Understanding Adaptive, Multiscale Temporal Integration In Deep Speech Recognition Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Natural signals such as speech are hierarchically structured across many different timescales, spanning tens (e.g., phonemes) to hundreds (e.g., words) of milliseconds, each of which is highly variable and context-dependent. While deep neural networks excel at recognizing complex patterns from natural signals such as speech, little is known about how deep speech recognition systems flexibly integrate across multiple timescales. Here, we show how a recently developed method for studying temporal integration in biological neural systems – the temporal context invariance (TCI) paradigm – can be used to understand temporal integration in deep neural networks. The method is simple: we measure responses to a large number of stimulus segments presented in two different contexts (i.e., surrounded by different segments). We then measure how long the segments need to be for the response to be mostly invariant to surrounding context stimuli. We applied this approach to understand how the widely used DeepSpeech2 model integrates temporal information in speech. We find that nearly all of the model units have a meaningful integration window, within which stimuli substantially alter the response and outside of which stimuli have little effect. We show that training causes these integration windows to diversify across layers, with higher-order layers integrating across substantially longer timescales. Moreover, by measuring integration windows for time-stretched/compressed stimuli, we demonstrate that there is a transition point midway through the trained network where integration windows become substantially more adaptive in the sense of becoming yoked to duration of stimulus structures (e.g., words) rather than absolute time. Our study reveals how the popular DeepSpeech2 model learns to flexibly integrate across the complex, multiscale structure that defines speech signals and introduces a simple toolbox for understanding temporal integration in black-box machine learning models.

1 Introduction

A central challenge of representing natural stimuli, such as speech and music, is that they are structured across many different timescales from milliseconds to seconds (Chomsky and Halle, 1968; Lerdahl and Jackendoff, 1985; Hickok and Poeppel, 2007). Speech, for instance, is composed of phonemes, syllables, and words spanning tens to hundreds of milliseconds, and the duration of these structures varies widely due to a myriad of factors (e.g., talker, speaking rate, prosody, etc.) (Figure 1). Intelligent systems, whether biological or artificial, must learn to flexibly integrate across these different timescales to derive meaning from natural signals. Deep neural networks (DNNs) have proven remarkably effective at learning complex temporal structures from natural signals, such as speech, and the representations learned by these networks show non-trivial similarities to those present in the brain (Kell et al., 2018). However, relatively little is known about how deep speech recognition systems flexibly integrate across multiple timescales. Here, we address this gap by using

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

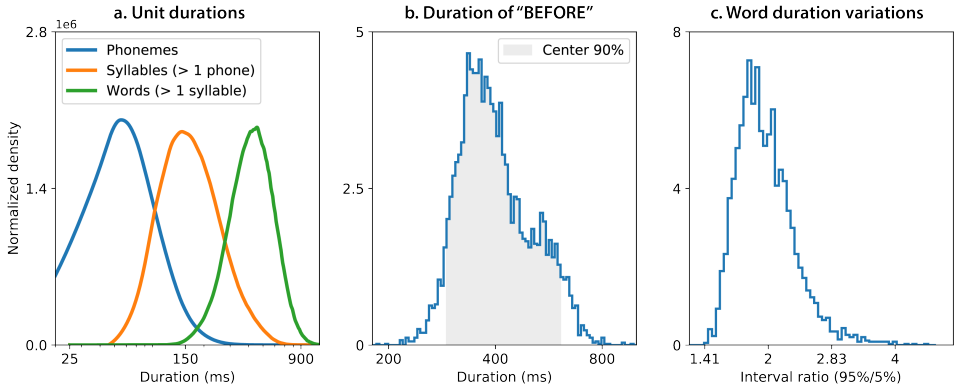

Figure 1: Duration statistics of speech structures in LibriSpeech corpus. (a) Distribution of durations for all phonemes, multi-phoneme syllables, and multi-syllable words. (b) Duration distribution for all utterances of the word "before" in the corpus. The central 90% interval is marked in gray. (c) Distribution of interval ratios across all words, measured by dividing the 95th percentile duration by the 5th percentile duration.

a recently developed method for studying temporal integration in biological neural systems – the temporal context invariance (TCI) paradigm (Norman-Haignere et al., 2020) – to probe how a widely used deep speech recognition system (DeepSpeech2) learns to integrate across multiple timescales.

We study temporal integration by measuring how training a DNN alters the integration window of the network's units. Integration windows are defined as the time window when stimuli alter the neural response and outside of which stimuli have little effect (Theunissen and Miller, 1995; Norman-Haignere et al., 2020). Because this definition is simple and general, integration windows can be used to measure and compare the analysis timescale of responses from different models, stimuli, and modalities. We have recently designed a simple method to estimate the integration window of an arbitrary response by presenting segments of natural stimuli (here speech) in two different random orders such that each segment occurs in two different contexts (i.e., is surrounded by different stimuli) (Figure 2a). If the integration window is less than the segment duration, there will be a moment when the response is the same across the two contexts. We can thus vary the segment duration in order to find the smallest segment duration which produces a context-invariant response. The random ordering of segments makes it possible to use a simple correlation measure to quantify at each moment in time the degree to which the response is context invariant (Figure 2b), as described in detail below. The goal of this paper is to introduce this paradigm to the machine learning community and show how it can be used to understand temporal integration in black-box machine learning models.

Our paper makes the following contributions:

1. We show how to estimate the integration window of a unit by measuring the degree to which its response depends upon a set of target stimuli vs. surrounding context stimuli. This paradigm provides a simple way to track the representation of an arbitrary set of stimuli as that representation evolves and decays over time.

2. We show that the units in all layers of the DeepSpeech2 model have a meaningful integration window within which stimuli substantially alter the response and outside of which stimuli have little effect.

3. We show that training causes the integration windows of different layers of the DeepSpeech2 model to specialize for different timescales, with substantially longer integration windows in higher-order layers of the network.

4. By measuring integration windows for time-stretched/compressed speech, we demonstrate that there is a transition point, midway through the network, where integration windows become substantially more dynamic, expanding or shrinking depending on the timescale of the features/structures in a stimulus.

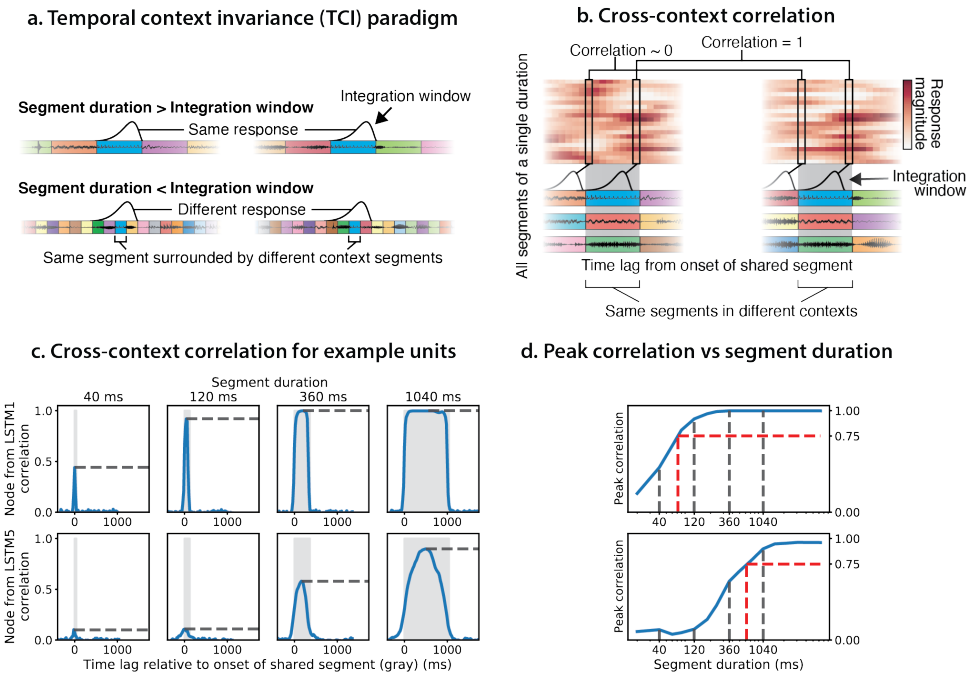

Figure 2: Temporal context invariance (TCI) paradigm. (a) Schematic of the paradigm used to measure integration windows. Segments of natural stimuli (here speech) are presented using two different random orderings. As a consequence, the same segment is surrounded by different context segments. If the segment duration is longer than the integration window (top panel), there will be a moment when the window is fully contained within each segment. As a consequence, the response at that moment will by definition be unaffected by the surrounding context segments. If the segment duration is shorter than the integration window (bottom panel), the integration window will always overlap the surrounding context segments, and they can therefore alter the response. The TCI paradigm estimates the minimum segment duration needed to achieve a context invariant response. This figure plots waveforms for an example sequence of segments that share the same central segment. Segment boundaries are demarcated by colored boxes. The hypothesized integration window is plotted above each sequence at the moment when it best overlaps the shared segment. (b) Schematic of the analysis used to assess context invariance for a single unit and segment duration. The unit response timecourses to all segments of a given duration are organized as a matrix: the segment-aligned response (SAR) matrix. Each row of the SAR matrix contains the response timecourse to a different segment, aligned to segment onset. We compute two versions of the SAR matrix, one for each context. The central segments are the same across SAR matrices from different contexts, but the surrounding segments differ. The gray region shows the time window when the shared segments are present. To determine if the response is context invariant, we correlate corresponding columns across SAR matrices from different contexts (the "cross-context correlation"). This analysis is schematized by the linked columnar boxes, which show the columns being correlated. For each box, we plot a schematic of the integration window at that moment time. At the start of the shared segments (first box pair), the integration window for a causal response will fall on the preceding contexts segments, which are random across contexts and so the correlation should be approximately zero. As the lag relative to segment onset increases, the integration will begin to overlap the shared central segment. If the integration window is less than the segment duration, there will be a lag when the response is the same across contexts and the correlation will be 1 (second box pair). For non-causal responses (not depicted), the integration window can overlap the shared segment before its onset. (c) The cross-context correlation is shown for two example units from different layers (LSTM1 and LSTM5). Each plot shows a different segment duration. The gray region shows the time interval when the shared segment was present (as in panel b). Black dashed lines show the peak of the cross-context correlation across all lags. (d) The peak correlation for all tested segment durations. Black dashed lines indicate the segment durations shown in panel c. The red dashed line indicates the segment duration needed to achieve a peak correlation value of 0.75, which is how integration windows were defined for each unit.

2 Relationship with prior work

Understanding latent representations of deep neural networks is a highly active topic of research. A variety of feature visualization techniques have been developed to understand image-trained networks (Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2015; Yosinski et al., 2015) with less work focused on understanding speech-trained networks (Huang et al., 2015; Krug and Stober, 2019; Beguš and Zhou, 2021). Probing tasks have been used to examine the representation of particular structures, such as phonemes, across different layers of speech recognition networks (Belinkov and Glass, 2017; Nagamine et al., 2015; Nagamine and Mesgarani, 2017), and a recent study examined the geometry of how invariant feature representations evolve over time (Stephenson et al., 2020). However, to our knowledge, little is known about how speech-trained networks learn to integrate across time, despite the centrality of temporal integration to the task of speech recognition.

A large body of work has focused on understanding the flow of temporal information in recurrent neural networks (Hochreiter and Schmidhuber, 1997; Pascanu et al., 2012; Saxe et al., 2013; Sussillo and Barak, 2013). Theoretical arguments suggest that gating is important to the ability of recurrent neural networks to adaptively integrate across different timescales (Tallec and Ollivier, 2018) with the bias term strongly influencing the scale of temporal integration (Mahto et al., 2020). However, there is no precise mapping between the bias term and the integration window of the unit for a complex stimulus like speech. Convolutional layers have a well-defined upper bound on their integration window that is determined by the size of the filter (and those of preceding layers), but training can dramatically change the effective integration window of the learned filters. Thus, there is a need for an experimental method that can be used to estimate the integration window of a nonlinear response to a complex stimulus like speech and can thus be used to compare integration windows across units, layers, and architectures.

Neuroscience has produced a wealth of experimental methods for studying sensory timescales. Scrambling methods have been used to examined selectivity for intact temporal structure at particular timescales (Hasson et al., 2008; Farbood et al., 2015; Overath et al., 2015). Many studies have examined the rate at which modulations in a stimulus are tracked by the neural response (Joris et al., 2004; Wang et al., 2008; Teng and Poeppel, 2020), sometimes referred to as the encoding window (Theunissen and Miller, 1995; Chen et al., 2012). A variety of model-based approaches have been used to estimate neural integration windows (Mahto et al., 2020), although these models often make strong assumptions about the type of response being measured such as assuming a quasi-linear mapping between a spectrogram and the response (Meyer et al., 2017; Khatami and Escabí, 2020). Chien and Honey (2020) recently introduced an experimental method that measures the build-up of context invariance over time after a stimulus change, and they used this method to analyze a neural language model (Chien and Honey, 2020). However, this method does not directly measure the time window in the stimulus that is needed to achieve an invariant response, which requires measuring responses to segments of different durations as is done with our TCI paradigm. Moreover, speech recognition presents many unique challenges that are not present in language modeling. For example, a key contribution of our paper is showing there is a transition from static, time-based integration to adaptive, structure-driven integration in a trained speech recognition model, a distinction that is not meaningful in a language model where the input is inherently structure-based (i.e., words).

3 Temporal context invariance

3.1 Stimuli

The TCI paradigm involves presenting a sequence of segments in two different random orderings such that each segment occurs in two different contexts (Norman-Haignere et al., 2020) (Figure 2a). We tested 16 segment durations, varying between 20 ms and 6.28 seconds in pseudo-logarithmic steps. From these segments, we created random 45-second sequences, which were input to the network. Each sequence contained segments of a single duration, and thus the number of segments was inversely proportional to the segment duration.

Segments were excerpted from the LibriSpeech corpus (clean set). We chose segments that were composed of words spoken at a moderate rate such that stretching or compressing the word would not place the word outside of its typical duration. To this end, we first selected all words spoken at least 100 times in the corpus. We then identified all utterances of these words within the central 10% of the

duration distribution for each word (45-55th percentile). From this set, we randomly selected 10,000 utterances from with a maximum of 50 utterances per word. For each word utterance, we excerpted 16 segments, one per segment duration, with the middle of the word at the center of the segment (if the segment duration was shorter than the word, we excerpted just the middle portion of the word).

Figure 1a shows the distribution of durations for phonemes, multi-phone syllables (computed using (Gorman, 2019)), and multi-syllable words in the LibriSpeech corpus (clean set), which shows the wide range of timescales these structures span (median durations are 70, 200, and 470 ms, respectively). Figure 1b shows the duration distribution for a single example word ("before") with the central 90% interval indicated (5-95th percentile), along with a histogram of the 90% interval ratio (95%/5%) across all words (Figure 1c). This plot shows that the duration of individual words varies widely with a median interval ratio of 1.92. Time alignments for phonemes and words of the LibriSpeech dataset were extracted from (Lugosch et al., 2019), which used the Montreal Forced Aligner to align the transcripts to sounds (McAuliffe et al., 2017).

3.2 DeepSpeech2 model

We applied our TCI paradigm to measure integration windows from a pre-trained DeepSpeech2 model (Amodei et al., 2016), which was trained on LibriSpeech using pytorch lightning (Naren, 2021). The DeepSpeech2 model consists of two convolutional layers (Conv1/2), 5 LSTM layers (LSTM1/2/3/4/5), and a softmax readout that is trained to predict graphemes using a CTC loss (Graves et al., 2006). We analyzed logarithmically transformed probabilities for the softmax readout (to avoid the high sparsity of the probability measures). We compared results from this pre-trained model with a randomly initialized model to assess effects of training.

3.3 Cross-context correlation

Our key analysis involves correlating the response of a unit to different stimulus segments across two different contexts (Figure 2a), an analysis we refer to as the "cross-context correlation". The cross-context correlation is applied separately to responses from different segment durations. First, we organize the response timecourses to all segments of a given duration as a matrix, which we refer to as the segment-aligned response (SAR) matrix (Figure 2b). Each row of the SAR matrix contains the response timecourse surrounding a single segment, aligned to segment onset. Different rows thus correspond to different segments and different columns correspond to different lags relative to segment onset. We compute two versions of the SAR matrix using the two different contexts for each segment. The central segment is the same across contexts, but the surrounding segments are different.

The goal of our analysis is to assess if there is a lag when the response is the same across contexts. We instantiate this idea by correlating corresponding columns across SAR matrices from different contexts. At segment onset (lag=0), the cross-context correlation should be near zero for a causal response, since the integration window must overlap the preceding segments, which are random across contexts. As time progresses, the integration window will start to overlap the shared segment, and the cross-context correlation should increase. Critically, if the integration window is less than the segment duration, there will be a lag where the integration window is fully contained within the shared segment, and the response will thus be the same across contexts, yielding a correlation of 1. For a non-causal response, we expect the same pattern, but the build-up of the cross-context correlation can begin before the onset of the shared segment.

We plot the cross-context correlation for segments of increasing duration for two example units from different layers of the DeepSpeech2 network (LSTM1 and LSTM5) (Figure 2c). For both units, the cross-context correlation started at zero, rose near the onset of the shared segment, and fell back to zero after the offset of the shared segment. The pre-onset rise of the correlation is due to the non-causal nature of the bidirectional LSTM layers. Critically, the peak of the cross-context correlation varied across segment durations and units. For the unit from the LSTM1 layer (top panel), the peak of the cross-context correlation was close to 1 for segment durations as short as 240 milliseconds. For longer segment durations, the cross-context correlation remained high for an extended duration indicating an extended time period in the stimulus during which the integration window of the unit remained within the shared segment and thus largely context-invariant. By comparison, the unit from the LSTM5 layer required much longer segment durations for the cross-context correlation to approach 1, and the build-up/fall-off of the cross-context correlation with lag was slower, implying a

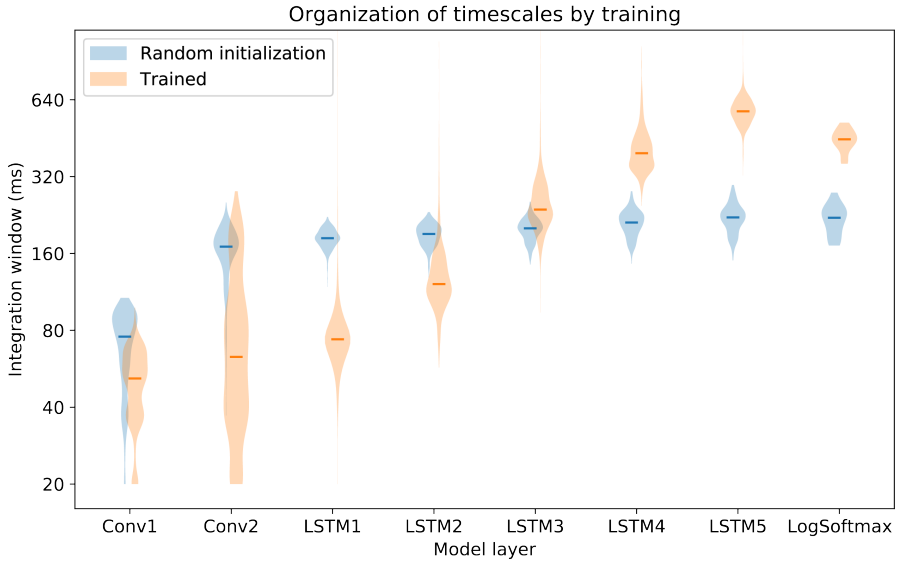

Figure 3: Effect of training on overall network integration windows. Violin plots show the distribution of integration windows across all units from each layer for a trained (orange) and randomly initialized model (blue). Horizontal lines indicate distribution medians..

substantially longer integration window. The general patterns described in these two example units were observed in virtually all units (Supplement Figure 1 shows several more units).

We quantified the integration window of each unit by measuring the peak of the cross-context correlation across lag as a function of the segment duration (Figure 2d). The units showed a clear pattern where the peak correlation rose steeply as the segment duration was increased, and then gradually approached an asymptotic value. This asymptotic value was high for nearly all units (Supplement Figure 2), indicating a mostly context-invariant response. We defined the integration window of each unit as the segment duration needed to achieve a threshold correlation value, which we set to 0.75 (estimated by linearly interpolating the correlation vs. segment duration curves shown in Figure 2d). Results were robust to the choice of threshold, though higher thresholds of course result in longer integration windows.

3.4 Effect of training

Figure 3 plots the distribution of integration windows for the units in each layer for both trained and random models. The integration windows from the random model varied little with layer, particularly amongst the five LSTM layers (median integration windows varied between 184 and 221 ms across the LSTM layers). In contrast, for the trained model we observed large differences between layers (median integration windows varied between 74 and 577 ms across the LSTM layers), yielding a highly significant difference between trained and untrained models for all layers (Wilcoxon rank-sum test, $p < 1e-7$). In particular, for the trained model we found that integration windows were substantially larger in higher layers of the network, with a nearly 11-fold increase between Conv1 (52 ms) and LSTM5 (577 ms). This pattern qualitatively mimics the hierarchical nature of temporal integration in biological auditory systems (Khatami and Escabí, 2020; Norman-Haignere et al., 2020).

4 Time- vs. structure-based integration

The duration of speech structures such as phonemes and words is highly variable (Figure 1). Thus, a natural question is whether the time windows measured from the network reflect a static time window or instead reflect the duration of the features/structures to which a unit responds. A simple way to address this question is to time-compress or time-stretch the stimulus such that all stimulus structures

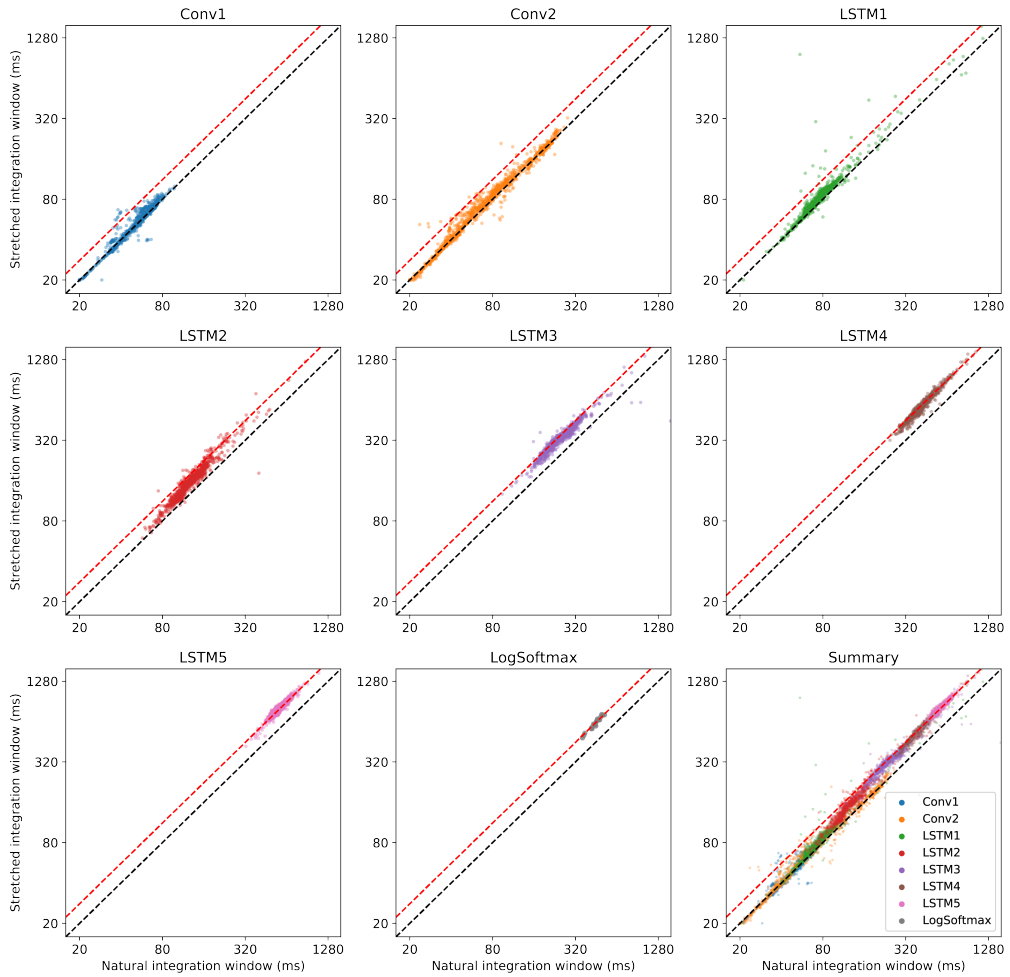

Figure 4: Integration windows for natural and time-stretched speech. For each layer, a scatter plot shows the integration window for stretched vs. natural speech. Dashed lines indicate what one would expect for a static, time-based integration window (black) as well as a fully adaptive, structure-driven window (red). The final panel summarizes the results by jointly plotting all layers on the same graph (colored by layer), which illustrates the progression from short, time-based windows to long, structure-driven windows.

get mapped to a new timescale. If the integration window is time-based, then the integration window should remain fixed, while if the integration window is structure-driven, it should appear to shift in proportion to the degree of stretching/compression. We note that natural timescale variation in speech does not involve uniform stretching/compression (Covell et al., 1998), but this manipulation is experimentally useful because it effectively changes the timescale of all stimulus structures, and thus we would expect a shift in the integration windows regardless of the particular structures a unit is tuned for.

Stretching and compression were implemented by interpolating the spectrogram computed by the DeepSpeech2 model. The stretched and compressed stimuli differed from the original stimuli by a factor of $\sqrt{2}$, which is well within the range of typical variation in word durations (Figure 1c). To assess performance for stretched/compressed speech, we used a simple measure of the word error rate (WER): whether the target word was present in the transcribed text (using 6.28 second segments centered on the target). WER was relatively low in all cases, but was best for natural speech (0.055) and worst for compressed speech (0.096), with stretched speech (0.075) only slightly worse than natural speech.

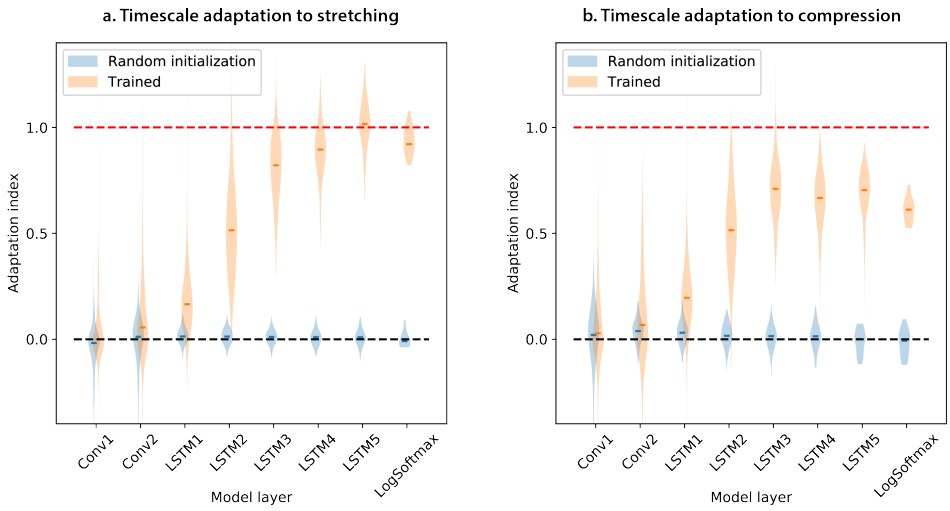

Figure 5: Effect of training on timescale adaptation for stretched and compressed speech. For each unit, we computed an adaptation index, reflecting the degree to which its integration window shifted with the magnitude of stretching/compression. A value of 0 indicates a static window, while a value of 1 indicates a fully adaptive, structure-driven window. Violin plots show the distribution of adaptation indices across all units from each layer. Results are plotted for trained (orange violins) and random models (blue violins) measured with either stretching (left panel) or compression (right panel). The violin plots show the center 99% of the distribution, thus discarding extreme outliers. Horizontal lines indicate distribution medians.

Figure 4 shows a scatter plot of integration windows measured for stretched vs. natural speech for all model layers, along with dashed lines indicating what one would expect from a fixed, time-based window (black) or an adaptive, structure-driven window (red). For each unit, we computed an adaptation index by measuring the ratio between integration windows for stretched/compressed vs. natural speech, normalized by the magnitude of stretching/compression ($\sqrt{2}$) (equation 1). Thus, an index of 0 indicates a static, time-based window, while a value of 1 indicates a fully adaptive, structure-driven window. Figure 5 plots this adaptation index across all layers for stretched (top panel) and compressed speech (bottom panel) from both trained (orange violins) and random (blue violins) networks.

$$adaptation\ index = \begin{cases} \frac{stretched - natural}{\sqrt{2} \times natural - natural} & \text{for stretched} \\ \\ \frac{natural - compressed}{natural - natural / \sqrt{2}} & \text{for compressed} \end{cases} \tag{1}$$

Integration windows from the random model were very similar for stretched and natural speech, indicating a static, time-based window for all layers. In contrast, for the trained model, we found there was a transition point between layers LSTM1 and LSTM3, where integration windows shifted from mostly time-based (median adaptation index of 0.17/0.20 in LSTM1 for stretched/compressed speech) to mostly structure-driven (median adaptation index of 0.82/0.71 in LSTM3 for stretched/compressed speech). The magnitude of adaptation was slightly larger for stretched vs. compressed speech in the later layers of the network (one-tailed Wilcoxon rank-sum test; $p > 0.3$ for Conv1 to LSTM2, $p < 1e - 7$ for LSTM3 to LogSoftmax), which suggests that the network was better able to adapt its integration window to slowed speech, consistent with its better performance for stretched (0.075 WER) vs. compressed speech (0.096 WER). These findings suggest that there are two modes of temporal integration: an initial time-based integration stage followed by a subsequent structure-based integration stage, each of which consists of multiple processing layers.

5 Conclusion

We have described a method for understanding how black-box machine learning models learn to integrate temporal information in complex natural signals like speech, inspired by recent work in studying temporal integration in biological neural systems (Norman-Haignere et al., 2020). We present segments of stimuli in different contexts by surrounding each segment with different context segments, and we determine the smallest segment duration outside of which responses are largely invariant to the surrounding context stimuli. We find that virtually all units in the popular DeepSpeech2 model have a meaningful integration window, inside of which stimuli substantially alter the response and outside of which stimuli have little effect. We show that training causes the integration windows from different network layers to specialize for different timescales with longer integration windows in higher-order layers. Moreover, we find that integration windows at later network layers become highly adaptive, yoked more to the duration of stimulus structures than absolute time. These findings provide a picture of how DeepSpeech2 learns to integrate temporal information in speech: first integrating across short, fixed time windows at early layers and then integrating flexibly across long timescales in order to detect time-independent structures.

We emphasize that our core method is simple and general. Nothing about our approach is specific to speech or the particular models tested. We thus believe that the approach described in this paper should be widely applicable to the understanding how intelligent systems integrate across the complex temporal structures that compose natural signals.

6 Limitations and future directions

The cross-context correlation effectively measures the overall integration window across a large number of segments. In practice, the integration window might vary from segment-to-segment particularly for complex nonlinear models. One way to test this possibility is to compare integration windows from two sets of segments, as was done here for stretched/compressed speech vs. natural speech.

The uniform stretching/compression used to assess adaptation is unnatural and thus it is possible that we might observe greater adaptation with more natural forms of non-uniform stretching/compression. Uniform stretching/compression is experimentally useful because it shifts the timescale of all stimulus structures, and thus should produce a shift in integration windows regardless of the particular stimulus structures a unit represents.

Here, we focused on analyzing DeepSpeech2, one of the most popular speech recognition systems. Future work could explore how our key findings (hierarchical adaptive integration) depends upon network architecture (e.g. convolution vs. recurrence), task (e.g. speech vs. speaker recognition) and stimulus type (e.g. speech vs. music). The TCI paradigm provides a simple and general tool to understand temporal integration across a wide range of models, tasks, and stimuli.

7 Broader impacts

Our findings raise questions about the extent to which biological and artificial systems use similar mechanisms to integrate across time. The hierarchy of temporal integration windows learned by DeepSpeech2 qualitatively mimics the hierarchy of integration windows observed in biological auditory systems (Khatami and Escabí, 2020; Norman-Haignere et al., 2020). A key long-term goal of this work is to build computational models that can mimic the mechanisms of temporal integration in the brain. Building accurate computational models of brain responses has many potential benefits, including making it possible to better understand how damage to the peripheral auditory systems (e.g., the cochlea or auditory nerve) effects downstream neural responses that underlie perception and behavior. We note that neural network training consumes large amounts of energy which could have a negative impact on the environment.

References

Amodei, D., R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun,

P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. *International conference on machine learning*, 173–182.

Beguš, G. and A. Zhou (2021). Interpreting intermediate convolutional layers of cnns trained on raw speech. *arXiv:210409489*.

Belinkov, Y. and J. Glass (2017). Analyzing hidden representations in end-to-end automatic speech recognition systems. *arXiv:170904482*.

Chen, C., H. L. Read, and M. A. Escabí (2012). Precise feature based time scales and frequency decorrelation lead to a sparse auditory code. *Journal of Neuroscience*, 32:8454–8468.

Chien, H.-Y. S. and C. J. Honey (2020). Constructing and forgetting temporal context in the human cerebral cortex. *Neuron*.

Chomsky, N. and M. Halle (1968). *The sound pattern of English*. Harper & Row.

Covell, M., M. Withgott, and M. Slaney (1998). Mach1: Nonuniform time-scale modification of speech. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, 349–352.

Farbood, M. M., D. J. Heeger, G. Marcus, U. Hasson, and Y. Lerner (2015). The neural processing of hierarchical structure in music and speech at different timescales. *Frontiers in neuroscience*, 9:157.

Gorman, K. (2019). Syllabify python module for syllabifying arpabet transcriptions. [Github repository] `https://github.com/kylebgorman/syllabify`.

Graves, A., S. Fernández, F. Gomez, and J. Schmidhuber (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *Proceedings of the 23rd international conference on Machine learning*, 369–376.

Hasson, U., E. Yang, I. Vallines, D. J. Heeger, and N. Rubin (2008). A hierarchy of temporal receptive windows in human cortex. *Journal of Neuroscience*, 28:2539–2550.

Hickok, G. and D. Poeppel (2007). The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8:393–402.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation*, 9:1735–1780.

Huang, J.-T., J. Li, and Y. Gong (2015). An analysis of convolutional neural networks for speech recognition. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4989–4993.

Joris, P. X., C. E. Schreiner, and A. Rees (2004). Neural processing of amplitude-modulated sounds. *Physiological reviews*, 84:541–577.

Kell, A. J. E., D. L. K. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*.

Khatami, F. and M. A. Escabí (2020). Spiking network optimized for word recognition in noise predicts auditory system hierarchy. *PLOS Computational Biology*.

Krug, A. and S. Stober (2019). Visualizing deep neural networks for speech recognition with learned topographic filter maps. *arXiv:191204067*, 6124–6128.

Lerdahl, F. and R. S. Jackendoff (1985). *A generative theory of tonal music.* MIT press.

Lugosch, L., M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio (2019). Speech model pre-training for end-to-end spoken language understanding. *INTERSPEECH-2019*.

Mahendran, A. and A. Vedaldi (2015). Understanding deep image representations by inverting them. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5188–5196.

Mahto, S., V. A. Vo, J. S. Turek, and A. Huth (2020). Multi-timescale representation learning in lstm language models. *arXiv:200912727*.

McAuliffe, M., M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger (2017). Montreal forced aligner: trainable text-speech alignment using kaldi. *INTERSPEECH-2017*.

Meyer, A. F., R. S. Williamson, J. F. Linden, and M. Sahani (2017). Models of neuronal stimulus-response functions: elaboration, estimation, and evaluation. *Frontiers in systems neuroscience*, 10:109.

Nagamine, T. and N. Mesgarani (2017). Understanding the representation and computation of multilayer perceptrons: A case study in speech recognition. *Proceedings of the 34th International Conference on Machine Learning 70*, 2564–2573.

Nagamine, T., M. L. Seltzer, and N. Mesgarani (2015). Exploring how deep neural networks form phonemic categories. *INTERSPEECH-2015*, 1912–1916.

Naren, S. (2021). Deepspeech2 implementation and pre-trained models (release v3). [Github repository] `https://github.com/SeanNaren/deepspeech.pytorch`.

Norman-Haignere, S. V., L. K. Long, O. Devinsky, W. Doyle, I. Irobunda, E. M. Merricks, N. A. Feldstein, G. M. McKhann, C. A. Schevon, A. Flinker, and N. Mesgarani (2020). Multiscale integration organizes hierarchical computation in human auditory cortex. *bioRxiv:2020.09.30.321687*.

Overath, T., J. H. McDermott, J. M. Zarate, and D. Poeppel (2015). The cortical analysis of speech-specific temporal structure revealed by responses to sound quilts. *Nature neuroscience*, 18:903–911.

Pascanu, R., T. Mikolov, and Y. Bengio (2012). Understanding the exploding gradient problem. *CoRR*.

Saxe, A. M., J. L. McClelland, and S. Ganguli (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:13126120*.

Stephenson, C., J. Feather, S. Padhy, O. Elibol, H. Tang, J. McDermott, and S. Chung (2020). Untangling in invariant speech recognition. *arXiv:200301787*.

Sussillo, D. and O. Barak (2013). Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25:626–649.

Tallec, C. and Y. Ollivier (2018). Can recurrent neural networks warp time? *arXiv:180411188*.

Teng, X. and D. Poeppel (2020). Theta and gamma bands encode acoustic dynamics over wide-ranging timescales. *Cerebral Cortex*, 30:2600–2614.

Theunissen, F. and J. P. Miller (1995). Temporal encoding in nervous systems: A rigorous definition. *Journal of Computational Neuroscience*, 2:149–162.

Wang, X., T. Lu, D. Bendor, and E. Bartlett (2008). Neural coding of temporal information in auditory thalamus and cortex. *Neuroscience*, 154:294–303.

Yosinski, J., J. Clune, A. Nguyen, T. Fuchs, and H. Lipson (2015). Understanding neural networks through deep visualization. *arXiv:150606579*.

Zeiler, M. D. and R. Fergus (2014). Visualizing and understanding convolutional networks. *European conference on computer vision*, 818–833.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Described in the limitations section.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The speech dataset and pre-trained models are publicly available and cited. Code implementing our cross-context correlation analysis will be made available prior to publication in NeurIPS.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] We used publicly available pretrained models.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] We plot the distribution of values using violin plots.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] We used pre-trained models and so the total amount of computation was minimal. We used NVIDIA Titan XP GPUs for model inference.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]