of mathematical notations.

1.5 Echo State Networks (ESNs)

Let us formalize an example ESN. Figure 1.3 depicts a typical ESN as it is driven by an input sequence u, possibly a speech recording, and elicits a high-dimensional response x, the sequence of internal states of the ESN's reservoir. An output layer is often added to ESNs, e.g., for time series classification, regression, or generation, but omitted here since the presented classification and concept extraction methods will only rely on the internal states of the network.

Let N be the number of internal neurons. N will typically be large relative to the dimensionality d of the input. The input weight matrix $W^{in} \in \mathbb{R}^{N \times d}$, the bias vector $b \in \mathbb{R}^N$, and the internal weight

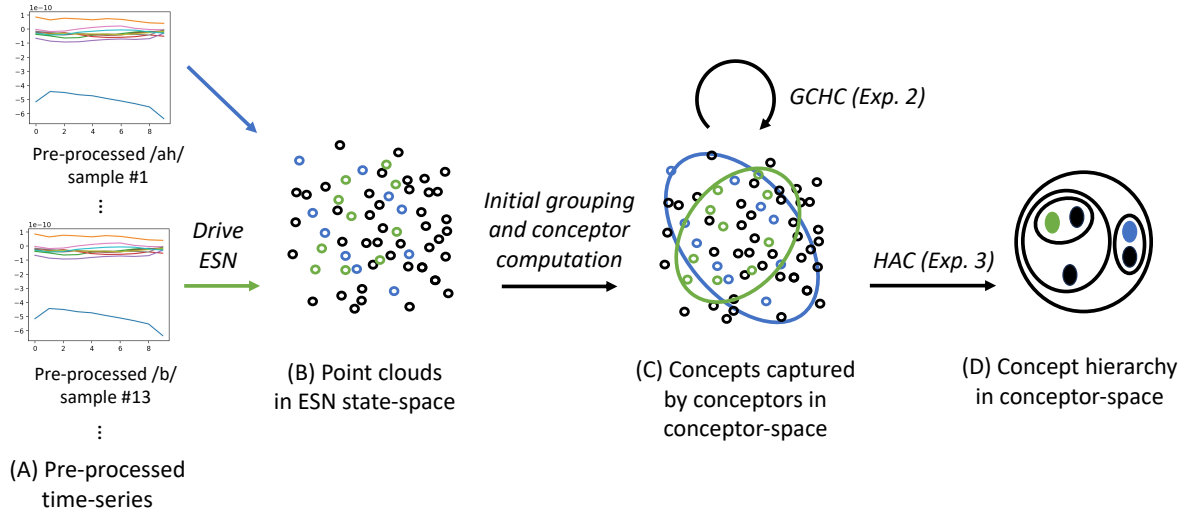

Figure 1.2: We are given time series, the pre-processed phoneme recordings from TIMIT (A). These time series drive an ESN, producing ESN responses, i.e., point clouds in state space (B). These responses are initially grouped into concepts, each captured by one conceptor (C). In Experiment 2, these concepts are refined using GCHC. In Experiment 3, these concepts are related in a hierarchy using HAC to extract higher-level concepts (E).

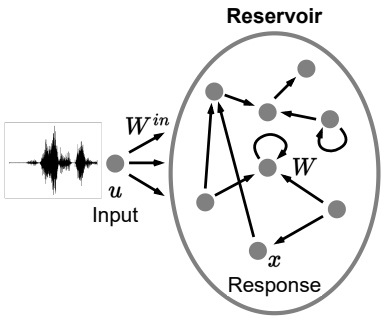

Figure 1.3: An ESN driven with an input sequence. Dots represent neurons. Arrows represent synaptic connections.

matrix $W \in \mathbb{R}^{N \times N}$ are randomly initialized, the latter of which will typically contain many zeros to implement the sparsity of the reservoir. When driven by a discrete input sequence u of length L, the ESN elicits a response, the internal state sequence x of dimensionality N and of the same length as the input. The ESN's update equation is that of classical discrete-time recurrent neural networks:

$$x(n+1) = \tanh(Wx(n) + W^{in}u(n+1) + b), \quad (1.1)$$

where $x(n)$ and $u(n)$ are the internal reservoir state and input column vectors at time step n and *tanh* is the hyperbolic tangent.

1.6 Conceptors

1.6.1 Definition and Intuition

Conceptors are a means to capture and manipulate the internal representations of ESN, among other features. Given a sequence of states $x = (x(1), ..., x(L))$, which may have arisen from running the above ESN, the conceptor matrix C computed

from x minimizes the following loss function $\mathcal{L}$:

$$\begin{aligned}\mathcal{L}(C) &= \sum_{n=1}^{L} ||x(n) - Cx(n)||^2/L + \alpha^{-2}||C||^2 \\ C &= \arg\min_C \mathcal{L}(C),\end{aligned} \tag{1.2}$$

where $\alpha \geq 0$ is the conceptor's aperture (further explained below). The conceptor C that minimizes $\mathcal{L}(C)$ may be analytically computed via the following procedure:

1. Concatenate the states in x column-wise in an $N \times L$ collection matrix $X = [x(1)|...|x(L)]$.
2. Compute the correlation matrix $R = XX'/N$.
3. Obtain the conceptor $C(R, \alpha) = R(R + \alpha^{-2}I)^{-1}$.

Intuitively, a conceptor can be considered the "'fingerprint' of the activity of [a] network" over a period of time (Jaeger, 2014b); this identifying capacity of conceptors is reflected in their loss function. Minimizing $\mathcal{L}$, the term $||x(n) - Cx(n)||^2$ nudges C toward realizing an identity mapping for the subspace populated by states x, while the regularization term $\alpha^{-2}||C||^2$ draws the conceptor toward the zero matrix, preventing it from becoming the identity matrix. This tension causes the conceptor to retain information along the axes that account for more of the variance of x, where the minimization of $||x(n) - Cx(n)||^2$ is most beneficial while filtering out information along the less relevant axes. The aperture parameter α controls the permissiveness of this filtering by regularizing the conceptor. In the *geometric interpretation*, reservoir states form a point cloud in state space $(-1, 1)^N$, whose shape the conceptor approximates as a hyperellipsoid; the ellipsoid's axes, corresponding to the singular-value-scaled singular vectors of the conceptor, align with the principal components of the point cloud.

1.6.2 Similarity Function

The similarity between two conceptors C_a and C_b shall be defined as:

$$\text{Sim}(C_a, C_b) = \frac{|(S_a)^{1/2}(U_a)'(U_b)(S_b)^{1/2}|^2}{|\text{diag}(S_a)||\text{diag}(S_b)|}, \tag{1.3}$$

where $U_a S_a (U_a)'$ is the SVD of C_a and $U_b S_b (U_b)'$ is the SVD of C_b. It is a function of the squared cosine similarity of the conceptors that measures the angular alignment between all pairings of singular vectors of the two conceptors weighted by the corresponding singular values.

1.6.3 Aperture

The aperture of a conceptor can be set during its computation, or when given a pre-computed conceptor C, its aperture can still be adapted by any factor of $\gamma > 0$ using the aperture-adaptation function φ that returns the aperture-adapted conceptor C_{new}:

$$C_{new} = \varphi(C, \gamma) = C(C + \gamma^{-2}(I - C))^{-1} \tag{1.4}$$

1.6.4 Logical Operators

Several logical operators have been meaningfully defined on conceptors. Given two conceptors C and B, we have the following definitions and semantics:

1. **Negation ($\neg$)**

   $$\neg C := I - C \tag{1.5}$$

   It returns a conceptor that describes the linear subspace complementary to that of C.

2. **Conjunction ($\wedge$)**

   $$C \wedge B := (C^{-1} + B^{-1})^{-1} \tag{1.6}$$

   $C \wedge B$ returns a conceptor that describes the intersection of the linear subspaces of C and B. This method relies on the inversion of B and C and thus fails when B or C contain singular values of 0. Such singular values may occur due to rounding or negating conceptors with unit singular values. Because these cases could not be prevented, a more robust definition of conjunction was used (Appendix A.1).

3. **Disjunction ($\vee$)**

   $$C \vee B := \neg(\neg C \wedge \neg B), \tag{1.7}$$

   by De Morgan's law. It returns a conceptor that describes the union of the linear subspaces of C and B.

2 Methods and Results

The Python code is available on `https://github.com/jorisptrs/Unsupervised-Conceptors`.

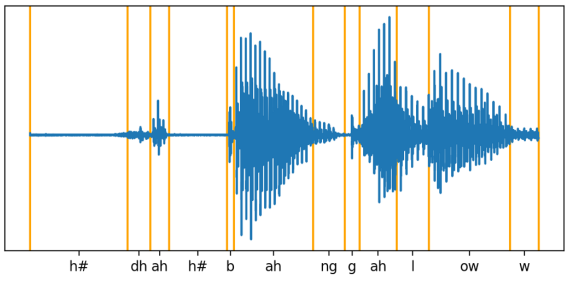

Figure 2.1: Example of the twelve first segments of one of TIMIT's sentence utterances.

2.1 Dataset

TIMIT was chosen as the data source for it features diverse and phonetically annotated speech signals. It comprises 6300 sentence utterances. Each of the 630 US-based native-English speakers (30% female and from eight dialect regions) read ten sentences: five phonetically-compact, three phonetically-diverse, and two dialect sentences. Each utterance comes with a phonetic transcription that indicates which of 64 phones is uttered at any time. Moreover, the corpus is pre-split into a training (73% of the utterances) and test set, used as such in Experiment 1. Experiments 2 and 3 relied solely on the training set.

2.2 Pre-processing

For all experiments, the following pre-processing steps were performed. The utterances were segmented according to the phonetic transcriptions into $n = 241225$ segments ($n_{\text{TIMIT}-\text{train}} = 177080$ and $n_{\text{TIMIT}-\text{test}} = 64145$), each a vocalization of one phone (Figure 2.1). The first $d = 13$ Mel Frequency Cepstral Coefficients (MFCCs) were extracted from each segment, consistent with previous literature (Bromberg, Qian, Hou, Li, Ma, Matthews, Moreno-Daniel, Morris, Siniscalchi, Tsao, et al., 2007). This representation ought to isolate the information most relevant to speech analysis. To compute the MFCCs, the Librosa Python library (McFee, Raffel, Liang, Ellis, Mcvicar, Battenberg, and Nieto, 2015) was used with one MFCC vector computed every 1 ms from a 2.5 ms long sliding window.

The resulting time series were normalized in amplitude and time. First, the amplitudes varied strongly across the channels (the lowest mean amplitude of an MFCC channel ($\mu_1 = -593.2$) is 24.2 times lower than that of the second lowest channel ($\mu_4 = -22.4$)). To provide each MFCC a similarly strong effect on the ESN dynamics under the identically distributed input weights, they were normalized to a range of $[-0.5, 0.5]$ across all samples. Second, the series were normalized in time by fitting each channel with a cubic spline and sampling it at $L = 10$ temporally equidistant points to account for differences in utterance speeds.

Lastly, the phonetic labels $p_{i(i=1,\ldots,c)}$ were mapped from the original set of 61 phones to a subset of 39 phonemes P. Initially proposed by Lee and Hon (1989), this mapping (Table A.1) amounts to folding stress-related variations and allophonic variations of phonemes (e.g., /em/ and /m/) into the same classes. The mapping helped achieve reasonable classification and clustering performances, feasible computations, and consistency with the previous literature (Karsmakers et al., 2007; Chatterji, 2022; Oh et al., 2021; Lopes and Perdigao, 2011).

Thus, the resulting data consists of tuples $D = \{(s_i, p_i)|i = 1, ..., c\}$ with MFCC time series s_i, phone labels $p_i \in P$, and the set of phones P after folding ($|P| = 39$). The ready-made train-test split from TIMIT was used, resulting in $D_{\text{TIMIT}-\text{train}}$ and $D_{\text{TIMIT}-\text{test}}$ of respective lengths $n_{\text{TIMIT}-\text{train}}$ and $n_{\text{TIMIT}-\text{test}}$.

2.3 Model

The following ESN setup was used in all three experiments. Its hyperparameters are summarized in Table 2.1. The ESN consisted of $N = 100$ neurons with a connection density of $r = 10\%$. The entries of W^{in} and b were randomly sampled from a standard normal distribution and rescaled by factors of $k_{W^{in}} = 1.0$ and $k_b = 0.6$, respectively. W was obtained by random sampling from a standard normal distribution and rescaling the result to a spectral radius of $\rho = 2.3$. The spectral radius of an internal weight matrix is its largest absolute eigenvalue. The larger ρ, the farther W scales the internal state during the state update along its first eigenvector, leading to more chaotic behavior. ρ was adapted by rescaling the initial internal weight matrix W_{old} to $W_{new} = \frac{\rho_{new}}{\rho(W_{old})} W_{old}$ where W_{new} has the desired spectral radius ρ_{new} instead of the previous ρ_{old}.

The above hyperparameters were picked by hand based on their effects on the accuracy in Experiment

Hyperparameter	Value
Number of neurons (N)	100
Connection density (r)	10%
Scaling factor for W^{in} ($k_{W^{in}}$)	1.0
Scaling factor for b (k_b)	0.6
Spectral radius (ρ)	2.3

Table 2.1: Final ESN hyperparameters.

1, previous research, and resource constraints. All parameters were initially set to the values used in the demonstration experiments of Jaeger (2014b) (Section 4.1, p. 161). Deviating, N was kept as large as possible under the available computational resources. Larger sizes would have likely improved performance after adapting the other hyperparameters but may have increased the risk for overfitting (Lukoševičius, 2012). The remaining hyperparameters, r, $k_{W^{in}}$, k_b, and ρ, were adjusted to maximize the validation accuracy of phoneme classification in Experiment 1. Moreover, automated hyperparameter optimization was attempted (see Appendix A.4) but not used due to its high computational cost and intractably slow convergence.

The resulting ESN was driven on each time series s_i ($i = 1, ..., c$), producing the reservoir state sequence x_i. Concretely, each run started from the same state $x(0)$ sampled once from a standard normal distribution so as not to introduce meaningless between-sample differences while providing the network with an initial excitation. The following states $x_i(t)$ ($t = 1, ..., L$) were computed via update Equation 1.1 and collected column-wise in the $N \times L$ matrix $X_i = [x_i(1)|...|x_i(L)]$ (this excludes the starting state). Concluding, an ESN response collection matrix X_i was computed for each s_i.

2.4 Experiment 1: Phoneme Classification

2.4.1 Objective

Experiment 1 performed conceptor-based phoneme classification on TIMIT to (a) optimize the ESN hyperparameters on the data for the subsequent unsupervised Experiments 2 and 3 and (b) demonstrate the method by Jaeger (2014b) on this dataset. For objective (a), I assume a positive relationship between the accuracy of a conceptor-based classifier at distinguishing concepts (i.e., classes) and the capacity of a conceptor-based clustering algorithm (used in Experiments 2 and 3) to extract these and other relevant concepts given the same data and using the same ESN. Both classification and clustering require the ESN to be sensitive to relevant, class-discriminating input differences. For Example, hyperparameters like the aperture (Jaeger, 2014b) and spectral radius (Yildiz et al., 2012; Chatterji, 2022) are commonly tuned based on dynamical features of the ESN response, like its energy or the Echo State Property, rather than its performance on later use-cases. Thus, tailoring the ESN to data via classification may also improve concept extraction on that data.

2.4.2 Conceptor-based Classification

The classifier takes an ESN response and outputs the assigned class. During training, the responses of each class are captured with two conceptors. When classifying an unseen input, its response is compared to each class's conceptors and assigned the class of highest *Evidence*, a measure of proximity. Thus, this method classifies time series based on ESN responses as captured using conceptors; the assigned label is equally applied to the response to the time series and the time series.

2.4.3 Dataset

The pre-processed original dataset $D_{\text{TIMIT}-\text{train}}$ was divided into a preliminary training set, $D_{\text{pre}-\text{train}}$, and a validation set, D_{val}, to validate hyperparameter and design choices. The split was 80/20, respectively, stratifying over phonemic classes. Once hyperparameters and methods were set, the classifier was retrained on the whole of $D_{\text{TIMIT}-\text{train}}$ and evaluated on $D_{\text{TIMIT}-\text{test}}$.

2.4.4 Training

Training amounted to computing a *positive conceptor* C_p^+ and *negative conceptor* C_p^- per class $p \in P$. Each class's positive conceptor captures the linear subspace that ESN states of that class tend to occupy and was computed as follows. Let η_p be the number of training instances of p. The state collection matrices corresponding to instances of p were concatenated column-wise into a class-level collection matrix $X_p = [X_1|X_2|...|X_{\eta_p}]$ from which C_p^+

was then computed with an initial aperture of $\alpha = 1$ by steps 2 and 3 of the procedure for conceptor computation (Section 1.6.1). These steps were repeated for each class, obtaining the set of preliminary positive conceptors $C^+_{pre} = \{C^+_p | p \in P\}$.

2.4.5 Aperture Optimization

After computing the conceptors in C^+_{pre} with an initial aperture of $\alpha = 1$, their apertures were adapted. First, as specified in Appendix A.3.1, the ∇-criterion was used to estimate the aperture at which the positive conceptors would be maximally sensitive to scalings of the underlying states, a proxy for discriminatory ability. Each positive conceptor was aperture-adapted to this aperture, resulting in aperture-optimized positive conceptors C^+.

2.4.6 Trace Normalization

However, classification using C^+ seemed biased. The trace of a conceptor, the sum of its eigenvalues, reflects the total variance or volume of the subspace it captures. Figure 2.2 shows at $x = 1$ that the conceptors in C^+ varied in their traces. Higher traces could arise from high-energy ESN states, such as those associated with particularly loud utterances. To illustrate the bias, let x be an arbitrary ESN state of unknown distribution (class). During classification, i.a., the Positive Evidence $E^+(x, p) = x' C^+_p x = x' USU' x$ is compared across classes $p \in P$, where USU' is the SVD of C^+_p. An increase in the singular values, reflected in $tr(C^+_p) = \sum_i S[i, i]$, would cause a higher expected Positive Evidence for p. Thus, the Positive Evidence component may cause classification to favor larger conceptors.

To mitigate this potential bias, I normalized the traces of the positive conceptors to a common target value tr_{target}, the mean trace among the conceptors:

$$tr_{target} = \frac{1}{|C^+|} \sum_{C \in C^+} \mathrm{tr}(C) \approx 57.23 \qquad (2.1)$$

Algorithm A.1 was developed to adapt each conceptor's trace to tr_{target} with an error tolerance of $\epsilon = 0.01$. Figure 2.2, at adaptation steps $x > 1$, shows the algorithm's normalizing effects over its iterations. The trace normalization led to a 0.1% increase in validation accuracy. This increase seems

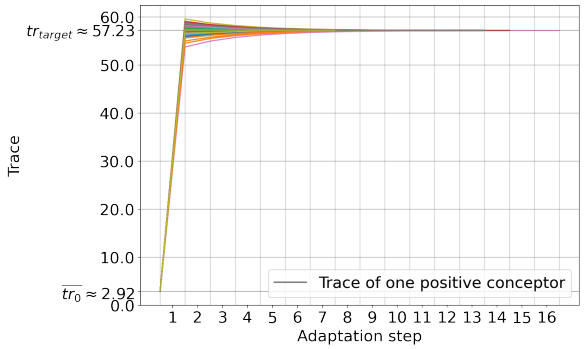

Figure 2.2: The traces of the positive conceptors in function of the adaption steps. The mean trace is initially at $\overline{tr_0}$. Step $x = 1$ is the aperture optimization based on the ∇-criterion. The increase in aperture caused the traces to increase and diverge. The remaining steps $x > 1$ correspond to the iterations of Algorithm A.1 that normalize the traces to target value tr_{target}.

insignificant, although significance could not be statistically verified on only one run. Following Occam's razor, the method was not used in the final run.

2.4.7 Negative conceptors

From the aperture-optimized positive conceptors in C^+, the set of negative conceptors $C^- = \{C^-_p | p \in P\}$ was computed. Each class's negative conceptor models the linear subspace complementary to the space occupied by the states of all other classes. In other words, it models the subspace that states from *none of the other classes* are expected to occupy. These semantics are reflected in their definition:

$$C^-_p = \neg \bigvee \{C^+_q | q \in P, q \neq p\}, \qquad (2.2)$$

where $\bigvee S$ is the associative disjunction of the $|S|$ conceptors of any set of conceptors S:

$$\bigvee S = ((C_1 \vee C_2) \vee C_3) \vee ... \, C_{|S|} \qquad (2.3)$$

Optimizing the apertures of the negative conceptors C^- decreased the validation accuracy by 0.97%, so it was not done in the final run.

2.4.8 Testing

The Combined Evidence was used to classify time series via the corresponding ESN responses. The

Combined Evidence $E(x, p)$ that some ESN state x corresponds to class p is a measure of similarity between that state and the positive and negative conceptors of class p. Concretely, it is the sum of a Positive Evidence $E^+(x, p)$, computed using the positive conceptor C_p^+, and a Negative Evidence $E^-(x, p)$, computed using the negative conceptor C_p^-:

$$\begin{aligned} E(x,p) &= E^+(x,p) + E^-(x,p), \\ E^+(x,p) &= x' C_p^+ x \\ E^-(x,p) &= x' C_p^- x \end{aligned} \tag{2.4}$$

$E(x, p)$ is large when x is close to the linear subspace modeled by conceptor C_p^+ but far from the linear subspace which the other conceptors model (see Equation 2.2).

To classify a point cloud X (a column-wise state collection matrix), the class that maximizes the mean Combined Evidence over all states (columns of the collection matrix) is assigned:

$$\arg\max_{p \in P} \frac{1}{L} \sum_{i=1}^{L} E(X[:, i], p), \tag{2.5}$$

where L is the width of X.

Finally, when given any time series s_{input}, it is assigned the class of maximal (mean) Combined Evidence for X_{input}, the ESN response to s_{input}.

2.4.9 Results

Table 2.2 shows the accuracies on the validation, training, and test sets, all significantly above chance. The confusion matrix in Figure A.1 of Appendix A.3.5 visualizes the classification rates across the classes and indicates a robust classification performance across classes. Like Jaeger (2014b), I repeated the experiment with enhanced inputs (Appendix A.5), which improved the training accuracy to 63.64% but lowered the test accuracy to 49.13%. Given the promising classification accuracies, I reused the ESN setup and conceptor-based classification mechanisms in Experiments 2 and 3, where I turn to the unsupervised extraction of concepts from ESN states.

Set	**Accuracy (%)**
Validation D_{val}	57.40
Training $D_{\text{TIMIT}-\text{train}}$	55.56
Test $D_{\text{TIMIT}-\text{test}}$	54.05
Chance	$\frac{100}{\lvert P \rvert} = 2.56$

Table 2.2: Validation, training, and test accuracies compared to chance.

2.5 Experiment 2: Concept Extraction by Conceptor-Based GCHC

2.5.1 Objective

Now blinded to the phonemic classes, the unlabeled ESN responses were clustered in four conditions that determined the parameters of GCHC, aiming to extract human-meaningful and coherent concepts without supervision.

2.5.2 Dataset

For computational constraints, the experiment was conducted using subsamples $D_{l,train} \subset D_{\text{TIMIT}-\text{train}}$ and repeated 10-fold ($l = 1, ..., 10$). Each $D_{l,train}$ encompassed a random subsample of time series with the following constraints. First, they stemmed from an independent sample 7 phonemes P_l to reduce selection bias. The cardinality of $|P_l| = 7$ aims to roughly align the difficulty with the simple condition of Lerato and Niesler (2012), which also performed clustering on TIMIT's phoneme recordings. Second, they stemmed from an independent sample of 48 speakers with an equal ratio across genders and dialect regions. This stratification aimed to represent the population evenly. Third, they stemmed from the phonetically compact sentences, where each phoneme appears in only a few phonetic contexts (e.g., /aa/ only before /f/), limiting phonetic variability. Given these constraints, 15 time series per phoneme in P_l were selected for $n' = 105$ time series in $D_{l,train}$, with the remaining time series $D_{l,test}$ held out for testing.

2.5.3 GCHC

Using each subsample $D_{l,train}$, GCHC was performed in four conditions. Next, I describe the algorithm, including parameters and procedure, followed by the conditions.

GCHC is a clustering algorithm that identifies K non-overlapping clusters (i.e., discrete concepts) among its inputs by iteratively relocating a set of cluster-defining centroids. Besides the number of clusters K, it is parametrized by:

1. A clustering set $D' = \{p_1, p_2, ..., p_n\}$ of the points to be clustered.
2. A centroid computation function centroid(Cl) that returns the centroid of a given cluster $Cl \subset D'$.
3. A dissimilarity function $d(p, \mu)$ that returns the dissimilarity between a point p and centroid μ. This function is assumed to be non-negative and monotonically increasing with dissimilarity. Symmetry is not assumed (cf. Sarmiento et al., 2019).

The algorithm minimizes loss function $\mathcal{L}_{GCHC}$:

$$\mathcal{L}_{GCHC} = \sum_{k=1}^{K} \sum_{p_i \in Cl_k} d(p_i, \text{centroid}(Cl_k)), \quad (2.6)$$

where Cl_k is the set of points in the k^{th} cluster.

To minimize this loss, the GCHC Algorithm 2.1 proceeds as follows. Centroids are initialized via K-means++ initialization. After randomly selecting a first initial centroid, this method selects each following initial centroid with probability proportional to its squared dissimilarity from the nearest already selected centroid. Compared to a random initialization, K-means++ initialization tends to produce more evenly spread initial centroids and converge faster and more consistently when applied to points in Euclidean space (Arthur and Vassilvitskii, 2007). After initialization, GCHC iteratively repeats the following steps:

- *Assignment step*: Each data point is assigned to the cluster with the least dissimilar centroid using $d(\cdot, \cdot)$.
- *Reassignment step*: To prevent empty clusters, which may occur after the *Assignment step* (Bradley and Fayyad, 1998), the most misfit point – the point with the largest dissimilarity from its current cluster's centroid – is reassigned to any empty clusters. Hence, all clusters have at least one data point from which a centroid can be computed.
- *Centroid update step*: The centroids are recalculated based on the newly formed clusters using the function centroid($\cdot$).

This process terminates once all centroids converge in their position or a maximum number of iterations is reached.

2.5.4 Conditions

The experiment was performed in four conditions that determine the parameters of GCHC:

- MFCC-Euclidean clustered time series directly. This condition acts as a baseline by directly clustering the time series by K-means, the special case of GCHC on points in Euclidean space.
- ESN-Euclidean clustered the earlier collected, but now unlabeled, ESN responses to the time series by K-means.
- ESN-Evidence clustered the ESN responses, but with conceptors as centroids and a derivate of the Combined Evidence as dissimilarity function.
- ESN-Hybrid combined the ESN-Euclidean and ESN-Evidence conditions.

Table 2.3 lists the arguments used as parameters (columns) across conditions (rows).

2.5.5 Runs

Algorithm 2.1 was run in each condition for 20 runs with different initializations to average out the effects of specific random cluster initialization choices. Thus, 20 folds $\times$ 20 runs = 400 runs were performed per condition. To limit the scope, the number of clusters was fixed at $K = 7$, consistent with the number of phonemic classes $|P_l|$.

2.5.6 Testing

The resulting clusters were evaluated using an *intrinsic* measure – based only on the data available during clustering – and two *extrinsic* measures – consulting the labels unavailable during clustering as ground truths. The intrinsic *mean intra-cluster dissimilarity* (MICD) aims to gauge the coherency of the extracted concepts, whereas the extrinsic *normalized mutual information* (NMI) and *cluster*

Condition	**Clustering Set** D'	**Centroid computation function** centroid(Cl_j)	**Dissimilarity function** $d(p, \mu_j)$
MFCC-Euclidean	MFCC time series: $\{s_i \| s_i \in D_{l,train}, s_i \in \mathbb{R}^{13\times10}\}$	Element-wise mean: $\frac{1}{\|Cl_j\|}\sum_{p_i\in Cl_j} p_i \in \mathbb{R}^{13\times10}$	Euclidean distance: $\|\|p-\mu_j\|\| \in [0,\infty)$
ESN-Euclidean	ESN responses: $\{X_i \| s_i \in D_{l,train}, X_i \in \mathbb{R}^{100\times10}\}$	Element-wise mean: $\frac{1}{\|Cl_j\|}\sum_{p_i\in Cl_j} p_i \in \mathbb{R}^{100\times10}$	Euclidean distance: $\|\|p-\mu_j\|\| \in [0,\infty)$
ESN-Evidence	ESN responses: $\{X_i \| s_i \in D_{l,train}, X_i \in \mathbb{R}^{100\times10}\}$	Tuple of positive and negative conceptors[a]: $(C_j^+, C_j^-) \in (\mathbb{R}^{100\times100}, \mathbb{R}^{100\times100})$	Reciprocal of Combined Evidence[b]: $\frac{1}{E(p,j)} \in [0,\infty)$
ESN-Hybrid	ESN responses: $\{X_i \| s_i \in D_{l,train}, X_i \in \mathbb{R}^{100\times10}\}$	Tuple of Euclidean- and conceptor-based centroids: $(\frac{1}{\|Cl_j\|}\sum_{p_i\in Cl_j} p_i, (C_j^+, C_j^-))$	Mean of Euclidean- and conceptor-based dissimilarities[b]: $\frac{\|\|p-\mu_j\|\|+\frac{1}{E(p,j)}}{2} \in [0,\infty)$

Table 2.3: Arguments across clustering conditions. [a]The ESN-Evidence centroid computation function resembles the training procedure of Experiment 1 with the clusters as classes. It returns a tuple of positive and negative conceptors representing class j with instances Cl_j. Note that implementing this function also relies on the members of the other clusters to compute the negative conceptors. [b]No division by zero was encountered.

classification accuracy (CCA) aim to assess human-meaningfulness of found concepts.

2.5.7 MICD

First, the MICD is the mean dissimilarity between the clusters' centroids and members across clusters and a measure of mean cluster cohesion. It resembles the loss function (Equation 2.6) but with additional normalization by K and each cluster's size. With normalization, clusters of varying sizes are considered equal, thus measuring the mean cluster cohesion, interpreted as the mean coherency of the extracted concepts. By its similarity to the loss, the MICD may also indirectly inform about convergence behavior; a steady reduction in MICD over iterations is expected as clusters become increasingly cohesive. Many alternative cohesion measures, like the within-cluster sum of squares or silhouette coefficient, were unsuitable because they assume a distance *metric* with criteria not fulfilled by the conceptor-based dissimilarity measures. Moreover, the MICD should not be compared between the conditions because they used different dissimilarity functions.

Concretely, given a clustering $Cl = \{Cl_1, Cl_2, ..., Cl_K\}$, where Cl_k is the set of points assigned to cluster k, the MICD is calculated as follows. For any cluster Cl_k, let the intra-cluster dissimilarity be the mean dissimilarity between its members p_i and its centroid centroid(Cl_k), where $d(p_i, \text{centroid}(Cl_k))$ is the dissimilarity function and depends on the condition. The MICD is then the mean of the intra-cluster dissimilarity values for all clusters:

$$MICD = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|Cl_k|}\sum_{p_i\in Cl_k} d(p_i, \text{centroid}(Cl_k)) \tag{2.7}$$

2.5.8 NMI

Second, the NMI is an extrinsic measure of the similarity between a clustering $Cl = \{Cl_1, Cl_2, ..., Cl_K\}$, where Cl_k is the set of points from the dataset $D_{l,train}$ assigned to cluster k, and the ground truth phonemic grouping $G = \{G_1, G_2, ..., G_{|P_l|}\}$, where G_p is the set of points with label p in the dataset

Algorithm 2.1 Generalized centroid-based hard clustering (GCHC) pseudocode

Require:
Number of clusters K
[1] Clustering set $D' = \{p_1, p_2, ..., p_n\}$
[2] Centroid computation function centroid(Cl)
[3] Dissimilarity function $d(p, \mu)$

Initialize K cluster centroids via the K-means++ initialization procedure: $\mu = \{\mu_1, \mu_2, ..., \mu_K\}$

while TRUE **do**
  Reset all clusters: $Cl = \{\emptyset, \emptyset, ..., \emptyset\}$

  Assignment step:
  for p_i in D' **do**
    $k \leftarrow \arg\min_{1 \leq j \leq K} d(p_i, \mu_j)$
    Assign p_i to cluster k: $Cl_k \leftarrow Cl_k \cup p_i$
  end for

  Reassignment step to avoid empty clusters:
  while any Cl_j is empty **do**
    Find point p_{max} in D' that is most dissimilar to the centroid of its current cluster.
    Move p_{max} from its old cluster to Cl_j. No p_{max} should be moved twice to prevent infinite loops.
  end while

  Centroid update step:
  for $j = 1$ to K **do**
    $\mu_j \leftarrow \text{centroid}(Cl_j)$
  end for

  if converged (cluster assignments did not change) or iteration limit reached **then**
    break
  end if
end while

return Set of clusters $Cl = \{Cl_1, Cl_2, ..., Cl_K\}$

$D_{l,train}$. Its values range from 0 for entirely dissimilar clusterings to 1 for identical clusterings. It was used to compare the found clusters with the phonemic classes of TIMIT, measuring their human-meaningfulness.

To compute the NMI, the mutual (shared) information I between Cl and G is normalized by the mean entropy (uncertainty) H within each clustering (Lerato and Niesler, 2012):

$$\begin{aligned} &I(G, Cl) \\ &= \sum_{G_p \in G} \sum_{Cl_k \in Cl} P(G_p \cap Cl_k) \log \frac{P(G_p \cap Cl_k)}{P(G_p)P(Cl_k)} \\ &H(Z) \\ &= -\sum_{Z_p \in Z} P(Z_p) \log P(Z_p), \text{ for some grouping } Z \\ &NMI(G, Cl) = \frac{I(G, Cl)}{\frac{1}{2}[H(G) + H(Cl)]} \end{aligned} \tag{2.8}$$

2.5.9 CCA

Third, the CCA measures the classification accuracy of the conceptors derived from the clusters on new phonemes. Like the NMI, the CCA is an extrinsic measure of cluster alignment with the ground truth phonemic classes. However, whereas the NMI focuses on the cluster members (i.e., the concept instances), the CCA focuses on conceptor-based representations of the clusters. The CCA is a practical attempt at interpreting clusters as human-meaningful concepts on par with phonemic classes by measuring phoneme classification performance. For computation, it was assumed that each cluster corresponded to exactly one of the phonemic classes in P. Clusters were matched with classes via the Kuhn-Munkres algorithm (Plummer and Lovász, 1986, mentioned in Song et al., 2013), using the `linear_sum_assignment` function from the SciPy library (Virtanen, Gommers, Oliphant, Haberland, Reddy, Cournapeau, Burovski, Peterson, Weckesser, Bright, van der Walt, Brett, Wilson, Millman, Mayorov, Nelson, Jones, Kern, Larson, Carey, Polat, Feng, Moore, VanderPlas, Laxalde, Perktold, Cimrman, Henriksen, Quintero, Harris, Archibald, Ribeiro, Pedregosa, van Mulbregt, and SciPy 1.0 Contributors, 2020). Concretely, this algorithm finds the match between clusters and classes

that globally maximizes the cumulative cardinality of the intersections between matched clusters and classes. Then, Experiment 1 was essentially replicated; a conceptor-based classifier was trained on the clusters to classify the respective matched class and tested on $D_{l,test}$.

2.5.10 Results

Table 2.4 compares the scores of the extrinsic performance measures (columns) of the clustering results across conditions (rows) averaged across runs. Two additional rows were added, Random clusters and Dataset classes, with the scores of a random clustering and a clustering that perfectly matched the ground truth class labels, respectively. All conditions significantly exceeded random clustering in NMIs and CCAs. A one-way repeated measures ANOVA revealed a significant effect of the condition on NMI ($F(3, 57) = 8.57$, $p < 0.001$). However, a Tukey post-hoc analysis revealed no significant difference pairwise differences between the conditions ($p > 0.05$). For CCA, the ANOVA revealed no significant effect of condition ($F(3, 57) = 1.19$, $p = 0.32$).

Condition	NMI	CCA
MFCC-Euclidean	0.488	0.459
ESN-Euclidean	0.468	0.455
ESN-Evidence	0.488	0.472
ESN-Hybrid	0.465	0.455
Random clusters	0.0871	0.251
Dataset classes	1.0000	0.700

Table 2.4: Mean extrinsic scores of final clusterings between conditions.

Figure 2.3 plots the NMIs over the iterations for each condition, averaged across runs. The NMIs increased across conditions during the first iterations but then stagnated.

Figure 2.4 illustrates the normalized MICDs over the iterations, averaged across runs for each condition. They were normalized by scaling to a unit range to accommodate the different original ranges of each condition. An exponential decay in MICDs is observed across all conditions. More than 70% of the change in MICDs and NMIs occurred across conditions within the first three iterations. The occasional peaks and progressive increases in variances

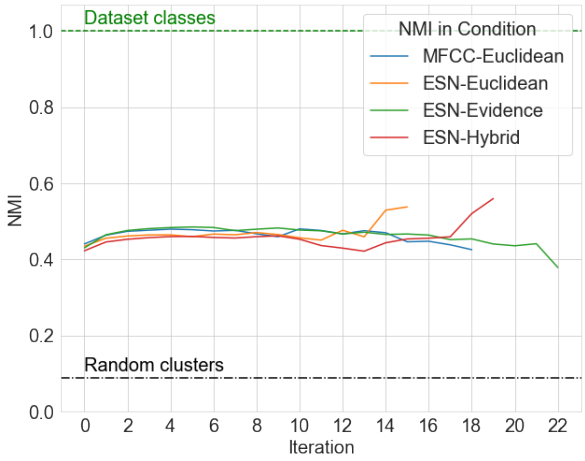

Figure 2.3: The NMIs of the clusterings over the iterations averaged across runs. For reference, horizontal lines indicate the NMIs of randomly initialized clusters and the dataset's classes.

can be attributed to the gradual termination of runs, causing changes in the samples underlying the mean computation (Appendix A.6.2).

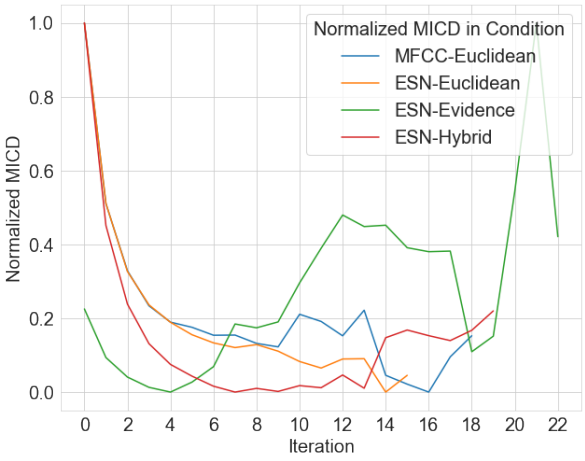

Figure 2.4: The MICDs of the clusterings over the iterations averaged across runs.

2.6 Experiment 3: Concept Extraction by Conceptor-Based HAC

2.6.1 Objective

This experiment aims to extract a hierarchy of concepts from the ESN responses to phonemes captured by the positive conceptors in C^+ from Experiment 1, using an adaptation of HAC. Whereas Experiment 2 extracted non-overlapping concepts among ESN states pre-grouped by phoneme segments, Experiment 3 extracted concept hierarchies among ESN states pre-grouped by phoneme classes.

2.6.2 HAC

I first describe HAC and then elaborate on how its three parameters were set to adapt it to conceptors. HAC is a clustering algorithm that identifies a cluster hierarchy among its inputs by iteratively grouping the inputs into higher-level clusters. The conceptor-adapted HAC is parametrized by:

1. A clustering set. Its points are classically in Euclidean space. However, here, they were the positive conceptors from Experiment 1, each representing one phoneme in conceptor space:

$$D_{HAC} = C^+ = \{C_p^+ | p \in P\} \tag{2.9}$$

2. A dissimilarity measure between two points p_i and p_j. Here, the difference from 1 of the conceptor similarity of p_i and p_j was used:

$$d_{HAC}(p_i, p_j) = 1 - \mathrm{Sim}(p_i, p_j) \tag{2.10}$$

   This function fulfills several desiderata for a dissimilarity measure: non-negative, symmetric, and zero for identical inputs.

3. A linkage function $d_{link}(Cl_i, Cl_j)$ that returns the dissimilarity between two clusters Cl_i and Cl_j. Here, the mean pairwise dissimilarity between the points in the two clusters (average linkage) was used:

$$d_{link}(Cl_i, Cl_j) = \frac{1}{|Cl_i||Cl_j|} \sum_{p_x \in Cl_i, p_y \in Cl_j} d_{HAC}(p_x, p_y) \tag{2.11}$$

   This linkage function tends to produce more balanced clusterings by considering the average rather than extreme values like some alternatives (Manning, Raghavan, and Schütze, 2008).

After defining the parameters, the HAC Algorithm 2.2 proceeds as follows. One cluster is initialized per data point. Then, HAC iteratively merges the clusters with the smallest dissimilarity d_{link} into a new cluster. This process terminates when only one cluster remains, resulting in a binary tree in which each node corresponds to a cluster formed by merging two children.

Algorithm 2.2 Hierarchical agglomerative clustering (HAC) pseudocode

Require:
[1] Set of points $D' = \{p_1, p_2, ..., p_n\}$
[2] Dissimilarity function $d(p_i, p_j)$
[3] Linkage function $d_{link}(Cl_i, Cl_j)$

Initialize a cluster for each point: $Cl \leftarrow D'$

while number of clusters > 1 **do**
  Find the two most similar clusters:
    $Cl_i, Cl_j \leftarrow \underset{Cl_i \neq Cl_j}{\arg\min}\, d_{link}(Cl_i, Cl_j)$
  Update clusters in Cl:
    Remove Cl_i and Cl_j
    Add $Cl_{merged} = Cl_i \cup Cl_j$
end while

2.6.3 Results

Figure 2.5 depicts the resulting cluster hierarchy as a dendrogram. Each leaf on the left represents a phoneme. Clusters emerge toward the right, representing increasingly abstract concepts. The abscissa of the links corresponds to the dissimilarity between the corresponding children.

Several overlaps could be identified between the HAC phoneme clustering results and phonetic groups depending on the choice of phonetic model. I begin by comparing the clusters based on the manner of production as provided by TIMIT (Figure 1.1). Table 2.5 depicts their overlaps with phonetic groups in the left column and their associated phonemes in the right column. Each group also corresponds to a HAC cluster except **bold** phonemes that were moved between sibling clusters. The two primary clusters encompass consonants (top) and vowels

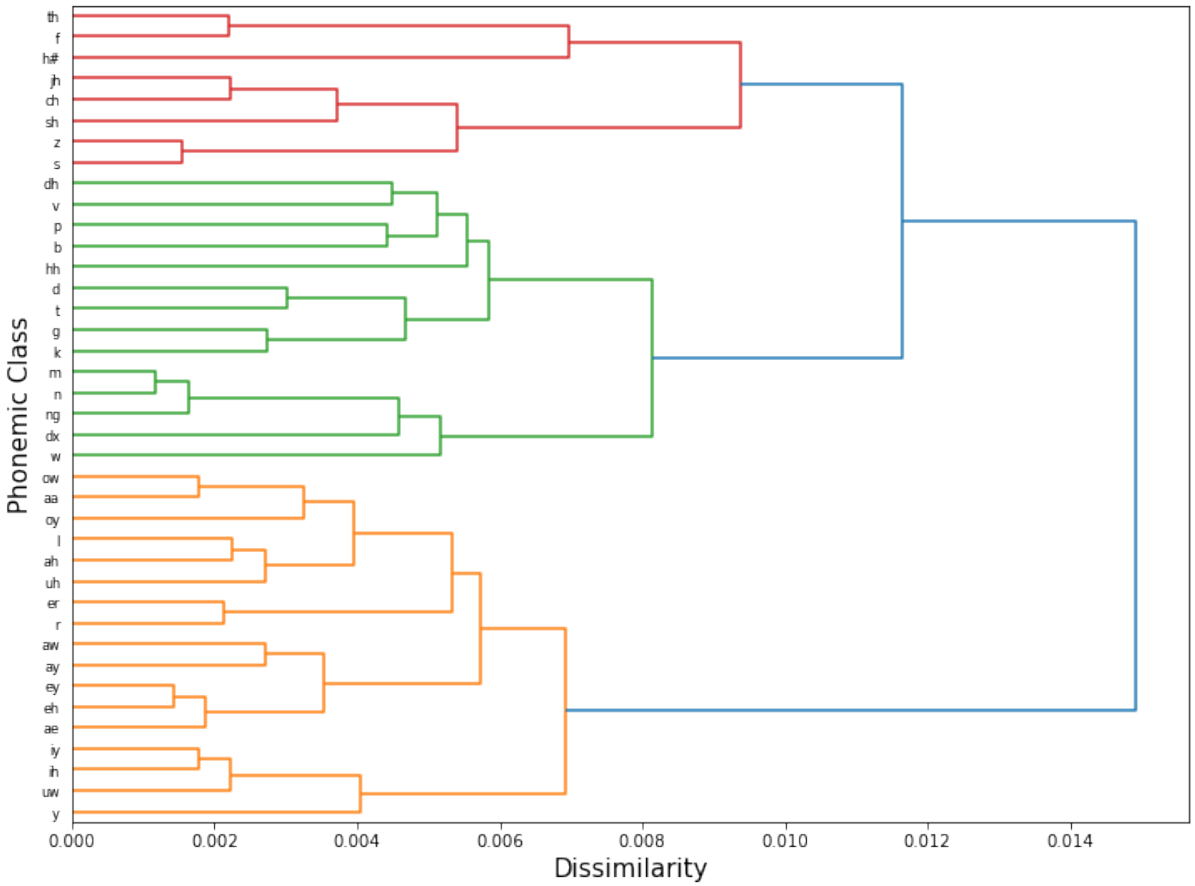

Figure 2.5: Dendrogram of the concept hierarchy from clustering phoneme-representing conceptors with the adapted HAC.

(bottom, orange), with a dissimilarity of about 0.015. Within the consonants cluster, five subgroups can be identified. The red subcluster corresponds to fricatives and affricates (enclosed), both produced with air friction. The green subcluster encompasses stops (top) and nasals (bottom). Within the vowel cluster of Figure 2.5, no significant correspondence between the sub-clusters and tongue positions or other articulatory features (like in Pfeifer and Balik (2011)) is apparent.

The phonemes in the Mixed group were considered separately because their articulatory features resemble both vowels and consonants. The group contains the glides /l/ and /r/, semivowels /w/ and /y/, and liquid /hh/. These instances were moved since no cluster exclusively corresponded to the mixed group. Lastly, the silence /h#/ was considered separately, like in TIMIT's categorization.

3 Discussion

The present work explored conceptor-based clustering for extracting coherent and human-meaningful concepts from ANN activations without supervision. The developed methods were demonstrated on ESN responses to pre-processed phoneme utterances from TIMIT.

3.1 Experiment 1

Experiment 1 served to (a) optimize the ESN on the data in preparation for the unsupervised Experiments 2 and 3 and (b) demonstrate conceptor-based time series classification on TIMIT.

Group		Phonemes of Group
Consonants	Fricatives	th f sh
		z s **dh v**
	Affricatives	jh ch
	Stops	p b d
		t g k **dx**
	Nasals	m n ng
Vowels		ow aa oy
		ah uh er
		aw ay ey eh
		ae iy ih uw
Mixed		**l r w y hh**
Silence		h#

Table 2.5: Phonetic categories (based on the manner of production) among the concepts that resulted from HAC. Bold phonemes were moved from sibling clusters into their current place.

Regarding objective (b), a testing accuracy of 54.05% was achieved, significantly above chance, suggesting that conceptor-based time series was successfully demonstrated on TIMIT. A difference of 1.51% to the training accuracy suggests a slight overfitting.

Unfortunately, a performance comparison to previous phoneme classification experiments on TIMIT is impeded by differences in data selection, preprocessing, and testing. For example, Karsmakers et al. (2007), holding the leading testing accuracy of 78.4%, pre-processed the data into 181-dimensional vectors by combining, among other features, 24 MFCCs and their first and second-order derivatives, which differs from my 13×10 dimensional MFCC time series.

More insightful conclusions can be drawn by comparison to previous conceptor-based time series classification methods. First, the current method, although lower in accuracy than Jaeger's (2014b) 99.9%, points to the applicability of conceptor-based classification to more complex datasets; With three times as many classes and greater phonetic variation, the pre-processed TIMIT dataset was more complex than their Japanese Vowels dataset. Likewise, Vlegels (2022) and Bartlett et al. (2019) distinguished between only 14 and 3 classes, respectively.

Further lessons lie in the attempted methodological deviations from Jaeger (2014b). The first attempted deviation was to normalize the traces of the positive conceptors C^+ across classes to mitigate a potential classification bias. However, since this intervention did not seem to increase validation accuracy significantly, it was not used in the final run. Future work should further investigate the described bias and its proposed remedy, empirically by statistically testing the effect of normalizing the trace on accuracy with cross-validation or analytically by verifying whether the proposed reasoning is sound, especially in the classification over multiple classes and with Negative Evidences. The second deviation from Jaeger (2014b) was not to optimize the apertures of the negative conceptors. This deviation was used in the final run since it improved validation accuracy significantly. Repeating Jaeger's (2014b) experiment is advisable to validate the benefits of the intervention.

Moreover, the attempt to further increase classification accuracy by modifying the data (Appendix A.5) was unsuccessful. This attempt involved concatenating the input and response before processing. Although this change led to a significant increase in training accuracy, testing accuracy decreased, suggesting overfitting. A possible reason for why this overfitting occurred for me, but not in Jaeger (2014b), is the difference in the dimensionality of the classified vectors z. My vectors were $(13 + 40) \times 10 = 530$-dimensional, compared to 88 dimensions in their setup. The information within the data might not have sufficed to learn the relevant patterns across such a high-dimensional space; the data required for effective learning increases exponentially with input dimensionality (curse of dimensionality (Bishop and Nasrabadi, 2006)). More generally, this issue highlights the complexity of regularizing a conceptor-based classifier (with $\frac{N(N+1)}{2}$ parameters to regularize with symmetric conceptors: quadratic in N) compared to alternatives like classification via a trained ESN output layer (with $|P| \times N$ output weight matrix parameters to regularize: linear in N). Thus, unless overfitting can be prevented, this modification seems to offer no benefit and even adds computational cost.

The fulfillment of motivation (a) rests on an assumption to be verified. The results suggest that the ESN could effectively represent the data such

that its phonemes could be distinguished during classification. Based on the earlier assumption, this discriminatory capacity of classification transfers to clustering on the same data and with the same ESN. The ESN was the same, and while the data pools changed for Experiment 2, these changes seem negligible. First, experiment 2 only used the phonetically compact sentences, but this constraint likely favored clustering by decreasing intra-class variability. Second, although Experiment 2 relied on subsets of seven of the original 39 phonemes, these subsets were randomly sampled. Moreover, by tuning the network to the pre-established phonemic classes, a confirmation bias may have acted in favor of their extraction, improving extrinsic validation scores. However, this bias likely had a minimal and uniform impact across all classes and conditions. Experiment 3, which used the same conceptors (i.e., data) as Experiment 1, does not face these concerns.

In conclusion, Experiment 1's significant classification accuracy points to the greater applicability of conceptor-based classification to more complex datasets. It also suggests that the involved ESN, with its hyperparameters, represented the data well enough to extract phonemic and super-phonemic concepts in Experiments 2 and 3.

3.2 Experiment 2

Experiment 2 explored the potential for unsupervised concept extraction from ANN activations in four variants of GCHC: MFCC-Euclidean (baseline K-means clustering on time series), ESN-Euclidean (K-means on ESN responses), ESN-Evidence (conceptor-based with Combined Evidence as the dissimilarity function), and ESN-Hybrid (a hybrid of ESN-Euclidean and ESN-Evidence).

On the one hand, all conditions outperformed random clustering by a significant margin in both extrinsic measures, NMI and CCA. This result suggests that GCHC, including its conceptor-based variants, could extract significantly human-meaningful concepts when evaluated against the phonemes of TIMIT.

On the other hand, although the first ANOVA indicated that at least one condition significantly differed from the others in its effect on NMI, the Tukey test did not reveal significant differences between any pairs of conditions. Thus, the experiment provided insufficient evidence to tell which method was more effective in extracting human-meaningful concepts. Similarly, the second ANOVA showed a 32% chance of observing the CCA results under the null hypothesis of no effect of condition. Future studies could replicate this experiment on larger datasets to address this uncertainty.

The striking similarity across conditions suggests high mutual information between ESN states and their inputs, providing evidence of their nature as high-dimensional non-linear expansions, echos, of the inputs (cf. Lukoševičius, 2012). In contrast, the activations of trained ANNs are attuned to specific features of the inputs. Thus, the activations of trained ANNs may encode concepts that GCHC may extract neither from inputs (MFCC-Euclidean) nor activations (ESN-Euclidean) but from conceptors (ESN-Evidences). Thus, I hypothesize that training favors the extrinsic performance of conceptor-based concept extraction.

Moving to the intrinsic measure, the exponential decay of MICDs across conditions (Figure 2.4) indicates that mean cluster cohesion increases as expected over iterations but at a decreasing rate. The occasional increases in MICDs could be attributed to the gradual termination. This course is consistent with the stagnation of NMIs (Figure 2.3) and suggests the proper convergence of GCHC and its ability to extract increasingly coherent and human-meaningful concepts over the iterations.

In conclusion, the experiment suggests significant potential in GCHC, including its conceptor-based adaptations, for extracting human-meaningful and coherent concepts. However, more research is needed to clarify whether there are significant differences to the Euclidean-based approaches.

3.3 Experiment 3

Experiment 3 adapted HAC in search of super-phonemic concepts among the 39 conceptor-represented phonemes. A strong resemblance between the extracted concepts and established phonetic categories suggests human-meaningfulness and coherency.

The resemblance was more apparent to categorizations based on manner and place of production than alternative phonetic features. Since the ESN and its concepts could have only represented acoustic information from isolated segments, this result underlines the dominant influence of place and man-

ner of production on acoustics suggested by the phonetic (International Phonetic Association, 1999; Garofolo et al., 1993; Rabiner, 1978) and reflected by the brain's representation of phonemic speech (Mesgarani et al., 2014; Shepard, 1980).

The only phonemes to be moved among sibling concepts for a neat alignment with production manner categories were among TIMIT's Fricatives and Semivowels and Glides (the latter also includes liquid /hh/). The hierarchical phoneme clustering of Oh et al. (2021) similarly struggled with these categories. The misalignment of fricatives likely resulted from their noisy acoustics caused by friction. Similarly, the misfit of semivowels, glides, and liquid reflects their intermediate articulatory properties. Therefore, the algorithm's errors may have been caused by and confirm phonetic properties.

In conclusion, the proposed conceptor-adapted HAC has the potential for extracting human-meaningful and coherent concepts, and the results reflected some established perspectives from phonetics.

3.4 Euclidean to Conceptor-based Clustering

In retrospect, the following changes enabled the adaptation of K-means and HAC, two classical clustering algorithms operating in Euclidean space, to conceptor-based representations. On the one hand, their mean and Euclidean distance functions could have also been applied to conceptor matrices as element-wise operations. On the other hand, this could have interfered with the geometric interpretation and the definition of conceptors; the formalism conceives conceptors as hyperellipsoids in state space and as "regularized identity maps" (Jaeger, 2014b), as captured by their loss. These aspects are crucial for, i.a., the semantics of their operators. However, the mean and Euclidean distance lack such a clear geometric interpretation. Moreover, adding two conceptors, as done by the mean, does not necessarily return a new conceptor. Thus, for consistency with the original framing of conceptors, the following adaptations were made in ESN-Evidence and ESN-Hybrid of Experiment 2 and in Experiment 3:

1. The mean became disjunction with aperture adaptation in the *Centroid update step* of Experiment 2. In classical K-means, cluster centroids are computed via the mean, which collects the information of member points by summation and normalizes it by division by cluster size. By analogy, the information of member conceptors was collected by disjunction and normalized through aperture adaptation.
2. The Euclidean distance became the reciprocal of the Combined Evidence in the *Assignment step* of Experiment 2 and the difference from 1 of the conceptor similarity in Experiment 3. Classical K-means assigns points to clusters by Euclidean distance to the respective centroids. By analogy, conceptor points were assigned by the reciprocal of the Combined Evidence, $1/E(\cdot,\cdot)$, for the respective centroid. Moreover, in classical HAC, the distance between two clusters is computed by the mean Euclidean distance of their members. By analogy, the distance between conceptor clusters was computed by the mean of their members' dissimilarity, $1 - Sim(\cdot,\cdot)$.

The ESN-Hybrid condition is worth highlighting. By combining the ESN-Euclidean and ESN-Evidence, the hybrid carried information about both centers and spreads of clusters; the centroids of ESN-Euclidean are cluster means, and the centroids of ESN-Evidence are conceptors that capture cluster variance. Therefore, I expected it to benefit from a richer cluster representation than its constituents. The high hopes for the hybrid were backed by its resemblance to affine conceptors, an extension proposed by Jaeger (2014b), which accounts for both the variance and the mean of state clouds in the conceptor computation. By not forcing the ellipsoids to be centered at the origin, these affine conceptors may more accurately model state clouds. However, the insignificant performance differences between the ESN-Hybrid and any of the other conditions cast doubt on its benefits. If the still outstanding development of affine conceptors succeeds, they could potentially enable a simpler, computationally cheaper, and more effective means of integrating information about cluster means and variances.

Other adaptations would also have been possible. For instance, Mossakowski et al. (2019) used the fuzzy generalization of the Löwner ordering to cluster conceptors. Mossakowski et al. (2019) lacking validation, Experiment 3 provides more rigorous evidence that clustering conceptors can extract coher-

ent concepts. Regarding the dissimilarity, however, an informal adaptation of Experiment 3's method to that dissimilarity function seemingly led to less human-meaningful taxonomies on the present task. A dedicated study could aim to elucidate the differences between our dissimilarity measures. Ultimately, choosing a dissimilarity function that identifies meaningful and coherent concepts can require the consideration of the properties of the input, neural network, and task (Shepard, 1980).

3.5 Computational Creativity

I will now discuss the implications of this study in light of the two primary motivations. The first motivation was to advance neuro-symbolic approaches to computational creativity. Previous computational approaches with creative potential relied on the availability of pre-established or manually identified concepts (Jaeger, 2014a; Lake et al., 2019; Fabi et al., 2021; Hofstadter et al., 1987; Hofstadter and Mitchell, 1994; Ha and Eck, 2017). The developed methods pave the way toward overcoming this reliance by autonomously extracting relevant concepts; their subsequent combination, morphing, or analogy could generate new relevant concepts. Implementations might draw on traditional cognitive models of creativity, the various mechanisms available for conceptors, and the (re)generative capacities of ESNs with output layers (Jaeger, 2014a). A similar case as for creativity may hold for other cognitive functions (Sheth et al., 2023); various conceptual-symbolic models of cognition exist that future research could attempt to apply to data-dynamical content with unsupervised concept extraction methods as a bridge.

3.6 Unsupervised Conceptors

Regarding the second motivation, the present method could extend the conceptor formalism to unsupervised applications and enhance supervised ones. The current task differs from previous ones. Unlike the supervised task of computing the single conceptor that best captures a group of states, Experiments 2 and 3 aimed to find a set of conceptors that collectively best capture an ungrouped set of states. By pivoting toward unsupervised conceptors, this attempt could broaden the applications of conceptors, for example, by enhancing supervised approaches (time series classification like in Experiment 1, Jaeger (2014b), Bartlett et al. (2019), and Vlegels (2022), and recognition like in Chatterji (2022)); extracted concepts could be used as in Oh et al. (2021), which informed and improved phoneme classification on TIMIT by training additional classifiers on super-phonemic concepts extracted by hierarchical clustering. More research is needed to explore the formation and use cases of unsupervised conceptors.

3.7 Practical Applications

Additional applications result from the present method's capacity to represent various types of time-extended data, including brain data, ANN activities, and other time series. First, it could integrate with previous studies on concept-extraction from brains (Kriegeskorte and Kievit, 2013; Lin, Mur, Kietzmann, and Kriegeskorte, 2019; Tucciarelli et al., 2019; Balkenius and Gärdenfors, 2016; Shepard, 1980; Chang et al., 2010; Laakso and Cottrell, 2000), building on the insights of Bartlett et al. (2019) for applying conceptors to brain data. Thus, it could contribute to better understanding the conceptual structures by which the human brain represents and categorizes stimuli. In this context, the conceptor-based representation may open possibilities like ordering the extracted concepts by their abstraction relationships (Mossakowski et al., 2019; Jaeger, 2014b; Bricman et al., 2022).

Second, in the context of ANN explainability, the approach could improve on previous clustering-based concept extraction methods (Fel et al., 2024; Liu and Arik, 2020; Ghorbani et al., 2019; Song et al., 2013) by offering new ways of analyzing the concepts post-extraction. For example, an extracted concept's neural pattern may be reactivated for inspection by projecting the NN's state into the corresponding conceptor's subspace. Moreover, conceptor-represented concepts could be compared to user-queried concepts (cf. Kim et al., 2018) and among each other, where a possible pipeline could extract base concepts by Experiment 2's method and relate them by Experiment 3's method (i.e., analyzing the representational geometry (Kriegeskorte and Kievit, 2013)). Similarly, Nested State Clouds (Bricman et al., 2022) may relate the concepts extracted by my methods in a semantic graph.

Third, the current methods may offer performance and efficiency benefits for time series clustering. As

mentioned, the high mutual information between inputs and ESN responses enables the mutual transfer of clusterings and classifications. Thus, the methods of Experiments 2 and 3, although formulated with a focus on neural activations, can be applied to cluster the input time series (cf. Estevan, Wan, and Scharenborg, 2007, Atencia, Gallicchio, Joya, and Micheli, 2020). The most similar study, Lerato and Niesler (2012), was outperformed in NMIs by about an order of magnitude across conditions. This comparison is based on the NMI of their HAC at seven clusters in their simple condition. We used an equivalent definition of the NMI. While this tenfold improvement suggests a significant contribution to the field of time series clustering, it may have not only resulted from differences in clustering methods; perhaps normalizing phoneme segments to 10-step time series was superior to their dynamic time warping in this context. Moreover, restricting the data to vowels, which I did not do for consistency with my other experiments, may have added difficulty to Lerato and Niesler's (2012) task.

Moreover, my method can be more efficient than other time series clustering approaches. For example, dynamic time warping, used to compute distances between variable-length time series for clustering (Lerato and Niesler, 2012), scales quadratically with sample length. Conceptors, however, can compress variable-length time series, and even groups of time series, to constant-sized objects; the conceptor computation is linear in their (cumulative) length, and subsequent conceptor comparisons are independent of it. However, since the present method dealt with the variability in phoneme segment lengths by normalizing them in time, their original lengths did not affect time complexity once pre-processed, and more research is needed to explore this efficiency benefit.

3.8 Time Efficiency and Convergence

The time complexity of the GCHC algorithm in Experiment 2 may be expressed in terms of the clustering set size n, number of clusters K, input dimensionality d, ESN dimensionality N and number of iterations required for convergence T. The implementation includes the K-means++ initialization and then repeats the *Assignment* and *Centroid update steps* for T iterations. The complexity of each component depends on the conditions, as shown in Table 3.1.

Condition	**Dissimilarity computation**	***Assignment step***
MFCC-Euclidean	Euclidean distance in $O(d)$	$O(nKd)$
ESN-Euclidean	Euclidean distance in $O(N)$	$O(nKN)$
Conceptor-based	Combined Evidence in $O(N^2)$	$O(nKN^2)$
	Centroid computation	***Centroid update step***
MFCC-Euclidean	Mean in $O(d)$	$O(nd)$
ESN-Euclidean	Mean in $O(N)$	$O(nN)$
Conceptor-based	Conceptor computation in $O(N^3)$	$O(KN^3)$

Table 3.1: The complexity classes of the main steps of GCHC (third column) and their most expensive computations (second column) across conditions (rows).

The dissimilarity computation dominates the *Assignment step*, while the centroid computation dominates the *Centroid update step*. With the current parameters, both computations are significantly more expensive in the conceptor-based conditions, which involve matrix multiplications and inversions of complexity $O(N^3)$; in contrast, the Euclidean-based conditions scale linearly with input or ESN dimensionality. Concluding, GCHC has time complexities of:

- $O(TnKd)$ for MFCC-Euclidean
- $O(TnKN)$ for ESN-Euclidean
- $O(TKN^3)$ for conceptor-based conditions

In practice, Euclidean-based conditions may be favored for their computational efficiency. A middle way could be to identify clusters using a Euclidean-based representation, exploiting its efficiency, and then compute conceptors for each cluster, thereby still profiting from the geometric interpretability and mechanisms available with a conceptor-based

representation. The insignificant between-condition differences in extrinsic performances facilitate this practical flexibility.

The adapted HAC algorithm in Experiment 3 begins by computing $O(n^2)$ pairwise dissimilarities, each involving the conceptor similarity function with matrix multiplications and inversions in $O(N^3)$. This first step amounts to $O(n^2N^3)$ operations. Then, the algorithm, over $n-1$ iterations, finds the two closest clusters to merge using $O(n^2)$ simple float comparisons. This second step amounts to $O(n^3)$ operations. With the current parameters, the adapted HAC algorithm is dominated by the initial dissimilarity computation, resulting in a time complexity of $O(n^2N^3)$. Thus, when comparing the conceptor-based clustering approaches, GCHC scales better than HAC with the clustering set size.

Convergence could be guaranteed in some conditions. In Experiment 2, methods akin to GCHC typically assume a distance function that fulfills the triangle inequality for convergence (Banerjee, Merugu, Dhillon, Ghosh, and Lafferty, 2005). This assumption holds for the Euclidean, but not the conceptor-based conditions, which use Combined Evidences. Nonetheless, the consistently and continuously decaying MICDs (Appendix A.6.2) suggest convergence across conditions to local optima. HAC makes no such assumptions on the dissimilarity function (Jain, 1988), guaranteeing convergence even with the conceptor-based dissimilarity function in Experiment 3.

3.9 Limitations and Future Directions

Several limitations of the present study remain to be addressed by future research. Confounding variables introduced additional difficulty. Classes of variables like gender, dialect region, and co-articulatory context (especially in Experiment 2, which was constrained to phonetically compact sentences) may have competed with phonemes and other phonetic categories for clusters. Future studies could use larger and more diverse datasets to dilute the effects of confounding variables while assessing generalizability.

Moreover, the methods extracted concepts from the representations of pre-segmented signals. Further research could generalize it to unsegmented signals, potentially building on the attempt in Appendix A.7, to extend its applicability.

Lastly, one may object that the phonemic concepts extracted in this work stretch or oversimplify the perhaps more popular conception of concepts as high-level entities like "cow". However, this distinction dissolves at a neural level; concepts may differ semantically, but structurally, they seem equally reducible to observable patterns of neural activity (according to physicalism). Focusing on the smallest possible concepts of speech allowed for more methodological exploration without losing control. Nonetheless, future research could apply the presented method to more abstract concepts, such as those represented by large language models. Human feedback could be consulted to evaluate the meaningfulness of extracted higher-level concepts (Ghorbani et al., 2019). In this attempt, alternative clustering algorithms or dissimilarity measures may also bear potential; soft clustering, for example, might enable the extraction of fuzzy concepts (cf. Chatterji, 2022, Mossakowski et al., 2019) by assigning states a degree of membership.

4 Conclusions

In summary, Experiment 1 successfully demonstrated conceptor-based phoneme classification and tuned the ESN for the subsequent concept extraction experiments. Experiment 2 found that a generalized centroid-based hard clustering, when paired with conceptors, can extract human-meaningful and coherent concepts, although the conceptor-based approach had no significant edge over the simpler and more efficient ESN- and time series-based conditions. Experiment 3 adapted hierarchical agglomerative clustering to conceptors, extracting human-meaningful and mostly coherent concepts with a strong correspondence to established phonetic categories related to place and manner of articulation.

Collectively, these findings suggest that conceptor-based clustering can extract human-meaningful and coherent concepts from the activations of ANNs without supervision. While promising use cases lie in brain concept extraction, enhancing supervised conceptor-based methods, computational creativity, time-series clustering, and ANN explainability, the most critical next research step seems to be applying the present methods to trained architectures

to verify the differences between conceptor-, ESN response- and input-based approaches to concept extraction.

References

David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

Miguel Atencia, Claudio Gallicchio, Gonzalo Joya, and Alessio Micheli. Time series clustering with deep reservoir computing. In *International Conference on Artificial Neural Networks*, pages 482–493. Springer, 2020.

Christian Balkenius and Peter Gärdenfors. Spaces in the brain: From neurons to meanings. *Frontiers in psychology*, 7:1820, 2016.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. Clustering with bregman divergences. *Journal of machine learning research*, 6(10), 2005.

Madeleine Bartlett, Daniel Hernandez Garcia, Serge Thill, and Tony Belpaeme. Recognizing human internal states: A conceptor-based approach. *arXiv preprint arXiv:1909.04747*, 2019.

Sara D Beach, Ola Ozernov-Palchik, Sidney C May, Tracy M Centanni, John DE Gabrieli, and Dimitrios Pantazis. Neural decoding reveals concurrent phonemic and subphonemic representations of speech across tasks. *Neurobiology of Language*, 2(2):254–279, 2021.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Valentina Borghesani and Manuela Piazza. The neuro-cognitive representations of symbols: the case of concrete words. *Neuropsychologia*, 105: 4–17, 2017.

Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer, 1998.

Paul Bricman, Dr Herbert Jaeger, and Dr Jacolien van Rij-Tange. Nested state clouds: Distilling knowledge graphs from contextual embeddings. Bachelor's thesis, University of Groningen, 8 2022.

Ilana Bromberg, Qian Qian, Jun Hou, Jinyu Li, Chengyuan Ma, Brett Matthews, Antonio Moreno-Daniel, Jeremy Morris, Sabato Marco Siniscalchi, Yu Tsao, et al. Detection-based asr in the automatic speech attribute transcription project. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

Gijs Joost Brouwer and David J Heeger. Categorical clustering of the neural representation of color. *Journal of Neuroscience*, 33(39):15454–15465, 2013.

Edward F Chang, Jochem W Rieger, Keith Johnson, Mitchel S Berger, Nicholas M Barbaro, and Robert T Knight. Categorical speech representation in human superior temporal gyrus. *Nature neuroscience*, 13(11):1428–1432, 2010.

Satchit Chatterji. Cut the carp! using context to disambiguate similar signals using conceptors. Bachelor's thesis, University of Groningen, 8 2022.

Radwa El Shawi. Conceptglassbox: Guided concept-based explanation for deep neural networks. *Cognitive Computation*, pages 1–14, 2024.

Yago Pereiro Estevan, Vincent Wan, and Odette Scharenborg. Finding maximum margin segments in speech. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–937. IEEE, 2007.

Brian S. Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. Wiley, Chichester, West Sussex, United Kingdom, 5th edition, 2011. ISBN 978-0-470-74991-3.

Sarah Fabi, Sebastian Otte, and Martin V Butz. Compositionality as learning bias in generative rnns solves the omniglot challenge. In *Learning to Learn-Workshop at ICLR 2021*, 2021.

Thomas Fel, Victor Boutin, Louis Béthune, Rémi Cadène, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach

to unifying automatic concept extraction and concept importance estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren. Timit acoustic-phonetic continuous speech corpus. Technical Report LDC93S1, Linguistic Data Consortium, Philadelphia, 1993. URL `https://catalog.ldc.upenn.edu/LDC93S1`.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.

Mara Graziani, An-phi Nguyen, Laura O'Mahony, Henning Müller, and Vincent Andrearczyk. Concept discovery and dataset exploration with singular value decomposition. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

Douglas R Hofstadter and Melanie Mitchell. The copycat project: A model of mental fluidity and analogy-making. In Keith J Holyoak and John A Barnden, editors, *Advances in connectionist and neural computation theory*, volume 2, pages 31–112. Ablex Publishing Corporation, Norwood, NJ, 1994.

Douglas R Hofstadter, Melanie Mitchell, and Robert Matthew French. *Fluid concepts and creative analogies: A theory and its computer implementation*. University of Michigan, Cognitive Science and Machine Intelligence Laboratory, 1987.

Alexander G Huth, Shinji Nishimoto, An T Vu, and Jack L Gallant. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 76(6):1210–1224, 2012.

Alexander G Huth, Wendy A De Heer, Thomas L Griffiths, Frédéric E Theunissen, and Jack L Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532 (7600):453–458, 2016.

International Phonetic Association. *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, 1999. ISBN 978-0-521-63751-0. URL `https://www.cambridge.org/core/books/handbook-of-the-international-phonetic-association/1E4CD39BACE0D72A156E4322AE45EF19`.

Herbert Jaeger. From continuous dynamics to symbols. In *Dynamics, synergetics, autonomous agents: Nonlinear systems approaches to cognitive psychology and cognitive science*, pages 29–48. World Scientific, 1999.

Herbert Jaeger. Conceptors: an easy introduction. *arXiv preprint arXiv:1406.2671*, 2014a.

Herbert Jaeger. Controlling recurrent neural networks by conceptors. *arXiv preprint arXiv:1403.3369*, 2014b.

AK Jain. Algorithms for clustering data, 1988.

Peter Karsmakers, Kristiaan Pelckmans, Johan AK Suykens, and Hugo Van hamme. Fixed-size kernel logistic regression for phoneme classification. In *INTERSPEECH*, pages 78–81, 2007.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

Seonggyeom Kim and Dong-Kyu Chae. What does a model really look at?: Extracting model-oriented concepts for explaining deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Nikolaus Kriegeskorte and Rogier A Kievit. Representational geometry: integrating cognition, computation, and the brain. *Trends in cognitive sciences*, 17(8):401–412, 2013.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, page 4, 2008.

Aarre Laakso and Garrison Cottrell. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76, 2000.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

K-F Lee and H-W Hon. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11):1641–1648, 1989.

Lerato Lerato and Thomas Niesler. Investigating parameters for unsupervised clustering of speech segments using timit. In *Twenty-Third Annual Symposium of the Pattern Recognition Association of South Africa*, page 83, 2012.

Baihan Lin, Marieke Mur, Tim Kietzmann, and Nikolaus Kriegeskorte. Visualizing representational dynamics with multidimensional scaling alignment. *arXiv preprint arXiv:1906.09264*, 2019.

Yu-han Liu and Sercan O Arik. Explaining deep neural networks using unsupervised clustering. *arXiv preprint arXiv:2007.07477*, 2020.

Carla Lopes and Fernando Perdigao. Phoneme recognition on the timit database. In Ivo Ipsic, editor, *Speech Technologies*, chapter 14. IntechOpen, Rijeka, 2011. doi: 10.5772/17600. URL `https://doi.org/10.5772/17600`.

Mantas Lukoševičius. A practical guide to applying echo state networks. *Neural Networks: Tricks of the Trade: Second Edition*, pages 659–686, 2012.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Single-link, complete-link & average-link clustering. `https://nlp.stanford.edu/IR-book/completelink.html`, 2008. Accessed: 2024-10-18.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

Jessica Maye and LouAnn Gerken. Learning phonemes without minimal pairs. In *Proceedings of the 24th annual Boston university conference on language development*, volume 2, pages 522–533, 2000.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt Mcvicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In Kathryn Huff and James Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 18–24, Austin, TX, July 2015. doi: 10.25080/Majora-7b98e3ed-003.

Nima Mesgarani, Connie Cheung, Keith Johnson, and Edward F Chang. Phonetic feature encoding in human superior temporal gyrus. *Science*, 343 (6174):1006–1010, 2014.

Robert M Mok and Bradley C Love. A non-spatial account of place and grid cells based on clustering models of concept learning. *Nature communications*, 10(1):5685, 2019.

Till Mossakowski, Razvan Diaconescu, and Martin Glauer. Towards fuzzy neural conceptors. *IfCoLog Journal of Logics and their Applications*, 6(4):725–744, 2019. URL `https://collegepublications.co.uk/ifcolog/?00033`.

Fernando Nogueira. Bayesian optimization, 2014. URL `https://github.com/fmfn/BayesianOptimization`. Accessed: October 19, 2024.

Donghoon Oh, Jeong-Sik Park, Ji-Hwan Kim, and Gil-Jin Jang. Hierarchical phoneme classification for improved speech recognition. *Applied Sciences*, 11(1):428, 2021.

Vaclav Pfeifer and Miroslav Balik. Comparison of current frame-based phoneme classifiers. *Advances in Electrical and Electronic Engineering*, 9, 12 2011. doi: 10.15598/aeee.v9i5.545.

Michael D Plummer and László Lovász. *Matching theory*, volume 121. Elsevier, 1986.

Lawrence R Rabiner. *Digital processing of speech signals*. Pearson Education India, 1978.

Lance J Rips, Edward E Smith, and Douglas L Medin. 11 concepts and categories: Memory,

meaning, and metaphysics. *The Oxford handbook of thinking and reasoning*, page 177, 2012.

Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. Neuro-symbolic artificial intelligence. *AI Communications*, 34(3):197–209, 2021.

Auxiliadora Sarmiento, Irene Fondón, Iván Durán-Díaz, and Sergio Cruces. Centroid-based clustering with $\alpha\beta$-divergences. *Entropy*, 21(2):196, 2019.

Roger N Shepard. Multidimensional scaling, tree-fitting, and clustering. *Science*, 210(4468):390–398, 1980.

Amit Sheth, Kaushik Roy, and Manas Gaur. Neurosymbolic artificial intelligence (why, what, and how). *IEEE Intelligent Systems*, 38(3):56–62, 2023.

Chunfeng Song, Feng Liu, Yongzhen Huang, Liang Wang, and Tieniu Tan. Auto-encoder based data clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 18th Iberoamerican Congress, CIARP 2013, Havana, Cuba, November 20-23, 2013, Proceedings, Part I 18*, pages 117–124. Springer, 2013.

Raffaele Tucciarelli, Moritz Wurm, Elisa Baccolo, and Angelika Lingnau. The representational space of observed actions. *elife*, 8:e47686, 2019.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Jamie Vlegels. Multivariate time series classification using conceptors: Exploring methods using astronomical object data. Bachelor's thesis, University of Groningen, 2022.

Izzet B Yildiz, Herbert Jaeger, and Stefan J Kiebel. Re-visiting the echo state property. *Neural networks*, 35:1–9, 2012.

A Appendix A

A.1 Robust Conceptor Conjunction

$$C \wedge B := (P_{R(C)\cap R(B)}(C^\dagger + B^\dagger - I)P_{R(C)\cap R(B)})^\dagger \tag{A.1}$$

This more robust definition of conceptor conjunction was used. An algorithm for computing projector matrix $P_{R(C)\cap R(B)}$ is given on pp. 174-175 of Jaeger (2014b).

A.2 Dataset

Table A.1 lists the phone labels, their frequencies within the processed TIMIT dataset, and the corresponding phonemes.

A.3 Additions to Experiment 1

A.3.1 Aperture Optimization using the ∇-criterion

We are given a set of conceptors $C = \{C_p | p \in P\}$ representing a set of classes P and computed with initial apertures of $\alpha = 1$. The objective is to maximize the sensitivity to differences in the underlying ESN dynamics since we assume this to improve the expected classification performance on similarly distributed testing data. This objective is operationalized by the ∇-*criterion*, a function of conceptor C and candidates aperture adaptation factor γ [¶]. It returns the gradient of the Frobenius norm of the aperture-adapted conceptor with respect to the logarithm of γ:

$$\nabla(C, \gamma) = \frac{d}{d\ \log(\gamma)} ||\varphi(C, \gamma)||^2 \tag{A.2}$$

Intuitively, this gradient corresponds to the sensitivity of conceptor C to changes in aperture (i.e., scalings of the underlying ESN states, see p. 49 of Jaeger (2014b)). In Experiment 1, the optimal aperture γ_p was approximated for each conceptor C_p by sweeping through 200 candidate values $\gamma_{candidate}$ in the interval $[0.001, 500)$ on a logarithmic scale; logarithmic, for the optimal value, was expected on the lower end of the interval. γ_p was set to the $\gamma_{candidate}$ that maximized a numerical approximation of $\nabla(C_p, \gamma_{candidate})$. This derivative was numerically approximated using a finite forward difference

[¶] In this case, γ equals the resulting aperture α_{new}, since $\gamma = \frac{\alpha_{new}}{\alpha}$ and the current aperture $\alpha = 1$.

	Phone	Folded	#Training	#Test
	iy		6953	2710
	ih	ix	13693	4654
	eh		3853	1440
	ae		3997	1407
	ah	ax-h ax	6291	2343
	uw	ux	2463	750
	uh		535	221
	aa	ao	6004	2289
	ey		2282	806
	ay		2390	852
	oy		684	263
	aw		729	216
	ow		2136	777
	l	el	6752	2699
	r		6539	2525
	y		1715	634
	w		3140	1239
	er	axr	5453	2183
	m	em	4027	1573
	n	en nx	8762	3112
	ng	eng	1368	419
	ch		822	259
	jh		1209	372
	dh		2826	1053
	b		2181	886
	d		3548	1245
	dx		2709	940
	g		2017	755
	p		2588	957
	t		4364	1535
	k		4874	1614
	z		3773	1273
	v		1994	710
	f		2216	912
	th		751	267
	s		7475	2639
	sh	zh	2389	870
	hh	hv	2111	725
	h# (silence)	dcl tcl kcl bcl pcl pau epi q gcl	39467	14021
Σ	39	22	177080	64145

Table A.1: Phone labels and their frequencies. Column 1: Phone classes. Column 2: Phones originally present in TIMIT but folded into a class with the left-adjacent phone. Columns 3 and 4: Number of speech samples of each class. Bottom row: Sums of classes or samples.

with a step size of $\Delta\gamma = 10^{-4}$. $|P| = 39$ values γ_p ($p \in P$) resulted. Finally, the apertures of all positive conceptors were adapted using the mean $\gamma_{opt} = \frac{1}{|P|}\sum_{p\in P}\gamma_p \approx 133.98$ resulting in the set of aperture-optimized conceptors C_{opt}.

A.3.2 Trace Normalization

We are given a conceptor C and a target trace tr_{target}. The objective is to adapt C to tr_{target}, maintaining its semantics as much as possible. In the following Algorithm A.1, the trace of C is adapted indirectly through its aperture; the algorithm iteratively adapts the aperture by a factor of the current trace error ratio until reaching the target trace. This indirection is effective since for the conceptors in C^+_{opt} (and any other *soft* conceptors with at least one singular value strictly between 0 and 1), the trace and aperture are positively related, as demonstrated in Proposition 1 (Appendix A.3.3). This indirection seemed necessary since it was unclear how to otherwise adapt a conceptor's trace while preserving its idempotency ($C = C^2$) and most of its semantics.

Algorithm A.1 Adapt the trace of a conceptor

Require:
Conceptor C whose trace is to be adapted to tr_{target}
Target trace tr_{target}
Error tolerance ϵ

while TRUE **do**
  if $|tr_{target} - tr(C)| < \epsilon$ **then**
    break
  end if
  $\gamma \leftarrow tr/\mathrm{tr}(C)$
  $C \leftarrow \varphi(C, \gamma)$
end while
return C

A.3.3 Proposition 1

Let C be a conceptor with aperture $\alpha \in (0, \infty)$ and singular values $s_1, ..., s_N$ with at least one $s_i \in (0, 1)$. Let conceptor $C_{new} = \varphi(C, \alpha_{new}/\alpha)$ be the aperture-adaptation of C with new aperture $\alpha_{new} \in (0, \infty)$ and trace $tr(C_{new})$. Then, $\frac{dtr(C_{new})}{d\alpha_{new}} > 0$.

A.3.5 Additional Results of Experiment 1

The confusion matrix in Figure A.1 shows the classification rates across the phonemic classes. For every phoneme, their correct classification rate (diagonal entries) was higher than the misclassification rate as any individual other class (off-diagonal entries), suggesting a robust, above-chance classification performance across all phonemic classes. Error rates seem elevated within groups with a shared manner of production, such as vowels (top left quadrant) and consonants (bottom right quadrant).

A.4 Additions of Experiment 2: Hyperparameter Optimization

Hyperparameters k_b, $k_{W^{in}}$, r, and ρ were also tuned automatically via Bayesian optimization using the `Bayesian Optimization` python package (Nogueira, 2014). Bayesian optimization was preferred over a more straightforward grid search since, considering the high computational complexity of training the classifier (about 30 minutes on my computer), the reduced number of training steps outweighed the overhead added by the Bayesian optimizer. Specifically, the optimization objective was to maximize the testing accuracy of phoneme classification in Experiment 1. After ten initial exploration steps, 40 optimization steps were taken. At each optimization step, a set of hyperparameters is sampled from a promising region of the hyperparameter space, aiming to maximize an estimated surrogate $\bar{f}$ for the unknown objective function $f : (\rho, k_{W^{in}}, k_b, r) \longrightarrow$ accuracy. The surrogate estimate is improved after training and testing the phoneme classifier with these hyperparameters on the training set (with a train-test split). For a detailed review of Bayesian optimization, see Frazier (2018). The hyperparameter space was restricted to:

- Bias scaling parameter $b \in (0, 2)$
- Input weight scaling parameter $k_{W^{in}} \in (0.01, 0.99)$
- Spectral radius $r \in (0.01, 4)$
- Internal weight density $\rho \in (0.01, 1)$

Figure A.2 shows the hyperparameter and corresponding accuracies across optimization iterations. The optimizer seems not to have converged within

A.3.4 Proof of Proposition 1

Proposition 3 of Jaeger (2014b) provides the singular values of $\varphi(C, \alpha_{new}/\alpha)$ in function of C's singular values. I substituted them in the second line:

$$\begin{aligned}
tr(C_{new}) &= tr(\varphi(C, \alpha_{new}/\alpha)) \\
&= \sum_{i=1}^{N} \begin{cases} \frac{s_i}{s_i + \alpha_{new}^{-2}\alpha^2(1-s_i)} & \text{for } 0 < s_i < 1 \\ s_i & \text{otherwise} \end{cases} \\
dtr(C_{new})/d\alpha_{new} &= \sum_{i=1}^{N} \begin{cases} d\frac{s_i}{s_i + \alpha_{new}^{-2}\alpha^2(1-s_i)}/d\alpha_{new} & \text{for } 0 < s_i < 1 \\ ds_i/d\alpha_{new} & \text{otherwise} \end{cases} \\
&= \sum_{i=1}^{N} \begin{cases} \frac{0 - s_i(-2\alpha^2(1-s_i)\alpha_{new}^{-3})}{(s_i + \alpha_{new}^{-2}\alpha^2(1-s_i))^2} & \text{for } 0 < s_i < 1 \\ 0 & \text{otherwise} \end{cases} \\
&= \sum_{i=1}^{N} \begin{cases} \frac{2\alpha^2(1-s_i)s_i}{\alpha_{new}^3(s_i + \alpha_{new}^{-2}\alpha^2(1-s_i))^2} & \text{for } 0 < s_i < 1 \\ 0 & \text{otherwise} \end{cases} \\
&> 0,
\end{aligned}$$

since $\alpha > 0$, $\alpha_{new} > 0$, and $0 < s_i < 1$ for at least some i.

the allocated 60 iterations. Given the intractability of this method, parameters were eventually picked by hand.

A.5 Extension of Experiment 1: Inclusion of Input States

Experiment 1 was repeated, slightly adapting its methods to better account for non-stationary time series sources. Jaeger (2014b) demonstrated that conceptors might be used to classify signals produced by stationary and non-stationary processes. Stationary processes produce the same kind of signal (with the same probability distribution) over time; for example, white noise or sin waves result from stationary processes. Meanwhile, non-stationary processes change their properties over time, leading to signals like speech whose probability distributions change over time. Previously, the order within a sequence of states was lost when deriving a conceptor. However, this temporal order is relevant for non-stationary sources like speech production and may be valuable during classification. Jaeger (2014b) approached this limitation by unrolling the ESN response $x(n)_{n=1,...,L}$ into a vector z reserving a dimension for each step in time. Moreover, the input signal s is appended to z for additional information. $z = [x(0); s(0); x(1); s(1); ...; x(L); s(L)]$. For z, the same classification procedure applies. New hyperparameters were picked by hand; the ESN size was reduced to $N' = 40$ neurons due to the larger computational complexity of this method, but the same density of $r' = 10\%$ was used. Scaling factors were changed to $k'_{W^{in}} = 1.5$ and $k'_b = 0.2$. A spectral radius of $\rho' = 1.5$ was used. A training accuracy of 63.64% and a test accuracy of 49.13 were reached.

A.6 Additions to Experiment 2

A.6.1 K-means++ Initialization

Algorithm A.2 contains the pseudocode of K-means++ algorithm for improved centroid initialization.

A.6.2 Unaveraged MICDs

Figure A.3 plots the MICDs of all runs and conditions. They all continuously decreased, indicating the consistent convergence of GCHC toward cohesive clusters.

A.7 Extension of Experiment 2: Segmentation

The following is an informal generalization of the method of ESN-Evidence from Experiment 2 to

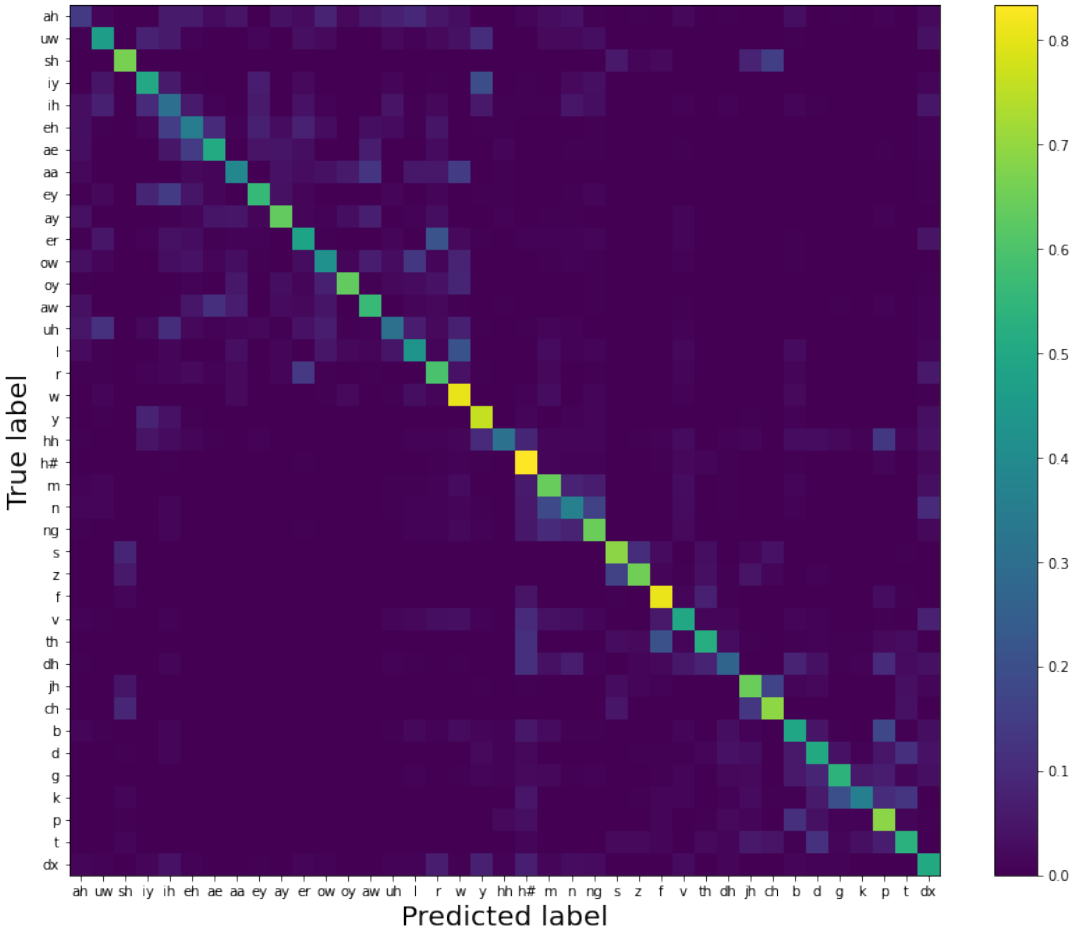

Figure A.1: Multi-class confusion matrix of the classification results. The colors represent the relative frequencies of the predictions (x-axis) made for each class (y-axis).

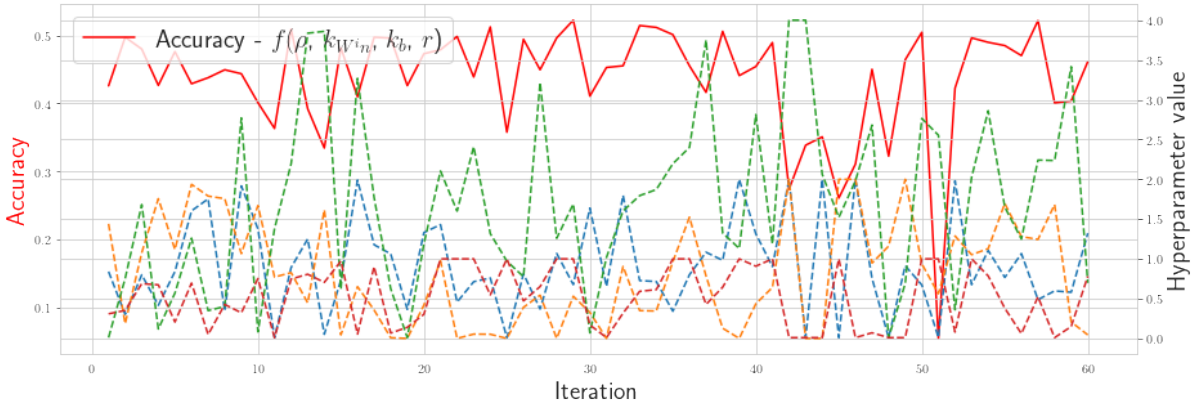

Figure A.2: The accuracy (red) for hyperparameters configurations (dotted lines) across iterations of Bayesian optimization.

Algorithm A.2 K-means++ initialization of centroids.

Require:
Number of clusters K
Set of points $D = \{p_1, p_2, ..., p_n\}$
Dissimilarity function $d(p_i, p_j)$

Choose first centroid μ_1 uniformly at random from D

for $k = 2$ to K **do**
For each point p_i, compute squared dissimilarity from nearest centroid$^{\|}$:
$d_{min}(p_i) = \min_{0<j<k} d(p_i, \mu_j)^2$

Choose $p_i \in D$ as the next centroid μ_k with probability $\frac{d_{min}(p_i)}{\sum_{j=1}^{n} d_{min}(p_j)}$
end for

return Set of centroids $\mu = \{\mu_1, \mu_2, ..., \mu_K\}$

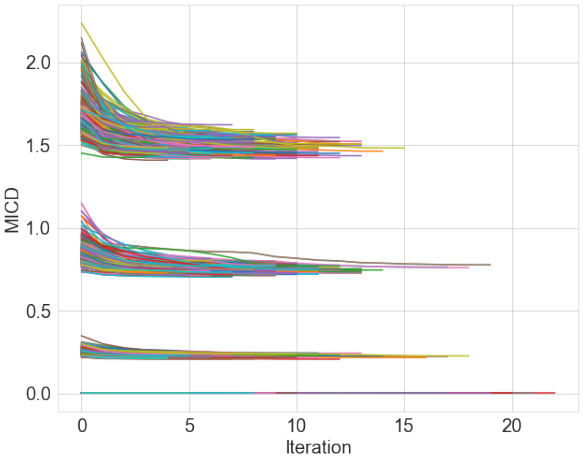

Figure A.3: Unaveraged MICDs of all runs across iterations.

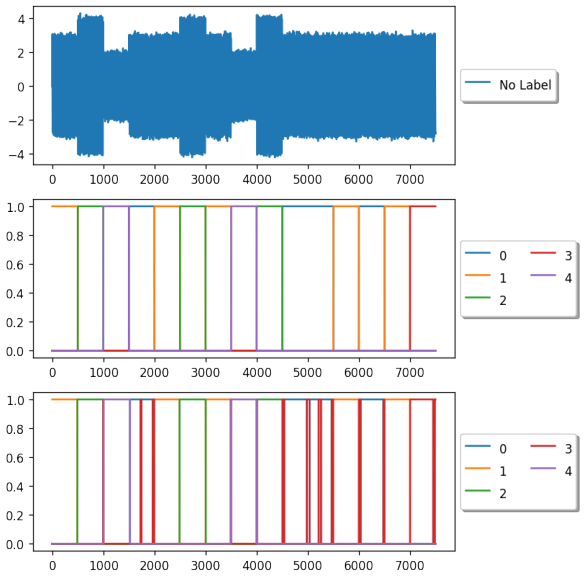

Figure A.4: Generated time series (first subfigure), its ground truth classes (second subfigure), and the clusters assigned by the algorithm (third subfigure). The x-axis represents the time steps and the y-axis the signal values.

unsegmented time series. The input time series was a mixture of Gaussian noise and a succession of sin waves with random amplitudes and frequencies (see first subfigure of Figure A.4). An ESN processed them. Then, GCHC was used similarly to ESN-Evidence on the individual ESN states. However, now, groups of states were clusters. Their conceptors were the centroids. Assignments were done using the Combined Evidence, averaged over states. To incentivize continuous segments, a Gaussian filter in time was applied to the Combined Evidences during the *Assignment Step*, so that states that are closer in time would be more likely to be assigned the same cluster.

The significant overlaps between found clusters (third subfigure of Figure A.4) and ground truth classes (second subfigure of Figure A.4) suggest the potential of this method for concept extraction from unsegmented time series and simply for time series segmentation.

B Appendix B

Notation	Meaning
A' or x'	Transposes of matrix A or vector x
I	$N \times N$ identity matrix, N to be inferred from context
$[x\|y]$	Matrix resulting from the column-wise concatenation of vectors x and y
$[x; y]$	Matrix resulting from the row-wise concatenation of vectors x and y
$A[:, y]$	Vector corresponding to the y'th column of matrix A
$A[x, :]$	Vector corresponding to the x'th row of matrix A
$A[x, y]$	Element corresponding in the x'th row and y'th column of matrix A
$\|S\|$	Cardinality of set S
$\|\|A\|\|$	Frobenius norm of matrix A
$\text{diag}(A)$	Vector containing the main diagonal of matrix A
$\|x\|$	Magnitude of vector x
$A^\dagger$	Pseudo-inverse of square matrix A
$R(A)$	Range of matrix A
$tr(A)$	Trace of square matrix A
P_S	$N \times N$ Projector matrix on the linear subspace S of $\mathbb{R}^N$, N to be inferred from context
$S \cap Z$	Intersection of linear spaces S and Z

Table B.1: Nomenclature