Contents

1 ON-LSTM

1.1 My takeaways

1. The key designs of ON-LSTM.
   1. ordered neurons implemented through a new activation `cumax()`.
   2. introduces a vector of master input and forget gates.
      - master input/forget gates ensure that some neurons are always more frequently updated than other neurons.
2. No theoretical analysis. The model architecture design is from intuition and inspiration.

1.2 The problem to address in this paper

1. Natural language is hierarchically structured.
2. There are some existing research and evidence show that LSTM with sufficient capacity potentially implements syntactic processing mechanisms by ***encoding the tree structure implicitly***.

This paper proposes to design a better architecture ***equipped with an inductive bias towards learning such latent tree structures***.

1.3 Approach: ON-LSTM

1.3.1 1. Ordered Neurons

- *Ordered neurons* is an inductive bias that forces neurons to represent information at different time-scales.
- Neurons are split into high-ranking and low-ranking neurons.

low-ranking neurons	high-ranking neurons
contains long-term or global information that lasts for several time steps to the entire sentence	encodes local information that lasts only one or a few time steps.

The differentiation between high-ranking and low-ranking neurons is learned in a completely data-driven fashion by controlling the update frequency of single neurons 1. to erase (or update) high-ranking neurons, the model ***should first erase (or update) all lower-ranking neurons***. 1. low-ranking neurons are always updated more frequently than high-ranking neurons others, and order is pre-determined as part of the model architecture.

1.3.2 2. ON-LSTM

Recap standard LSTM equations:

$$\mathbf{f}_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{1}$$
$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) \tag{2}$$
$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) \tag{3}$$
$$\mathbf{c}_t = \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) \tag{4}$$
$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t) \tag{5}$$

- Standard LSTM:
  - $\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{c}_t$.
  - forget gate $\mathbf{f}_t$ controls erasing on cell states
  - input data $\mathbf{i}_t$ controls writing on cell states
  - cell states act indepently on each neurons.
- Modifications ***ON-LSTM*** maded to standard LSTM:
  - replace the update function for the cell state $\mathbf{c}_t$.
  - make input and forget gate for each neuron dependent on others

1.3.2.1 2.1 active function `cumax`

The `cumax` activation function: $\hat{g}$ which is the cumulative sum of softmax.

$$\mathbf{g} = \text{cumsum(softmax(...))}$$

1. $\mathbf{g} = [0, ..., 0, 1, ..., 1]$ is a binary gate that splits the cell into two segments: the 0-segment and the 1-segment. As a result, the model can apply different update rules on the two segments.
2. $\mathbf{g}$ is the expectation of the binary gate g.
   - ideally, $\mathbf{g}$ should take the form of a discrete variable, but computing gradients when a discrete variable is included in the computation graph is not trivial.
   - $\mathbf{g}s$ hare is a continuous relaxation.

1.3.2.2 2.2 structured gating

1. Introduces two new gates: *master forget gate* $\tilde{f}_t$ and *master input gate* $\tilde{i}_t$

$$\tilde{f}_t = \text{cumax}(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{t-1} + b_{\tilde{f}}) \tag{6}$$
$$\tilde{i}_t = 1 - \text{cumax}(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{t-1} + b_{\tilde{f}}) \tag{7}$$

$\tilde{f}_t$	$\tilde{i}_t$
controls the erasing behavior	controls the writing behavior
monotonically increasing	monotonically decreasing

- master gates only focus on coarse-grained control.
- it is computationally expensive and unnecessary to model them with the same dimensions as the hidden states.
- the paper sets $\tilde{f}_t$ and $\tilde{t}_t$ to bo $D_m = \frac{D}{C}$ where C is the chunk size factor.
- each dimension of the master gates are repeated C times before the element-wise multiplication with LSTM's origian forget gates f_t and input gates i_t.

2. New update rules for c_t based on master gates

$$\omega_t = \tilde{f}_t \circ \tilde{i}_t \tag{8}$$

$$\hat{f}_t = f_t \circ \omega_t + (\tilde{f}_t - \omega_t) = \tilde{f}_t \circ (f_t \circ \tilde{i}_t + 1 - \tilde{i}_t) \tag{9}$$

$$\hat{i}_t = i_t \circ \omega_t + (\tilde{i}_t - \omega_t) = \tilde{i}_t \circ (i_t \circ \tilde{f}_t + 1 - \tilde{f}_t) \tag{10}$$

$$c_t = \hat{f}_t \circ c_{t-1} + \hat{i}_t \circ \hat{c}_t \tag{11}$$

- ω_t represents the overlap of $\tilde{f}_t$ and $\tilde{i}_t$

2 References

1. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks