Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Recently, recurrent neural networks (RNNs) have been augmented with "attention" mechanisms which compute a fixed-length representation of entire sequences. We propose a simplified model of attention which is applicable to feed-forward neural networks and demonstrate that it can solve some long-term memory problems (specifically, those where temporal order doesn't matter). In fact, we show empirically that our model can solve these problems for sequence lengths which are both longer and more widely varying than has been shown for RNNs.

1 Models for Sequential Data

Many problems in machine learning are best formulated using sequential data, i.e. data where a given observation may be dependent on previous observations. Such problems can be coarsely classified as sequence transduction (producing a new sequence given an input sequence), sequence classification (producing a single label or vector from an entire sequence), or sequence generation (producing a sequence from no input) tasks. Appropriate models for these tasks must be able to capture temporal dependencies in sequences, potentially of arbitrary length.

1.1 Recurrent Neural Networks

One such class of models are recurrent neural networks (RNNs). Recent advances in optimization techniques [], initialization methods [], and model architectures [] have facilitated the success of RNNs in a wide variety of fields, including machine translation [], speech recognition [], handwriting recognition [], ... This variety of applications demonstrates their applicability to most machine learning tasks involving sequences. In their simplest form, RNNs can be considered as a function f whose output at time t depends on input x_t, the previous state h_{t-1}, and the model's learnable parameters θ:

$$h_t = f(x_t, h_{t-1}, \theta)$$

In the supervised setting, the parameters θ are optimized with respect to a loss function which measures the performance of f. A common approach is to use backpropagation through time [], which "unrolls" the RNN over time steps to compute the gradient of θ with respect to the loss.

Despite the success of RNNs, their recursive nature makes them hard to both optimize and parallelize. Because the same function f is applied repeatedly over time, the gradient with respect to θ can easily explode or vanish []. The use of gating architectures [], sophisticated optimization techniques [], gradient clipping [], and/or careful initialization [] can help mitigate this issue. However, these approaches do not solve this problem, and as a result RNNs are in practice typically only applied in tasks where sequential dependencies span at most hundreds of time steps []. Very long sequences can also make training computationally inefficient due to the fact that RNNs must be evaluated sequentially and cannot be fully parallelized.

Attention

Simplified attention

2 Toy Long-Term Memory Problems

Different tasks

Previous results

2.1 Fixed-Length Experiment

2.2 Variable-length Experiment

3 Limitations