1.1 Citation

Neural Machine Translation in Linear Time by Nal Kalchbrenner, Lasse Espeholt, Karen

Simonyan, Aaron van den Oord, Alex Graves, Koray Kavukcuoglu

(Submitted on 31 Oct 2016 (v1), last revised 15 Mar 2017 (this version, v2))

Published in:
· Proceeding
NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems
Pages 4797-4805

1.2 Task

Neural networks for machine translation either have running time that is super linear in the length of the source and target sequences, or they process the source sequence into a constant size representation, burdening the model with a memorization step. Both of these drawbacks grow more severe as the length of the sequences increases
To overcome this they presented a neural translation model, the ByteNet, and a neural language model, the ByteNet Decoder, that aimed at addressing these drawbacks. The ByteNet used convolutional neural networks with dilation for both the source network and the target network. The ByteNet connected the source and target networks via stacking and unfolded the target network dynamically to generate variable length output sequences. They viewed the ByteNet as an instance of a wider family of sequence-mapping architectures that stack the sub-networks and used dynamic unfolding. The sub-networks themselves may be convolutional or recurrent.

1.3 Data

They did raw character-level machine translation on the NewsTest English-German WMT data.
The number of sentences were 3003, number of german-words were 63078, number of English-words were 67624, number of distinct german-words were 13930 and number of distinct English words were 10458

1.4 Approach

ByteNet architecture is composed of a target network that is stacked on a source network and generates variable-length outputs via dynamic unfolding. The target network, referred to as the ByteNet Decoder, is a language model that is formed of one-dimensional convolutional layers that use dilation and are masked. The source network processes the source string into a representation and is formed of one-dimensional convolutional layers that use dilation but are not masked.

Each sentence is padded with special characters to the nearest greater multiple of 25. Each pair of sentences is mapped to a bucket based on the pair of padded lengths for efficient batching during training. Sub-BN learns bucket-specific statistics that cannot easily be merged across buckets, this was tackled by circumventing this issue by simply searching over possible target intervals as a first step during decoding with a beam search; each hypothesis uses Sub-BN statistics that are specific to a target length interval.

1.5 Evaluation

They took BLEU points to compare the results that they achieved.
They evaluated the ByteNet on raw character-level machine translation on the English-German WMT benchmark. The ByteNet achieved a score of 18.9 and 21.7 BLEU points on, respectively, the 2014 and the 2015 test sets; these results approach the best results obtained with other neural translation models that have quadratic running time

The ByteNet used in the experiments had 15 residual blocks in the source network and 15 residual blocks in the target network. As in the ByteNet Decoder, the residual blocks were arranged in sets of five with corresponding dilation rates of 1,2,4,8 and 16. They used residual blocks with ReLUs and Sub-BN. The number of hidden units d was 892. The size of the kernel in the source network was 1×5, whereas the size of the masked kernel in the target network was 1×3. They used bags of character n-grams as additional embeddings at the source and target inputs: for n>2 they pruned all n-grams that occur less than 500 times. For the optimization they used Adam with a learning rate of 0.003.