Sparsity Emerges Naturally in Neural Language Models

Citation for published version:

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Abstract
Concerns about interpretability, computational resources, and principled inductive priors have motivated efforts to engineer sparse neural models for NLP tasks. If sparsity is important for NLP, might well-trained neural models naturally become roughly sparse? Using the Taxi-Euclidean norm to measure sparsity, we find that frequent input words are associated with concentrated or sparse activations, while frequent target words are associated with dispersed activations but concentrated gradients. We find that gradients associated with function words are more concentrated than the gradients of content words, even controlling for word frequency.

1 Introduction
Researchers in NLP have long relied on engineering features to reflect the sparse structures underlying language. Modern deep learning methods promised to relegate this practice to history, but have not eliminated the interest in sparse modeling for NLP. Along with concerns about computational resources (Chen et al., 2016; Narang et al., 2017b) and interpretability (Murphy et al., 2012; Subramanian et al., 2018), human intuitions continue to motivate sparse representations of language. For example, some work applies assumptions of sparsity to model latent hard categories such as syntactic dependencies (Padó and Lapata, 2007) or phonemes (Cotterell and Eisner, 2018). Niculae and Blondel (2017) found that a sparse attention mechanism outperformed dense methods on some NLP tasks; Narang et al. (2017a) found sparsified versions of LMs that outperform dense originals. Attempts to engineer sparsity rest on an unstated assumption that it doesn’t arise naturally when neural models are learned. Is this true?

Using a simple measure of sparsity, we analyze how it arises in different layers of a neural language model in relation to word frequency. We show that the sparsity of a word representation increases with exposure to that word during training. We also find evidence of syntactic learning: gradient updates in backpropagation depend on whether a word’s part of speech is open or closed class, even controlling for word frequency.

2 Methods
Language model. Our LM is trained on a corpus of tokenized, lowercased English Wikipedia (70/10/20 train/dev/test split). To reduce the number of unique words (mostly names) in the corpus, we excluded any sentence with a word which appears fewer than 100 times. Those words which still appear fewer than 100 times after this filter are replaced with <UNK>. The resulting training set is over 227 million tokens of around 19.5K types.

We use a standard 2-layer LSTM LM trained with cross entropy loss for 50 epochs. The pipeline from input $x_{t-1}$ at time step $t - 1$ to predicted output distribution $\hat{x}_t$ for time $t$ is described in Figure 1, illustrating intermediate activations $h^t_e$, $h^t_1$, and $h^t_2$. At training time, the network observes $x_t$ and backpropagates the gradient updates $\bar{h}^t_e$, $\bar{h}^t_1$, $\bar{h}^t_2$, and $\bar{x}_t$.

The embeddings produced by the encoding layer are 200 units, and the recurrent layers have 200 hidden units each. The batch size is set to forty, the maximum sequence length to 35, and the dropout ratio to 0.2. The optimizer is standard SGD with clipped gradients at $\ell_2 = 0.25$, where the learning rate begins at 20 and is quartered whenever loss fails to improve.

Measuring sparsity. We measure the sparsity of a vector $v$ using the reciprocal of the Taxi-Euclidean norm ratio (Repetti et al., 2015). This measurement has a long history as a measurement of sparsity in natural settings (Zibulevsky...
Figure 1: LM architecture for target word distribution $\hat{x}_t$, showing gradient updates from observed word $x_t$.

Figure 2: Average sparsity $\chi(h^2_t)$ over all training epochs (x-axis), for target words $x_t$ occurring more than 100k times in training. Target words are sorted from most frequent (bottom) to least frequent (top).

3 Experiments

Sparsity is closely related to the behavior of a model: If only a few units hold most of the mass of a representation, the activation vector will be highly concentrated. If a neural network relies heavily on a small number of units in determining its predictions, the gradient will be highly concentrated. A highly concentrated gradient is mainly modifying a few specific pathways. For example, it might modify a neuron associated with particular inputs like parentheses (Karpathy et al., 2015), or properties like sentiment (Radford et al., 2017).

**Representations of Target Words.** Our first experiments look at the relationship of sparsity to target word $x_t$. Gradient updates triggered by the target are often used to identify units that are relevant to a prediction (Li et al., 2015), and as shown in Figure 2, gradient sparsity increases with both the frequency of a word in the corpus and the overall training time. In other words, more exposure leads to sparser relevance. Because the sparsity of $\tilde{h}^2$ increases with target word frequency, we measure not sparsity itself but the Pearson correlation, over all words $w$, between word frequency and mean $\chi(h)$ over representations $h$ where $w$ is the target:

$$\rho(w) = \text{corr}_w(\mu_{t:x_t=w}(\chi(h_t)), \text{freq}(w))$$

Here (Figure 3a) we confirm that concentrated gradients are not a result of concentrated activations, as activation sparsity $\chi(h^2)$ is not correlated with target word frequency.

The correlation is strong and increasing only for $\rho(w)$. The sparse structure being applied is therefore particular to the gradient passed from the softmax to the top LSTM layer, related to how a word interacts with its context.

**The Role of Part of Speech.** Figure 4 shows that $\rho(w)$ follows distinctly different trends for open POS classes and closed classes. To associate words to POS, we tagged our training corpus with spacy; we associate a word to a POS only if the majority (at least 100) of its occurrences are tagged with that POS. We see that initially, frequent words from closed classes are highly concentrated, but soon stabilize, while frequent words from open classes continue to become more concentrated throughout training. Why?

Closed class words clearly signal POS. But open classes contain many ambiguous words, like

1 ADJ, ADV, INTJ, NOUN, PROPN, VERB
2 ADP, AUX, CCONJ, DET, PART, PRON, SCONJ
3 https://spacy.io/
Figure 3: Correlation between mean sparsity of a word’s representation and word frequency. Vertical dashed lines indicate when the optimizer has rescaled the step size.

Figure 4: \( \rho \left( h^2 \right) \), evaluated over vocabulary from open and closed classes of POS.

Figure 5: Mean sparsity of \( \chi(h^2) \) after 50 epochs, for words occurring more than 1k times in the train set.

“report”, which can be a noun or verb. Open classes also contain many more words in general. We posit that early in training, closed classes reliably signal syntactic structure, and are essential for shaping network structure. But open classes are essential for predicting specific words, so their importance in training continues to increase after part of speech tags are effectively learned.

The high sparsity of function word gradient may be surprising when compared with findings that content words have a greater influence on outputs (Kádár et al., 2016). However, those findings were based on the impact on the vector representation of an entire sentence after omitting the word. Khandelwal et al. (2018) found that content words have a longer window during which they are relevant, which may explain the results of Kádár et al. (2016). Neither of these studies controlled for word frequency in their analyses contrasting content and function words, but we believe this oversight is alleviated in our work by measuring correlations rather than raw magnitude. Because \( \rho \left( h^2 \right) \) is higher when evaluated over more frequent words, which also tend to be function words (see Figure 5), we further control for the effect of frequency by including a measurement of trends in a sample of 120 words each from open and closed classes (Figure 4). This sample was selected by sorting all open and closed class words by frequency, then choosing a range of each sorted list with a similar average frequency.

**Representations of Input Words.** We next looked at the vector representations of each step in the word sequence as a representation of the input word \( x_{t-1} \) that produced that step. We measure the correlation with input word frequency:

\[
\rho \left( h \right) = \text{corr}_w \left( \mu_{t,x_{t-1}=w} \left( \chi(h_t) \right), \text{freq}(w) \right)
\]

Here (Figure 3b) we find that the view across training sheds some light on the learning process. While the lower recurrent layer quickly learns sparse representations of common input words, \( \rho \left( h^1 \right) \) increases more slowly later in training and is eventually surpassed by \( \rho \left( h^2 \right) \), while gradient sparsity never becomes significantly correlated with word frequency. Li et al. (2016) studied the
activations of feedforward networks in terms of the importance of individual units by erasing a particular dimension and measuring the difference in log likelihood of the target class. They found that importance is concentrated into a small number of units at the lowest layers in a neural network, and is more dispersed at higher layers. Our findings suggest that this effect may be a natural result of the sparsity of the activations at lower layers.

We relate the trajectory over training to the Information Bottleneck Hypothesis of Shwartz-Ziv and Tishby (2017). This theory, connected to language model training by Saphra and Lopez (2018), proposes that the earlier stages of training are dedicated to learning to effectively represent inputs, while later in training these representations are compressed and the optimizer removes input information extraneous to the task of predicting outputs. If extraneous information is encoded in specific units, this compression would lead to the observed effect, in which the first time the optimizer rescales the step size, it begins an upward trend in $\rho \rightarrow$ as extraneous units are mitigated.

4 Potential Explanations

Why do common target words have such concentrated gradients with respect to the final LSTM layer? A tempting explanation is that the amount of information we have about common words offers high confidence and stabilizes most of the weights, leading to generally smaller gradients. If this were true, the denominator of sparsity, gradient $\ell_1$, should be strongly anti-correlated with word frequency. In fact, it is only ever slightly anti-correlated (correlation $>-1$). Furthermore, the sparsity of the softmax gradient $\chi(\bar{x})$ does not exhibit the strong correlation seen in $\chi(h^2)$, so sparsity at the LSTM gradient is not a direct effect of sparse logits.

However, the model could still be “high confidence” in terms of how it assigns blame for error during common events, even if it is barely more confident overall in its predictions. According to this hypothesis, a few specialized neurons might be responsible for the handling of such words.

Perhaps common words play a prototyping role that defines clusters of other words, and therefore have a larger impact on these clusters by acting as attractors within the representation space early on. Such a process would be similar to how humans acquire language by learning to use words like ‘dog’ before similar but less prototypical words like ‘canine’ (Rosch, 1999). As a possible mechanism for prototyping with individual units, Dalvi et al. (2019) found that some neurons in a translation system specialized in particular word forms, such as verb inflection or comparative and superlative adjectives. For example, a common comparative adjective like ‘better’ might be used as a reliable signal to shape the handling of comparatives by triggering specialized units, while rarer words have representations that are more distributed according to a small collection of specific contexts.

There may also be some other reason that common words interact more with specific substructures within the network. For example, it could be related to the use of context. Because rare words use more context than common words and content words use more context than function words (Khandelwal et al., 2018), the gradient associated with a common word would be focused on interactions with the most recent words. This would lead common word gradients to be more concentrated.

It is possible that frequent words have sparse activations because frequency is learned as a feature and thus is counted by a few dimensions of proportional magnitude, as posited by Li et al. (2016).

5 Potential Applications

Understanding where natural sparsity emerges in dense networks could be a useful guide in deciding which layers we can apply sparsity constraints to without affecting model performance, for the purpose of interpretability or efficiency. It might also explain why certain techniques are effective: for example, in some applications, summing representations together works quite well (Hill et al., 2016). We hypothesize that this occurs when the summed representations are sparse so there is often little overlap. Understanding sparsity could help identify cases where such simple ensembling approaches are likely to be effective.

Future work may develop ways of manipulating the training regime, as in curriculum learning, to accelerate the concentration of common words or incorporating concentration into the training objective as a regularizer. We would also like to see how sparsity emerges in models designed for specific end tasks, and to see whether concentration is a useful measure for the information compression predicted by the Information Bottleneck.
References


