Sparse Attention with Linear Units

Citation for published version:
<https://aclanthology.org/2021.emnlp-main.523>

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:

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.
Sparse Attention with Linear Units

Biao Zhang\textsuperscript{1}  Ivan Titov\textsuperscript{1,2}  Rico Sennrich\textsuperscript{3,1}

\textsuperscript{1}School of Informatics, University of Edinburgh
\textsuperscript{2}ILLC, University of Amsterdam
\textsuperscript{3}Department of Computational Linguistics, University of Zurich

B.Zhang@ed.ac.uk, ititov@inf.ed.ac.uk, sennrich@cl.uzh.ch

Abstract

Recently, it has been argued that encoder-decoder models can be made more interpretable by replacing the softmax function in the attention with its sparse variants. In this work, we introduce a novel, simple method for achieving sparsity in attention: we replace the softmax activation with a ReLU, and show that sparsity naturally emerges from such a formulation. Training stability is achieved with layer normalization with either a specialized initialization or an additional gating function. Our model, which we call Rectified Linear Attention (ReLA), is easy to implement and more efficient than previously proposed sparse attention mechanisms. We apply ReLA to the Transformer and conduct experiments on five machine translation tasks. ReLA achieves translation performance comparable to several strong baselines, with training and decoding speed similar to that of the vanilla attention. Our analysis shows that ReLA delivers high sparsity rate and head diversity, and the induced cross attention achieves better accuracy with respect to source-target word alignment than recent sparsified softmax-based models. Intriguingly, ReLA heads also learn to attend to nothing (i.e. ‘switch off’) for some queries, which is not possible with sparsified softmax alternatives.\textsuperscript{1}

1 Introduction

Attention models (Bahdanau et al., 2015) have been hugely successful recently, with Transformer (Vaswani et al., 2017) in particular, advancing state of the art on various tasks, such as machine translation (Bojar et al., 2018), document summarization (Liu and Lapata, 2019) and speech processing (Chiu et al., 2018), and delivering a large impact on a broad range of NLP tasks via large-scale self-supervised pretraining (Devlin et al., 2019).

\textsuperscript{1}Source code is available at https://github.com/bzhangGo/zero.

At the core of attention is a mechanism that dynamically highlights relevant context features for a given query input. In the vanilla softmax-based attention model (Vaswani et al., 2017, SMA), this is achieved by imposing a categorical distribution constraint on the query-context relevance (i.e. attention) scores, implemented with the softmax activation (see Figure 1(a)). SMATT produces dense distributions, assigning some small amounts of attention even to irrelevant features. This complicates the analysis of the information flow in the model, and has led researchers to study sparse alternatives, which often lead to improved model performance and/or interpretability (Correia et al., 2019). Efforts in this category include designing fixed sparsity patterns (Raganato et al., 2020; Child et al., 2019) and creating sparsified softmax variants (Martins and Astudillo, 2016; Peters et al., 2019). However, these methods also have drawbacks. Fixed sparsity patterns lack flexibility and generalize poorly across tasks. Sparsified softmax variants often depend on complex inference algorithms (e.g., requiring the sorting operation), which reduces their efficiency.

In this paper, we propose rectified linear attention (ReLA) to alleviate the above problems. ReLA
uses ReLU rather than softmax as an activation function for attention scores, abandoning the probabilistic constraint. ReLU is inherently sparse since negative activations are dropped, and we will show that such sparse behaviour indeed emerges during training. In contrast to softmax activations, the output of ReLU can be any non-negative value, providing extra flexibility. To stabilize gradients and ease model convergence, we apply layer normalization together with a specialized initialization or a gating mechanism. Figure 1(b) shows ReLA, and also contrasts it with SMATT.

ReLA is an easy-to-implement drop-in replacement for SMATT that requires no specialized operations or inference processes. Note that the behaviour of ReLA is data-driven, and it does not enforce a constant attention mass or sparsity level across queries, even allowing for null attention (all attention scores are zero) for some queries. We provide experimental results for ReLA with Transformer on five machine translation tasks, along with an in-depth analysis on WMT14 English-German task. Our contributions are summarized below:

- We propose ReLA, a drop-in SMATT alternative, that learns sparse attention automatically with high flexibility and efficiency.
- Experiments on five translation tasks show that ReLA achieves comparable translation performance, with similar training/decoding speed to SMATT, but is substantially faster than sparsified softmax baselines.
- Our analysis shows that ReLA delivers high sparsity rate, high head diversity, and better accuracy than all baselines with respect to source-target word alignment. We also observe the emergence of attention heads with a high rate of null attention, only activating for certain queries. For some heads, this null rate can also indicate the quality of sentence pairs.

2 Related Work

ReLA ensures sparsity in attention. An alternative solution in this direction is to develop sparsified softmax alternatives, such as sparsemax (Martins and Astudillo, 2016; Malaviya et al., 2018), entmax (Peters et al., 2019; Correia et al., 2019), fusedmax (Niculae and Blondel, 2017), and hashing/clustering-based variants (Roy et al., 2020; Kitaev et al., 2020). These models often require dedicated algorithms for forward and backward propagation, at the cost of a significant computational overhead. Another strategy is to manually define sparse patterns inspired by task-specific attention analysis. Raganato et al. (2020) corroborated the feasibility of fixed patterns for Transformer encoder in translation. Child et al. (2019) introduced local and strided patterns to scale SMATT up to very long inputs. Unlike data-driven approaches, whether these patterns could generalize to different tasks and settings is still an open question.

In contrast, ReLA is both data-driven and efficient. In this respect, our work shares similarity with the explicit sparse Transformer (Zhao et al., 2019) which also delivers faster speed but still depends on top-k sorting as in sparsemax and entmax with k, a tunable hyperparameter. Note that all the above mentioned methods follow the categorical distribution constraint on attentions, while ReLA goes beyond. Thus, unlike ReLA, none of them enables null attentions.

A different type of linear attention model is proposed by Katharopoulos et al. (2020) and Choromanski et al. (2020), who aim at reducing the $O(n^2)$ complexity in SMATT. These models behave fundamentally differently from ReLA, because they eliminate the token-wise modeling rather than introducing sparsity.

The explanatory power of standard attention weights is hotly debated (Wiegreffe and Pinter, 2019; Jain and Wallace, 2019). Much of the criticism stems from the observation that low attention scores do not always imply irrelevance of the corresponding feature, as the information can still flow and its influence can be large (e.g., due to the large magnitude of the corresponding features). In contrast, sparse variants, including ReLA, assign exact zeroes, ensuring that the information flow from the corresponding features within the attention component is cut completely. Even with standard attention, prior studies show some evidence that attention partially reflects linguistic properties. In machine translation, the encoder-decoder attention captures the source-target word alignment to a certain degree (Ghader and Monz, 2017), with recent work further strengthening this via specific induction methods (Ding et al., 2019; Kobayashi et al., 2020; Chen et al., 2020). We apply analysis techniques from previous work to analyze our
models.

3 Background: Attention in Transformer

Many variants of attention mechanism have been developed since its first proposal (Bahdanau et al., 2015; Luong et al., 2015). In this paper, we focus on the one used by Transformer, namely multi-head scaled dot-product attention (MHATT), in an encoder-decoder setup. Given query inputs $X \in \mathbb{R}^{n \times d}$ and a sequence of context items $Y \in \mathbb{R}^{m \times d}$, each head in MHATT summarizes query-relevant context information as follows:

$$\text{SMATT}(X, Y) = \alpha V,$$

with $\alpha = \text{Softmax}(f(Q, K^T))$ ,

$$f(Q, K) = Q^T K W_\theta,$$

$$f(Q, K^T) = f(Q, K).$$

where $Q \in \mathbb{R}^{n \times d}$, $K \in \mathbb{R}^{d \times m}$, $V \in \mathbb{R}^{H \times d}$, and $\alpha \in \mathbb{R}^{n \times m}$ is the attention weight, which estimates the degree of relevance between one query input and each context. The softmax normalizes the scores and ensures that the attention weights $\alpha$ define a categorical distribution. $f(\cdot)$ is a scoring function. Different attention mechanisms make different choices for $f(\cdot)$, but the use of softmax, or its sparsified variants, is universal.

SMATT in Transformer adopts the scaled dot product for $f(\cdot)$, which is further extended by MHATT to allow for parallel attentions in different sub-spaces over the same inputs:

$$\text{MHATT}(X, Y) = [\text{SMATT}_1, \ldots, \text{SMATT}_H] W_o,$$

where $[\cdot, \cdot]$ denotes the concatenation operation, $H$ is the number of heads, $W_o \in \mathbb{R}^{Hd_h \times d}$ are output transformation parameters, and $d = H d_h$.

In the encoder-decoder framework, MHATT is used in three different ways: Encoder Attention, Decoder Attention and Cross Attention, modeling intra-source, intra-target, and source-target dependencies, respectively. Transformer performs layered MHATT with residual connection and layer normalization (Ba et al., 2016) to handle variations of token-wise dependencies. The learning of MHATT is guided by the training objective, often without direct supervision.

4 Rectified Linear Attention

We argue that the use of the softmax function in SMATT (Eq. 1) has two undesirable consequences:

- The attention mass is densely distributed over all context items, even the ones that are intuitively irrelevant.
- The attention mass for each query is constant, although the relevance of context may vary.

Both potentially hamper interpretability and even performance.\(^3\)

As an alternative to sparsified softmax variants (Peters et al., 2019; Correia et al., 2019), we go one step further and consider whether the softmax, or broadly the categorical distribution, could be avoided completely.

Model Structure We offer an answer to the question by proposing rectified linear attention (ReLA). ReLA abandons the distribution assumption and adopts linear activation instead. It is formulated as follows (see Figure 1(b) for illustration):

$$\text{ReLA}(X, Y) = \text{LN}(\alpha V),$$

with $\alpha = \text{ReLU}(f(Q, K^T))$, where $f(\cdot)$ denotes any scoring function as in Eq. 1, $\text{LN}(\cdot)$ denotes variants of layer normalization (Ba et al., 2016; Zhang and Sennrich, 2019), and $\text{ReLU}(\cdot) = \max(0, \cdot)$ is the rectified linear unit. Note here, we describe our model by assuming only one attention head for clarity. In the multi-head ReLA, we impose the normalization $\text{LN}(\cdot)$ on the concatenated head representation rather than each single head separately.

Unlike SMATT, ReLA prunes out all negative scores of low query-context relevance, automatically ensuring the sparse property of the attention weight $\alpha \in \mathbb{R}^{n \times m}$. Besides, ReLA allows for null attention, where it assigns zero scores to all context items (i.e. some rows of $\alpha$ are zero vectors), effectively switching off the corresponding attention head for certain queries. Nevertheless, the outputs of ReLU in Eq. 3 are often of different scales and varied variance, causing gradient instability and also optimization failure.

Stabilization with Normalization A common strategy in deep learning to stabilize neuron activations is to apply layer normalization $\text{LN}(\cdot)$ (Ba

\(^3\)As an anecdotal example, Voita et al. (2018) performed an analysis of attention to previous sentences in MT, and found that the model has learned to generally attend to the end-of-sentence symbol as a way to ignore context. While this might be an effective strategy for instances where context matters little, this reduces the interpretability of attention.
We follow this strategy and normalize each representation \( z \in \mathbb{R}^{d_h} \) in the attention outputs \((\alpha \mathbf{V})\) with root mean square layer normalization (Zhang and Sennrich, 2019, RMSNorm):

\[
\text{LN}(z) = \text{RMSNorm}(z) = \frac{z}{\text{RMS}(z)} \odot g, \tag{4}
\]

where \( \odot \) denotes the element-wise multiplication, \( \text{RMS}(\cdot) \) calculates the root mean square statistic, and \( g \in \mathbb{R}^{d_h} \) is the gain parameter, usually initialized at 1. We adopt RMSNorm rather than the vanilla LayerNorm (Ba et al., 2016) for ReLA because it avoids the re-centering constraint, being more flexible and computationally simpler.

Although RMSNorm largely smooths gradients, our preliminary experiments show that ReLA still suffers from unstable gradients during early training, delivering suboptimal convergence. We propose two solutions, corresponding to two variants of ReLA, to solve this problem by down-scaling ReLA’s activations.

**ReLA-\(i\)** changes the initialization of the gain parameter \( g \) in RMSNorm with a uniform xavier initializer: \( g \sim \mathcal{U}(\sqrt{\frac{3}{d_h}}, \sqrt{\frac{3}{d_h}}) \).

**ReLA-\(g\)** adds a simple gating function to the normalization:

\[
\text{LN}(z) = \sigma (w \odot z) \odot \text{RMSNorm}(z), \tag{5}
\]

where \( \sigma(\cdot) \) denotes the sigmoid function, and \( w \in \mathbb{R}^{d_h} \) is a trainable parameter.

We compare their performance in our experiments. The only overhead due to ReLA, compared to SMAT, is the added normalization layer and it is marginal. ReLA is a drop-in replacement of SMAT, and we apply it to Transformer for all three types of attention.

### 5 Experiments

#### Settings

We take machine translation as the testbed. We conduct experiments on five datasets of varied language pairs and training data sizes, including WMT14 English-German (Bojar et al., 2014, En-De, 4.5M training instances), WMT14 English-French (Bojar et al., 2014, En-Fr, 36M), WMT18 English-Finnish (Bojar et al., 2018, En-Fi, 3.3M), WMT18 Chinese-English (Bojar et al., 2018, Zh-En, 25M), and WMT16 Romanian-English (Bojar et al., 2016, Ro-En, 608K). We evaluate on the official test set from the corresponding year (e.g. newstest2014 for WMT14), and regard the previous year’s test set as the development set (e.g. newstest2013 for WMT14). We preprocess all datasets using the byte pair encoding algorithm (Sennrich et al., 2016) with 32K merging operations. We report detokenized case-sensitive BLEU (Papineni et al., 2002) implemented by SacreBLEU (Papineni et al., 2002) implemented by SacreBLEU (Post, 2018), and also show tokenized case-sensitive BLEU with multi-bleu.perl for ablation studies.

**Model Configuration** We use the Transformer base setting for experiments: model dimension \( d = 512 \), head number \( H = 8 \), head dimension \( d_h = 64 \), 6 layers and FFN size of 2048 (Vaswani et al., 2017). We apply dropout to the residual connections and attention weights, with a rate of 0.1. We tune model parameters using Adam (Kingma and Ba, 2015, \( \beta_1 = 0.9, \beta_2 = 0.98 \)) with label smoothing of 0.1. We schedule the learning rate following Vaswani et al. (2017) with a warmup step of 4K. Each training batch contains around 25K target tokens. For decoding, we average the last 5 checkpoints and adopt beam search with a beam size of 4 and length penalty of 0.6.

Apart from softmax-based SMAT, we consider two additional baselines: sparsemax (Martins and Astudillo, 2016) and 1.5-entmax (Peters 2018, Zh-En, 25M), and WMT16 Romanian-English (Bojar et al., 2016, Ro-En, 608K). We evaluate on the official test set from the corresponding year (e.g. newstest2014 for WMT14), and regard the previous year’s test set as the development set (e.g. newstest2013 for WMT14). We preprocess all datasets using the byte pair encoding algorithm (Sennrich et al., 2016) with 32K merging operations. We report detokenized case-sensitive BLEU (Papineni et al., 2002) implemented by SacreBLEU (Papineni et al., 2002) implemented by SacreBLEU (Post, 2018), and also show tokenized case-sensitive BLEU with multi-bleu.perl for ablation studies.

<table>
<thead>
<tr>
<th>ID</th>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baseline (softmax)</td>
<td>26.9</td>
</tr>
<tr>
<td>2</td>
<td>1 + sparsemax</td>
<td>26.4</td>
</tr>
<tr>
<td>3</td>
<td>1 + 1.5-entmax</td>
<td>26.7</td>
</tr>
<tr>
<td>4</td>
<td>1 + ReLU alone</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>4 + RMSNorm</td>
<td>26.0</td>
</tr>
<tr>
<td>6</td>
<td>1 + ReLA-(i)</td>
<td>26.5</td>
</tr>
<tr>
<td>7</td>
<td>1 + ReLA-(g)</td>
<td>26.6</td>
</tr>
<tr>
<td>8</td>
<td>7 + LayerNorm</td>
<td>26.6</td>
</tr>
<tr>
<td>9</td>
<td>7 + GeLU</td>
<td>26.5</td>
</tr>
<tr>
<td>10</td>
<td>7 + Leaky ReLU</td>
<td>26.5</td>
</tr>
<tr>
<td>11</td>
<td>7 + Encoder Attention Only</td>
<td>26.3</td>
</tr>
<tr>
<td>12</td>
<td>7 + Decoder Attention Only</td>
<td>27.0</td>
</tr>
<tr>
<td>13</td>
<td>7 + Cross Attention Only</td>
<td>27.0</td>
</tr>
</tbody>
</table>

Table 1: SacreBLEU (tokenized BLEU in brackets) for different models and settings on WMT14 En-De. GeLU: Gaussian error linear unit (Hendrycks and Gimpel, 2016); Leaky ReLU: leaky rectified linear unit (Xu et al., 2015). Baseline: Transformer. “-”: optimization failed, where training loss didn’t decrease. Higher BLEU indicates better result.

**Note** here we set \( \text{fan}_\text{in} = \text{fan}_\text{out} = d_h \).
Decoding Speedup

0.0
0.5
1.0
5
10
20
30
40
50
Source Length

Figure 2: Decoding speedup as source length increases on WMT14 En-De. We divide the newstest2014 testset into 10 disjoint groups uniformly according to source length. Results are averaged over 3 runs.

<table>
<thead>
<tr>
<th>Model</th>
<th>#Params</th>
<th>ΔTrain</th>
<th>ΔDecode</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>72.31M</td>
<td>1.00×</td>
<td>1.00×</td>
</tr>
<tr>
<td>sparsemax</td>
<td>72.31M</td>
<td>0.26×</td>
<td>0.54×</td>
</tr>
<tr>
<td>1.5-entmax</td>
<td>72.31M</td>
<td>0.27×</td>
<td>0.49×</td>
</tr>
<tr>
<td>ReLA-g</td>
<td>72.34M</td>
<td>0.93×</td>
<td>0.98×</td>
</tr>
</tbody>
</table>

Table 2: Number of parameters (#Params) and running efficiency for different models on WMT14 En-De. ΔTrain: speedup per training step measured on 500 steps with about 25K target tokens per batch. ΔDecode: translation speedup on newstest2014 with a batch size of 1. We perform 3 runs on a single GeForce GTX 1080 Ti and report average results for the speedups. Higher speedup indicates better efficiency.

5.1 Translation Results

We start with an ablation study for ReLA on WMT14 En-De. Results are given in Table 1.

Ablation on ReLA’s Architecture At the heart of ReLA is its replacement of softmax with ReLU. But simply applying ReLU to SMATT increases gradient instability, resulting in training failure (4). Applying layer normalization to the outputs of the attention model alleviates this problem, albeit sacrificing 0.9 detokenized BLEU ((1)→(5)). By contrast, the proposed solutions, ReLA-i and ReLA-g, yield a detokenized BLEU score of 26.5 (6) and 26.6 (7) respectively, narrowing the quality gap against the baseline. ReLA-g performs slightly better than ReLA-i (+0.1 detokenized BLEU) and on par with 1.5-entmax (-0.1 detokenized BLEU), partially due to the increased gating parameters. In the following experiments and analysis, we mainly report results with ReLA-g (i.e. 7).

RMSNorm vs. LayerNorm Results show that replacing RMSNorm with LayerNorm (7→8) leads to no quality improvement (-0.13 tokenized BLEU). We adopt RMSNorm for ReLA due to its efficiency.

ReLU vs. its Variants We also attempted some smoothed variants of ReLU, such as GeLU (Hendrycks and Gimpel, 2016) and Leaky ReLU (Xu et al., 2015). Results show that these variants (9, 10) yield worse performance than ReLU (-0.1 detokenized BLEU). Dropping those low-relevance attention scores, as ReLA does, benefits translation.

ReLA for Different Attention Types By default, we apply ReLA to all attention sublayers. As shown in Section 3, Transformer includes three types of attentions with different functionalities. We study next how ReLA performs when applied to each attention separately. Results show that incorporating ReLA into the decoder self-attention (12) or encoder-decoder cross attention (13) yields quality gains over Baseline (+0.1 detokenized BLEU). By contrast, only sparsifying encoder-side attentions with ReLA leads to a big quality reduction (-0.6 detokenized BLEU). We argue that the encoder self-attention requires denser token-wise modeling to induce informative features of the source input for translation, compared to the other two attention types, echoing with the findings of Correia et al. (2019).

Efficiency Analysis Table 2 shows the results. Sparsemax and 1.5-entmax run more than 3 and 1.8 times slower than Baseline (softmax) at training and decoding, respectively. We ascribe this to the dedicated inference procedure (such as sorting) both methods require in order to discover the best sparsity patterns (Peters et al., 2019), which reduces efficiency. By contrast, the computation in ReLA-g is much simpler, and training and decoding speed is comparable to the baseline.

Besides, we offer an analysis about the impact of source length on decoding speed. Curves in Figure 2 show consistent efficiency trend across different lengths: ReLA translates slightly slower than Baseline but at least 1.8 times faster than sparsemax and 1.5-entmax.

We also notice that Correia et al. (2019) and Zhao et al. (2019) reported better training efficiency.

et al., 2019; Correia et al., 2019).5 We implement all models with Tensorflow (version 1.13.1).

Notes

5https://gist.github.com/justtheurist/60167e7a95221586be315ae527c3c95
6Note we apply ReLA-g to all attention sublayers so as to avoid the interference of other attention variants. This allows us to fully examine the effectiveness of ReLA.
Table 3: SacreBLEU scores for different models on more WMT translation tasks. Best scores are highlighted in **bold**.

<table>
<thead>
<tr>
<th>Model</th>
<th>WMT14 En-Fr</th>
<th>WMT18 En-Fi</th>
<th>WMT18 Zh-En</th>
<th>WMT16 Ro-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>37.2</td>
<td><strong>15.5</strong></td>
<td>21.1</td>
<td>32.7</td>
</tr>
<tr>
<td>sparsemax</td>
<td>37.5</td>
<td>15.1</td>
<td>19.2</td>
<td><strong>33.5</strong></td>
</tr>
<tr>
<td>1.5-entmax</td>
<td><strong>37.9</strong></td>
<td><strong>15.5</strong></td>
<td>20.8</td>
<td>33.2</td>
</tr>
<tr>
<td>ReLA-g</td>
<td><strong>37.9</strong></td>
<td>15.4</td>
<td>20.8</td>
<td>32.9</td>
</tr>
</tbody>
</table>

Figure 3: Average sparsity rate over heads at each layer for different attention models and types on the WMT14 En-De test set. Larger sparsity rate indicates that more attention scores are exactly zero.

Table 4: Comparison of FLOPs between softmax-based SMA and ReLA-g.

<table>
<thead>
<tr>
<th>Model</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>$3HT^2 - HT$</td>
</tr>
<tr>
<td>ReLA-g</td>
<td>$HT^2 + 10Td + T$</td>
</tr>
</tbody>
</table>

for sparsemax and 1.5-entmax than our results in Table 2. This is due to implementation difference. We re-tested the efficiency of different approaches using Pytorch with an in-house Transformer codebase, and worked with the official entmax implementation. We observe that the training efficiency gap becomes much narrower, where sparsemax, 1.5-entmax and ReLA yield a speedup of $0.87 \times$, $0.90 \times$ and $0.95 \times$, respectively. Although speedups vary across implementations, ReLA shows consistently higher computational efficiency than these sparsified softmax variants.

Why ReLA-g Is Slower Than Softmax? Table 2 and Figure 2 show that ReLA-g runs slower than Baseline. This is because ReLA-g is not just an activation function as in softmax. Apart from ReLU, ReLA-g also includes a gated RMSNorm layer which brings in extra computational overhead. This becomes clearer as we show their FLOPs in Table 4, where $T$ denotes the sequence length.

Take Transformer base ($H = 8, d = 512$) as an example. For translation tasks where sequences often contain fewer than 100 tokens, the FLOPs of softmax is lower than that of ReLA-g ($239K < 592K$, at $T = 100$). But ReLA-g has better scalability with respect to sequence length and would benefit long-sequence modeling ($23.99M > 13.12M$, at $T = 1000$).

Results for More Language Pairs Table 3 summarizes the results. Overall, performance of ReLA-g is competitive to the baseline, with BLEU differences ranging from -0.3 detokenized BLEU (Zh-En) to +0.7 detokenized BLEU (En-Fr), suggesting that ReLA generalizes to different (similar or distant) language pairs. Average performance is 0.5 detokenized BLEU higher than that of sparsemax, and 0.1 detokenized BLEU below that of 1.5-entmax.

5.2 Attention Analysis

Although different sparsified SMATT models achieve comparable translation performance, their learned attention weights $\alpha$ often have different characteristics. In this section, we quantitatively analyze these weights on WMT14 En-De. We first define *layer attention*, the weights averaged over the heads in one layer, to ease our following study. Besides, we obtain the word-level attention weights by merging its subword-level counterparts following Zenkel et al. (2019). We train each model three times with different random seeds on WMT14 En-De and report the average results.

Attention Sparsity The ability to automatically induce sparse attention is one of the key characteristics of ReLA. We next report the sparsity rates, i.e. the fraction of attention weights exactly equal
to 0. We calculate the average sparsity rate over heads for each layer.

Results are shown in Figure 3. We observe that the cross attention has the highest sparsity rate on average, resonating with the fact that word alignment is sparse. Self-attention at lower encoder/decoder layers often has a higher sparsity rate, particularly for sparsemax and 1.5-entmax. In ReLA-$g$, we find that its sparsity rate for the decoder self- and cross-attention tends to increase with layer depth, while that of the encoder self-attention fluctuates. Overall, ReLA-$g$ produces attentions of similar but slightly higher (lower) sparsity rate than 1.5-entmax (sparsemax), learned automatically without any constraint. Note softmax-based SMATT only produces dense attentions, i.e., a sparsity rate of 0.

**Cross Attention vs. Word Alignment** We experiment with the publicly available De-En evaluation set\(^9\) and evaluate the alignment quality with alignment error rate (Och and Ney, 2000, AER). We study normal attention and shifted attention following previous work (Chen et al., 2020; Kobayashi et al., 2020). The former explores attention weights corresponding to decoder outputs (i.e. $\alpha$ in Eq. 1 and 3); the latter, by contrast, skips the weights at the first decoding step, i.e. $\alpha[1:]$, to offset the left padding to the decoder inputs made for auto-regressive generation in Transformer.

Figure 4 shows the results. Regardless of the attention type (normal or shifted), attention re-

\(^9\)https://www-i6.informatik.rwth-aachen.de/goldAlignment/

![Figure 4: AER scores for different models with normal attention (left, $\alpha$) and shifted attention (right, $\alpha[1:]$). Solid curves correspond to the best head per layer; dashed curves are for layer attention. Lower AER score indicates better alignment quality.](image)

<table>
<thead>
<tr>
<th>Model</th>
<th>Normal Attention</th>
<th>Shifted Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AoL</td>
<td>AoH</td>
</tr>
<tr>
<td>softmax</td>
<td>75.95</td>
<td>67.51</td>
</tr>
<tr>
<td>sparsemax</td>
<td>78.17</td>
<td>67.88</td>
</tr>
<tr>
<td>1.5-entmax</td>
<td>78.82</td>
<td>67.69</td>
</tr>
<tr>
<td>ReLA-$g$</td>
<td>64.46</td>
<td>61.64</td>
</tr>
</tbody>
</table>

Table 5: Average AER scores over layers for different models. AoL: average for layer attention; AoH: average for the best head. Best scores are highlighted in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>Enc</th>
<th>Dec</th>
<th>Cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax</td>
<td>0.56</td>
<td>0.31</td>
<td>0.45</td>
</tr>
<tr>
<td>1.5-entmax</td>
<td>0.84</td>
<td>0.39</td>
<td>0.52</td>
</tr>
<tr>
<td>ReLA-$g$ ($\tau=1.00$)</td>
<td>1.26</td>
<td>0.81</td>
<td>0.78</td>
</tr>
<tr>
<td>ReLA-$g$ ($\tau=0.50$)</td>
<td>0.97</td>
<td>0.65</td>
<td>0.69</td>
</tr>
<tr>
<td>ReLA-$g$ ($\tau=0.25$)</td>
<td>0.92</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>ReLA-$g$ ($\tau=0.10$)</td>
<td>0.92</td>
<td>0.66</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 6: Average head diversity score over layers for different models. Enc: encoder attention; Dec: decoder attention; Cross: cross attention. $\tau$: temperature used for the re-normalization.

**Head Diversity** We evaluate head diversity with a generalization of Jensen-Shannon divergence following (Correia et al., 2019) to reflect disagreements between heads. For ReLA-$g$, we re-normalize its attention scores via softmax, and regard the null attention as a special one-hot distribution putting all probability mass to a dummy zero vector, i.e. entropy of 0.

Figure 5 shows the results. We observe that the heads of the encoder self-attention exhibit much higher disagreement than those of the other two attention types for all sparsified attention models. Overall, heads in ReLA-$g$ are in less agreement than with the sparsified softmax alternatives, in most cases across different attention types. This indicates that ReLA-$g$ is capable of inducing heads with different roles (Voita et al., 2019).

Note we convert the attention scores of ReLA-$g$ into categorical distribution via softmax for diver-
JS Divergence
Encoder Attention
Decoder Attention
Cross Attention

Figure 5: Jensen-Shannon (JS) Divergence over heads at each layer for different attention models and types on the WMT14 En-De test set. Higher JS Divergence indicates higher head diversity.

Figure 6: Average null rate over heads at each layer for different attention types in ReLA-g on the WMT14 En-De test set. Null attention corresponds to attention of all zero scores. Dashed curves stand for the results on the hallucinated test set where target sentences are randomly shuffled. Hallucination pairs are assigned with significantly higher null rate for the cross-attention across different layers.

Null Attention

One important feature distinguishing ReLA from (sparsified) softmax is that ReLA allows for null attention where all keys are assigned an attention weight of 0, effectively deactivating the attention head for this query. Figure 6 analyzes the null rate of different attention types, i.e. the fraction of query tokens associated with all zero attention scores. Note all softmax-based variants have a null rate of about or exactly 0.

We find that the encoder self-attention has few null attentions, suggesting that encoder prefers denser connections and also compact information flow. The decoder self-attention yields more null attentions for deeper layers. Together with the findings from Figure 3, it shows that the lower decoder self-attention layers model dense dependency with previous target tokens, while the dependencies in the upper ones become sparser. The cross-attention shows the most interesting phenomenon: an obvious peak at the middle layer. Attention at this layer shows the highest sparsity (Figure 3) with a large null rate variance of 0.256 (over heads), low head disagreement (Figure 5), but best AER score (right, bottom figure in Figure 4). Notice that attentions in ReLA-g are of high diversity. The layer attention at each layer has almost no null attentions.

Diving deeper into these null attentions as shown in Figure 7, we observe diverse specializations: head 0 and 7 capture source-target alignments with varying degrees of sparsity; head 2 has a null rate of 83%, and tends to fire after producing a verb (null rate 0% after AUX, 23% after VERB), attending tokens in the corresponding clause; head 3 has a null rate of 95%, but regularly fires after comma (null rate 0.03%), attending to the relevant source context (the clause boundary has said that in this example). Extra attention matrices are shown in Appendix A.

Is Null Attention Meaningful?

Apart from heads that have learned some sparse specialization, we also find that null attention can be informative for some cross-attention heads which learn an alignment. Specifically, the null rate increases for sentence pairs of low quality where many target tokens lack relevant source translations (see Appendix B). In order to quantify this effect, we create a hallucinated test set with target references randomly shuffled for comparison. The dashed curves in Figure 6 show that ReLA-g associates such hallucination pairs with clearly higher null rate for the cross-attention across different layers.

We next average the null rate of the cross-attention over layers and utilize this metric to rank null attention evaluation. Such re-normalization might have a large impact on the final diversity results. We next explore this impact by adding some temperatures (τ) to α, i.e. ατ (in Eq. 3) before applying softmax. Smaller temperatures will enforce smoothness into the output distribution, so alleviating the emergence of peaked distributions. Table 6 shows the results. The temperature indeed affects the diversity results but does not eliminate the diversity gap, and the diversity of ReLA gradually converges as τ goes smaller.

Null Attention

One important feature distinguishing ReLA from (sparsified) softmax is that ReLA allows for null attention where all keys are assigned an attention weight of 0, effectively deactivating the attention head for this query. Figure 6 analyzes the null rate of different attention types, i.e. the fraction of query tokens associated with all zero attention scores. Note all softmax-based variants have a null rate of about or exactly 0.

We find that the encoder self-attention has few null attentions, suggesting that encoder prefers denser connections and also compact information flow. The decoder self-attention yields more null attentions for deeper layers. Together with the findings from Figure 3, it shows that the lower decoder self-attention layers model dense dependency with previous target tokens, while the dependencies in the upper ones become sparser. The cross-attention shows the most interesting phenomenon: an obvious peak at the middle layer. Attention at this layer shows the highest sparsity (Figure 3) with a large null rate variance of 0.256 (over heads), low head disagreement (Figure 5), but best AER score (right, bottom figure in Figure 4). Notice that attentions in ReLA-g are of high diversity. The layer attention at each layer has almost no null attentions.

Diving deeper into these null attentions as shown in Figure 7, we observe diverse specializations: head 0 and 7 capture source-target alignments with varying degrees of sparsity; head 2 has a null rate of 83%, and tends to fire after producing a verb (null rate 0% after AUX, 23% after VERB), attending tokens in the corresponding clause; head 3 has a null rate of 95%, but regularly fires after comma (null rate 0.03%), attending to the relevant source context (the clause boundary has said that in this example). Extra attention matrices are shown in Appendix A.

Is Null Attention Meaningful?

Apart from heads that have learned some sparse specialization, we also find that null attention can be informative for some cross-attention heads which learn an alignment. Specifically, the null rate increases for sentence pairs of low quality where many target tokens lack relevant source translations (see Appendix B). In order to quantify this effect, we create a hallucinated test set with target references randomly shuffled for comparison. The dashed curves in Figure 6 show that ReLA-g associates such hallucination pairs with clearly higher null rate for the cross-attention across different layers.

We next average the null rate of the cross-attention over layers and utilize this metric to rank diversity evaluation. Such re-normalization might have a large impact on the final diversity results. We next explore this impact by adding some temperatures (τ) to α, i.e. ατ (in Eq. 3) before applying softmax. Smaller temperatures will enforce smoothness into the output distribution, so alleviating the emergence of peaked distributions. Table 6 shows the results. The temperature indeed affects the diversity results but does not eliminate the diversity gap, and the diversity of ReLA gradually converges as τ goes smaller.

Null Attention

One important feature distinguishing ReLA from (sparsified) softmax is that ReLA allows for null attention where all keys are assigned an attention weight of 0, effectively deactivating the attention head for this query. Figure 6 analyzes the null rate of different attention types, i.e. the fraction of query tokens associated with all zero attention scores. Note all softmax-based variants have a null rate of about or exactly 0.

We find that the encoder self-attention has few null attentions, suggesting that encoder prefers denser connections and also compact information flow. The decoder self-attention yields more null attentions for deeper layers. Together with the findings from Figure 3, it shows that the lower decoder self-attention layers model dense dependency with previous target tokens, while the dependencies in the upper ones become sparser. The cross-attention shows the most interesting phenomenon: an obvious peak at the middle layer. Attention at this layer shows the highest sparsity (Figure 3) with a large null rate variance of 0.256 (over heads), low head disagreement (Figure 5), but best AER score (right, bottom figure in Figure 4). Notice that attentions in ReLA-g are of high diversity. The layer attention at each layer has almost no null attentions.

Diving deeper into these null attentions as shown in Figure 7, we observe diverse specializations: head 0 and 7 capture source-target alignments with varying degrees of sparsity; head 2 has a null rate of 83%, and tends to fire after producing a verb (null rate 0% after AUX, 23% after VERB), attending tokens in the corresponding clause; head 3 has a null rate of 95%, but regularly fires after comma (null rate 0.03%), attending to the relevant source context (the clause boundary has said that in this example). Extra attention matrices are shown in Appendix A.

Is Null Attention Meaningful?

Apart from heads that have learned some sparse specialization, we also find that null attention can be informative for some cross-attention heads which learn an alignment. Specifically, the null rate increases for sentence pairs of low quality where many target tokens lack relevant source translations (see Appendix B). In order to quantify this effect, we create a hallucinated test set with target references randomly shuffled for comparison. The dashed curves in Figure 6 show that ReLA-g associates such hallucination pairs with clearly higher null rate for the cross-attention across different layers.

We next average the null rate of the cross-attention over layers and utilize this metric to rank
the WMT14 En-De training corpus (top ranked samples have lower null rate). We randomly sample 100 cases from the top 10K candidates, and another 100 from the bottom 10K for manual analysis. We observe clear quality difference between these two groups: sentence pairs with a low null rate are predominantly good translations (~95% correct), whereas sentence pairs with a high null rate are predominantly mistranslations (~1% correct). Bad translations include sentence pairs with the wrong output language and semantically mismatched sentence pairs. Most interestingly, this null rate metric is sensitive to insertion errors, which are difficult to detect via traditional corpus filtering methods. We note previous work that used attention statistics to identify mistranslations (Rikters and Fishel, 2017), but find null attention more easily interpretable than more subtle changes in attention distribution.

6 Conclusion and Future Work

In this paper, we have presented rectified linear attention (ReLA), a novel softmax-free sparse attention model. ReLA avoids the categorical distribution assumption for attention, and, due to using ReLU as activation function, prunes out all negative attention scores and produces sparse attention. To stabilize model training, we add a normalization layer to attention outputs with a specialized initialization or gating structure. ReLA is data-driven, computationally efficient and is a drop-in replacement of SMATT. Experiments on five machine translation tasks with Transformer demonstrate ReLA’s effectiveness in delivering comparable translation quality to softmax-based baselines. Results also show that ReLA is substantially faster than sparsemax and 1.5-entmax at training and decoding. The attentions learned by ReLA correspond better to word alignment, with high head diversity and sparsity rate. Also, in contrast to alternative sparse attention approaches, ReLA produces null attentions, i.e. a head can assign a total attention of zero for some queries, leading to highly specialized attention heads and showing potential to indicate translation quality.

In the future, we will apply ReLA to other neural models and tasks. We are interested in scaling ReLA to very long inputs (Child et al., 2019; Kitaev et al., 2020), or multi-source architectures where the relevance of each source may vary. In its current formulation, the sparsity level of ReLA emerges from the threshold in ReLU which prunes negative scores. We are interested in ways to manipulate the level of sparsity, or make the threshold differentiable.

Acknowledgments

We thank the reviewers for their insightful feedback. Rico Sennrich acknowledges support of the Swiss National Science Foundation (MUTAMUR; no. 176727).

References


Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In Proceedings of the


A Examples for Null Attention

B Null Attention for Diverse-quality Examples
Figure 8: Null-attention examples at the 3rd cross attention layer. All examples come from the WMT14 En-De test set.
Figure 9: Null-attention examples for multi-clause sentences at the 3rd cross attention layer. All examples come from the WMT14 En-De test set.

Figure 10: Null-attention at the 3rd cross attention layer for the 7-th head. Example comes from the WMT14 En-De training corpus. Top row shows high-quality examples, while bottom row shows low-quality ones. Low-quality examples include insertion errors and mistranslations, which increase the null rate.