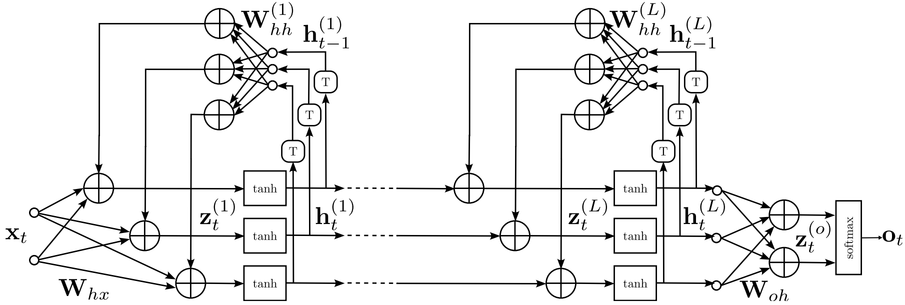
5. Sequence modeling with recurrent neural networks

5.3. Deep Recurrent Neural Networks and Bidirectional Neural Networks

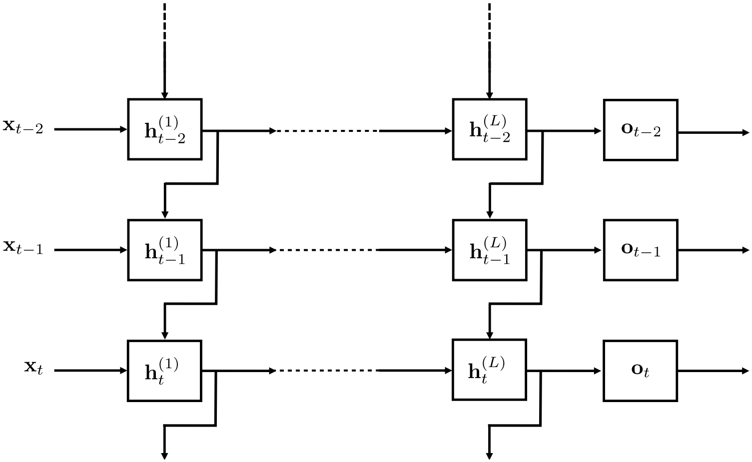
Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup



- The deep RNN follows the same idea as the one in the MLP or CNN: construct a sequence of hidden states $\mathbf{h}^{(l)}$, $1 \leq l \leq L$

Deep Recurrent Neural Networks

Unrolled GRNN

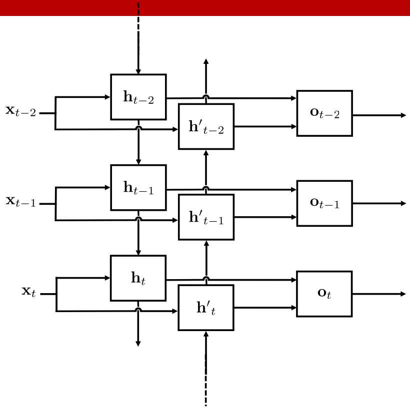


- The GRNN can be represented in a compact, unrolled way.

- The DRNN has a training that is similar to the one of the standard RNN.
- The generalization of the training can be found by computing the gradient with respect to each hidden state $\mathbf{h}_t^{(l)}$ and then the expression of the backpropagated error.
- Each weight matrix $\mathbf{W}_{hh}^{(l)}$ is trained with the product of the backpropagation error and its input $\mathbf{h}_t^{(l-1)}$.
- The derivation is left as an exercise for the student.

Bidirectional Recurrent Neural Networks

Structure



- Bidirectional RNNs make sense when the prediction tasks can extract information of input patterns in both time directions.
- A common example is the task of part of speech detection.

- The BRNN has forward and backward state equations.

$$\begin{aligned}\mathbf{h}_t &= \tanh\left(\mathbf{W}_{hx}^\top \mathbf{x}_t + \mathbf{W}_{hh}^\top \mathbf{h}_{t-1} + \mathbf{b}_h\right)\\ \mathbf{h}'_t &= \tanh\left(\mathbf{W}_{h'x}^\top \mathbf{x}_t + \mathbf{W}_{h'h'}^\top \mathbf{h}'_{t+1} + \mathbf{b}_{h'}\right)\end{aligned} \tag{1}$$

- And the output is computed as

$$\mathbf{o}_t = \operatorname{softmax}\left(\mathbf{W}_{oh}^\top \mathbf{h}_t + \mathbf{W}_{oh'}^\top \mathbf{h}'_t + \mathbf{b}_o\right) \tag{2}$$

- The BRNN has also a training that is similar to the one of the standard RNN, so the derivation is left as an exercise for the student.