RNN Bitesized Pieces

Reuben Brasher

March 21, 2022

RNNs

RNN with attention

Basic RNNs

- Any feedforward NN architecture can be used to define RNN

Basic RNNs

- Any feedforward NN architecture can be used to define RNN
- For each time t let x_t be a feature vector

Basic RNNs

- Any feedforward NN architecture can be used to define RNN
- For each time t let x_t be a feature vector
- Concatenate with previous output and feed into net

$$y_t = F\left(x_t, y_{t-1}\right)$$

LSTM and GRU

"Long short-term memory" Hochreiter and Schmidhuber, 1997
"Empirical evaluation of gated recurrent neural networks on sequence modeling" Chung et al., 2014

LSTM and GRU secret sauce

- Entrywise multiplication of two previous layers outputs

$$(x \odot y)_i = x_i y_i$$

LSTM and GRU secret sauce

- Entrywise multiplication of two previous layers outputs

$$(x \odot y)_i = x_i y_i$$

- Called gates because they are conintuous analogs of boolean and gates. If x and y are strictly 1 or 0, then

$$x \wedge y = x \times y$$

LSTM suggestive names

- Activation, $h_t^j = o_t^j \tanh\left(c_t^j\right)$.

LSTM suggestive names

- Activation, $h_t^j = o_t^j \tanh\left(c_t^j\right)$.
- Output gate, $o_t^j = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t\right)$

LSTM suggestive names

- Activation, $h_t^j = o_t^j \tanh\left(c_t^j\right)$.
- Output gate, $o_t^j = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t\right)$
- Memory cell, $c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j$

LSTM suggestive names

- Activation, $h_t^j = o_t^j \tanh\left(c_t^j\right)$.
- Output gate, $o_t^j = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t\right)$
- Memory cell, $c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j$
- Memory content, $\tilde{c}_t^j = \tanh\left(W_c x_t + U_c h_{t-1}\right)$

LSTM suggestive names

- Activation, $h_t^j = o_t^j \tanh\left(c_t^j\right)$.
- Output gate, $o_t^j = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t\right)$
- Memory cell, $c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j$
- Memory content, $\tilde{c}_t^j = \tanh\left(W_c x_t + U_c h_{t-1}\right)$
- Forget gate, $f_t^j = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right)$

LSTM suggestive names

- Activation, $h_t^j = o_t^j \tanh\left(c_t^j\right)$.
- Output gate, $o_t^j = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t\right)$
- Memory cell, $c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j$
- Memory content, $\tilde{c}_t^j = \tanh\left(W_c x_t + U_c h_{t-1}\right)$
- Forget gate, $f_t^j = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right)$
- Input gate, $i_t^j = \sigma\left(W_i x_t + U_i h_{t-1} V_i c_{t-1}\right)$

LSTM rewritten

- Activation, $h_t = o_t \odot \tanh\left(c_t\right)$.

LSTM rewritten

- Activation, $h_t = o_t \odot \tanh(c_t)$.
- Output gate, $o_t = \sigma(A_o(x_t, h_{t-1}, c_t))$

LSTM rewritten

- Activation, $h_t = o_t \odot \tanh(c_t)$.
- Output gate, $o_t = \sigma(A_o(x_t, h_{t-1}, c_t))$
- Memory cell, $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$

LSTM rewritten

- Activation, $h_t = o_t \odot \tanh(c_t)$.
- Output gate, $o_t = \sigma(A_o(x_t, h_{t-1}, c_t))$
- Memory cell, $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
- Memory content, $\tilde{c}_t = \tanh(A_c(x_t, h_{t-1}))$

LSTM rewritten

- Activation, $h_t = o_t \odot \tanh(c_t)$.
- Output gate, $o_t = \sigma(A_o(x_t, h_{t-1}, c_t))$
- Memory cell, $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
- Memory content, $\tilde{c}_t = \tanh(A_c(x_t, h_{t-1}))$
- Forget gate, $f_t = \sigma(A_f(x_t, h_{t-1}, c_{t-1}))$

LSTM rewritten

- Activation, $h_t = o_t \odot \tanh(c_t)$.
- Output gate, $o_t = \sigma(A_o(x_t, h_{t-1}, c_t))$
- Memory cell, $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
- Memory content, $\tilde{c}_t = \tanh(A_c(x_t, h_{t-1}))$
- Forget gate, $f_t = \sigma(A_f(x_t, h_{t-1}, c_{t-1}))$
- Input gate, $i_t = \sigma(A_i(x_t, h_{t-1}, c_{t-1}))$

GRU suggestive names

- Activation, $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$.

GRU suggestive names

- Activation, $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$.
- Update gate, $z_t = \sigma\left(A_z\left(x_t, h_{t-1}\right)\right)$

GRU suggestive names

- Activation, $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$.
- Update gate, $z_t = \sigma\left(A_z\left(x_t, h_{t-1}\right)\right)$
- Candidate activations, $\tilde{h}_t = \tanh\left(A\left(x, r \odot h_{t-1}\right)\right)$

GRU suggestive names

- Activation, $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$.
- Update gate, $z_t = \sigma\left(A_z\left(x_t, h_{t-1}\right)\right)$
- Candidate activations, $\tilde{h}_t = \tanh\left(A\left(x, r \odot h_{t-1}\right)\right)$
- Reset gate, $r_t = \sigma\left(A_r\left(x_t, h_{t-1}\right)\right)$

Sequence to sequence with attention

"Neural machine translation by jointly learning to align and translate" Bahdanau, Cho, and Bengio, 2014

Attention Layers

- Let x_j be the input sequence and h_j encoding by RNN.

Attention Layers

- Let x_j be the input sequence and h_j encoding by RNN.
- Let y_i be the target sequence, and s_i a hidden state.

Attention Layers

- Let x_j be the input sequence and h_j encoding by RNN.
- Let y_i be the target sequence, and s_i a hidden state.
- $$s_i = f\left(s_{i-1}, y_{i-1}, c_i\right)$$

Attention Layers

- Let x_j be the input sequence and h_j encoding by RNN.
- Let y_i be the target sequence, and s_i a hidden state.
-
$$s_i = f\left(s_{i-1}, y_{i-1}, c_i\right)$$
- c_i, called the context vector is
$$c_i = \sum_j \alpha_{ij} h_j$$

Attention Layers

- Let x_j be the input sequence and h_j encoding by RNN.
- Let y_i be the target sequence, and s_i a hidden state.
- $$s_i = f\left(s_{i-1}, y_{i-1}, c_i\right)$$
- c_i, called the context vector is
$$c_i = \sum_j \alpha_{ij} h_j$$
- α_{ij} is the importance of h_j for s_i
$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_k \exp\left(e_{ik}\right)}$$
where $e_{ij} = a\left(a_{i-1}, h_j\right)$.

References I

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473*.

Chung, Junyoung et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.