Examining the performance of different contextual representations in a canonical language model

Brandon G. Jacques [1] Per B. Sederberg [1*]

Abstract

Driven by advances in computer engineering, model architectures, and training methods, the field of Natural Language Processing (NLP) has reached new heights of performance. Currently utilizing short-term buffers to represent the context in which a word is experienced, these models have lagged behind recent developments in the understanding of human memory. Finite representations of context ignore the fact that the temporal scale in which words are predictive of each other can go up to hundreds of words apart(H. W. Lin & Tegmark, 2017). In this paper, we leverage recent developments in the understanding of memory to augment the performance of a canonical NLP model with a compressed representation of context that contains many time-scales of information. We show that the Timing from Inverse Laplace Transform (TILT) representation, a neurally plausible way of compressing history utilizing leaky integrators, can function as a drop-in replacement for a buffer representation in a canonical language model to increase performance without adding computational complexity or increasing the size of the overall model.

Affiliations

[1] Department of Psychology, University of Virginia, USA

[*] Corresponding Author, Per B. Sederberg, 434-924-5725 (phone), pbs5u@virginia.edu (email)

Keywords

- Artificial Neural Networks
- Statistical Language Modeling
- Compressed Memory
- Neurally Plausible Representation

Introduction

Combining the fields of ML, statistics, linguistics, and others, Natural Language Processing (NLP) is the study of how machines can be used to process large

amounts of naturalistic language data. The goals of each NLP model differ wildly, from automatically translating text from one language to another (P.-C. Chang, Galley, & Manning, 2008; Luong, Pham, & Manning, 2015), to autocompleting a text message based on one's own typing habits (Ghosh & Kristensson, 2017). For these predictive models, neural networks are trained to learn the joint probability function of a word occurring within a given context, or last few words. One problem with these models is that the representation of context they use is typically finite and small, which means the model will only use a few of the most recent words when making its predictions. However, the temporal scale in which words are predictive and informative of each other is unbounded for human language (H. W. Lin & Tegmark, 2017; Voss & Clarke, 1975; Wagenmakers, Farrell, & Ratcliff, 2004). From word to word, paragraph to paragraph, both long and short range temporal correlations between words help to inform an observer of the context in which each word has been experienced.

Today's state-of-the-art models tend to either ignore long-term mutual information between words by utilizing a short-term buffer representation of context (Devlin, Chang, Lee, & Toutanova, 2019), or learn exactly how long some words should be kept around in an internal representation of context (Kuncoro et al., 2018; Vaswani et al., 2017). The former can be problematic because it forces the model to miss out on learning from words further into the past than their context is able to represent. The latter is problematic because it requires a Recurrent Neural Network (RNN), and these networks are subject to a number of issues including exploding/vanishing gradients, hardware incompatibilities due to a large number of parameters, as well as memory constraints (Y. Bengio, Simard, & Frasconi, 1994; Pascanu, Mikolov, & Bengio, 2012). It is also the case that more modern RNNs like the Long-Short Term Memory (LSTM) layers take significant amounts of resources to learn the scale in which information should be kept around and, once learned, can only operate on those scales. We should, however, be able to leverage recent developments in our understanding of human cognition to use a neurally inspired representation of context that represents information across both short and long time-scales, and avoids the pitfalls of RNNs.

Memory theorists have been trying to understand the mechanisms behind human memory for centuries. Until somewhat recently, a finite, first-in-first-out (FIFO) buffer was a sufficient short term memory representation in these models to capture a wide range of episodic memory phenomena (Atkinson & Shiffrin, 1968), which is likely why it was adopted by ML as the default representation of context. However, driven by the emergence of long-range recency and contiguity effects (M. W. Howard & Kahana, 1999), modern theorists have since proposed that context consists of an exponentially-decaying representation of the recent past (M. W. Howard & Kahana, 2002; Sederberg, Howard, & Kahana, 2008). It soon became clear that in order to capture behavioral phenomena over a broad range of temporal scales, this single exponential decay would need to be extended to a family of exponentially-decaying functions, each with a different decay rate. Formally, Shankar & Howard (2011) put forth a formal model that observes how

a family of exponentially-decaying functions operates as a Laplace transform of experience, which can be inverted to recover a compressed representation of what happened when in the past. This context can be mathematically formulated and fit within the physiological limits of the brain, as well as evolutionary pressures, and is called Timing from Inverse-Laplace Transforms (TILT) (M. W. Howard & Eichenbaum, 2013; M. W. Howard & Shankar, 2017; M. W. Howard et al., 2014; M. W. Howard, Shankar, Aue, & Criss, 2015; Shankar & Howard, 2011). This representation of context is neurally plausible, meaning neurons could perform all the necessary mathematical functions, and compressed in the way that human memory tends to be, meaning the recent past is has better temporal accuracy than the distant past but the distant past is still present in the representation. There have since been many studies that have found neural evidence for this representation in many brain areas including the hippocampus and lateral prefrontal cortex (Cruzado, Tiganj, Brincat, Miller, & Howard, 2019; Eichenbaum, 2014; M. W. Howard, 2017; Salz et al., 2016; Tiganj, Cromer, Roy, Miller, & Howard, 2018).

In this paper, we examine the utility of the TILT representation of context within the framework of a canonical language model. This model, developed by Bengio, Ducharme, Vincent, & Janvin (2003), utilized a FIFO buffer as the context representation that can be easily replaced with TILT without having to change the base model. This will let us compare how well the model performs with its buffer representation relative to that of the same model with a TILT context representation that spans more time into the past with the identical memory footprint, and see which model performs better with a measure of performance known as perplexity.

Methods

We examined three different representations of context using the Bengio et al. (2003) Multi-Layer Perceptron language model as a means to evaluate their abilities to model a collection of documents known as the Brown Corpus. First, we explored the default representation of Bengio's MLP, the FIFO buffer. This representation is commonly used across ML as a way to imbue a model with temporal information in its input because it has perfect memory of the order in which things happened. Secondly, we implemented a representation proposed by Shankar and Howard that has been shown to exist in the brain (M. W. Howard et al., 2015; Shankar & Howard, 2011, 2012). Unlike the temporally perfect, but finite, representation of a buffer, this TILT representation uses leaky integrators to compress the entire history of words a model has experienced. Thirdly, we implemented a conceptual midway point between the two representations, the TILT-Buffer-MLP, to better understand which aspects of the two representations are driving differences in performance. We then compare all three representations' performances as they are trained and tested on their ability to predict the next word with different input sizes and MLP parameters, as well as their ability to

predict words even further into the future.

Bengio's Multi-Layer Perceptron

Bengio et al. (2003) utilized a Multi-Layer Perceptron (MLP) neural network model, one of the best-performing variants was referred to as MLP09 in the paper, and the model is illustrated in Figure 1. The goal of Bengio's MLP is to take the context of the last few words and generate a probability distribution of words that might occur after it. The strength of this model lies in its ability to learn both a semantic representation of each word, called an embedding, at the same time as learning to construct a predictive probability distribution over subsequent words. Learning these semantic features allows the MLP to learn how to map the presence of a word, as well as all semantically related words, onto what words might be occur next, instead of learning a mapping between individual words, which would require learning many times more weights between layers of the networks. This distinction is non-trivial because it allows their models to train much faster than their embeddingless counterparts and also be more flexible. For example learning for the context "her dog ate the food" would be represented similarly to the context "his cat ate the tuna" and both contexts might predict the word "quickly". Not surprisingly, almost all modern NLP models currently utilize an embedded feature space representation (Bengio et al., 2003; P.-C. Chang et al., 2008; Kuncoro et al., 2018; Luong et al., 2015; Vaswani et al., 2017). With this straightforward model as a backdrop, we can examine different representations of context while maintaining an equal playing field when comparing their performances.

Representations of Context

Buffer representation model: Buffer-MLP

The Bengio et al. (2003) MLP utilizes a FIFO buffer of the last few words to represent the context in which the model is experiencing the next word, and we will use its performance as a benchmark to measure other representations against. Buffers have commonly been used in ML as a way to represent the context, or recent history, in a task. Buffers are useful for tasks that require high temporal fidelity of information because they are a perfect representation of the order in which events occurred. Since a buffer's size, and therefore temporal span, is finite and must be picked a priori, one must be cognizant of the scale in which task relevant information exists over. Any information that exists outside the range of the buffer that could be used to solve the problem will be ignored. Increasing the buffer size isn't a perfect solution, though, because as a buffer gets bigger, the number of weights for the neural network will also increase. This increase in model complexity will make it harder to train, and sometimes can lead to overfitting that causes worse performance.

The Buffer representation for the Buffer-MLP is implemented as a buffer of 1-hot vectors of size 1xV, which uniquely distinguish the presence of one word

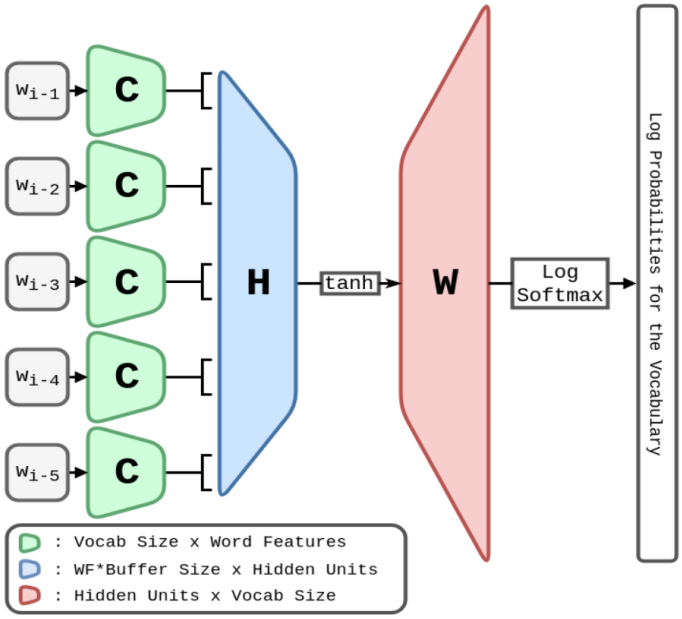

Figure 1: *Baseline Bengio et al. Multi-Layer Perceptrion Illustration* This model is used to take a buffer representation of the last few words and turn it into a probability of what word might be coming next. First, the model transforms each word present in each buffer position into its specific embedding space representation via the matrix C. C is learned independently of any temporal information and replicated across all words in the buffer. After the entire buffer is transformed, the feature vectors are concatenated together and passed through the hidden layer H with a tanh activation function. From here, the nodes are brought back to vocabulary space via the matrix W, where a log softmax turns the activation across all words in the vocabulary into a log probability of the likelihood each word would occur next in the sentence. w_i.

vs all other words, where V is the size of the vocabulary for the selected corpus. Every word in the vocabulary is assigned a position in that vector, and each buffer position contains a vector containing only the word that occurred some number of words in the past. This vector is used to index into the embedding matrix, C, such that only the word present at that moment into the past will get its features extracted from the matrix. In Figure 2, we see the activation levels of each buffer position as a function of when the word happened in the past. Only the word that happened exactly at that point in time in the past has an activation value of 1.0, and all other words that happened at all other times in the past have an activation value of 0.0. This neural network has 2 major parameters that we will be examining in this paper. The first is the size of the embedding matrix, C. Increasing the size of this matrix will allow the model to create higher dimensional semantic representations of each word. The second is the size of the hidden layer, H. Increase this size of this layer will allow the model to create more complex combinations of features across the different input positions.

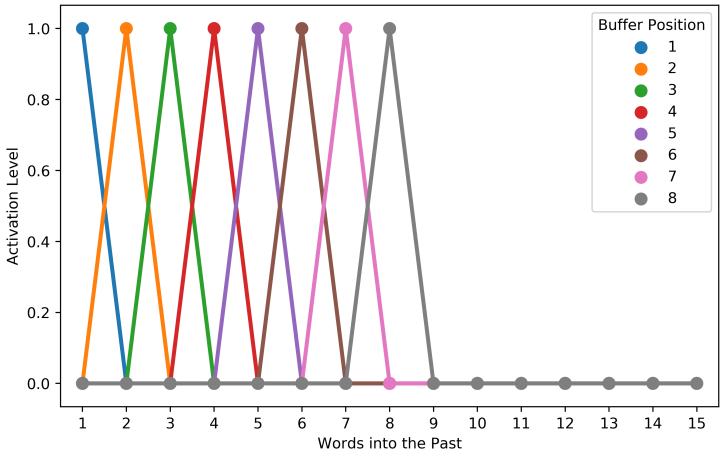

Figure 2: *Activation level of a one feature across Buffer positions* In this representation, temporal information is approximated by number of words into the past. Here we have a FIFO buffer that is keeping track of the order in which words occurred. Each position within the buffer is exactly one *word* timestep apart, and will be maximally activated at exactly one moment in the past. This also means that a buffer can only hold as much of the past in it as it has buffer positions, and that any words that occur even one word further into the past will not be present in this context representation.

TILT representation model: TILT-MLP

The TILT representation was first defined by Shankar & Howard (2011) as they set out to join theories of timing and episodic memory. They describe

a stimulus-agnostic mechanism for compressing an entire history leading up to the present moment. TILT is built with 3 features in mind. First, the mechanism must be able to represent a wide range of time scales of information. Second, events that occur further into the past experience compression that reduces temporal specificity for those events when compared to the more recent events. Third, this mechanism must be able to evolve through time without any outside influence other than the most recent instantaneous input. The TILT representation achieves this with a family of exponential decays with different decay rates, which function as the Laplace transform of the input signal. Then, TILT recreates a compressed version of the entire history of the input signal up until now through an approximation of the inverse Laplace transform. Thus, through an exponential decay instantiated with a family of leaky-integrators followed by a single linear operation, we are able to create a representation of context that spans much further into the past than a buffer with the exact same memory footprint.

When it is paired with language models, the TILT representation will track the history of every word in the vocabulary of a corpus. Because the TILT representation is functional over continuous time, we have to convert the discrete order of words within a corpus into continuous space. To do so, we present every word to TILT for the same amount of *time*, and every word occurs an equal amount of time apart. Thus, for the sake of simplicity, we will use numbers of words and amount of time into the past interchangeably. In contrast to a Buffer representation, which is a series of equally spaced in time containers with one word each, a TILT representation is a series of temporal receptive fields centered at a time into the past, called a τ^*s. Instead of having just one word present within them, each τ^* contains information about every word that occured within its receptive field, and the amount that each word is activated within that τ^* is proportional to how many times that word occurred.

The model portion of TILT-MLP is identical in structure to the Buffer-MLP, but it uses a set of logarithmically spaced τ^*s to represent the current history of what words were present at what times right before the next word. We are determining the input size to the model to be exactly how many τ^*s we include. For each of the τ^* indexes, there will be a vector of size 1xV that will have values in it associated with how close a word occurred in the past to the center of that τ^*s receptive field. For example, the last word to be presented to TILT would be highly active in the very first τ^*, but a word that occurred 10 presentations into the past wouldn't have any activation at all in the first τ^*. This also means that each τ^* index will have values indicating that many different words were present within their receptive fields. In figure 3, we show what the shape of each receptive field would look like. The first τ^*, which is centered around 1 word time step into the past, has activation values mostly for the words that occurred 1 to 3 words into the past. Later τ^*s will have activation from many more words, but will always have a lower maximum activation level than the earlier τ^*s. This is because their receptive field is much more spread out than the earlier τ^*s.

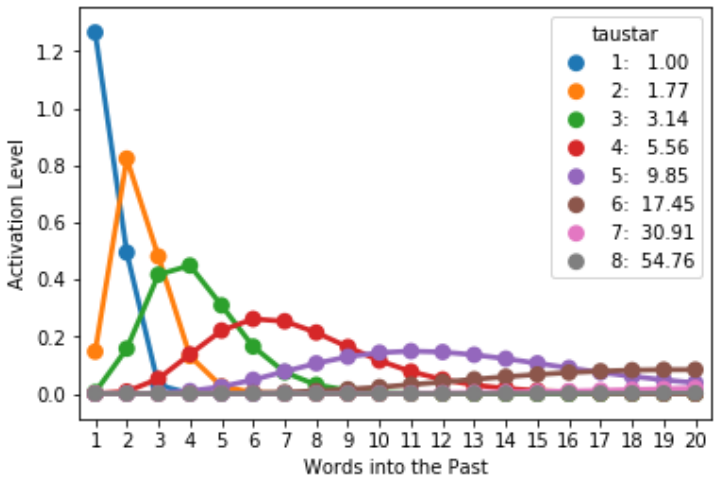

Figure 3: *Activation level of a one feature across TILT τ^*s* For this representation, each τ^* position has an entire activation function, instead of just being maximally active for exactly one position in the past. Each color represents a different τ^*, and each τ^* is a temporal receptive field that is centered around a different, logarithmically spaced, distance into the past. For example, τ^* 4 is centered around 5.56 words into the past, which means if a word occurred about 5 or 6 words ago it would be maximally active within that τ^*. Not only that, but τ^* 4 has activation for words that happened both earlier and later than 5 to 6 words into the past, and the further a word occurred from the center of the receptive field, the less activated it is. Each τ^* further into the past has a larger receptive field, allowing TILT to represent the gist information, meaning feature full but a lack of temporal specificity, of the distant past while also very finely representing the more recent past.

Because each τ^* can have activation values from multiple words in it, they will index into C slightly differently than how a buffer does. Instead of just extracting the exact feature vector of one word from C, they will construct a weighted combination of all of the words in a τ^* where the weighting depends on how activated that word was within that index.

Midway point between Buffer and TILT: TILT-Buffer-MLP

To establish a middle ground between the TILT and Buffer representations, we introduced the TILT-Buffer representation. Since even the first τ^* contained information about several words in it, we needed a way to verify that strength of the TILT model is coming from the long range correlations. Each position in the buffer is a snapshot of what the first τ^* of TILT looked like some number of words ago. Like TILT-MLP, it will turn activation levels of the words into a weighted combination of feature space representations, but like the Buffer-MLP, each position in the buffer will contain information a fixed distance further into the past. FIgure 4 shows an example of what this representation might look like when plotted as word activation as a function of how long they occurred in the past. This model will allow us to identify the key features that separate performance of models with the TILT and Buffer representations. Comparing the Buffer-MLP and TILT-Buffer-MLP allows us to see how a model with a weighted combination of features performs when compared to a model that only sees one word's features at a time, while comparing the TILT-MLP and the TILT-Buffer-MLP allows us to see the effectiveness of equally spaced receptive fields, like a buffer, compared to logarithmically spaced receptive fields, a key feature of TILT.

Training and Testing

In order to understand the operational limits of the different representations outlined above, we explored the parameter space of the representation/model combinations and examined how changing those parameters affected model performance.

We trained all of our models on the same corpus that was used by Bengio et al. (2003), the Brown corpus (Francis & Kucera, 1964). This corpus has over 1 million words in around 500 documents with roughly 2000 words each. We only analyzed words that appeared more than 5 times in the entire corpus, the same amount as Bengio et al. (2003), to ensure that our models only try to learn to predict or represent words that show up in the documents enough times to form a semantic understanding of the word. This resulted in a vocabulary size of ~13000 words. In every experiment presented below we trained on the first ~800,000 words in the Brown corpus.

We implemented all of our models in Python using Pytorch (Paszke et al., 2017), and optimized models with Pytorch's version of Stochastic Gradient Descent (SGD) algorithm (Sutskever, Martens, Dahl, & Hinton, 2013) with a learning

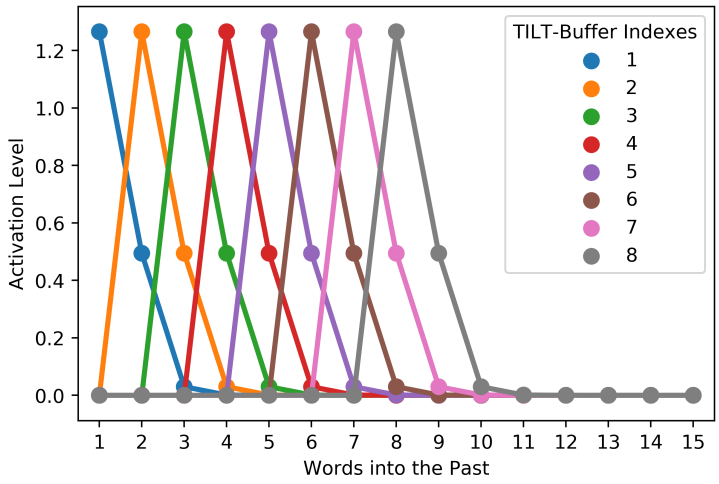

Figure 4: *Activation level of a one feature across TILT-Buffer indexes* Meant to be a bridge between the TILT and Buffer representations, this representation is just a buffer of the first τ^* in TILT. Each color represents a different buffer position, and each index is just what the first τ^* looked liked some number of words into the past.

rate of 1e-3. We trained all models on 20 viewings of the entire training set of documents, in order to keep things as similar as possible to the training regiment outlined by Bengio et al. (2003). During both training and testing, we cleared out the context representation at the start of each document because inter-document mutual information isn't likely to exist in the Brown corpus.

We tested the models in 4 different experimental scenarios where we modified the parameters of each MLP (TILT-MLP, Buffer-MLP, TILT-Buffer) to see which models performed the best in each test. Evaluating the models consisted of calculating a measure of perplexity, or how well the model predicts the correct target item, over the course of the ~200,000 words following the training set. Perplexity can be thought of as how many choices your model has narrowed down its prediction of the next word(Brown, Pietra, Mercer, Pietra, & Lai, 1992). Lower perplexity indicates a better performing model. We measured perplexity with the following equation:

$$perplexity = e^{\frac{-1}{N}\sigma lnq(w_i,c)},$$

where $q(w_i, c)$ is the proposed probability assigned w_i given context c by the model, and N is the number of words in the test set.

Results

We wanted to explore the full parameter spaces of these representations, as well as put them through different tasks, in order to determine where each representation flourished and where they failed. Firstly, we wanted to see if we could replicate the results presented by Bengio et al. (2003) and see how the TILT representation measured up against it. Replicating these results with the Buffer-MLP model would ensure that we did not do anything incorrectly in our implementation, as well as give us an appropriate baseline against which to measure the performance of TILT-MLP.

Experiment 1: Perplexity of Different Input Sizes

Firstly, we tested the Buffer-MLP against the TILT-MLP at an input size of 5. For the duration of the experiment, we mimicked the parameters of the best performing Bengio et al. (2003) standard MLP. They achieved a perplexity score of 276 in their paper, and our Buffer-MLP achieved a score of 262. This decrease in perplexity is likely due to the efficiency of the Pytorch implementation of SGD. Regardless, we were not concerned about this difference because this experiment is more about comparing the Buffer-MLP with the TILT-MLP. We felt satisfied with our ability to recreate Bengio's model, and moved on to test TILT-MLP. To maintain parity and comparability between the models, we used the first 5 indexed τ^*s to achieve an input size of 5 for our model. The TILT-MLP would then contain the same amount of learnable weights as the Buffer-MLP, allowing us to directly compare performance. After training, TILT-MLP achieved a perplexity score of 244, better than the Buffer-MLP. This successful increase in performance showed us evidence that the TILT representation is, in fact, of use in this problem space, but to be thorough we tested both representations with inputs spanning sizes 1 to 8. It could have been that we picked the perfect amount of τ^*s, so we wanted to investigate the performance of both of these models at various input sizes. We then trained a series of Buffer-MLP and TILT-MLP models, from input sizes 1 to 8, to see where, if at all, the Buffer-MLP might surpass the TILT-MLP in terms of perplexity performance. The results of which are displayed in 5.

At every input size, the TILT representation outperforms the Buffer representation. We first note that the TILT representation shows success even with a size of 1 compared to a Buffer representation of size 1. If the TILT representation was not achieving comparable performance to the Buffer representation at this input size, then that would have been evidence that the Bengio MLP could not utilize a compressed history when learning how to represent and predict words. Instead, the success of a TILT model over the Buffer model at size 1 is evidence that the feature space embedding matrix, C, could be learned via probing C with a weighted combination of words rather than just a simple one-hot-vector probing of C. We also see that the TILT representation continues to achieve lower perplexity than the Buffer representation at higher input sizes. Both the

Buffer and TILT are seeming to asymptote around input size 8, at 254 and 235 perplexity, respectively. This provides evidence that the longer time scales in TILT (6+) still have information in them that can help the model predict the next word. Witnessing TILT-MLP outperforming the Buffer-MLP during these tests made us wonder which part of the TILT representation contributed most to these results. In other words, was it the fact that the TILT representation indexed into C as a weighted combination of words, instead of the 1-hot vectors with the buffer representation, that lead to TILT-MLP doing better than the Buffer-MLP?

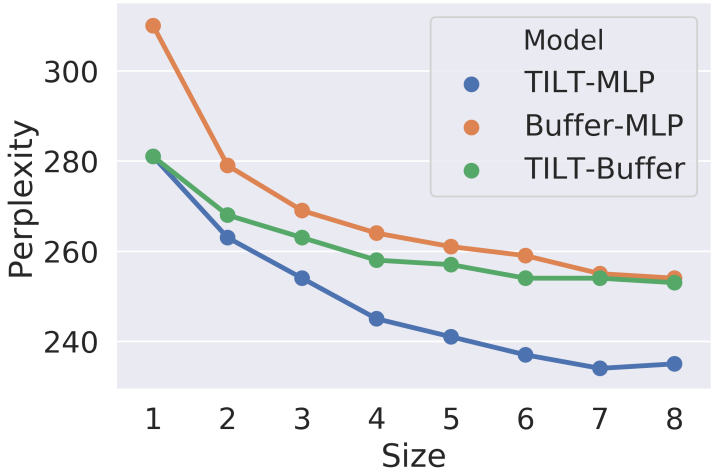

Figure 5: *Experiment 1. Perplexity Vs Model Size* As input size increases, so does the amount of time each model is able to look into the past. Perplexity decreases the further into the past each model is able to look, but TILT-MLP beats the other models at every input size. TILT-MLP is likely able to out-perform the other models due to the logarithmic spacing of its temporal receptive fields. Each additional input position for TILT-MLP looks much further into the past than the other models.

This lead us to test the representation called TILT-Buffer-MLP. TILT-Buffer-MLP, as described above, indexes into C the same way that the TILT-MLP does. We hypothesized that, since a TILT of size 1 still spans multiple words in time,, a TILT-Buffer-MLP could potentially achieve similar results at all input sizes, invalidating our use of scale invariance. By putting the first τ^* in a buffer, we would be able to show TILT's larger temporal span was really what was driving the boost to performance. What we ended up seeing was the TILT-Buffer-MLP performing similarly to TILT-MLP at low input sizes, and then similarly to Buffer-MLP at higher input sizes. This was likely due to the TILT-Buffer-MLP not extending as far into the past as the TILT-MLP with its higher input sizes. Now we were able to more confidently say that it was a combination of the TILT

representation's weighted indexing into C, as well as the larger but compressed temporal span, that allowed the TILT-MLP to outperform the Buffer-MLP.

Experiment 2: Predicting further into the future

Pascanu et. al. suggested that a representation that contained longer-term memory would be more successful for predicting further into the future (Pascanu et al., 2012). We hypothesized that TILT-MLP performed better than the Buffer-MLP of a similar size because TILT's ability to represent larger scales of time. In experiment 2 we ran a test where the models were trained to predict what word would occur some number of words into the future, and the results are displayed within 6. We kept the size of the models static, with an input size of 5, hidden layer of 100, and feature space representations of 30. The results show the performance of both representations got worse the further into the future they tried to predict. We also saw that TILT-MLP performed better than the Buffer-MLP at every number of words into the future. This is to be expected because TILT is able to represent more of the past than a buffer with the same input size. These results gave us more evidence that TILT is able to not only represent things that happened long ago in the past in a sparse way, but also to represent this past in a useful way for the model. From here, we wanted to explore the base model parameters a bit more. Bengio et al. (2003) had explored the parameter space of their MLP and found that 30 and 100 were the best sizes for their model. It is the case that TILT-MLP could actually start to fail if we modified the model parameters, so we wanted to explore how the various sizes of these parameters affected the performance of both our Buffer-MLP and the TILT-MLP.

Experiment 3: Investigation into model parameters

In experiment 3, we examined how model performances changed when increasing the size of the different layers within them, and if changing these parameters affected either model differently. In particular, we wanted to increase the size of the embedding space C and/or increase the size of the hidden layer H. We knew that increasing the size of the models would in turn make the models more complex, but we wanted to know if, keeping all other variables the same, TILT-MLP would still perform better than Buffer-MLP as it did in Experiments 1 and 2. The results of this experiment are shown in Figure 7. For reference, we added the perplexity measure for the model size 5 runs from experiment 1 for each grouping. We saw the general trend of a decreasing perplexity when increasing either the hidden layer size or the embedding size, but overall TILT-MLP experienced the greater decrease.

TILT-MLP achieved its lowest perplexity score of 233 when increasing the size of both the hidden layer and the embedding space to 200 and 60, respectively. Independently, increasing the size of the hidden layer or the embedding space would decrease perplexity, but increasing both only marginally decreased perplexity.

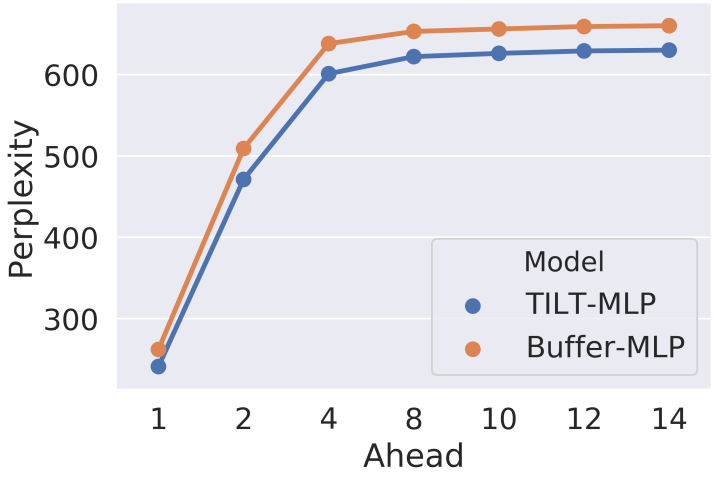

Figure 6: *Experiment 2: Models Predicting further into the Future* Both models were trained with an input size of 5 and 20 epochs through the corpus. The models get worse the further ahead they are trying to predict, but TILT-MLP is able to beat the Buffer-MLP regardless of how far ahead it is predicting.

This is likely due to us keeping the training time the same across all of these tests. The greatest decrease in perplexity was attributed to increasing the size of the hidden layer, however. The only time we saw TILT-MLP perform worse than its default parameter baseline was when we drastically increased the size of the hidden layer to 700, but this is likely due to the fact that larger models require more training epochs. The Buffer-MLP saw very slight decreases in perplexity when modifying the model parameters. It achieved its lowest perplexity score of 257 by increasing the size of the hidden layer to 200 and not modifying the size of the embedding space. Increasing the size of both layers, the Buffer-MLP performed roughly the same with a score of 258. Increasing the size of these layers did not, however, allow the Buffer-MLP to achieve anywhere close the the perplexity level of the TILT-MLP. Because of this, we decided to exclude the Buffer-MLP from the massive hidden layer tests, but we suspect that the perplexity would, like the TILT-MLP, worsen rather than improve performance, due to how large the models would be.

With these results, we saw no indication that the model parameters of the Buffer-MLP could be modified to make it perform better than the TILT-MLP. In addition, both models' perplexity scores dropped the most by increasing the size of the hidden layer of the model. This was especially true for TILT-MLP. Increasing the size of the hidden layer leads to increasing the number of connections between different indexes within each representation of context. The improvement in performance due to this increase in hidden layer size shows us that the models are leveraging as much information as possible from their

representations of context onto predicting the next word. From this point on, we felt that it was no longer necessary to examine the Buffer-MLP. Instead, we wanted to know just how vital the compressed history aspect of the TILT representation was to its success in the previous tests.

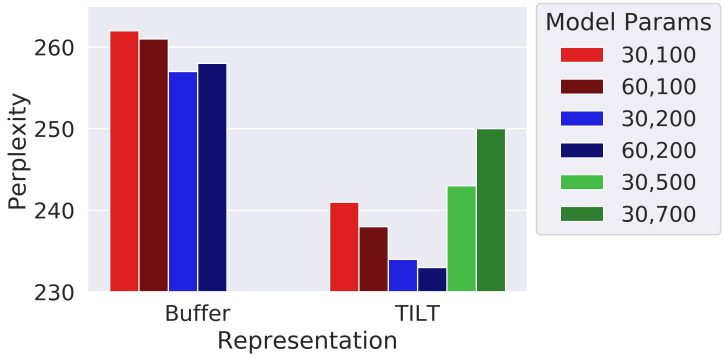

Figure 7: *Experiment 3. Perplexity Vs Layer Sizes* We modified the internal parameters of the MLP and trained it again for both the TILT and the Buffer representations. To give the models equal footing, all of the above tests were done with the same amount of training epochs, and with an input size of 5. All of the different model sizes in the legend are of the following format: feature-space size, hidden layer size. Increasing the size of the hidden layer increased the performance of both representations, whereas increasing the size of the feature space only provided a minor increase in performance to both. The performance boost was larger with the TILT representation than with the Buffer representation.

Based on the previous experiments, we have seen that adding additional τ^*s which represent events even further into the past increases the performance of your model. We don't know, however, how useful these later τ^*s are independent of their earlier counterparts. Examining an edge case of the TILT representation where the more recent τ^*s are not present would give us a look into exactly where this representation starts to fail, given that we have only really seen one time where it starts to fail in the previous experiment.

Experiment 4: Determining the importance of long range correlations

Instead of adding additional τ^*s, we wanted to focus on how TILT-MLP performed using only the longer time-scale τ^*s. Based on work by H. W. Lin & Tegmark (2017), we know that human language, much like human experience, contains information on both long and short time scales that help us predict and understand the world around us. If the later τ^*s contain any predictive information on their own, and are therefore a meaningful recreation of the distant

past for predicting the future, then we should see performance better than that of the uniformed baseline perplexity of the corpus. This baseline, where the predictive probability assigned to each word is just the number of occurrences of that word divided by the number of words in the training set, would achieve a perplexity score of 853. Figure 8 shows that these models are very reliant on the most recent τ^*s. Even removing just the first τ^*, we see a sharp increase in perplexity, which equates to a sharp decline in performance, and every model after that gets worse. Importantly, we saw that even the longer τ^*s still have some predictive information in them. This is in line with Lin and Tegmark; when examining written or spoken human language, there is important contextual information about the topics at hand that span many different numbers of words into the past. The fifth τ^* on its own is still able to have some semblance of a prediction as to what words might be coming next based solely on its compressed representations of all of the words within its temporal receptive field.

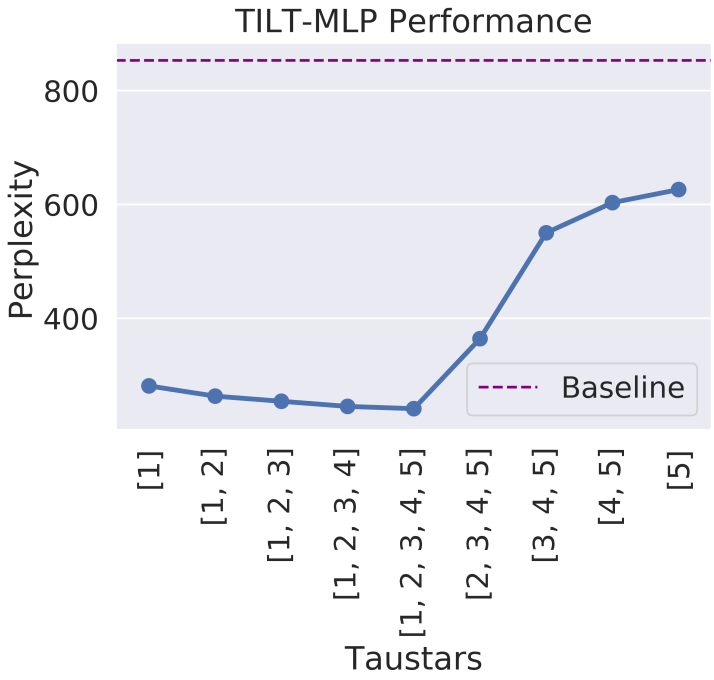

Figure 8: *Experiment 4: TILT-MLP performance with subsets of τ^*s* Displayed are the results from Experiment 1 and Experiment 4 together. In experiment 4 we are removing the most recent τ^*s to view the relevance of longer range τ^*s to performance. Experiment 1's results showed us that including more τ^*s decreased perplexity. As you take away the more recent τ^*s from the model, it is still able to function, albeit not nearly as well. This implies that there is still useful information in those long range τ^*s that is important for this task of predicting the next word.

Discussion

In this paper, we have shown that a neurally-inspired representation of context can improve a canonical language model's performance on a variety of tests. At all input sizes, model layer sizes, and amount of time further predicting into the future, the TILT representation outperformed the Buffer representation. We attribute the success of this representation to TILT's log-spaced compression, which allows it to simultaneously span much longer time-scales and maintain temporal specificity of short time-scale events. The results herein place the TILT representation as a definitive drop in replacement for a buffer representation in the domain of NLP, and, by extension, any neural network modeling time series data with long-term temporal correlations. Since long-term correlations are so prevalent in the human experience (H. W. Lin & Tegmark, 2017), the likelihood of finding additional applications for the TILT representation is high.

An added benefit of the TILT representation is scale-invariance, which frees the researcher from knowing a priori over what duration information is relevant for making model predictions. One of the ways in which machine learning scientists imbue models the ability to learn temporal relationships without a Buffer is with Recurrent Neural Networks (RNNs), sometimes in the form of Long Short Term Memory (LSTM) models. Unfortunately, there are problems with these recurrent methods, as they have serious trade offs in computational complexity, hardware compatibility, and internal memory footprint (Pascanu et al., 2012). Utilizing a TILT representation, however, allows the model to concentrate all of its learning on how to transform a scale-invariant representation of the past into the desired output. A TILT representation, religating any recurrence to the representation itself and not within the model, is also not subject to any of the exploding/imploding gradient problems that often occur while training RNNs.

Despite the benefits listed above, the TILT representation will struggle when applied to data that require high temporal specificity of the extreme distant past. Once information enters those longer time-scale τ^*s, temporal specificity is almost completely lost. We believe this weakness will eventually be eliminated, however, with future model developments that involve multiple TILT representation layers in a hierarchy or by relying on more features present in our full memory systems. For instance, TILT on its own lacks the ability to reinstate previously experienced histories that are associated with the current context, which is a way in which humans are able to learn new concepts extremely quickly (M. W. Howard, Jing, Rao, Provyn, & Datey, 2009). Formalizing the mechanisms of human memory will, eventually, lead to developing faster and more efficiently trained neural networks that could recover detail from past events, much like jumping back in time with episodic memory, to help make more accurate predictions of the future..

In the future, we hope to utilize additional formalized mechanisms of memory, attention, and perception, to improve neural network models. Explicitly, we hope to examine methods related to mental time travel, or the brain's ability to

return itself to a previously experienced state, in order to boost memory and learning. This, in addition to the TILT representation of context, would allow an NLP to compare its current context to previously experienced contexts for guiding predictions. The model would then be able to determine which context words were the same across presentations of the target word, strengthening the associations between those words, and which context words are novel, potentially weakening the associations between those words since they don't co-occur together all that often. This context reinstatement could allow an NLP to train much more quickly, i.e. with many fewer runs through the training set of words, because each time the model experiences a word, it will be re-experiencing and learning from every context it ever experienced that word in at the same time.

References

Atkinson, R., & Shiffrin, R. (1968). Human Memory: A Proposed System and its Control Processes. In *Psychology of Learning and Motivation* (Vol. 2, pp. 89–195). https://doi.org/10.1016/S0079-7421(08)60422-3

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, *3*, 1137–1155. Retrieved from http://dl.acm.org/citation.cfm?id=944919.944966

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Brown, P. F., Pietra, V. J. D., Mercer, R. L., Pietra, S. A. D., & Lai, J. C. (1992). An Estimate of an Upper Bound for the Entropy of English. *Comput. Linguist.*, *18*(1), 31–40. Retrieved from http://dl.acm.org/citation.cfm?id=146680.146685

Chang, P.-C., Galley, M., & Manning, C. (2008, June). *Optimizing Chinese Word Segmentation for Machine Translation Performance*. 224–232. https://doi.org/10.3115/1626394.1626430

Cruzado, N. A., Tiganj, Z., Brincat, S. L., Miller, E. K., & Howard, M. W. (2019). Conjunctive representation of what and when in monkey hippocampus and lateral prefrontal cortex during an associative memory task. *bioRxiv*, 709659. https://doi.org/10.1101/709659

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [Cs]*. Retrieved from http://arxiv.org/abs/1810.04805

Eichenbaum, H. (2014). Time cells in the hippocampus: A new dimension for mapping memories. *Nature Reviews Neuroscience*, *15*(11), 732–744. https://doi.org/10.1038/nrn3827

Francis, W. N., & Kucera, H. (1964). *Brown Corpus Manual*. Retrieved from

http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM

Ghosh, S., & Kristensson, P. O. (2017). Neural Networks for Text Correction and Completion in Keyboard Decoding. *arXiv:1709.06429 [Cs]*. Retrieved from http://arxiv.org/abs/1709.06429

Howard, M. W. (2017). Temporal and spatial context in the mind and brain. *Current Opinion in Behavioral Sciences*, *17*, 14–19. https://doi.org/10.1016/j.cobeha.2017.05.022

Howard, M. W., & Eichenbaum, H. (2013). The hippocampus, time, and memory across scales. *Journal of Experimental Psychology. General*, *142*(4), 1211–1230. https://doi.org/10.1037/a0033621

Howard, M. W., & Kahana, M. J. (1999). Contextual variability and serial position effects in free recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(4), 923–941. https://doi.org/10.1037/0278-7393.25.4.923

Howard, M. W., & Kahana, M. J. (2002). A Distributed Representation of Temporal Context. *Journal of Mathematical Psychology*, *46*(3), 269–299. https://doi.org/10.1006/jmps.2001.1388

Howard, M. W., & Shankar, K. H. (2017). Neural scaling laws for an uncertain world. *arXiv:1607.04886 [Physics, Q-Bio]*. Retrieved from http://arxiv.org/abs/1607.04886

Howard, M. W., Jing, B., Rao, V. A., Provyn, J. P., & Datey, A. V. (2009). Bridging the gap: Transitive associations between items presented in similar temporal contexts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *35*(2), 391–407. https://doi.org/10.1037/a0015002

Howard, M. W., MacDonald, C. J., Tiganj, Z., Shankar, K. H., Du, Q., Hasselmo, M. E., & Eichenbaum, H. (2014). A Unified Mathematical Framework for Coding Time, Space, and Sequences in the Hippocampal Region. *Journal of Neuroscience*, *34*(13), 4692–4707. https://doi.org/10.1523/JNEUROSCI.5808-12.2014

Howard, M. W., Shankar, K. H., Aue, W. R., & Criss, A. H. (2015). A distributed representation of internal time. *Psychological Review*, *122*(1), 24–53. https://doi.org/10.1037/a0037840

Kuncoro, A., Dyer, C., Hale, J., Yogatama, D., Clark, S., & Blunsom, P. (2018). LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1426–1436. https://doi.org/10.18653/v1/P18-1132

Lin, H. W., & Tegmark, M. (2017). Critical Behavior in Physics and Probabilistic Formal Languages. *Entropy*, *19*(7), 299. https://doi.org/10.3390/e19070299

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *arXiv:1508.04025 [Cs]*. Retrieved

from http://arxiv.org/abs/1508.04025

Pascanu, R., Mikolov, T., & Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. *arXiv:1211.5063 [Cs]*. Retrieved from http://arxiv.org/abs/1211.5063

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic Differentiation in PyTorch. *NIPS Autodiff Workshop*.

Salz, D. M., Tiganj, Z., Khasnabish, S., Kohley, A., Sheehan, D., Howard, M. W., & Eichenbaum, H. (2016). Time Cells in Hippocampal Area CA3. *Journal of Neuroscience*, *36*(28), 7476–7484. https://doi.org/10.1523/JNEUROSCI.0087-16.2016

Sederberg, P. B., Howard, M. W., & Kahana, M. J. (2008). A context-based theory of recency and contiguity in free recall. *Psychological Review*, *115*(4), 893–912. https://doi.org/10.1037/a0013396

Shankar, K. H., & Howard, M. W. (2011). A Scale-Invariant Internal Representation of Time. *Neural Computation*, *24*(1), 134–193. https://doi.org/10.1162/NECO_a_00212

Shankar, K. H., & Howard, M. W. (2012). Optimally fuzzy temporal memory. *J. Mach. Learn. Res.*, *14*, 3785–3812.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In S. Dasgupta & D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning* (pp. 1139–1147). Retrieved from http://proceedings.mlr.press/v28/sutskever13.html

Tiganj, Z., Cromer, J. A., Roy, J. E., Miller, E. K., & Howard, M. W. (2018). Compressed timeline of recent experience in monkey lPFC. *bioRxiv*, 126219. https://doi.org/10.1101/126219

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [Cs]*. Retrieved from http://arxiv.org/abs/1706.03762

Voss, R. F., & Clarke, J. (1975). "1/ f noise" in music and speech. *Nature*, *258*(5533), 317–318. https://doi.org/10.1038/258317a0

Wagenmakers, E.-J., Farrell, S., & Ratcliff, R. (2004). Estimation and interpretation of 1/f noise in human cognition. *Psychonomic Bulletin & Review*, *11*(4), 579–615. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1479451/