# Image Captioning with Sparse LSTM

Yujun Lin
Tsinghua University
linyy14@mails.tsinghua.edu.cn

Song Han
Stanford University
songhan@stanford.edu

Yu Wang
Tsinghua University
yu-wang@tsinghua.edu.cn

William J. Dally
Stanford University
NVIDIA
dally@stanford.edu

## ABSTRACT

Long Short-Term Memory (LSTM) is widely used to solve sequence modeling problems, for example, image captioning. We found the LSTM cells are heavily redundant. We adopt network pruning to reduce the redundancy of LSTM and introduce sparsity as new regularization to reduce overfitting. We can achieve better performance than the dense baseline while reducing the total number of parameters in LSTM by more than 80%, from 2.1 million to only 0.4 million. Sparse LSTM can improve the BLUE-4 score by 1.3 points on Flickr8k dataset and CIDER score by 1.7 points on MSCOCO dataset. We explore four types of pruning policies on LSTM and visualize the sparsity pattern, weight distribution of sparse LSTM and analyze the pros and cons of each policy.

## 1 INTRODUCTION

Automatically describing an image has been an appealing task in recent years. Many innovative, efficient architectures proposed to solve this problem are based on Long Short-Term Memory (LSTM) model for its fascinating effects (Karpathy & Li, 2015; Vinyals et al., 2016; Xu et al., 2015). However, memory bandwidth of the hardware affects inference performance of LSTM, since the matrix-vector multiplication is memory bounded. Along with the storage and energy cost, it brings great challenges to mobile deployment. As claimed in Han et al. (2015) and See et al. (2016), network pruning has a compelling effect on reducing the number of parameters of neural networks.

Inspired by these works, we utilize network pruning to reduce the size of LSTM networks. At the same time, we introduce sparsity as a regularization to help deal with the overfitting during training. We experimented four simple types of pruning policies; and unlike existing work which prunes a pre-trained model in Han et al. (2015), we merge pre-train and pruning stages to shorten training time and show how this technique works with LSTM networks. We managed to improve the performance with proper pruning policy. Compared with NeuralTalk2 (Karpathy & Li, 2015) baseline, on Flickr8k dataset (Cyrus et al., 2010), we improve Bleu-4 score from 18.2 to 19.5. On MSCOCO dataset (Lin et al., 2014), we improve CIDER score from 91.4 to 93.1.

## 2 SPARSE TRAINING ON LSTM NETWORKS

The core of training sparse LSTM networks is pruning: removing small connections. During model initialization, a binary mask full of one is allocated to each weight in LSTM. Training still calls the dense linear algebra library, but we multiply the weight with its corresponding mask after optimizer update step in every iteration. The sparsity, i.e. the proportion of zeros in the mask, determines how many connections are trimmed.

We explore the following choices of the pruning policy: whether sparsity is consistent or not, when to start pruning, when to update masks and based on sort algorithm (Han et al., 2016b) or threshold algorithm (Narang et al., 2017). Simple pruning policies can be divided into four types as shown in Figure 1. Type I is to prune the weights from the second iteration with fixed sparsity. Type II is similar to Type I except that it starts from the second epoch. Type III begins pruning from the second epoch and increases the sparsity step by step while training. All these three types of policies update masks every iteration based on the sort algorithm. Type IV uses a continuously increasing threshold to discard smaller weights and updates masks at regular intervals.
Figure 1: Four types of pruning policies

Type I and Type II have only one hyper-parameter, the sparsity $k$. There are two more hyper-parameters in Type III, the period $f$ to increase sparsity and the epoch $ee$ to stop growing sparsity. Type IV has four hyper-parameters: $si$ is the iteration to start pruning; $ei$ is the iteration to end updating masks; $f$ is the period to update masks; $s$ is the slope to increase threshold.

3 EXPERIMENTS

We run all our experiments with NeuralTalk2 (Karpathy & Li, 2015). It uses a CNN (VGG net) as feature extractor, and the following LSTM network to generate image descriptions. For evaluation, we adopt two widely used datasets: Flickr8k (Cyrus et al., 2010) and MSCOCO (Lin et al., 2014). We select separately two sets of 5K random images from MSCOCO validation dataset as our validation and test sets and use them to report results in the following section. The baseline model we used is trained with default hyper-parameters. It achieved BLEU 1-4 score of [59.3, 40.8, 27.2, 18.2] on Flickr8k test dataset and a CIDER score of 91.4 on MSCOCO test dataset. To explore the impact of pruning on LSTM, we only prune the weights in the LSTM, not the CNN.

Figure 2 shows in detail the learning curves on MSCOCO dataset with sparsity 80%. For Type I and II, the damage on performance brought by pruning is recovered by fine-tuning CNN feature extractor. For Type III, the final rise of sparsity results in a surge of validation loss but soon the loss falls near the baseline. As training goes further, validation loss curves of sparse models decline slowly to a slightly lower level than baseline. It shows that the sparsity of weight matrix contributes to reducing overfitting as regularization.

Figure 2: The validation loss and CIDER score curves for the baseline and sparse training on MSCOCO dataset. The sparsity is 80% (20% non-zero) for all sparse cases.

3.1 IMAGE CAPTION COMPARISON

Table reports the results on two datasets when we pruned 80% of origin LSTM size. Since BLEU score is not a perfect metric, we also visualize some images and corresponding captions in Flickr8k test dataset in Figure 3. Pruning mainly improves originally-worse captions in the baseline, like image 1, 4, 5. More image caption results generated by four types of sparse LSTMs are provided in the appendix. Therefore, although the capacity of LSTM is significantly reduced, the model itself is not badly damaged and somewhat improves its performance. We can roughly have a performance ranking: Type I $>$ Type II $\approx$ Type IV $>$ Type III $\approx$ Baseline.

The default hyper-parameters are provided on [https://github.com/karpathy/neuraltalk2](https://github.com/karpathy/neuraltalk2).
Table 1: Results on Neuraltalk2 (Sparsity: 80%)

<table>
<thead>
<tr>
<th>Type</th>
<th>Flickr8k</th>
<th>MSCOCO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU-1</td>
<td>BLEU-2</td>
</tr>
<tr>
<td>Baseline</td>
<td>59.3</td>
<td>40.8</td>
</tr>
<tr>
<td>Sparse Type I</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60.4 (+1.1)</td>
<td>42 (+1.2)</td>
</tr>
<tr>
<td>Sparse Type II</td>
<td>60.4 (+1.1)</td>
<td>41.7 (+0.9)</td>
</tr>
<tr>
<td>Sparse Type III</td>
<td>60.7 (+1.4)</td>
<td>42.3 (+1.5)</td>
</tr>
<tr>
<td>Sparse Type IV</td>
<td>60.2 (+0.9)</td>
<td>42.0 (+1.2)</td>
</tr>
</tbody>
</table>

Figure 3: Visualization of image captions generated by four sparse LSTM models.

3.2 Visualizing the Sparsity Pattern

We visualize the sparsity pattern of LSTM and can easily distinguish the four gates of LSTM (i, f, o, g) in Figure 4. Different gates are allocated with different sparsity ratio. We also observe that the dynamic range of weight values will expand after pruning. This is reasonable because pruning reduces the number of connections between layers and thus the neural network enhances valuable connections while training. In sorting-based pruning, the shape of the main lobe of weight distribution remains similar to baseline except for a deep notch around zero value. In Type IV which is based on threshold algorithm, pronounced bimodality appears with less expansion on the range of weights than others. These sharp and concentrated peaks help further quantization for network compression [Han et al., 2016a].

Figure 4: Visualization of sparsity pattern and weight distribution in LSTM. (Sparsity: 80%)

4 Conclusion

We explored four types of pruning policy for image caption LSTM to decrease the model size and improve quality. Type I and II have only one hyper-parameter and thus are the easiest way to reduce the number parameters in LSTM and can even achieve better performance on image captioning. Type I can improve BLEU-4 score by 1.3 points on Flickr8k dataset and CIDER score by 1.7 points on MSCOCO dataset. Both Type III and IV can maintain baseline performance, but they require more dedicated work on designing hyper-parameters. We also show that pruning weights in LSTM during the initial training will save training time compared with fine-tuning on the pre-trained model.
References


A More Examples of Image Captioning after Four Types Pruning on Neuraltalk2 (Images from Flickr8k Test Set)

Unrelated to the image

Baseline: a young boy jumping into a swimming pool
Sparse Type I: a young boy is running through a grassy field
Sparse Type II: a black dog is running through a grassy field
Sparse Type III: a black dog is running through the grass
Sparse Type IV: a black dog is running through a grassy field

Somewhat related to the image

Baseline: a boy in a red shirt is swinging on a swing
Sparse Type I: a little boy in a blue shirt is swinging on a swing
Sparse Type II: a boy in a red shirt is swinging on a swing
Sparse Type III: a boy in a red shirt is swinging on a swing
Sparse Type IV: a child in a blue shirt is swinging on a swing

Somewhat related to the image

Baseline: two black dogs are running through the snow
Sparse Type I: two black dogs are playing in the snow
Sparse Type II: two black dogs are playing in the snow
Sparse Type III: two black dogs are playing in the snow
Sparse Type IV: two black dogs are playing in the snow

Describes with minor error

Baseline: a boy in a red shirt is jumping off a swing
Sparse Type I: a boy in a blue shirt is jumping off a swing
Sparse Type II: a boy in a blue shirt is jumping off a swing
Sparse Type III: a boy in a blue shirt is jumping off a swing
Sparse Type IV: a boy in a blue shirt is jumping off a swing

Describes without errors

Baseline: a man is standing on top of a cliff overlooking a lake
Sparse Type I: a dog is running through the water
Sparse Type II: a brown dog is running through the water
Sparse Type III: a brown dog is running through the water
Sparse Type IV: a brown dog is running through the water
*Unrelated to the image*

- Baseline: a woman in a white shirt and black pants is standing in front of a crowd
- Sparse Type I: a woman wearing a white shirt and black hat is standing in front of a crowd
- Sparse Type II: a woman in a white shirt and black pants is standing in front of a crowd
- Sparse Type III: a woman in a blue shirt and sunglasses smiles
- Sparse Type IV: a woman wearing a black shirt and white pants is standing on the sidewalk

*Somewhat related to the image*

- Baseline: a young boy wearing a red shirt is smiling.
- Sparse Type I: a young boy with a blue shirt is smiling.
- Sparse Type II: a young boy wearing a blue shirt is standing on a UNK.
- Sparse Type III: a young boy wearing a blue shirt is looking at the camera.
- Sparse Type IV: a young boy is wearing a blue shirt

*Describes with minor error*

- Baseline: a group of people are standing on a snowy hill
- Sparse Type I: a group of people is standing in front of a building
- Sparse Type II: a group of people is standing in front of a brick wall
- Sparse Type III: a group of children are playing in a field
- Sparse Type IV: a group of people are standing in a field

*Describes without errors*

- Baseline: a brown dog is digging in the sand
- Sparse Type I: a brown dog with a red collar is standing in the water
- Sparse Type II: a brown dog with a collar is standing in the water
- Sparse Type III: a brown dog is running through the water
- Sparse Type IV: a brown dog is standing in the water

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline:

-Sparse Type IV:
- Sparse Type III:
- Sparse Type II:
- Sparse Type I:
- Baseline: