THE GEOMETRY OF INTEGRATION IN TEXT CLASSIFICATION RNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the widespread application of recurrent neural networks (RNNs) across a variety of tasks, a unified understanding of how RNNs solve these tasks remains elusive. In particular, it is unclear what dynamical patterns arise in trained RNNs, and how those patterns depend on the training dataset or task. This work addresses these questions in the context of a specific natural language processing task: text classification. Using tools from dynamical systems analysis, we study recurrent networks trained on a battery of both natural and synthetic text classification tasks. We find the dynamics of these trained RNNs to be both interpretable and low-dimensional. Specifically, across architectures and datasets, RNNs accumulate evidence for each class as they process the text, using a low-dimensional attractor manifold as the underlying mechanism. Moreover, the dimensionality and geometry of the attractor manifold are determined by the structure of the training dataset; in particular, we describe how simple word-count statistics computed on the training dataset can be used to predict these properties. Our observations span multiple architectures and datasets, reflecting a common mechanism RNNs employ to perform text classification. To the degree that integration of evidence towards a decision is a common computational primitive, this work lays the foundation for using dynamical systems techniques to study the inner workings of RNNs.

1 INTRODUCTION

Modern recurrent neural networks (RNNs) can achieve strong performance in natural language processing (NLP) tasks such as sentiment analysis, document classification, language modeling, and machine translation. However, the inner workings of these networks remain largely mysterious.

As RNNs are parameterized dynamical systems tuned to perform specific tasks, a natural way to understand them is to leverage tools from dynamical systems analysis. A challenge inherent to this approach is that the state space of modern RNN architectures—the number of units comprising the hidden state—is often high-dimensional, with layers routinely comprising hundreds of neurons. This dimensionality renders the application of standard representation techniques, such as phase portraits, difficult. Another difficulty arises from the fact that these are monolithic systems trained end-to-end. Instead of modular components with clearly delineated responsibilities that can be understood and tested independently, neural networks could learn an intertwined blend of different mechanisms needed to solve a task, making understanding them that much harder.

Recent work has shown that modern RNN architectures trained on binary sentiment classification learn low-dimensional, interpretable dynamical systems (Maheswaranathan et al., 2019). These RNNs were found to implement an integration-like mechanism, moving their hidden states along a line of stable fixed points to keep track of accumulated positive and negative tokens. Later, Maheswaranathan & Sussillo (2020) showed that contextual processing in these networks, e.g. handling word sequences like "not good", was built on top of the line-integration mechanism, employing an additional subspace which the network entered upon encountering a modifier word (e.g. "not" and "very"). The understanding achieved in those works suggest the potential of the dynamical systems perspective. However it remains to be seen if this perspective can shed light on RNNs in more complicated settings, beyond binary sentiment classification.

This work applies the dynamical systems perspective to understand how RNNs perform various multi-class text classification tasks, including document classification, review star prediction (from

one to five), and emotion tagging. Because text classification is a fundamental task in natural language processing, and the underlying evidence-integration mechanisms are core computational primitives, precisely understanding how RNNs solve text classification may prove useful in understanding more complex NLP problems.

We find the network dynamics in these text classification tasks to be low-dimensional and interpretable. All architectures use manifolds of attractive fixed points to accumulate evidence for each class. When class labels are categorical (e.g. "sports" or "business"), the dimensionality of the integration manifold simply reflects the number of classes in the dataset. Integration is also the dominant mechanism when class labels are ordered (as in star prediction); here, however, the manifolds case are two-dimensional in both three- and five-star settings. We show that this dimensionality reflects underlying correlations in the training dataset, and show that simple count statistics in the raw dataset is sufficient to explain the observed dimensionality. This unified picture allows us to also understand the integration manifolds in the the case of emotion tagging, which unlike the previous tasks we study allows each phrase to contain multiple labels.

In studying these different tasks, we rely heavily on synthetic datasets, which offer a controllable setting to study network dynamics. We develop synthetic data for each type of classification task; networks trained on these datasets exhibit similar dynamics and geometry as their natural counterparts.

Related work Apart from the recent studies mentioned in the introduction—which to our knowledge are the only ones to study discrete-time RNNs in language tasks using dynamical systems analysis—the dynamical properties of *continuous-time* RNNs have been extensively studied (Vyas et al., 2020) for their potential connections to neural computation in biological systems.

A related body of work, interpretability of neural language models, is reviewed thoroughly in Belinkov & Glass (2018). Common methods of analysis include training auxiliary classifiers (e.g., part-of-speech) on RNN trajectories to probe the network's representations; use of challenge sets to capture wider language phenomena than seen in natural corpora; and visualization of hidden unit activations as in Karpathy et al. (2015) and Radford et al. (2017).

2 SETUP

Models We study three common RNN architectures: LSTMs (Hochreiter & Schmidhuber, 1997), GRUs (Cho et al., 2014), and UGRNNs (Collins et al., 2016). We denote their n-dimensional hidden state and d-dimensional input at time t as $\mathbf{h}_t$ and $\mathbf{x}_t$, respectively. The function that applies hidden state update for these networks will be denoted by F, so that $\mathbf{h}_t = F(\mathbf{h}_{t-1}, \mathbf{x}_t)$. The network's hidden state after the entire example is processed, $\mathbf{h}_{\mathrm{T}}$, is fed through a linear layer to get N output logits for each label: $\mathbf{y} = \mathbf{W}\mathbf{h}_{\mathrm{T}} + \mathbf{b}$. We call the rows of $\mathbf{W}$ 'readout vectors' and denote the readout corresponding to the ith neuron by $\mathbf{r}_i$, for $i = 1, \ldots, N$. Throughout the main text, we will present results for the GRU architecture. Qualitative features of results were found to be constant across all architectures; additional results for the LSTM and UGRNN are given in Appendix E.

Tasks The classification tasks we study fall into three categories. In the ***categorical*** case, samples are classified into non-overlapping classes, for example "sports" or "politics". By contrast, in the ***ordered*** case, there is a natural ordering among labels: for example, predicting a numerical rating (say, out of five stars) accompanying a user's review. Like the categorical labels, ordered labels are exclusive. Some tasks, however, involve labels which may not be exclusive; an example of this ***multi-labeled*** case is tagging a document for the presence of one or more emotions. A detailed description of the natural and synthetic datasets used is provided in Appendices C and D, respectively.

Linearization and eigenmodes Part of our analysis relies on linearization to render the complex RNN dynamics tractable. This linearization is possible because, as we will see, the RNN states visited during training and inference lie near approximate fixed points $\mathbf{h}^*$ of the dynamics—points that the update equation leave (approximately) unchanged, i.e. for which $\mathbf{h}^* \approx F(\mathbf{h}^*, \mathbf{x})$.[1] Near these points, the dynamics of the displacement $\Delta\mathbf{h}_t := \mathbf{h}_t - \mathbf{h}^*$ from the fixed point $\mathbf{h}^*$ is well

[1] Although the fixed points expression depends on the input $\mathbf{x}$, throughout this text we will only study fixed points under a network's average input which is usually well approximated by $\mathbf{x} = 0$ (see Appendix A).

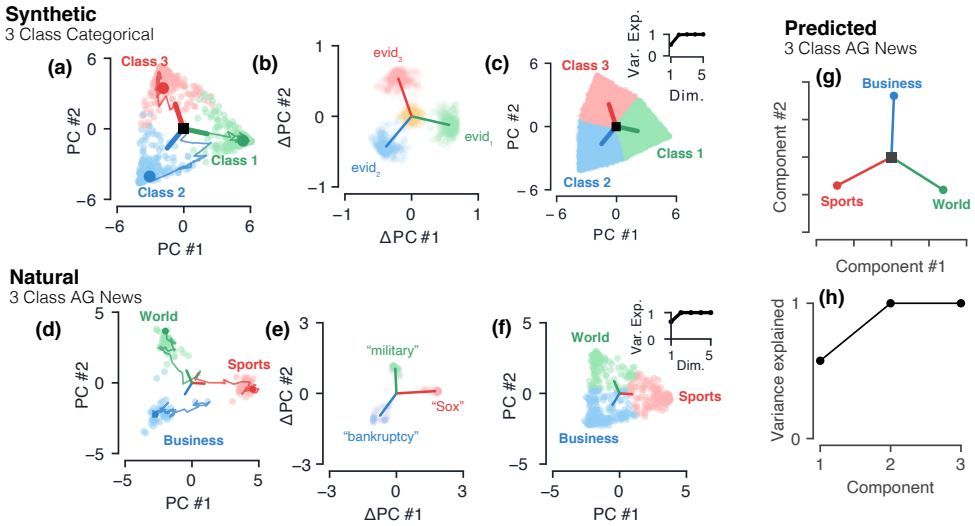

Figure 1: **Results from training networks on 3-class categorical data.** **(a, d)** Final hidden states of many test samples, colored by their label, with a few example hidden states trajectories shown explicitly. The initial state $\mathbf{h}_0$ is shown as a black square, and the three thick solid lines are the three readouts, colored by their respective class. **(b, e)** The hidden deflections from the three words in the vocabulary, with the average deflection of each shown as a solid line. **(c, f)** Fixed points, colored by their predicted label. The inset shows the variance explained as a function of number of PC dimensions. **(g)** The LSA score vectors projected into the top two variance dimensions. **(h)** Percentage of variance explained versus number of dimensions for LSA.

approximated by the linearization

$$\Delta \mathbf{h}_t \approx \mathbf{J}^{\text{rec}}\big|_{(\mathbf{h}^*,\mathbf{x}^*)} \Delta \mathbf{h}_{t-1} + \mathbf{J}^{\text{inp}}\big|_{(\mathbf{h}^*,\mathbf{x}^*)} (\mathbf{x}_t - \mathbf{x}^*)\,, \tag{1}$$

where we have defined the recurrent and input Jacobians $J^{\text{rec}}_{ij}(\mathbf{h},\mathbf{x}) := \partial F(\mathbf{h},\mathbf{x})_i/\partial h_j$ and $J^{\text{inp}}_{ij}(\mathbf{h},\mathbf{x}) := \partial F(\mathbf{h},\mathbf{x})_i/\partial x_j$, respectively (see Appendix A for details).

In the linear approximation, the spectrum of $\mathbf{J}^{\text{rec}}$ plays a key role in the resulting dynamics. Each eigenmode of $\mathbf{J}^{\text{rec}}$ represents a displacement whose magnitude either grows or shrinks exponentially in time, with a timescale τ_a determined by the magnitude of the corresponding (complex) eigenvalue λ_a via the relation $\tau_a := |\log|\lambda_a||^{-1}$. Thus, eigenvalues within the unit circle thus represent stable (decaying) modes, while those outside represent unstable (growing) modes. The Jacobians we find in practice almost exclusively have stable modes, most of which decay on very short timescales (a few tokens). Eigenmodes near the unit circle have long timescales, and therefore facilitate the network's storage of information.

Latent semantic analysis For a given text classification task, one can summarize the data by building a matrix of word or token counts for each class (analogous to a document-term matrix (Manning & Schutze, 1999), where the documents are classes). Here, the i, j entry corresponds to the number of times the ith word in the vocabulary appears in examples belonging to the jth class. In effect, the column corresponding to a given word forms an "evidence vector", i.e. a large entry in an particular row suggests strong evidence for the corresponding class. Latent semantic analysis (LSA) (Deerwester et al., 1990) looks for structure in this matrix via a singular value decomposition (SVD); if the evidence vectors lie predominantly in a low-dimensional subspace, LSA will pick up on this structure. The top singular modes define a "semantic space": the left singular vectors correspond to the projections of each class label into this space, and the right singular vectors correspond to how individual tokens are represented in this space.

Below, we will show that RNNs trained on classification tasks pick up on the same structure in the dataset as LSA; the dimensionality and geometry of the semantic space predicts corresponding features of the RNNs.

Regularization While the main text focuses on the interaction between dataset statistics and resulting network dimensionality, regularization also plays a role in determining the dynamical structures.

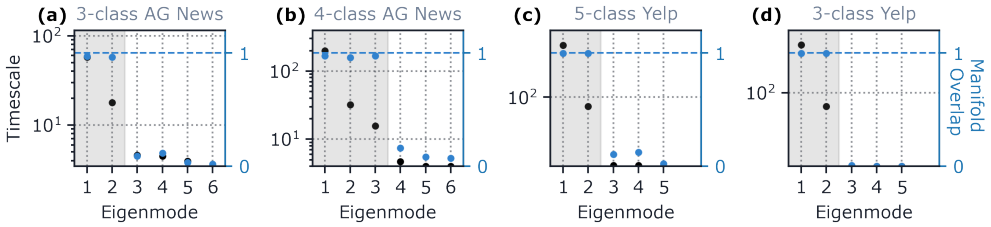

Figure 2: **Evidence integration is accomplished via eigenmodes with long timescales, aligned with the fixed-point manifold**. For each task, only a few modes (highlighted in gray) do not decay appreciably on the time scale of the average document length. These integration modes are aligned with, and create motion within, the fixed-point manifold. We see **(a)** three integration modes in four-class categorical tasks, **(b)** two integration modes in three-class categorical tasks, and two integration modes in both **(c)** three-class ordered and **(d)** five-class ordered tasks.

In particular, strongly regularizing the network can reduce the dimensionality of the resulting manifolds, while weakly regularizing can increase the dimensionality. Focusing on ℓ_2-regularization, we document this effect for the synthetic and natural datasets in Appendicies D.1 and F, respectively.

3 Results

3.1 Categorical classification yields simplex attractors

We begin by analyzing networks trained on categorical classification datasets, with natural examples including news articles (AG News dataset) and encyclopedia entries (DBPedia Ontology dataset). We find dynamics in these networks which are largely low-dimensional and governed by integration. Contrary to our initial expectations, however, the dimensionality of the network's integration manifolds are *not* simply equal to the number of classes in the dataset. For example, rather than exploring a three-dimensional cube, RNNs trained on 3-class categorical tasks exhibit a largely *two*-dimensional state space which resembles an equilateral triangle (Fig. 1a, d). As we will see, this is an example of a pattern that generalizes to larger numbers of classes.

Synthetic categorical data To study how networks perform N-class categorical classification, we introduce a toy language whose vocabulary includes $N+1$ words: a single evidence word 'evid_i' for each label i, and a neutral word 'neutral'. Synthetic phrases, generated randomly, are labeled with the class for which they contain the most evidence words (see Appendix D for more details). This is analogous to a simple mechanism which classifies documents as, e.g., "sports" or "finance" based on whether they contain more instances of the word "football" or "dollar".

The main features of the categorical networks' integration manifolds are clearly seen in the 3-class synthetic case. First, the dynamics are low-dimensional: performing PCA on the set of hidden states explored from hundreds of test phrases reveals that more than 97% of its variance is contained in the top two dimensions. Projected onto these dimensions, the set of network trajectories takes the shape of an equilateral triangle (Fig. 1a). Diving deeper into the dynamics of the trained network, we examine the deflections, or change in hidden state, $\Delta\mathbf{h}_t$, induced by each word. The deflections due to evidence words 'evid_i' align with the corresponding readout vector $\mathbf{r}_i$ at all times (Fig. 1b). Meanwhile, the deflection caused by the 'neutral' word is much smaller, and on average, nearly zero. This suggests that the RNN dynamics approximate that of a two-dimensional integrator: as the network processes each example, evidence words move its hidden state within the triangle in a manner that is approximately constant across the phrase. The location of the hidden state within the triangle encodes the integrated, relative counts of evidence for each of the three classes. Since the readouts are of approximately equal magnitude and align with the triangle's vertices, the phrase is ultimately classified by whichever vertex is closest to the final hidden state. This corresponds to the evidence word contained the most in the given phrase.

Natural categorical data Despite the simplicity of the synthetic categorical dataset, its working mechanism generalizes to networks trained on natural datasets. We focus here on the 3-class AG News dataset, with matching results for 4-class AG news and 3- and 4-class DBPedia Ontology

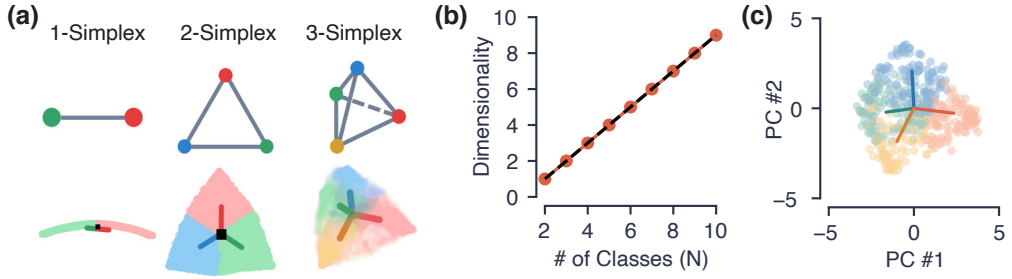

Figure 3: **Higher-dimensional simplexes in N-class categorical classification.** **(a)** Various simplex fixed-point manifolds seen in synthetic data for $N = 2, 3, 4$, colored by predicted label. **(b)** In red, the dimensionality of fixed-point manifolds from synthetic data as a function of number of classes, N. Here, dimensionality is the number of dimensions needed to capture more than 95% of the variance in the fixed-point space. Each datapoint is an average over 10 initializations. The black dotted line is the dimensionality of an $(N-1)$-simplex. **(c)** The fixed point manifold and readouts for 4-class AG news, colored by predicted label.

in Appendix E. Hidden states of these networks, as in the synthetic case, fill out an approximate equilateral triangle whose vertices once again lie parallel to the readout vectors (Fig. 1d). While these results bear a strong resemblance to their synthetic counterparts, the manifolds for natural datasets are, unsurprisingly, less symmetric.

Though the vocabulary in natural corpora is much larger than the synthetic vocabulary, the network still learns the same underlying mechanism: by suitably arranging its input Jacobian and embedding vectors, it aligns an input word's deflection in the direction that changes relative class scores appropriately (Fig. 1e). Most words behave like the synthetic word "neutral", causing little movement within the plane; certain words, however, (like "football") cause a large shift toward a particular vertex (in this case, "Sports"). Again, the perturbation is relatively uniform across the plane, indicating that the order of words does not strongly influence the network's prediction.

In both synthetic and natural cases, the two-dimensional integration mechanism is enabled by a manifold of approximate fixed points near the network trajectories, which allow the network to maintain its position in the absence of new evidence. As the position withing the plane encodes the network's integrated evidence, this maintenance is essential. In all 3-class categorical networks, we find a planar, approximately triangle-shaped manifold of fixed points which lie near the network trajectories (Fig. 1c, f); vertices of this manifold align with the readout vectors. PCA reveals the dimensionality of this manifold to be identical to that of the hidden states.

The fixed points' stability is crucial to ensure robust storage of the integrated evidence. We check for stability directly by linearizing the dynamics around each fixed point and examining the spectra of the recurrent Jacobians. Almost all of the Jacobian's eigenvalues well within the unit circle, corresponding to perturbations which decay on the timescale of a few tokens. Only two modes, which lie within the fixed-point plane, are capable of preserving information on timescales on the order of the mean document length (Fig. 2a). This linear stability analysis confirms our picture of a two-dimensional attractor manifold of fixed points; the network dynamics quickly suppress activity in dimensions outside of the fixed-point plane. Integration, i.e. motion within the fixed-point plane is enabled by two eigenmodes with long time constants.

LSA predictions Intuitively, the two-dimensional structure in this three-class classification task reflects the fact that the network tracks relative score between the three classes to make its prediction. To see this two-dimensional structure quantitatively in the dataset statistics, we apply latent semantic analysis (LSA) to the dataset, finding a low-rank approximation to the evidence vectors of all the words in the vocabulary. This analysis (Fig. 1h) shows that two modes are necessary to capture the variance, just as we observed in the RNNs. Moreover, the class vectors projected into this space (Fig. 1g) match exactly the structure observed in the RNN readouts. The network appears to pick up on the same structure in the dataset's class counts identified by LSA.

General N-class categorical networks The triangular structure seen in the 3-class networks above is an example of a general pattern: N-class categorical classification tasks result in an $(N-1)$-dimensional simplex attractor (Fig. 3a). We verify this with synthetic data consisting of up to 10

classes, analyzing the subspace of $\mathbb{R}^n$ explored by the resulting networks. More than 95% of the variance of the hidden states is contained in $N-1$ dimensions, with the subspace taking on the approximate shape of a regular $(N-1)$-simplex centered about the origin (Fig. 1b).[2] The readout vectors, which lie almost entirely within this $(N-1)$-dimensional subspace, align with the simplex's vertices. Mirroring the 3-class case, dynamics occur near a manifold of fixed points which also shaped like an $(N-1)$-simplex. This simplex geometry reflects the fact that to classify between N classes, the network must track $N-1$ scalars: the relative scores for each class.

As a natural example of this simplex, the full, 4-class AG News dataset results in networks whose trajectories explore an approximate 3-simplex, or tetrahedron. The fixed points also form a 3-dimensional tetrahedral attractor (Fig. 3c). Additional results for 4-class natural datasets, which also yield tetrahedron attractors, are shown in Appendix E.

3.2 Ordered classification yields plane attractors

Having seen networks employ simplex attractors to integrate evidence in categorical classification, we turn to ordered datasets, with Yelp and Amazon review star prediction as natural examples. Star prediction is a more finely-grained version of binary sentiment prediction that RNNs solve by integrating valence along a one-dimensional line attractor (Maheswaranathan et al., 2019). This turns out not to be the case for either 3-class or 5-class star prediction.

For a network trained on the 5-class Yelp dataset, we plot the two-dimensional projection of RNN trajectories while processing a test batch of reviews as well as the readout vectors for each class (Fig. 4d). Similar results for 3-class Yelp and Amazon networks are in Appendix E. The top two dimensions capture more than 98% of the variance in the explored states: as with categorical classification tasks, the dynamics here are largely low-dimensional. A manifold of fixed points that is also planar exists nearby (Fig. 4f). The label predicted by the network is determined almost entirely by the position within the plane. Additionally, eigenmodes of the linearized dynamics around these fixed points show two slow modes with timescales comparable to document length, separated by a clear gap from the other, much faster, modes (Fig. 2c, d). These two integration modes lie almost entirely within the fixed-point plane, while the others are nearly orthogonal to it.

These facts suggest that — in contrast to binary sentiment analysis — 5-class (and 3-class) ordered networks are *two-dimensional*, tracking two scalars associated with each token rather than simply a single sentiment score. As an initial clue to understanding what these two dimensions represent, we examine the deflections in the plane caused by particular words (Fig. 4f). These deflections span two dimensions—in contrast to a one-dimensional integrator, the word 'horrible' has a different effect than multiple instances of a weaker word like "poor." These two dimensions of the deflection vector seem to roughly correspond to a word's "sentiment" (e.g. good vs. bad) and "intensity" (strong vs. neutral). In this two-dimensional integration, a word like 'okay' is treated by the network as evidence of a neutral (e.g., 3-star) review.

Inspired by these observations, we build a synthetic ordered dataset with a word bank $\{\text{amazing}, \text{good}, \text{okay}, \text{bad}, \text{awful}, \text{neutral}\}$, in which each word now has a separate sentiment and intensity score.[3] Labels are assigned to phrases based on both its total sentiment and intensity; e.g., phrases with low intensity and sentiment scores are classified as "3 stars", while phrases with high positive sentiment and high intensity are "5 stars" (see Appendix D.2 for full details). Networks trained on this dataset correspond very well to the networks trained on Amazon and Yelp datasets (Fig. 4a-c). Dynamics are largely two-dimensional, with readout vectors fanning out in the plane from five stars to one. Deflections from individual words correspond roughly to the sentiment and intensity scores, and the underlying fixed-point manifold is two-dimensional.

More generally, the appearance of a plane attractor in both 3-class and 5-class ordered classification shows that in integration models, relationships (such as order) between classes can change the dimensionality of the network's integration manifold. These relationships cause the LSA evidence vectors for each word to lie in a low-dimensional space. As in the previous section, we can see this

[2] A 1-simplex is a line segment, a 2-simplex a triangle, a 3-simplex a tetrahedron, etc. A simplex is *regular* if it has the highest degree of symmetry (e.g. an equilateral triangle is a regular 2-simplex).

[3] Interestingly, a synthetic model that only tracks sentiment fails to match the dynamics of natural ordered data for $N > 2$. We take this as further evidence that natural ordered datasets classify based on *two*-dimensional integration. This simple model still produces surprisingly rich dynamics that we detail in Appendix D.2.

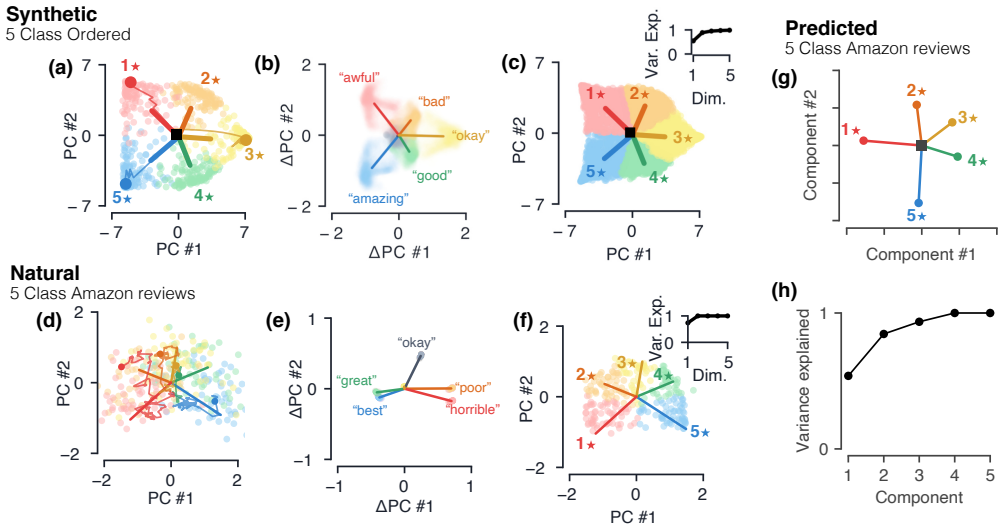

Figure 4: **Results from training networks on 5-class ordered data. (a, d)** Final hidden states of many test samples, colored by their label, with a few example hidden states trajectories shown explicitly. The initial state $\mathbf{h}_0$ is shown as a black square, and the three thick solid lines are the three readouts, colored by their respective class. **(b, e)** The hidden deflections from the three words in the vocabulary, with the average deflection of each shown as a solid line. **(c, f)** Fixed points, colored by their predicted label. The inset shows the variance explained as a function of number of PC dimensions. **(g)** The LSA score vectors projected into the top two variance dimensions. **(h)** Percentage of variance explained versus number of dimensions for LSA.

low-dimensional in the dataset statistics itself using LSA, showing that two singular values explain more than 95% of the variance (Fig. 4f). Thus, the planar structure of these networks, with dimensions tracking both (roughly) sentiment and intensity, is a consequence of correlations present in the dataset itself.

3.3 Multi-labeled classification yields independent attractors

So far, we have studied classification datasets where there is only a single label per example. This only requires networks to keep track of the *relative* evidence for each class, as the overall evidence does not affect the classification. Put another way, the softmax activation used in the final layer will normalize out the total evidence accumulated for a given example. This results in networks that, for an N-way classification task, need to integrate (or remember) at most $N-1$ quantities as we have seen above. However, this is not true in multi-label classification. Here, individual class labels are assigned independently to each example (the task involves N independent binary decisions). Networks trained on this task *do* need to keep track of the overall evidence level.

To study how this changes the geometry of integration, we trained RNNs on a multi-label classification dataset, GoEmotions (Demszky et al., 2020). Here, the labels are emotions and a particular text may be labeled with multiple emotions. We trained networks on two reduced variants of the full dataset, only keeping 2 and 3 labels. The results for 3 labels are detailed in Appendix E.5. For the 2-class version, we found that RNNs learned a 2-dimensional integration manifold where the readout vectors span a two-dimensional subspace (Fig. 5d), rather than a one-dimensional line as in binary classification. Across the fixed point manifold, there were consistently two slow eigenvalues (Fig. 5e), corresponding to the two integration modes. Similar to the previous datasets, increasing ℓ_2 regularization would (eventually) compress the dimensionality, again measured using the participation ratio (Fig. 5f).

The synthetic version of this task classifies a phrase as if it were composed of two independent sentiment analyses (detailed in Appendix D.3). This is meant to represent the presence/absence of a given emotion in a given phrase, but ignores the possibility of correlations between certain emotions. After training a network on this data, we find a low-dimensional hidden-state and fixed-point space that both take on the shape of a square (Fig. 5a, c). The deflections of words affecting independent labels act along orthogonal directions (Fig. 5b).

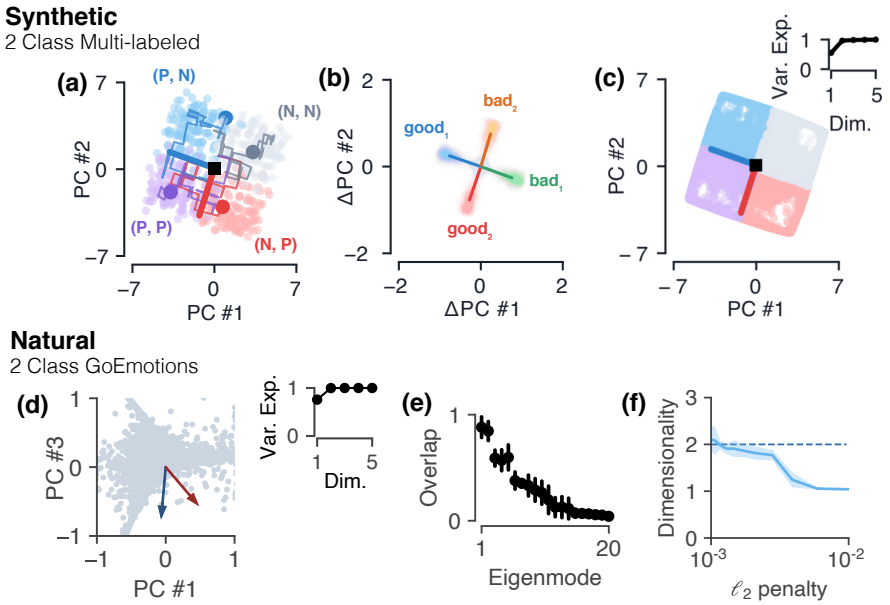

Figure 5: **Networks trained on multi-labeled synthetic (a-c) and natural (d-f) data with two labels.** **(a)** Final hidden states of a synthetic network, colored by their label, with a few example trajectories. The two thick solid lines are readout vectors. The network classifies examples using a 2D plane attractor. **(b)** The hidden deflections from the three words in the vocabulary, with the average deflection of each shown as a solid line. **(c)** Fixed points, colored by their predicted label; these form a square (cf. the triangle in Fig. 1). The inset shows the variance explained. **(d)** Fixed points (gray circles) and readout vectors for an RNN trained on the GoEmotions dataset. Inset shows that two dimensions are needed to explain the variance seen in the fixed point manifold. **(e)** At these fixed points, there are two eigenvalues with slow time constants, whose eigenvectors overlap with the 2D subspace; hence the system has a 2D plane attractor. **(f)** Analysis of the dimensionality (measured using the participation ratio) of networks trained with different amounts of ℓ_2 regularization. For a broad range of values, the networks learn a 2D attractor; if they are over-regularized, the dimensionality drops.

These results suggest that integration manifolds are also found in RNNs trained on multi-labeled classification datasets. Moreover, the geometry of the corresponding fixed points and readouts is different from the exclusive case; instead of an $(N-1)$-dimensional simplex we get an N-dimensional hypercube. Again, this makes intuitive sense given that the networks must keep track of N independent quantities in order to solve these tasks.

4 Discussion

In this work we have studied text classification RNNs using dynamical systems analysis. We found integration via attractor manifolds to underlie these tasks, and showed how the dimension and geometry of the manifolds were determined by statistics of the training dataset. As specific examples, we see $(N-1)$-dimensional simplexes in N-class categorical classification where the network needs to track relative class scores; 2-dimensional attractors in ordered classification, reflecting the need to track sentiment and intensity; and N-dimensional hypercubes in N-class multi-label classification.

We hope this line of analysis—using dynamical systems tools to understand RNNs—builds toward a deeper understanding of how neural language models perform more involved tasks in NLP, including language modeling or translation. These tasks cannot be solved by a pure integration mechanism, but it is plausible that integration serves as a useful computational primitive in RNNs more generally, similar to how line attractor dynamics serve as a computational primitive on top of which contextual processing occurs (Maheswaranathan & Sussillo, 2020).

REFERENCES

Yonatan Belinkov and James R. Glass. Analysis methods in neural language processing: A survey. *CoRR*, abs/1812.08951, 2018.

Francesco Camastra and Alessandro Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on pattern analysis and machine intelligence*, 24(10): 1404–1407, 2002.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

Jasmine Collins, Jascha Sohl-Dickstein, and David Sussillo. Capacity and trainability in recurrent neural networks, 2016.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41 (6):391–407, 1990.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. Goemotions: A dataset of fine-grained emotions. pp. 4040–4054, 01 2020. doi: 10.18653/v1/2020.acl-main.372.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, pp. 777–784, 2005.

Niru Maheswaranathan and David Sussillo. How recurrent networks implement contextual processing in sentiment analysis. *arXiv preprint arXiv:2004.08013*, 2020.

Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. In *Advances in Neural Information Processing Systems 32*, pp. 15696–15705. Curran Associates, Inc., 2019.

Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

Itamar Procaccia and Peter Grassberger. Measuring the strangeness of strange attractors. *Physica. D*, 9(1-2):189–208, 1983.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444, 2017.

Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 649–657. Curran Associates, Inc., 2015.