university of groningen / faculty of science and engineering

Hard Clustering of ESN Dynamics Using Conceptors

Bachelor's Project Thesis

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

Abstract: For data analysis, explainability, and neuro-symbolic integration, it can be of interest to identify groups and hierarchies within the activity dynamics of recurrent neural networks (RNNs). Using conceptors to represent RNN dynamics, this study aims to cluster the dynamics of Echo State Networks (ESNs), a variant of RNNs. Conceptors were computed from the ESN-response to phoneme recordings of the TIMIT dataset. These conceptors were then used to perform phoneme classification, clustering with an adaptation of K-means, and hierarchical clustering with average linkage. Conceptor-based phoneme classification reached a reasonable accuracy. Moreover, I show that conceptors may be used for clustering neural ESN activities into distinct groups and hierarchies that resemble existing phonetical taxonomies. I conclude that conceptors are well-suited to classifying and clustering ESN-dynamics and the time-series that induced these dynamics.

1 Introduction

1.1 Background

1.1.1 Teaser

Representation is a central aspect of biological and artificial intelligence (AI). Intelligent systems represent contents and operate on these representations to exhibit useful behavior (cf. (Kriegeskorte and Kievit, 2013)). The representational *contents* could range from sensory stimuli like sounds or images to abstract concepts like "representations". For example, a recurrent neural network (RNN) may be said to represent input speech signals (contents) as within its neural response pattern. Representation and function are dependent (cf. Hebbian learning); the performance of humans and AI systems in cognitive tasks like recall [(Miller, 1956), (Chase and Simon, 1973)], perception [], or creativity [] significantly depends on the task representation. Hence, representations are relevant for both explanation and development of intelligent systems.

Spaces Studies on representations tend to define *content spaces* (CS) spanning all relevant representational contents and *representational spaces* (RS) spanning all relevant representational *patterns*. The CS (e.g., color, sound, physical, stimulus, input, conceptual spaces) are inter-representationally objective; Figure 1.1 illustrates, among other aspects, how the same contents may be mapped to patterns in different RS's via *encoders*. However, the *represented information*, the mutual information between a content and its representational pattern, can vary between representations in function of the encoder. A *decoder* refers to the inverse of the encoder.

Symbolic and neural representations We generally distinguish the *symbolic* and *neural* representational formalisms. Both use patterns to represent content. I refer to the patterns of symbolic representations as *symbols* and the patterns of neural representations as *responses*. First, They differ in that the form of symbols is arbitrary; No structural relation between the form of symbols and represented contents (referents) is required for the function of the symbolic system. Their arbitrary form can make symbolic representation abide to high-level, abstract, and deterministic processes. Symbolic representations may appear in digital computers, many good old fashioned AI (GOFAI) systems, formal logics, and linguistics and can be transparent and explainable. Meanwhile, the form of neural responses is not arbitrary; contents encoded via a (simulated)

neural structure. While such structures may be capable of extracting relevant information from high-dimensional, dynamic, or non-deterministic contents like sensory data, the resulting neural representations, dubbed "black-boxes" often fall short in transparency.

This study aims to develop tools for the analysis of the *dynamic* neural representations (DNRs) found within recurrent neural networks (RNNs) for three main reasons. First, DNRs are relevant for their applicability to the ubiquitous class of time-extended contents like speech and video. Second, as RNNs are increasingly applied in impactful domains, explainability is demanded; yet, the DNRs of state-of-the-art RNNs often fall short in transparency, because of their often non-linear and chaotic encoders (i.e., their dynamics) and high dimensionality (including time). Third, insights about DNRs may fuel the improvement and development of their systems. The motivations for this study will be further outlined in Section.

Study of stimuli representations in the brain To this end, I turn toward cognitive neuroscience and psychology. Studies investigated how the human brain represents contents ranging from individual visual and auditory stimuli to concepts. Given individual stimuli, two representational stages are generally considered. First, the stimuli are encoded into *sensory neural representations* (in low-level sensory brain areas) which encode their purely sensory information. This sensory representation unfolds into a *semantic neural representation* (in high-level brain areas) Borghesani and Piazza (2017). Thus, a two-staged mapping occurs from points in CS to points in a sensory RS to points in a semantic RS. The *representational geometry* captures the relationship of contents within a RS.

Representational geometries can be reconstructed. Behavioral or neuroimaging data can quantify the dissimilarity between points in RS's. Representational dissimilarity matrices (RDMs) capture the pairwise dissimilarities. Clustering algorithms are especially useful to reconstruct the geometry of the points in RS from RDMs. For example, it may be found that the uttered /cow/ and written "cow" are encoded as two very dissimilar sensory representations. However, their semantic representations in Wernicke's area may be similar since they are symbols that refer to the same concept.

Abstracting away from the representations of individual stimuli, concepts representational geometries are analyzed to decode how Concepts The sensory patterns ow abstract concepts are represented, the geometries among stimuli are considered

Symbols are arbitrary associations of groups of stimuli. Multiple stimuli are clustered.

- Borghesani and Piazza (2017): Can introduce representations in the brain, the spaces used on the example of symbols; leads over nicely to speech representation - Huth, Nishimoto, Vu, and Gallant (2012): How objects and action categories are mapped in the brain, Semantic space - Balkenius and Gärdenfors (2016): Didn't read it but it seems to give an overarching theory of representational geometries and ties in with NNs and clustering - Op de Beeck, Wagemans, and Vogels (2001): Different "psychological spaces", study of representational geometry of stimuli

Representational geometries After mapping stimuli (from CS) into neural RS, a common next step is to search for *representational geometries*, structures among representations. - Define representational geometries - Kriegeskorte and Kievit (2013): A great overview on representational geometries - follow it tightly RSA is often performed to validate these structures. - Kriegeskorte, Mur, and Bandettini (2008) RSA as a way to bridge between spaces and make representations comparable. Possibly useful if I also perform RSA. - Appelhoff, Hertwig, and Spitzer (2022): Simple Stimulus representational geometry analysis using RSA in brains. Simple study to name as an example in the brain. Perhaps also mention in "the nature of the representation" - Representational similarity analysis (RSA) measures the correlates between models of stimuli, representations, and meanings. The physical proximity of representation (e.g., V1) to content (e.g., a color stimulus) makes it possible to validate any representation models. We cannot expect the relational structures to be isomorphic across the input and representational domains.

How contents are encoded is captured in the *coding scheme* of the representation. These (from localist to distributed and dynamic/temporal to static coding schemes). The coding schemes determine the dimensions of the resulting neural RS.

link may be as attractors, geometries, patterns....

Neuro-symbolic approaches The field of AI has long sought to integrate symbolic and neural representations in *neuro-symbolic* frameworks. Notably,

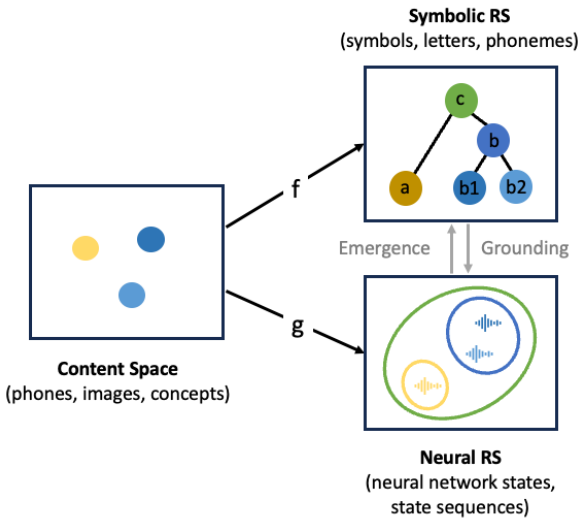

Figure 1.1: Three relevant spaces in the neuro-symbolic conceptor formalism on the example of the brain. Contents are points in content space. They are encoded into a symbolic RS (top) and a neural RS (the corresponding neural responses in Wernicke's area). From the neural representations symbols emerge and populate the symbolic RS on the right. These symbols are said to ground in their corresponding neural "substrates". Several theories on this exact neuro-symbolic correspondence exist; for example, patterns, geometrical and topological features, or point attractors among neural dynamics have been interpreted as symbols. This model can be applied to various processing and abstraction levels where representation occurs in neural systems [cf. Herbert].

the conceptor formalism Jaeger (2014a,b) provides a "symbolic lens" on neural network states. A *conceptor* is a (positive semi-definite) matrix capturing the hyper-sphere of one or more neural network states occupy in state space. Thereby, a conceptor "fingerprints" neural states by their geometry. Meanwhile, conceptors (and the associated neural representation) can be treated like symbols with means for compositionality, abstraction, and logics (see Section 1.7.2 and Jaeger (2014b)) among others. Since its publication in 2014b, the formalism has been applied both in the context of biological and artificial neural networks.

However, conceptors are just one (very versatile) means for capturing the geometry among neural representations.

Motivation By similar motivations, this study uses the neuro-symbolic conceptor formalism to analyzing neural representations.

1.1.2 Explainability

First, understanding and explainability **Challenges in explainability** challenges are equally a chance... **Development** From similar motivations as hofstadter... this ought to be an *exploratory step* toward the development of cognitive functions like creativity... - From this and more work, a motivation of using neuro-symbolic approaches to higher cognitive functions like creativity. My personal motivation for using the conceptor formalism in this project was that creativity.

Neur repr of symbols - Mao, Gan, Kohli, Tenenbaum, and Wu (2019): Tenenbaum, the neuro-symbolic concept learner; a good "bad" example of artificial bridging neuro and symbols - Ha and Eck (2017): Example of how study of representations in latent space; when symbols are known. But limitations; what if no prior information?

The nature of neural dynamic representation; spiking patterns, and symbols - Van Gelder (1998): Representations through dynamics - Jaeger (1996): Representations as dynamics symbols - Jaeger (1999): Possibilities of representing constant entities in dynamic representations - Perlovsky (2007): Symbols in language vs cognition Result: Still unknown how concepts are represented in the brain. Symbols.

1.2 Brain

1.3 Clustering

Clusters as an instance of representational geometry - clustering methods for analyzing representational geometries To analyze the representations in the brain, clusters have been searched for. Hard clustering aims to identify a set of K hard (distinct) clusters among its input data.

Hard clustering has been applied using various dissimilarity functions An iterative relocation scheme to find a set of clusters and centroids that minimize a dissimilarity function. and coined centroid-based parametric hard clustering. - Tucciarelli, Wurm,

Baccolo, and Lingnau (2019): very related study on the representational geometry of actions - Banerjee, Merugu, Dhillon, Ghosh, and Lafferty (2005): a generalized approach that works with any Bregman divergence - Hierarchical average linkage (HAC)

Representations of symbols in the brain Beyond individual stimuli, *classical symbols* are a much studied class of representational content. Symbols are entities that refer other entities, or referents. The classical definition of symbols by semiotics (Peirce and Buchler, 1902) and later, computer science and AI Newell and Simon (1975) frames them as arbitrary in form; the physical form of the utterance [dog] bears no relation to the symbol's semantics, the referenced (set of) dog(s). Given their ubiquity in spoken, written, and visual language, their representation has been subject of study. Borghesani and Piazza (2017) provides a summary: of that study.

Cognitive neuroscience attempts to explain how these cognitive symbols arise in sub-symbolic brains. From a sub-symbolic perspective, the brain is a network of neurons. When this dynamical system is stimulated, for example, by sensory inputs, patterns of neural firing result. These firing patterns may be considered symbols if they are associated with a specific entity. Symbol acquisition, then, is the process of reinforcing these firing patterns (through Hebbian learning) face to the repeated exposure to instances of the referenced entity. For example, when a child is first exposed to speech, every sound signal will trigger an arbitrary firing pattern. As the child repeatedly hears the phones [t^h], [t], and [ɾ] in similar linguistic contexts, a single pattern of neural firing becomes associated with them. This pattern corresponds to the cognitive symbol, the phoneme, /t/. After symbol acquisition, the pattern is no longer tied to any specific instance but represents a whole subset among the possible speech sounds and sensory experiences in general). Notably, children learn to distinguish phonemes without external supervision; No explicit instructions are needed let alone possible since only later in development do phonemes become the building blocks of words and enable the understanding and production of language Maye and Gerken (2000). So, symbols correspond to sub-symbolic firing patterns acquired through unsupervised learning.

1.4 TIMIT

Representation of Speech/phonemic data, TIMIT dataset The domain of human speech is commonly used to study representations. First, human speech is highly relevant as a main medium for communication. Second, it allows the study of how symbolic structures are represented in the neural dynamics of the brain. On the one hand, speech (spoken language) and its neural representation are time-extended (dynamic) signals. On the other hand, speech is symbolically structured. Grammars de-/presrcribe how syllables, words, sentences, etc. are to be formed.

Phonemes are the mental representations of phones, the smallest still distinguishable units of speech sound. For example, the phoneme /t/ is a symbol of the English-proficient mind and the mental representation of the phones [t^h], [t], and [ɾ]*, each of which encompasses a subset of speech sounds. The phones that a phoneme represents are its *allophones*. Communication through speech is possible, among many other factors, for the speech representations of the communicators coincide; The representations of English speakers generally feature the same 44 phonemes representing the same set of phones with some variability across dialects. Note that phonemes are language specific; Every language uses a different set of symbols to represent its particular speech sounds. Thus, cognition seems to rely on symbolic representations.

- Shepard (1980) best simple representational geometry study of stimuli in the brain, uses TIMIT, may also go in other paragraphs

The developed methods will be demonstrated on phonemic speech recordings. As mentioned, Phonetics – the branch of linguistics concerned with speech – groups speech sounds into phones (sound units) and phones into phonemes (symbols); yet, further above-symbol organizations exist. For example, Figure 1.2 depicts a possible taxonomy of the phonemes present in the TIMIT corpus which contains clusters like vowels and nasals†.

The availability of these phonetic systems makes

*Phonemes are broadly transcribed using a slanted bracket. Phones are narrowly transcribed by a square bracket. Phonic transcriptions will be written using the International Phonetic Alphabet (IPA).

†The TIMIT labels, also shown in Figure 1.2 and listed in Table A.1, are from IPA, although not written in Greek letters. The translation keys are given in the documentation.

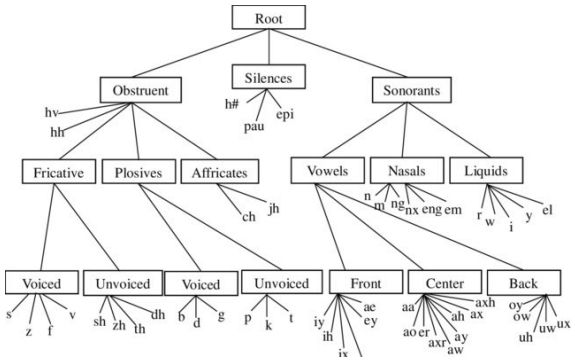

Figure 1.2: A taxonomy of all phones present in TIMIT Pfeifer and Balik (2011)

phoneme recordings a good domain for evaluating clustering algorithms. On the one hand, these phonemic classes and taxonomies can be used as ground truth clusterings to determine the performance of the developed clustering algorithms. [maybe connect the following sentence with analogy to child] On the other hand, the results from this study may provide an empirical verification or falsification of phonetic theory; whereas the groups and taxonomies of phones were largely developed based on their articulatory features, the following clustering Experiments will investigate whether these groups can be found on a purely acoustic basis. [Phonemes clustering literature on TIMIT (reference to methods to avoid redundancy)] [transition]

1.5 Study of neural representations in AI

XAI: In AI, however, few methods for identifying how ANNs represent stimuli have been developed. A large group is available in the area of Explainable AI (XAI)... - Study of stimuli representations in AI, XAI (without conceptors) **Representational geometries in NNs** - Lu, Chen, Pillow, Ramadge, Norman, and Hasson (2018): NNs of different architectures share representational geometries - Naitzat, Zhitnikov, and Lim (2020): Analyzing NN representational geometries via topologies

On TIMIT:

But how to cluster states, analyze representations, potentially form the pre-used classes? - Xu and Tian (2015), *slides "clustering intro.pdf*: clustering overview - Bricman, Jaeger, and van Rijn Tange (2022), Mossakowski, Diaconescu, and Glauer (2019): clustering experiments - : clusters within network states - Estevan, Wan, and Scharenborg (2007), Atencia, Gallicchio, Joya, and Micheli (2020): application driven experiments where network states are clustered to actually cluster inputs. (Somewhere I should also mention Yildiz, Jaeger, and Kiebel (2012) for the ESP) - Jaeger (2014a): Creativity application; what if we can generate the movements of stick figures conceptor experiment. IN this section we can already mention the experiment and refer to the gap. - Omniglot challenge - Fabi, Otte, and Butz (2021) (Maybe also Lake, Salakhutdinov, and Tenenbaum (2019)): Learning symbols automatically in RNNs

1.6 Conceptors

In few instances conceptors have used unsupervised fashion to analyze representations and their geometries Conceptors have already been used for describing representations. An interpretation of this is to see it as a repr geometry analysis that assumes an elliptoid base structure within a single sample - Jaeger (2017): Jaeger discribing conceptors as an "interpretable observer for the learnt representation". Also shows things that can be done once structure is known. - Conceptor space as a semantic space: Conceptors filter out the important features of the neural activities. It is like in experiment where the semantic space is described as a space consisting only of the important semantic dimensions. "The natural account of the "meaning" of a matrix conceptor C is the shape of the neural state cloud it is derived from.". So, conceptors can capture the geometry of a representation.

Conceptors for symbols Unsupervised learning tasks typically involve analyzing the structure of data without given classes or labels. For example, one may seek to identify important clusters (groups), hierarchies of clusters, axes, or relationships among a set of data points. In the following cases, conceptors were used to identify initially unknown structures or relationships among neural network state clouds for explainability and data analysis. Bricman et al. (2022) used conceptors for neural network explainability. The developed technique, Nested State Cloud, extracts a knowledge graph from the latent space encodings of a set of 100 symbols (e.g., "apple"). Each symbol encompassed

multiple instances, contexts in which they appear (e.g., "The apple fell."), all of which had different embeddings. Conceptors served as a compact representation of each symbol's embeddings and allowed them to be semantically related in the graph. Similarly, Mossakowski et al. (2019) used conceptors to represent and then organize classes of network states. This time, the symbols (i.e., classes) were Japanese speakers, and the network states were an ESN's responses to their speech samples. Using the fuzzy generalization of the Löwner ordering as a dissimilarity function on the speaker-specific conceptors, the symbols were organized in a tree by hierarchical agglomerative clustering. The authors mention the potential application in classification, but whether the resulting hierarchy truly coincided with any features of speakers like dialectal groups remained unknown. [One more example] These studies analyze structures between classes of reservoir states, but this means that they, like the supervised classification algorithm, start from pre-established symbols. I shall refer to these as *above-symbol structures*, for they are at a level of abstraction above that of symbols.

The identification of both below- and above-symbol symbol structures resembles classical clustering tasks. Clustering refers to the unsupervised task of organizing a data set into groups, or *clusters*, such that instances within the same cluster are more similar to each other than to those of other clusters. This similarity may be determined by a dissimilarity measure like the Euclidean distance. Clustering could play a significant role in finding below-symbol structures relevant to unsupervised symbol identification. By feeding the states of a neural network into a clustering algorithm, they can be grouped into clusters, each of which can then be treated as a symbol with the cluster members being the representations of that symbol. The analogy holds, for similarity is expected among states of the same symbol and dissimilarity between states of different symbols. Conceptors may serve to represent and evaluate the clusters. Conceptors can compactly represent and compare sets of network states and thereby act as centroids, i.e., exemplar-based representations of the symbols Bricman et al. (2022). Hence, the process of unsupervised symbol identification will be seen as a form of clustering, where each cluster corresponds to a symbol and is represented by a conceptor. Similarly, clustering algorithms seem equally applicable to the analysis of above-symbol structures by pre-grouping the network states by symbol according to a priori memberships.

What is unknown, and summary of motivations: Why is it relevant to know classes? What can we do by unsupervised identification? 1. **classification of representations (using conceptors)** A common use-case of conceptors with ESNs has been supervised time series classification (Jaeger, 2014b). An instance of this method will be outlined in Experiment 2.4, but the guiding idea is the following. Under the echo state property (ESP), an ESN responds uniquely to its inputs; i.e., there is a functional relationship between inputs and responses [**Why is it not bijective?**] [**Can I leave this out?**]. More concretely, with this functional relationship between inputs and responses, the responses can be seen as high-dimensional embedding of the input. Hence, any label assigned to a reservoir response is taken to equally apply to the input sequence that induced that response. To train the classifier, one captures the reservoir responses of each class using a conceptor. To classify, the new response is assigned the class whose conceptor indicates the highest similarity [**or correlation?**] to the previously seen responses of the class. Jaeger, Lukoševičius, Popovici, and Siewert (2007) demonstrated this method in speaker recognition on the *Japanese Vowels* dataset to a perfect accuracy, a result that promisingly exceeded several previous attempts [cite...]. [Describe the advantages of this classification method.] Building on this experiment, Vlegels (2022) showed that competitive accuracies in the classification of non-stationary time-series may be achieved using several variations of this methods. Moreover, Chatterji (2022) adapted the method to time-series *recognition*. The difference between classification and recognition is that the former estimate the class of pre-segmented signals (one label per segment), whereas the latter estimates the class present within unsegmented signals (multiple labels per signal over time) (Lopes and Perdigao, 2011). Concretely, Chatterji (2022) demonstrated their method on the recognition of phonemes on the TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT), the dataset also used in the current study. Concluding, in the supervised tasks of time series classification and recognition, the classes of interest are given, but they may not always be. -

Bartlett, Garcia, Thill, and Belpaeme (2019) (Simple recognition task in humans) - Sha and Saul (2006), Schilpp (2021): TIMIT classification - Wang, Severa, and Rothganger (2019): example of TIMIT representational confusion matrix during classification; a useful cross-link perhaps - Chatterji (2022): TIMIT recognition using conceptors and ESNs - Jaeger (2014b), Vlegels (2022): Classification using conceptors - Oh, Park, Kim, and Jang (2021): hierarchical phoneme classification on TIMIT - Lee and Hon (1989): Phoneme recognition (benchmark score I think) 2. **Pragmatics: BPTT, Unavailability of classes may indicate need for further training** Clustering RNN dynamics could help explain what groups of activity are present in the network. For example, both algorithms could be helpful tools for interpreting BPTT-trained RNNs. For example, Experiment 2 could show how classes become more distinct or numerous within the RNN (form activity clusters) while training. 3. **Creativity and co** - My initial motivation! - Jaeger (2014a): Creativity application; what if we can generate the movements of stickmen conceptor experiment - Lake et al. (2019) Fabi et al. (2021): Omniglot application - Hofstadter, Mitchell, and French (1987): Letter spirit (Chapter 10) - Hofstadter (2001): Analogies are core to cognition, argue that this could be a step to analyzing the components (symbols) among which analogies can be made (to which the Doug's tools that are full of assumptions can be applied) - Transfer learning, generalizability, recombination: Move to more general and creative AI systems via the tools of conceptors 4. **Explainability** - Global Explainability - Interpretability: Understand the presence of classes within the blackbox (not only based on outputs, accuracy). In most applications, we require networks to adopt certain external symbols (classes cat, dog, sounds [a], [b], letters "a", "b). They are man-made structures of the network's inputs and outputs. The network learns a function that relate these symbols to each other. To some degree its internal representation will feature these symbols. But what symbols does the network come up with to do so? *How* does it fulfill its design purpose, i.e. minimize its cost function? Its internal symbols might be very different to the symbols of in and outputs! .... Explainability. - Robustness and Adversarial Attacks: Understanding the geometry can also be crucial for identifying vulnerabilities to adversarial attacks. If representations of different classes are too close geometrically, small perturbations can lead to misclassifications. 5. Neuroscience Implications: For researchers in neuroscience, understanding the representational geometry in artificial networks can provide hypotheses or models for studying the brain. It can help in understanding how neurons and neural circuits represent and categorize sensory inputs.

However, but neural representations in AI lack still similar interpretability methods... no clustering/hierarchy method

While This research advantage is reflected in the historical delay of the usage of the term "black-box" in the context of ANNs (1980-today) vs brains (behaviorism). The topic of this study will be to develop methods for studying clusters in the representations of ANNs. Given the advantages of symbolic methods, we adopt a neuro-symbolic perspective/formalism. This may also guide toward an integration of cognitive functions and the neural substrates.

The study of representations is crucial to explaining and developing neural networks. First, The geometries, dependencies, patterns in representation of contents determines and constrains the possible behavior. E.g. something unrepresented cannot be used. If a representation echos an input, patterns among the representation may even reveal facts about the input. Third, ...

1.6.1 RQ

How to identify distinct and meaningful clusters within an RNN's representation of stimuli?

1.6.2 Proposed Solution

ESNs The proposed methods will be demonstrated on ESNs. ESNs are a type of RNN of particularly large size (many internal neurons), low density (low connections-to-neurons ratio), and random and untrained internal- and input weights (Yildiz et al., 2012). Given these features, ESNs perform a high-dimensional non-linear expansion of their input signals, that, under the right circumstances (echo state property), uniquely *echo* of these inputs (cf. Lukoševičius (2012)). Figure 1.3 depicts a typical ESN as it is driven by an input sequence u, possibly a speech sample, and elicits a high-dimensional response x, the sequence of internal states of the ESN's reservoir. An output layer is sometimes added to ESNs for (re)productive purposes but omitted

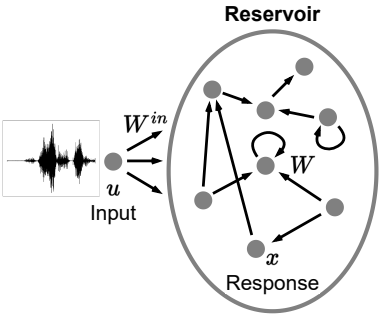

Figure 1.3: Sketch of an ESN driven with an input signal. State vector x is a high-dimensional response to the input signal u.

here since only the internal states of the network will be of interest for the study of neural representations in ESNs.

ESNs were used for three main reasons. - No training required - Very complex - The ground truth clusters and classes among the input should serve to extrinsically validate clusters and classifications. Therefore, the representational contents need to coincide with the inputs. ESN's have the property of providing a high-dimensional expansion of their inputs; by echoing the input, a large information overlap is to be expected. conceptors that offer a symbolic representation of the sub-symbolic firing patterns.

The developed methods will be demonstrated on phonemic speech recordings of TIMIT.

Experiment 1 is a phoneme classification task used to tune the ESN to the data; it ensures that the ESN can represent the data and use this representation for classification.

For clustering ESN responses to phoneme recordings, the ESN must be suited for the task. This makes the following assumption; Given an ESN, the more accurately a conceptor-based classifier can distinguish activity groups, the better these classes can be retrieved by an unsupervised clustering algorithm. The classification and clustering performances seem correlated because, for many hyperparameters like the aperture and spectral radius, a sweet spot seems to exist, where the ESN's signal-to-noise ratio is maximized and where the network is neither over- nor under-excited, that can be determined independent of the application in clustering or classification. Thus, I assume that tailoring the ESN to the TIMIT data and classification task will improve clustering on the same data. The hyperparameters were optimized in Experiment 1 on time series classification. Moreover, performing the classification task in Experiment 1 has the positive side-effect of demonstrating the application of conceptors to the phoneme classification on the TIMIT dataset. The following classification method builds on that of Jaeger (2014b). To my knowledge, 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) mentioned in Lopes and Perdigao (2011)].

Old - Study the representational geometry of phonemes inside ESNs We set up a network such that we can expect the representational geometry / its states to form clusters in alignment with classes of the stimuli. Focus on *segments* Then, Kmeans and HAC clustering - Evaluation is made by implicitly drawing the parallel to humans; symbolic classes **Phonemes, TIMIT** - Emphasize how your research builds upon, differs from, or addresses gaps in these previous studies. - Quality evaluation - the resulting conceptor matrix acts as a recognizer for that symbols. 3 methods. the one with conceptors wins. - Old: An unsupervised method for identifying relevant symbols within a ESN activity. This requires definitions. A symbol is an activity pattern captured using a conceptor. The domain of ESN activity is a *set* of ESN states. Their order is not considered as it falls away using conceptors. *Relevance* is established by comparing to the intrinsic and extrinsic (human-made) features of the symbols of domain of application. Distinctiveness is assumed as a symbol-intrinsic property. There is no natural law to my knowledge that maintains this principles since symbols are human/brain-made structures. However, it seems to be assumable that instances of the symbol are similar to each other (cohesion) and distinct from the instances of other symbols (separation). First, when learning occurs, weight resources in a network are limited. It becomes necessary to differentiate between symbols. If instances of one symbol were too similar to instances of another, it would be computationally costly and often impractical for the cognitive system to distinguish between them. Second, share perceptual physical features among the associated referents (especiali stimuli). Third, by analogy to the human categories

/ semiotics. [... chat gpt] It also presupposes non-overlapping symbols, which is made for we consider only a single processing level. Definition of symbols as patterns or regularities/constants.

We consider a symbol a pattern consistently arises when its referents are active (active means currently presented or another symbol occurring) but rarely arises in presence to other stimuli. I take inspiration from two classical hard clustering algorithms, K-means and hierarchical agglomerative clustering (HAC). Hard clustering assigns each data point to one, and only one, cluster, enforcing a clear separation between clusters. In contrast, soft clustering, like that of Bricman et al. (2022), assigns data points a level of membership for each cluster, allowing for overlap between clusters. I assumed, in the spirit of traditional symbolic representation theory, hard clustering to be the better fit for the endeavor of automated symbol identification using symbols to represent discrete, separated entities instead of multiple entities to varying degrees. ESN dynamics will be hard-clustered by adapting K-means and HAC, two well-known hard clustering algorithms with guaranteed convergence. K-means partitions data into a number of non-overlapping clusters and will be used to group below-symbol instances into symbolic groups. Importantly, traditional K-means can cluster N data points within $O(N)$ enabling its use on larger data sets. Meanwhile, HAC returns a hierarchy of clusters and will be used to analyze above-symbol structures among pre-existing symbols. HAC requires $O(N^2)$ with optimal implementation. Both algorithms have previously been used for time series clustering by [...]. However, while ESN state sequences are time series, it is unclear whether the previous methods are suited to the clustering of high-dimensional non-linear dynamics. Moreover, previous time series clustering algorithms have scaled badly with long, potentially infinite, time series, were often offline (fail to adapt as new data points), or struggled to handle variable length time series, a limitation mediocrely solved using Dynamic Time Warping. These shortcomings were equally menacing time series *classification* algorithms but were resolved by use of conceptors; the similarity in methods gives hope to similarly resolve them in conceptor-based time series *clustering*.

- Old solution: Reiterating, a symbol was defined as a pattern of network (ESN) activity that represents some entity. Such activity patterns, or symbols, can be represented, identified, compared, or induced using conceptors. In the previous Experiment, each positive conceptor represented the ESN activity patterns characteristic for a set of states. phoneme (the symbol) that, in turn, was the ESN. From these clusters, conceptors can be derived that reflect a new set of symbols.

To compute conceptor for any symbol, a set of network states (a state cloud) in which the pattern is present is needed. These states are responses to signal instances of an abstract class whose defining concept (the entity) is represented by the symbol.

When deriving a conceptor for any symbol from a set of states, it is preferable for that symbol to be the main commonality between the states. Geometrically, this translates to having lots of variance along principal components and low variance

In the following two experiments, the generalized centroid-based hard clustering and HAC algorithms were adapted to cluster the ESN responses to the phonemic speech recordings. The general theme of both experiments was to (A) embed time series (the pre-processed phoneme recordings) as reverberations of an ESN (the response), (B) use conceptors or other centroids to represent these ESN reverberations and (C) group them into a set of meaningful clusters. Figure 1.4 depicts the steps from the original signals to ESN states to conceptors to clusters. Although the conceptors were subjected to the clustering algorithms, the tight relationship between signals, ESN responses, and conceptors presumably allowed the resulting clusters to generalize from conceptors to the other two representations. This generalization was exploited during cluster evaluation when comparing the found clusters against phonetic groups of the TIMIT speech samples.

Experiments 2 and 3 - Sarmiento, Fondón, Durán-Díaz, and Cruces (2019): generalized hard partitioning clustering - Xu and Tian (2015): clustering overview. Mention why average and non min/max linkage - Teh (2008): hierarchical clustering - *"TIMIT/ Processing"*: see Zotero

1.6.3 Additional motivations

Additional motivations drove this proposal.

While primarily clustering and classifying the ESN responses to time series, the clusterings and predictions coincide with those of the time series that underlie these dynamics. This relationship will

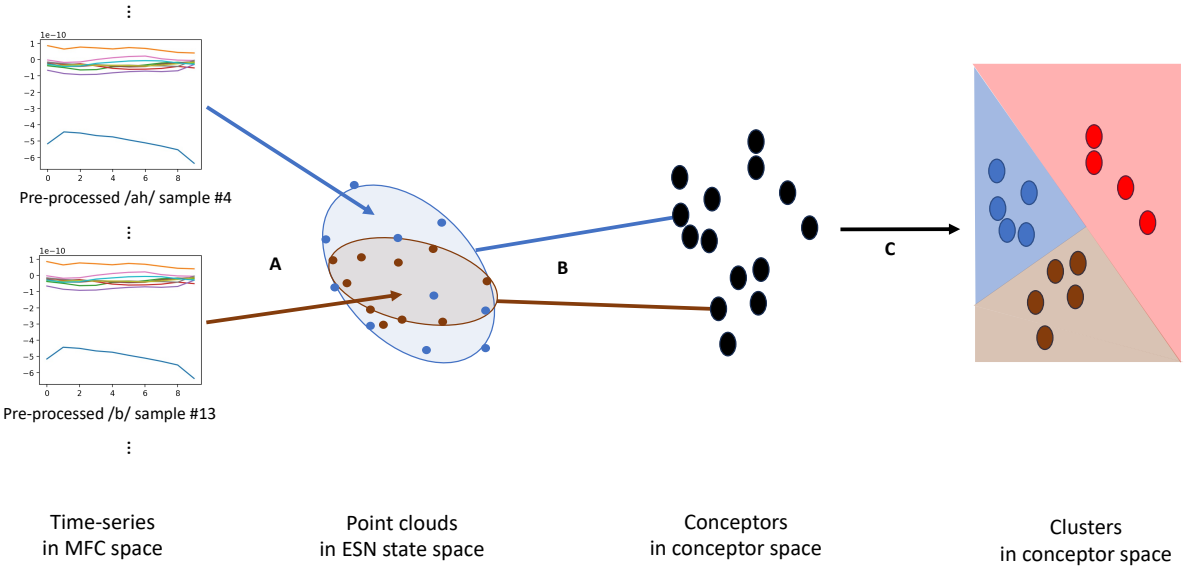

Figure 1.4: The steps from the MFC space of pre-processed speech signals to ESN state space by driving the ESN (A), followed by a transfer to conceptor space by computing conceptors (B), and finally the partitioning of the data into clusters (C). Note, while Experiment 2 aims to find partitions, Experiment 3 identifies hierarchies; thus, the right illustration would differ for Experiment 3. [Double check representation is truthful of apertures.]

prove useful in extrinsically evaluating the clusters. Moreover, the formalism of conceptors would thus be extended to time series clustering. In clustering, the conceptor-based methods will address several challenges of traditional time-series clustering.

Due to varying input lengths, clustering methods for embedding time-series data are rather limited (e.g., latent-space encodings). This paper shows that the clusters present in RNN dynamics can be tied back to clusters within the inputs used to produce these dynamics. - Time variability of data obstructs clustering and classification algorithms (Lukoševicius, Popovici, Jaeger, Siewert, and Park (2006), Tanisaro and Heidemann (2016) current approach to making ESNs time warping -¿ place it in the discussion maybe) Thus, Improving both classification and data clustering methods. Moreover, this extends the conceptor formalism.

1.6.4 Study limitiations

The following is not studied. First, we focus on cluster identification, not segmentation in time, nor application. Mitchell (2021) ("focus on syntax rather than semantics)". Not: - Representational topology - Time-continuous - Symbols like in PSS - The linguistic meaning of phonemes beyond their "sensory meaning" -¿ only perceptual features - Assumption of corresponding clusters with same K - Some generalizations are left to future research. Now we considered independent RNN runs (in response to pre-segmented speech). At each processing level, RNN activity can be seen as a flow from symbols (or symbol structures). Each symbol appears and disappears like an event. Identifying symbols in this perspective requires distinguishing when symbols and space (what pattern is activity). actually as events extended in space (its) and time (the information). Complete unsupervised symbol extraction would thus require - Segmentation - Association of segments - Autonomous applications (e.g., the task of identifying symbols in unsegmented speech)... Clustering only in space, not in time - Segmentation, although exploration of topic is appended

1.6.5 Key Insights / Findings

This demonstrates the correspondance of conceptor-based operation to

1.6.6 Structure

The next section provides formal definitions of ESNs and conceptors. The methods section begins with a description of the data, their pre-processing, and the collection of ESN representations of the data. Experiment 1 is a phoneme classification task used to tune the ESN to the data and demonstrate the efficacy of conceptor-based methods. In Experiment 2, the ESN's representation of individual stimuli (phonemic utterances) will be clustered. Experiment 3 will attempt to distill hierarchies from the ESN's representations of symbols (phonemes). The results are given after the methods of the respective experiments. Finally, all results will be discussed and an outlook application and future developments of the methods in the context of speech-processing, XAI, and neuro-symbolic AI will be provided.

1.7 Formal Definitions

The following sections on ESNs and conceptors were inspired by the detailed report Jaeger (2014b). Appendix B contains an index of mathematical notations used throughout the paper.

1.7.1 Echo State Networks

Let us formalize an example ESN like the one from Figure 1.3. Let N be the number of internal neurons. As mentioned, 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 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 time series input 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 RNNs:

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

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. Observe that the response x is a high-dimensional non-linear expansion of the input since $N >> d$ and tanh is applied.

1.7.2 Conceptors

Definition and Intuition Given any sequence of network states $x = (x(1), ..., x(L))$ that may have arisen from running the above ESN [‡], the conceptor matrix C computed from x minimizes the following loss function $\mathcal{L}$:

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

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 $R = XX'/N \approx \text{corr}(X)$.
3. Obtain the conceptor $C(R, \alpha) = R(R + \alpha^{-2}I)^{-1}$.

Conceptors can be considered "'fingerprint' of the activity of [a] network" over a period of time (Jaeger, 2014b). This potential for conceptors to uniquely identify or fingerprint a sequence of states is reflected in its loss function. When minimizing $\mathcal{L}$, the term $||x(n)-Cx(n)||^2$ nudges C toward realizing an identity mapping for the states $x(n)$. However, the regularization term $a^{-2}||C||^2$ attaches a cost to the magnitude of its entries, constraining the set of matrices available to realize this identity mapping. Thus, the conceptor is drawn away from the identity matrix toward the zero matrix, especially on those axes that can account for little of the variance of x, i.e., where the minimization of $||x(n) - Cx(n)||^2$ is less beneficial. The amount of regularization applied to the conceptor depends on its aperture α. The higher the aperture, the closer the resulting conceptor maps all the members of x to themselves. The lower the aperture, the more selective the conceptor becomes, realizing a "good" identity mapping only along the important axes that explain most of the sequence's variance.

A geometric perspective might extend this intuition. The reservoir states form a point cloud in state space $(-1, 1)^N$. Their conceptor can be represented as a hyperellipsoid that is of a similar shape as the point cloud; The ellipsoid's axes, given by the singular-value-scaled singular vectors of the conceptor, are similar to the principle components of the state clouds. However, the shape of the hyperellipsoid slightly deviates from that of the point cloud as the regularization *squashes* it along its shorter axes. With a low aperture, the hyperellipsoid would be very squashed, extended only along a few main axes. With a large aperture, the conceptor would be a little squashed and approach the unit hypersphere in $\mathbb{R}^N$ from inside.

This squashing in space may be seen as a spatial compression as it systematically reduces the variance of the point cloud along its less important axes. Furthermore, conceptors are a temporal compression of ESN state sequences. Mapping sequences of vectors in state-space to conceptor-space turns variable-length objects into constant-sized objects. When the state sequence is a time series, this mapping removes the temporal dimension, whence it may be seen as a temporal compression.

Similarity function Conceptors may be used to compute the similarities of state sequences. If C_a and C_b be the conceptors derived from the two sequences of states to be compared, their similarity may 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.

Aperture The aperture of a conceptor can be set during its computation, or when given a precomputed conceptor C, its aperture can still be adapted by any factor of $\gamma > 0$ using the aperture-

[‡] I consider *sequences* of network states for practicality, but they can be thought of as *sets* because their states need not necessarily be ordered; any ordinal information is lost during the computation of conceptors. Neither do the states need necessarily stem from the same ESN run.

adaptation function φ that returns the aperture-adapted conceptor C_{new}:

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

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

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

$$\neg C := I - C \quad (1.5)$$

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

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

$$C \wedge B := (P_{R(C)\cap R(B)}(C^{\dagger} + B^{\dagger} - I)P_{R(C)\cap R(B)})^{\dagger} \quad (1.6)$$

An algorithm for computing projector matrix $P_{R(C)\cap R(B)}$ is given on pp. 174-175 of Jaeger (2014b). $C \wedge B$ returns a conceptor that describes the intersection of the linear subspaces of C and B.

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

$$C \vee B := \neg(\neg C \wedge \neg B), \quad (1.7)$$

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

Although a simpler method for computing con- and disjunction exists, that 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 in practice, for example, due to rounding or through the negation of conceptors with unit singular values. Therefore, the above method is recommended whenever the absence of null and unit singular values cannot be ensured.

2 Methods and Results

2.1 Dataset

The TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT) was chosen as the data source, for it features diverse and phonetically annotated speech signals. TIMIT comprises 6300 sentence utterances. Each of the 630 US-based native-English speakers (from eight dialect regions and 30% of whom were female) 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 a test set used as such in Experiment 1. Experiments 2 and 3 relied solely on the training set.

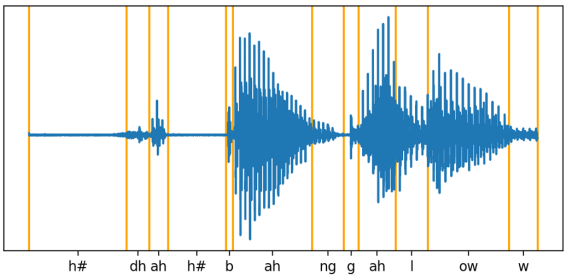

Figure 2.1: Example of the twelve first segments of one of the utterances.

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 ($n_{\text{TIMIT-train}} = 177080$ and $n_{\text{TIMIT-test}} = 64145$), each a vocalization of one phone (Figure 2.1). From each segment, the first $d = 13$ Mel Frequency Cepstral Coefficients (MFCCs) were extracted, 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 grant each MFCC a similarly strong effect on the ESN dynamics under the identically distributed input weights, each channel was normalized to a range of $[-0.5, 0.5]$ across all samples. Second, to account for differences in utterance speeds, the series were normalized in time by fitting each channel with a cubic spline and sampling it at $L = 10$ temporally equidistant points.

Lastly, the phonetic labels (transcriptions) p_i ($i = 1, ..., c$) 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. This was done to reach reasonable classification and clustering performances, feasible computations, and results comparable to previous studies that also used the mapping [add citations].

Thus, the resulting data consisted 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 giving $D_{\text{TIMIT-train}}$ and $D_{\text{TIMIT-test}}$ of respective lengths $n_{\text{TIMIT-train}}$ and $n_{\text{TIMIT-test}}$.

2.3 ESN

The following ESN setup was used for 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, which tends to lead to a more chaotic behavior **[Is "chaotic" non-binary?]**. ρ was adapted by rescaling the old (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}. [Effects on ESP]

All of the above hyperparameters were picked by hand based on their effects on the accuracy in Experiment 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). N was kept as a largest possible value still feasible under the available computational resources. Larger sizes would likely improve performance after adapting the other hyperparameters but may increase the risk for overfitting (Lukoševičius, 2012). The remaining hyperparameters, r, $k_{W^{in}}$, k_b, and ρ, were adjusted by hand with the objective to maximize the validation accuracy of phoneme classification in Experiment 1. Moreover, automated hyperparameters optimization was attempted but eventually not used due to its large computational cost and slow convergence that made waiting for its convergence intractable. Its method and results are in the Appendix.

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.

The resulting ESN was driven on each input signal s_i ($i = 1, ..., c$) producing the reservoir state sequence x_i ($i = 1, ..., c$). Concretely, each run started from the same state $x(0)$ sampled once from a standard normal distribution not to introduce meaningless between-sample differences while providing the network with an initial excitation. Indeed, using this normally distributed starting state led to a greater classification accuracy ([add correct value] ... on D_{val} using the final hyperparameters) in Experiment 1 than when using the null vector as the starting state **[Can/Should I just state this without reporting the experiment?]**. 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 training sample.

2.4 Experiment 1: Phoneme Classification

[**For this experiment 1's lengthiness, I suspect some parts might be better placed in the appendix (?)**] **Objective** Experiment 1 aimed to classify the pre-processed utterances using the corresponding ESN responses. The experiment's purpose is to (a) optimize the ESN hyperparameters for the subsequent clustering experiments and (b) evaluate and improve methods for conceptor-based time series classification. Concretely, the developed classifier takes as input the ESN's response to an utterance and outputs the assigned phoneme label. The assigned label is taken to equally apply to the ESN's response to the utterance and the utterance.

Data To optimize hyperparameters and take small design decisions, the pre-processed original dataset $D_{\text{TIMIT-train}}$ was initially divided into a preliminary training set, $D_{\text{pre-train}}$, and a validation set, D_{val}. Here, the split was 80/20, respectively, stratifying over phonemic classes. Once hyperparameters and methods were set, the classifier was retrained on the whole of $D_{\text{TIMIT-train}}$ and evaluated on $D_{\text{TIMIT-test}}$.

2.4.1 Training

Training amounted to computing one *positive conceptor* C_p^+ and one *negative conceptor* C_p^- per class $p \in P$. Each class' positive conceptor captures the linear state subspace that ESN states (from the responses to pre-processed phoneme utterances) of that class tend to occupy. For each class p, C_p^+ was computed as follows. Let η_p be the number of training instances of p. The state collection matrices corresponding to signals 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 computed with an initial aperture of $\alpha = 1$ by steps 2 and 3 of the procedure for conceptor computation (Section 1.7.2). This was repeated for each class to obtain the set of preliminary positive conceptors $C_{pre}^+ = \{C_p^+ | p \in P\}$.

Aperture adaptation After computing the conceptors in C_{pre}^+ with the initial aperture of $\alpha = 1$, their apertures were optimized. First, one new aperture α_{opt} was chosen for all positive conceptors. The objective was to maximize their sensitivity to differences in the underlying ESN dynamics which would likely improve their capacity to classify the data to come. This objective is quantified as the maximization of the ∇-*criterion*. Concretely, the ∇-*criterion* is defined in function of conceptor C and a candidate aperture adaptation factor γ [§]. It then returns the gradient of the Frobenius norm of the aperture-adapted conceptor with respect to the logarithm of γ:

$$\nabla(C, \gamma) = \frac{d}{d\ \log(\gamma)} ||\varphi(C, \gamma)||^2 \qquad (2.1)$$

Indeed, this corresponds to the magnitude at which the size $||C||^2$ of a conceptor C changes (sensitivity) with respect to its log aperture (scaling of the underlying ESN states, see p. 49 of Jaeger (2014b)). The optimal aperture, γ_p, was then approximated for each positive 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 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 $\gamma_{opt} = \frac{1}{|P|} \sum_{p \in P} \gamma_p \approx 133.98$. Let C_{opt}^+ be the resulting set of aperture-optimized positive conceptors.

However, classification using C_{opt}^+ may be biased. The conceptors in C_{opt}^+ have different traces as can be seen after the aperture optimization step at $x = 1$ in Figure 2.2. A conceptor's trace equals the sum its eigenvalues and represents the total variance, or volume, of the subspace captured by the conceptor. High singular values and a high trace may arbitrarily result from high energy ESN states (e.g., if the utterances of a class are particularly loud); Yet, this high energy class may be granted a classification advantage, because new ESN states are classified according to their proximity (*positive Evidence*) to the subspaces captured by the positive conceptors, *among other factors*. For illustration, take an arbitrary ESN state of unknown distribution (class). During classification, the positive evidences are com-

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

pared across classes. For some class $p \in P$, this positive evidence equals $E^+(x,p) = x'C_p^+x = x'USU'x$ where USU' is the SVD of C_p^+. A continuous increase of the singular values, roughly captured by $tr(C_p^+) = \sum_i S[i,i]$ causes p's expected positive evidence to increase. This argument is a simplification of the current classification method, and more work is needed to confirm if trace differences introduce bias in the multi-class and *combined evidence* setting. However, the measures taken to reduce the assumed effect, indeed, increased the validation accuracy.

To prevent any such bias, I normalize the positive conceptors in trace while maintaining their information. Therefore, the conceptor's traces set to a shared target value $tr_{target} = \bar{tr}$, the mean trace among the conceptors:

$$\bar{tr} = \frac{1}{|C^+|} \sum_{C \in C_{opt}^+} \mathrm{tr}(C) \approx 57.23 \qquad (2.2)$$

To adapt any conceptor's trace such target value with an error tolerance of ϵ (here, $\epsilon = 0.01$), Algorithm 2.1 was used. Let $\psi(C, tr_{target}, \epsilon)$ be the function computed by the algorithm that returns the trace-adapted conceptor. Since, in practice[¶], a conceptor's trace increases as its aperture increases (Proposition 2 in Appendix Section **??**) and I only know of a shape-preserving method for adapting the aperture (Equation 3), the Algorithm uses the latter to adapt the trace. Concretely, the iterative procedure adapts a conceptor's aperture by a factor of the current trace error ratio until reaching the target trace.

As shown in Figure 2.2 at $x > 1$, applying $\psi(\cdot, \bar{tr}, 0.01)$ to each of the conceptors in C_{opt}^+ normalizes their traces. Thus, via the above two aperture adaptation procedures, the apertures of the conceptors in C_{pre}^+ were optimized and their traces were normalized resulting in the final set of positive conceptors C^+.

Negative conceptors From C^+, the set of negative conceptors $C^- = \{C_p^- | p \in P\}$ was computed. Each class' negative conceptor models the linear state subspace that is complementary to the space occupied the states of all other classes. Or, equivalently, it models the subspace that states "from none

[¶]In practice, all conceptors in C_{opt}^+ have at least one singular value in $(0, 1)$: they are not *hard*.

Algorithm 2.1 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

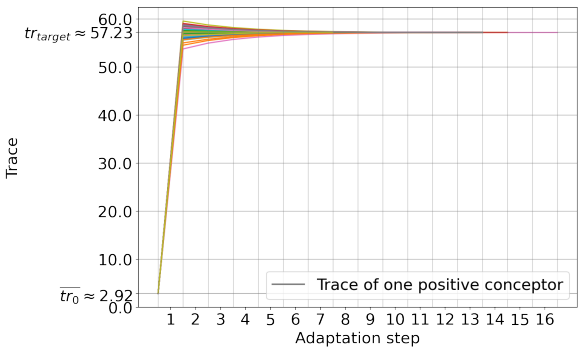

Figure 2.2: The traces of the positive conceptors in function of the adaption steps. The first adaptation step 1 is the aperture adaptation based on the ∇-criterion. The increase in aperture caused the traces to increase and diverge. The remaining steps ($x > 1$) are the results of normalizing the traces using Algorithm 2.1. Finally, these aperture adaptation steps resulted in an increase of the mean trace from an initial value of $\overline{tr_0}$ to approximately the target value tr_{target}.

of the other classes" are expected to occupy. These semantics are reflected in their formal definition:

$$C_p^- = \neg \bigvee \{C_q^+ | q \in P, q \neq p\} \tag{2.3}$$

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

$$\bigvee S = ((C_1 \vee C_2) \vee C_3) \vee ... \, C_{|S|} \tag{2.4}$$

Adapting the apertures of the resulting C_p^-, using the methods with which the positive conceptors were adapted, did not improve the validation accuracy.

2.4.2 Testing

The combined Evidence E was used to classify speech samples 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 combination 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^-:

$$\begin{aligned} E(x, p) &= E^+(x, p) + E^-(x, p), \\ E^+(x, p) &= x' C_p^+ x \\ E^-(x, p) &= x' C_p^- x \end{aligned} \tag{2.5}$$

$E(x, p)$ 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 2.3).

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

$$\arg\max_{p \in P} \frac{1}{L} \sum_{i=1}^{L} E(X[:, i], p), \tag{2.6}$$

where L is the width of X.

Finally, when given an MFCC time series s_{new} from the dataset D, its classification estimate is that of X_{new}, the earlier computed ESN response to s_{new}.

2.4.3 Results

Table 2.2 shows the resulting accuracies. For additional reference, the confusion matrix in Figure A.1 of Appendix A shows the classification rates across the classes. In an attempt to improve the accuracy, the experiment was repeated in slight adaptation to the type of data (Appendix A) which led to a training accuracy of 63.64% but test accuracy of 49.13%.

Domain	Accuracy (%)
Validation	55.66
Train	53.82
Test	53.83

Table 2.2: Validation, training, and test accuracies.

Transition In Experiment 1, an ESN setup (see Table 2.1) was found that could effectively represent time series as reflected in the accuracy achieved during conceptor-based classification. The same ESN setup and similar conceptor-based mechanisms were reused in Experiments 2 and 3 as I turn to unsupervised methods for identifying distinct symbols within ESN dynamics.

2.5 Experiment 2: Below-phoneme clustering

Objective Now blinded of any a-priori groups – the classes $|P|$ previously available during training – the unlabeled ESN responses to the MFCC time-series (pre-processed phoneme utterances) were clustered. Concretely, in Experiment 2 had three conditions for grouping the earlier computed ESN state clouds into K hard (non-overlapping) clusters and one baseline condition.

Dataset For computational constraints, the experiment was run on data subsamples $D_l \subset D_{\text{TIMIT-train}}$ but was repeated 10 times ($l = 1, ..., 10$). Each D_l encompassed utterances of an *independently* sampled set of 7 phonemes P_l to reduce selection bias. The cardinality of $|P_l| = 7$ ought roughly align the difficulty of the current task with the "simple" condition of a previous study, (Lerato and Niesler, 2012), that similarly performed hard

clustering on TIMIT's phoneme recordings. However, since (Lerato and Niesler, 2012) restricted the data to vowels, which I did not for consistency with Experiment 3, their task was likely more difficult. For each phoneme in P_l, 15 phoneme utterances were sampled, thus meeting the centroid-based clustering assumption of equally large clusters, for a total of $n' = 105$ utterances per subset. Concretely, each utterance was sampled from 48 random speakers with an equal ratio across genders and dialect regions (newly sampled for each fold l). This stratification aims to evenly represent the population and enable any found clusters to be evaluated on and generalized to new data. Moreover, only utterances from the phonetically compact sentences, that use each phonemes in a few phonetic contexts (e.g., /aa/ only before /f/), were considered to limit phonetic variability.

Clustering Scheme On each D_l, a generalized centroid-based clustering was performed in four conditions. I will proceed by describing the algorithm with its condition-dependent parameters and then elaborate on the conditions. Besides the desired number of clusters K, the generalized centroid-based clustering algorithm is parametrized by:

1. A set of points $D = \{p_1, p_2, ..., p_n\}$ to be clustered.
2. A centroid computation function centroid(Cl) that provides the centroid of a given cluster $Cl \subset D$.
3. A dissimilarity function $d(p_i, \mu_j)$ that provides the dissimilarity of a point p_i and centroid μ_j. During the assignment step, points are assigned to the cluster with the most similar centroid. This function is assumed to be non-negative and be monotonic increasing with dissimilarity. It may be asymmetric.

The algorithm minimizes the following loss function L:

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

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

L is minimized as follows. Centroids are initialized in a K-means++ fashion. With this method, the initial centroids are sampled from a distribution that aims to spread them evenly across point space. Compared to the random centroid-initialization of classical K-means, this method tends to converge faster and more consistently used on points in Euclidian space (Arthur and Vassilvitskii, 2007). Then, two steps are iteratively repeated. In the *Assignment step*, each data point is assigned to the cluster with the nearest centroid according to some distance function. In the *Centroid update step*, the centroids are recalculated based on the newly formed clusters. This process terminates once all centroids converged in their position or a maximum number of iterations is reached.

Empty clusters were actively prevented. It is theoretically possible for empty clusters to arise after the *Assignment step* in each condition. An explanation will be given in the discussion. In practice, this case only occurred in the conceptor-based conditions (*ESN-Evidence* and *ESN-Mixed*) defined below. Regardless, since no decent centroid can be found for empty clusters, this case was handled in an additional *Reassignment step* after the *Assignment step*. When an empty cluster arises, the most "misfit" point – the point with the largest dissimilarity to its current cluster's centroid – is reassigned to the empty cluster.

Conditions The experiment was performed in four conditions that would determine the parameters of the generalized centroid-based clustering algorithm:

- *MFCC-Euclidian* was a baseline where the MFCC time series were clustered using classical K-means(++).
- *ESN-Euclidian, ESN-Evidence*, and *ESN-Hybrid* clustered ESN state clouds, the responses to the MFCC time series.

First, in the *MFCC-Euclidian* condition, classical K-means was used to cluster the MFCC time series directly.

1. The set of points $D_{MFCC,l}$ comprises the $(d \times L) = (13 \times 10)$ vectorizations of the MFCC time series s_i from D_l:

$$D_{MFCC,l} = \{[s_i(0)|s_i(1), ..., s_i(L)] \,|\, s_i \in D_l,\, 0 < i \leq n'\} \tag{2.8}$$

Algorithm 2.2 Generalized centroid-based hard clustering algorithm

Require:
Number of clusters K
[1] Set of points $D = \{p_1, p_2, ..., p_n\}$
[2] Centroid computation function centroid(Cl)
[3] Dissimilarity function $d(p_i, p_j)$

Initialize K cluster centroids via the K-means++ 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 for 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} shall be moved twice to prevent infinite while loops.
  end while

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

  if cluster assignments did not change **then**
    break because converged
  end if
end while

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

2. Given a cluster $Cl \subset D_{MFCC,l}$, its centroid is the element-wise mean of its member points p_i:

$$\text{centroid}_{Eucl}(Cl) = \frac{1}{|Cl|} \sum_{p_i \in Cl} p_i \tag{2.9}$$

3. The distance from point p_i to cluster Cl is the Euclidian distance between p_i and Cl's centroid:

$$d_{Eucl}(p_i, Cl) = ||p_i - \text{centroid}_{Eucl}(Cl)|| \tag{2.10}$$

Second, in the *ESN-Euclidian* condition, the classical K-means clustered the ESN's responses to the MFCC time series.

1. The set of points $D_{ESN,l}$ contains the earlier collected, unlabeled, ESN responses to the MFCC time series s_i from D_l:

$$D_{ESN,l} = \{X_i | s_i \in D_l\}, \tag{2.11}$$

where X_i is the $(N \times L) = (100 \times 10)$ state collection matrix of the response to MFCC time series s_i.

2. Given a cluster Cl, its centroid is, again, the mean of its members centroid$_{Eucl}(Cl)$.

3. The distance from point p_i to cluster Cl is, again, the Euclidian distance between p_i and Cl's centroid $d_{Eucl}(p_i, Cl)$.

Third, in the *ESN-Evidence* condition, conceptors were used as centroids to represent the clusters. In essence, Experiment 1's classifier training procedure provides cluster centroids and its testing procedure acts to reassign points to clusters.

1. The set of points is, again, the ESN responses $D_{ESN,l}$.

2. Given a cluster Cl_j, its centroid is the tuple of the positive conceptor C_j^+ and negative conceptor C_j^- computed from its and the other clusters' members:

$$\text{centroid}_{Evid}(Cl_j) = (C_j^+, C_j^-) \tag{2.12}$$

Here, the clusters were construed as a set of classes. The same steps used to train the phoneme-classifier in Experiment 1 were applied to train a "cluster-classifier". C_j^+ and C_j^- are, then, the conceptors representing class j with training instances Cl_j.

3. The distance from point p to cluster Cl_j is the reciprocal of the combined Evidence $E(p, j)$ that p corresponds to cluster class j given the previously trained conceptors in $\text{centroid}_{Evid}(Cl_j)$:

$$d_{Evid}(p, \mu_j) = 1/\text{E}(p, j) \tag{2.13}$$

E indicates similarity on the range $(0, \infty)$; so, d_{Evid} also ranges in $(0, \infty)$ with larger values indicating a higher degree of dissimilarity between point p and the centroid μ_j of cluster Cl_j. No division by zero was encountered.

Fourth, in the *ESN-Hybrid* condition, the *ESN-Euclidian* and *ESN-Evidence* conditions were combined.

1. The set of points is, again, the ESN responses $D_{ESN,l}$.

2. Given a cluster Cl_j, its centroid is two-fold; It includes the mean of the states of the *ESN-Euclidian* condition and the positive and negative conceptors of the *ESN-Evidence* condition:

$$\begin{aligned}&\text{centroid}_{Hybrid}(Cl_j)\\ &= (\text{centroid}_{Eucl}(Cl_j), \text{centroid}_{Evid}(Cl_j))\end{aligned} \tag{2.14}$$

3. The distance from point p to cluster Cl_j is the mean of the distances used in the *ESN-Euclidian* and the *ESN-Evidence* conditions:

$$\begin{aligned}&d_{Hybrid}(p, Cl_j)\\ &= (d_{Eucl}(p, Cl_j) + d_{Evid}(p, Cl_j))/2\end{aligned} \tag{2.15}$$

$d_{Eucl}(p, Cl_j)$ and $d_{Evid}(p, Cl_j)$ are computed using their respective elements in $\text{centroid}_{Hybrid}(p, Cl_j)$.

Run With these arguments, Algorithm 2.2 was run in each condition, for 20 *trials* to reduce the effects of the random cluster initialization. Thus, 20 replicates $\times$ 4 conditions $\times$ 20 trials = 1600 trials were performed in total. To limit the scope of this study, only a number of $K = 7$ clusters, consistent with the number of phonemic classes $|P_l|$, were searched for.

Evaluation The resulting clusters were evaluated using one intrinsic measure, the *mean intra-cluster dissimilarity* (MICD), and two extrinsic measures, the *normalized mutual information* (NMI) and *cluster classification accuracy* (CCA).

First, the MICD is the mean dissimilarity between the clusters' centroids and member points across clusters. It is a measure of cluster cohesion ("tightness"), which contributes to clusters being *distinct*. In the interpretation of clusters as symbols (patterns of ESN activity), cluster cohesion translates to a similarity of symbol prototypes to symbol instances. Moreover, it resembles the optimization objectives of the algorithm in the different conditions (compare Equations 2.7 and 2.16), and a downward trend of the MICD over the iterations is expected. Thus, it was applied to inform about cluster cohesion and the convergence behavior of the algorithm. d_{Evid} not being a metric, it also does not fulfill the assumptions of many alternative intrinsic cluster quality measures like the within-cluster sum of squares or the silhouette coefficient. Importantly, the MICD should not be mistaken as a means to compare between conditions since these used different dissimilarity functions and data types. 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 (ICD) be the mean dissimilarity between its member points p_i and its centroid $\text{centroid}(Cl_k)$: 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 ICD values for all clusters:

$$MICD = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|Cl_k|}\sum_{p_i \in Cl_k} d(p_i, \text{centroid}(Cl_k)) \tag{2.16}$$

where K is the number of clusters.

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 assigned to cluster k, and the ground-truth phonemic groups $G = \{G_1, G_2, ..., G_{|P_l|}\}$, where G_p is the set of points with label p in the dataset D_l. Its values range from 0 for completely dissimilar clusterings and 1 for identical clusterings. It was used to compare the empirically found clusters with the phonemic classes of TIMIT. To compute the NMI, the mutual (shared) information I between Cl and

G is normalized by the mean entropy (uncertainty) H within each clustering:

$$I(G, Cl) = \sum_{G_p \in G} \sum_{Cl_k \in Cl} P(G_p \cap Cl_k) \log \frac{P(G_p \cap Cl_k)}{P(G_p)P(Cl_k)}$$

$$H(G') = -\sum_{G'_p \in G'} P(G'_p) \log P(G'_p), \text{ for some clustering } G'$$

$$NMI(G, Cl) = \frac{I(G, Cl)}{\frac{1}{2}[H(G) + H(Cl)]} \tag{2.17}$$

Third, I evaluated how accurately the conceptors derived from the clusters could classify new phonemes. This extrinsic measure, that I will refer to as *cluster classification accuracy* (CCA), ought to measure how accurately the acquired symbols this system could be used to represent certain entities, phoneme utterances in this case. Therefore, it was assumed that each cluster corresponded to one of the phonemic classes P. A bijective mapping of clusters to classes was made using Kuhn-Munkres algorithm to globally maximize the intersections (Plummer and Lovász (1986) mentioned in Song, Liu, Huang, Wang, and Tan (2013)). Then, Experiment 1 was essentially replicated; a conceptor-based classifier was trained on the clusters to classify the respective matched class. The classifier was tested on the phonetically diverse utterances of phonemes P_l that remained from the eight selected speakers.

Results Table 2.3 compares the mean scores of the extrinsic performance measures (columns) of the clustering results across conditions (rows). Two additional rows were added, 'Random clusters' and 'Dataset classes', with the scores to expect of a random clustering and a clustering that perfectly matched the true class labels, respectively. The scores in all conditions were significantly above random [**Any way to support the significance claim?**]. Between the conditions, *MFCC-Euclidian* outperformed the others in terms of NMI by very little. No significant differences in NMI could be found between the ESN-based conditions. The *ESN-Mixed* condition and the *MFFC-Euclidian* outperformed the others in their accuracy.

Figures 2.3 and 2.4 depict the normalized MICDs and NMIs over the iteration. Horizontal lines provide comparison to the scores of randomly initialized clusters and the "ground truth" scores of the dataset's classes. The MICDs were rescaled to a unit range for all condition, to visualize all of their values, initially of different ranges, in one plot. Moreover, the plotted values are the means across trials, excluding any already terminated trials. Since trials terminated at different iterations, the scores at the higher iterations are the means of *fewer* values which explains the increase in variance. For the MICDs, a downward trend exponentially decaying over the iterations can be observed for all conditions. Conversely, the NMIs across conditions increased over the first iteration, after which almost only very slight improvements occurred. Most change in the MICD and NMIs occured within the first three iterations (the large decline of the MICD in the *ESN-Evidence* condition after iteration 13 seems to be due to the increased variance).

Condition	**NMI**	**CCA**
MFCC-Euclidian	0.4934	0.5195
ESN-Euclidian	0.4774	0.5073
ESN-Evidences	0.4717	0.5040
ESN-Mixed	0.4757	0.5167
Random clusters	0.0871	0.2438
Dataset classes	1.0000	0.6971

Table 2.3: Mean scores of final clusterings between conditions.

2.6 Experiment 3: Above-phoneme

The second clustering experiment structures the conceptors C^+ in a hierarchy through an adaptation of HAC [add a reference to orginal paper]. Whereas in Experiment 2, clusters were identified among ESN states grouped (segmented) by phoneme utterance (more concrete subsymbols), Experiment 3 aims to identify cluster hierarchies among ESN states grouped by phoneme (more abstract subsymbols). The observation that lead to Experiment 3 is that many domains, like Phonetics, are conceptually structured as hierarchies or taxonomies of symbols of increasing abstraction (semantic space). Therefore, this experiment's purpose is to see if such hierarchical tree structures can be recovered from an ESN's representation of the domain (representa-

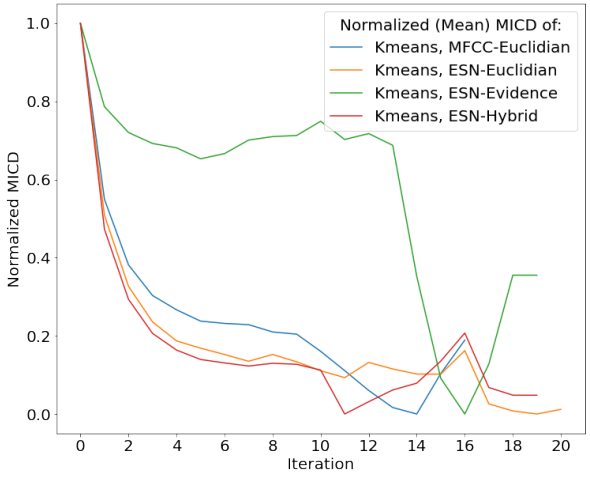

Figure 2.3: The MICDs of the clusterings over the iterations averaged across the trials. [Redo figure replacing "Kmeans" by "Clustering"]

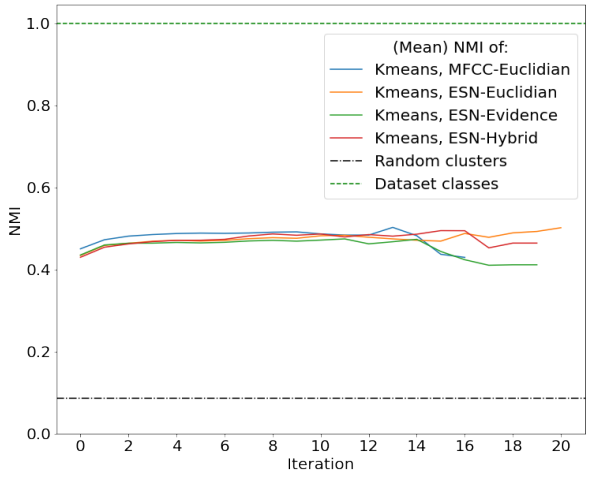

Figure 2.4: The NMIs of the clusterings over the iterations averaged across the trials. For reference, the ideal NMI of clusters corresponding to the dataset's classes (*Dataset*) and the baseline NMI of randomly initialized clusters (*Random*) are shown as horizontal lines.

tional space). The positive conceptors capture the geometry of the ESN's representation of phonemes.

Method I will proceed by describing the generalized HAC algorithm, then elaborate on its three parameters and their used arguments (values). HAC is a clustering algorithm aimed at identifying a hierarchy of clusters among the input data. The algorithm works by initializing one cluster per data point and then iteratively merging the two closest clusters until a single cluster or a desired number of clusters is obtained. The result is a hierarchical structure of nested clusters commonly visualized as a dendrogram.

Algorithm 2.3 Generalized HAC

[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

As shown by its pseudocode, the generalized HAC algorithm requires three parameters. These were set as follows to adapt it to the clustering of conceptors:

1. A set of points. These are classically in Euclidian space. Here, the set of points to be clustered consisted of one conceptor embedding per phonemic class of the corresponding samples in the training set. Therefore, the set of positive conceptors from Experiment 1 was reused:

$$D_{HAC} = C^+ = \{C_p^+ | p \in P\} \tag{2.18}$$

2. A dissimilarity measure between two points p_i and p_j. This is classically the Euclidian distance. Here, the reciprocal of the conceptor similarity of p_i and p_j was used:

$$d_{HAC}(p_i, p_j) = 1 - \text{Sim}(p_i, p_j) \tag{2.19}$$

This function fulfils several desirable properties for a dissimilarity measure: It is non-negative, symmetric, and equals 0 for identical inputs.

3. A function that provides the dissimilarity of two clusters. A common choice is the mean pairwise dissimilarity of the points in the two clusters. This variant of HAC is coined average linkage. Concretely, the linkage function $d_{link}(Cl_i, Cl_j)$ between clusters Cl_i and Cl_j is:

$$d_{link}(Cl_i, Cl_j) = \frac{1}{|Cl_i||Cl_j|} \sum_{p_x \in Cl_i, p_y \in Cl_j} d_{HAC}(p_x, p_y) \tag{2.20}$$

Results The resulting hierarchy is depicted as a dendrogram in Figure 2.5. Each leaf on the left represents a phonemic group. Moving toward the right, phoneme clusters emerge. The dissimilarities between merged child clusters is the abscissa of their link.

Several overlaps can be identified between the HAC phoneme clustering results and phonetic groups depending on the choice of phonetic model. I will compare the clusters with the classes provided by Pfeifer and Balik (2011) and shown in Figure 1.2, for their model is based on manner of production that seems the variable most correlated to the HAC clusters. Table 2.4 depicts these overlaps with the phonetic groups in the left column and their associated phonemes in the right column. Each group also corresponds to an HAC cluster with the exception of some **bold** phonemes that were moved between sibling clusters. The two primary clusters encompass consonants and vowels with a dissimilarity of about 0.015. Within the upper, 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 plosives (top) and nasals (bottom). However, instead of the manner of production, the place of production may be considered as the group-determining variable. Thus, several other overlaps of the subclusters of consonants with phonetic groups may be found; the *bilabial* stops /p/ and /b/, the *alveolar* stops /t/ and /d/, the *velar* stops /k/ and /g/, and the *alveolar* fricatives /s/ and /z/ were each assigned to exclusive subclusters.

Group		Phonemes of Group
Consonants	Fricatives	th f sh z s **dh v hh**
	Affricatives	jh ch
	Plosives	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**
Silence		h#

Table 2.4: Linguistic groups found within the phoneme clusters that resulted from HAC. Bold phonemes were moved from sibling clusters into their current place.

The lower cluster predominantly consists of vowel sounds. No significant correspondence between the sub-clusters of vowels and their pronunciation position or other articulatory features is apparent.

The phonemes in the Mixed group are considered separately, for their articulatory features resemble both vowels and consonants. The group contains the liquids /l/ and /r/ and semivowels /w/ and /y/. These instances all needed to be moved, since no cluster exclusively corresponded to the mixed group. Lastly, the silence /h#/ is considered separately like by Pfeifer and Balik (2011).

In summary, the table displays how the organization of phonemes derived from HAC coincides with traditional phonetic categories. The large resemblance suggests that the HAC algorithm produced relevant and mostly coherent clusters.

3 Discussion

Draft

3.1 Overview

[Paragraph: Recap] This study aimed to identify distinct and meaningful clusters within an ESN's representation of phonemic utterances....

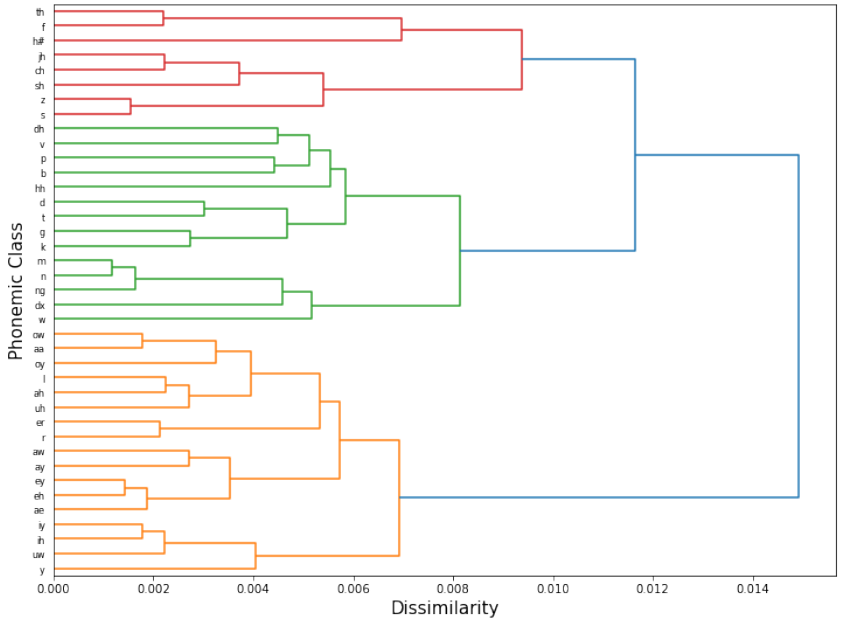

Figure 2.5: Dendrogram of the hierarchy that resulted from the adapted average linkage agglomerative clustering algorithm.

[Paragraph: Disclaimer] Importantly, the algorithms in the three experiments (except for the baseline of Experiment 2) used exclusively *ESN states*. Not speech signals, but the ESN's responses to these signals were the inputs to classification and clustering algorithms.

3.2 Discussion on Experiment 1: Phoneme Classification

[Paragraph: Purpose of Experiment 1] Experiment 1 served to setup an ESN that is sensitive to the data and demonstrate conceptor-based time-series classification

[Paragraph: Classification]
Findings [Mention results]
Implications [Comparison to phoneme classification benchmark and Jaeger's experiments.]

[Paragraph: Aperture Normalization]
Findings [Detail the significance of aperture adaptation in conceptor-based time series classification and allude to its implications for all experiments.]

- Methods of conceptor-based time series classification were inspired by (Jaeger, 2014b).
- Deviation to adapt explicitly only the apertures of the positive conceptors.
- Implications of normalization for classification.
- Figure reference depicting the normalization process.

The methods of conceptor based time series classification used in Experiment 1 were largely inspired by (Jaeger, 2014b) with the exception of the aperture adaptation. The first deviation here was to adapt explicitly only the apertures of the positive conceptors C^+..... [**normalizing/optimizing the aperture of the negative conceptors does not help**] It was made to avoid to give conceptors with larger singular values an advantage during classification. To understand this effect, consider the example of a 2×2 conceptor matrix A. [**...**] To avoid such classification errors, the apertures were iteratively adapted until approaching a target singular value sum using the following procedure. Indeed this normalization further improved classification.

It must be remarked that, by that process, the final apertures will deviate from the one originally chosen aperture via the ∇-criterion. Figure 2.2 depicts this normalization process for the classification

experiment.

[Paragraph: Z-Condition]

Findings Higher training, but lower test accuracy. Overfitting. Potential reasons for overfitting. Worse due to overfitting: Exponential reason (AI2). The size of z and the associated computational costs are larger than for the former method, and it tends to lead to better *training* classification accuracy, but much worse test accuracy indicating overfitting.

Implications Not useful.

3.3 Discussion on Experiments 2 and 3: Clustering Advantages and Challenges

[Paragraph: Purpose of Experiments 2 and 3]

[Paragraph: Findings from Experiment 2] Compare conditions quickly (distance considerations come later)

[Paragraph: Findings from Experiment 3]

3.4 Implications

[Paragraph: Clustering-related implications]

Re-explain General conceptor-based theme Emphasize that ESN activities are classified and clustered using conceptors, not the speech signals themselves.

[Paragraph: Obvious Advantages and Challenges]

- Advantages/disadvantages related to clustering methods.
- Pros and Cons slide of video (average linkage algorithm slide in particular)

[Paragraph: Time Efficiency and convergence] Time complexity of clustering algorithms: especially Experiment 2

[Paragraph: Comparison to traditional algorithms] **Conceptors as state embeddings** Discuss the role of conceptors as embeddings of ESN responses.

[Paragraph: On the compression aspect of conceptors / advantages & disadvantages / implications for data analysis] Highlight their significance in temporal and spatial compression. ESNs deal with time-extended inputs causing difficulties for preexisting clustering algorithms. Conceptors do not [...]. Moreover, they compress one or more network states into a mathematical object (a matrix). As constant-sized matrices, they do not scale with the number of time steps or modeled states. This is a leap for analyzing time series since previous algorithms typically scale poorly with longer samples; for example, computing the distance between time series in dynamic time warping (DTW) scales quadratically with sample length dynamic. Notwithstanding, the quality of conceptors depends on the number of states used for their computation [...]. [Comparing responses of different lengths would require workarounds like Dynamic Time Warping (DTW).] Moreover, ESNs deal with time-extended inputs causing several difficulties for preexisting clustering algorithms. For example, the euclidian distance can only be used to compare samples of different durations (for clustering algorithms that require distance functions) and have time-. They capture their information in a static mathematical compressing the temporal and spatial dimensions. The temporal and spatial compression realized by conceptors, allows them to be used as embeddings of ESN responses. Unifying responses in conceptor-space, independently of their length and maintaining only their most important information, allows state sequences to be related and compared. [Examples of the use of conceptors as embeddings]

[Paragraph: On the distance functions (of conceptors and ESN states)]

Why do we care about distance functions?

Challenge use of Euclidian distance functions in the context of neural networks. Highlight the significance of moving to conceptor space: Classically, these algorithms are used on data in Euclidian space and their components are well-defined for that space (mean, Euclidian distance, manhattan distance, etc.). However, the activities within the ESN are subjected to non-linear transformations like *tanh* at each time step as is the case for most neural networks. I hypothesize that Euclidian distance functions may not be well-suited to capturing the distances between network states. Moving to conceptor space will be an attempt to improve the clustering of ESN activities through a more adequate means for cluster representation, comparison, and assignment.

We attempt to solve this mismatch by mapping the network states to be clustered to conceptors since conceptors offer several tools for comparing and analyzing network states. By moving to conceptor space, the following adaptations were made.

1. plus becomes or: Firstly, the operations of Eu-

clidian vector space were changed.

2. division become aperture adaptation
3. Distance becomes the one proposed by Jaeger (2014b) OR...

[Paragraph: Problems with distance function] - (Lack of) semantics and implications - Convergence Concerns: Bregman divergence and cosine similarity. Assumptions of generalized clustering: Banerjee et al. (2005) bregman divergence, zero dist function when coinciding points and centroids. Importantly, this is Not a Bregman divergence, because cosine similarity does not fulfill the triangle inequality. Therefore, this algorithm is not guaranteed to converge. The current algorithm is not guaranteed to converge since it is an adaptation of K-Means that uses a non-Bregman divergence function (more on this in the comment under Experiment 2). Attempted explanation of swallowing problem. For experiments 1 and 3, there are no strong (bregman) assumptions, so the conceptor-based distance function is fine.

[Paragraph: Other aspects of Conceptor-based Clustering vs. Classical Methods]

[Paragraph: Symbols] - Implications for Neuro-symbolic integration - Unsupervised Symbol Identification - Symbols in time -¿ evolution and time-dependent clustering - Potential objection(s) Finally, one may object that, for unsupervised symbol acquisition, the use of hard clustering is very artificial and something akin to Hebbian learning should be preferred, something that resembles how the brain acquires symbols. However, often in computing, and especially reservoir computing for material constraints, learning may not be possible. Yet, one may want to identify the symbols prevalent in the "artificial mind".

[Paragraph: Implications for explainability] - Applications of findings in the context of neural networks, especially in terms of explainability - BPTT: To evaluate how well a classifier can distinguish between a set of classes, one may resort to performance metrics or investigate the degree of certainty in its predictions. However, such an evaluation relies only on the outputs and disregards the internal states that led to the predictions. For complex models or whole pipelines, it may be of interest to find groups or clusters present in the RNN's activations or within a subset thereof, e.g., corresponding to only one part of a pipeline. This study ought to be an exploratory step toward answering the following questions: How clustered are the activities within a NN? When, during training, do class differences in network activities arise?

3.5 Limitations

[Convergence]

[Misrepresentation of conceptors, mean vs variance]

[Paragraph: Categories/clusters may be due to co-articulation / correlations]

[Paragraph: Time] Issues with generalization to non-stationary signals. Challenges and implications of handling different utterance speeds and potential solutions. Normalization in time sufficient for good performance, but unknown if necessary. Potential solutions like adding leakiness to ESN to integrate time scales. Without normalization in time, performance decreases. How to handle different utterance speeds? Add leakiness to ESN to integrate time scales?

[Paragraph: ESP] The ESP might not be satisfied. Effects of spectral radius on ESP.]
Nonetheless, a phonetic interpretation about the speech signals will be made which, however, requires an assumption. Commonly, the echo state property is used to ensure a functional relationship between driver signal (phoneme recording) and RNN response Yildiz et al. (2012). However, for the echo state to be reached requires left-infinite or at least considerably long input sequences, for the starting state to be washed out. In our case, inputs will be downsampled to a length of 10 rendering the claim of the ESP impossible based on current theory. The analyzed states will most likely still correspond to the network's initial transient period. Thus, functionality between input and response cannot be guaranteed but only assumed. Hence, we shall nonetheless use the results of clustering and classification of ESN activities from the transient period for an interpretation about the input signals.

3.6 Future Directions

Potential future experiments based on findings and limitations .

4 Conclusions

[Summary of discussion. Emphasize the collective significance of experiments for RQ.]

References

Stefan Appelhoff, Ralph Hertwig, and Bernhard Spitzer. Eeg-representational geometries and psychometric distortions in approximate numerical judgment. *PLOS Computational Biology*, 18(12): e1010747, 2022.

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.

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

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.

William G Chase and Herbert A Simon. Perception in chess. *Cognitive psychology*, 4(1):55–81, 1973.

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

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.

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.

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

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

Douglas R Hofstadter. Analogy as the core of cognition. *The analogical mind: Perspectives from cognitive science*, pages 499–538, 2001.

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.

Herbert Jaeger. Dynamische symbolsysteme (dynamic symbol systems). Ph.d. thesis, University of Bielefeld, 11 1996.

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.

Herbert Jaeger. Using conceptors to manage neural long-term memories for temporal patterns. *The Journal of Machine Learning Research*, 18(1):387–429, 2017.

Herbert Jaeger, Mantas Lukoševičius, Dan Popovici, and Udo Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3):335–352, 2007.

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.

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.

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.

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`.

Qihong Lu, Po-Hsuan Chen, Jonathan W Pillow, Peter J Ramadge, Kenneth A Norman, and Uri Hasson. Shared representational geometry across neural networks. *arXiv preprint arXiv:1811.11684*, 2018.

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

Mantas Lukoševicius, Dan Popovici, Herbert Jaeger, Udo Siewert, and Residence Park. Time warping invariant echo state networks. *International University Bremen, Technical Report*, 2006.

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 Ellis, Matt Mcvicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. pages 18–24, 01 2015. doi: 10.25080/Majora-7b98e3ed-003.

George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63 (2):81, 1956.

Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

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`.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *The Journal of Machine Learning Research*, 21(1): 7503–7542, 2020.

Allen Newell and Herbert A Simon. Computer science as empirical inquiry: Symbols and search. In *ACM Turing award lectures*. 1975.

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.

Hans Op de Beeck, Johan Wagemans, and Rufin Vogels. Inferotemporal neurons represent low-dimensional configurations of parameterized shapes. *Nature neuroscience*, 4(12):1244–1252, 2001.

Charles Sanders Peirce and Justus Buchler. Logic as semiotic: The theory of signs. *Philosophical Writings of Peirce (New York: Dover Publications, 1955)*, page 100, 1902.

Leonid I Perlovsky. Symbols: Integrated cognition and language. In *Semiotics and intelligent systems development*, pages 121–151. IGI Global, 2007.

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.

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

Linus Schilpp. *Phoneme Classification and Alignment through Recognition on TIMIT*. PhD thesis, Ph. D. Thesis, Institute for Anthropomatics and Robotics Interactive Systems . . . , 2021.

Fei Sha and Lawrence K Saul. Large margin gaussian mixture modeling for phonetic classification and recognition. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

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

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.

Pattreeya Tanisaro and Gunther Heidemann. Time series classification using time warping invariant echo state networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 831–836. IEEE, 2016.

Yee Whye Teh. Hierarchical clustering. `http://videolectures.net/epsrcws08_teh_hc/`, January 23 2008. Slideset from the Sheffield EPSRC Winter School.

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

Tim Van Gelder. The dynamical hypothesis in cognitive science. *Behavioral and brain sciences*, 21(5):615–628, 1998.

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

Felix Wang, William M Severa, and Fred Rothganger. Acquisition and representation of spatiotemporal signals in polychronizing spiking neural networks. In *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*, pages 1–5, 2019.

Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193, 2015.

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

A Appendix

A.1 Dataset

Table A.1 lists the phone classes and their frequencies within the processed TIMIT dataset.

A.2 Additions to Experiment 1

A.2.1 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.2.3 Additional Results of Experiment 1

The confusion matrix in Figure A.1 shows the classification rates across the classes.

A.3 Additions of Experiment 2: Hyperparameter Optimization

Hyperparameters $k_b, k_{W^{in}}, r$ and ρ were also tuned automatically using bayesian optimization procedure of the Bayesian Optimization python package. The optimization objective was to maximize the testing accuracy of phoneme classification in Experiment 1. Concretely, after 10 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 trying to maximize an estimated surrogate $\bar{f}$ for the unknown objective function f $[f(\rho, k_{W^{in}}, k_b, r)$:=accuracy]. After training and testing the phoneme classifier with these hyperparameters on the training set (with a train-test split), the surrogate estimate is improved (for a detailed review of bayesian optimization, see Frazier). 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 $\rho \in (0.01, 1)\%$

	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 that were originally present in TIMIT but folded into a class with the left-adjacent phone. Columns 3 & 4: Number of speech samples of each class. Bottom row: Sums of classes or samples.

A.2.2 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 progress of the bayesian optimizer is depicted in Figure A.2. Bayesian optimization was preferred over the more straightforward grid search, since, under consideration of 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.

A.4 Extension of Experiment 1: Inclusion of input states

Experiment 1 was repeated in slight adaptation of its methods to the stationary type of data. Jaeger (2014b) demonstrated that conceptors may 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 are the results of stationary processes. Meanwhile, non-stationary processes change their properties over time leading to signals like speech whose probability distributions change over time. Meanwhile in the current classification method, the order within a sequence of states does not affect the resulting conceptor since it is lost when computing the correlation matrix. This works fine on signals from stationary sources where temporal order is of no relevance. However, for non-stationary sources, states can have different probabilities of occurring depending on their position in the sequence, information not accounted for in the current classification method. 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 test accuracy of 49.13 were reached.

A.5 Additions of Experiment 2

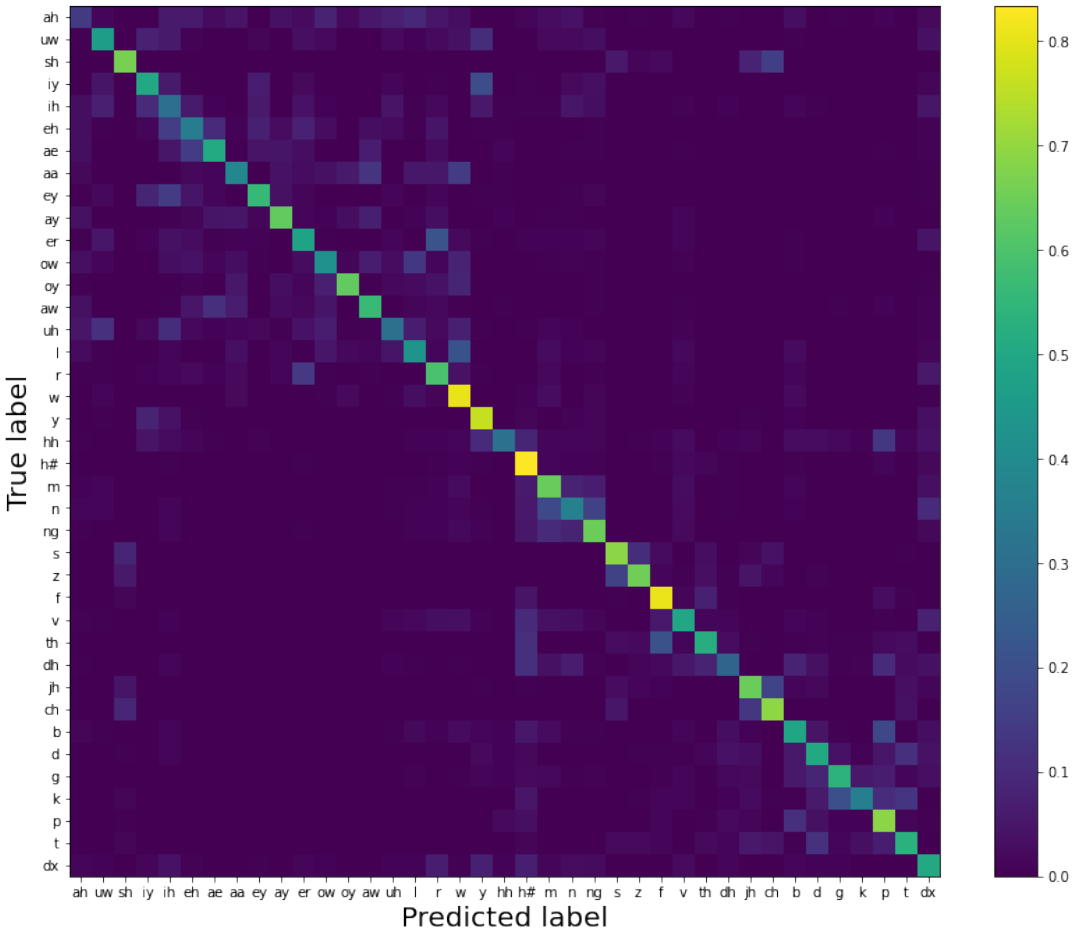

Figure A.1: Multi-class confusion matrix of the classification results. The colors represent the relative frequencies of the predictions (x-axis) made for each of the classes (y-axis). For every phoneme, their rate of correct classification (diagonal entries) was higher than the rate of misclassification to any of other classes (off-diagonal entries) suggesting an above-chance classification accuracy across all phonemic classes. Error rates seem elevated within groups of shared articulatory features like vowels (top left quadrant) and consonants (bottom right quadrant) [Elaborate if necessary].

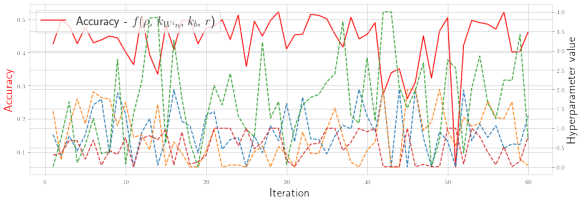

Figure A.2: Hyperparameter tuning with the bayesian optimizer and accuracy [Still no progress seems to occur. This experiment will need to be repeated.]

Algorithm A.1 K-means++ improved 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 distance to 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 $\mu = \{\mu_1, \mu_2, ..., \mu_K\}$

B Appendix

B.1 Mathematical notations

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: List of some of the mathematical notations used throughout the paper.