Graves 2013, "Generating Sequences with Recurrent Neural Networks"

Johannes Bausch and Jack Kamm

14 November 2017

Outline

1. RNN and LSTM
2. Generating text
3. Generating handwriting

Table of Contents

RNN

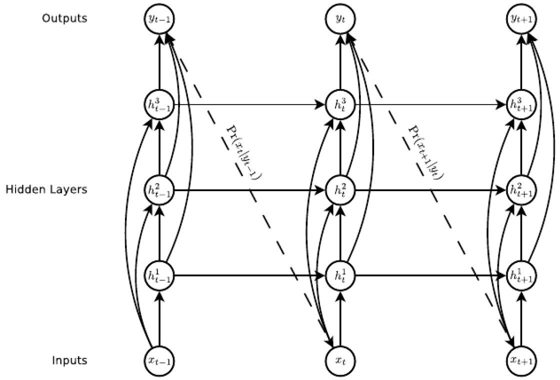


Figure 1: **Deep recurrent neural network prediction architecture.** The circles represent network layers, the solid lines represent weighted connections and the dashed lines represent predictions.

$$h_t^1 = \mathcal{H}(W_{ih^1} x_t + W_{h^1 h^1} h_{t-1}^1 + b_h^1)$$
$$h_t^n = \mathcal{H}(W_{ih^n} x_t + W_{h^{n-1} h^n} h_t^{n-1} + W_{h^n h^n} h_{t-1}^n + b_h^n)$$
$$\hat{y}_t = b_y + \sum_{i=1}^{N} W_{h^n y} h_t^n$$
$$y_t = \mathcal{Y}(\hat{y}_t)$$

Likelihood of input, loss function:

$$\mathbb{P}(\mathbf{x}) = \prod_{t=1}^{T} \mathbb{P}(x_{t+1} \mid y_t)$$
$$\mathcal{L}(\mathbf{x}) = -\sum_{t=1}^{T} \log \mathbb{P}(x_{t+1} \mid y_t)$$

LSTM

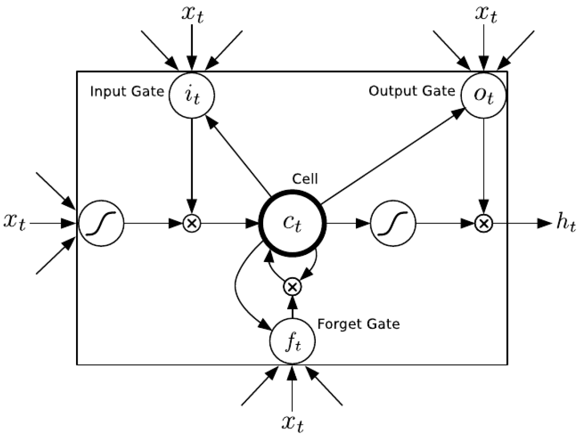


Figure 2: **Long Short-term Memory Cell**

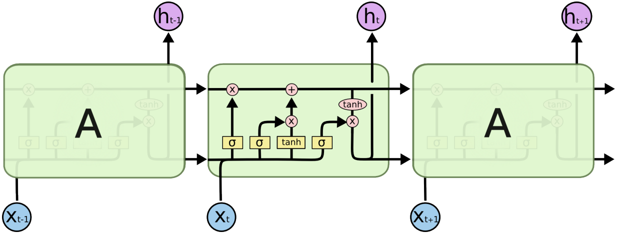
Instead of

$$h_t^n = \mathcal{H}(W_{ih^n} x_t + W_{h^{n-1}h^n} h_t^{n-1} + W_{h^n h^n} h_{t-1}^n + b_h^n)$$

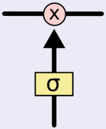
do we have instead the following?

$$h_t^n, c_t^n = \mathcal{H}(x_t, c_t^{n-1}, h_t^{n-1}, h_{t-1}^n)$$

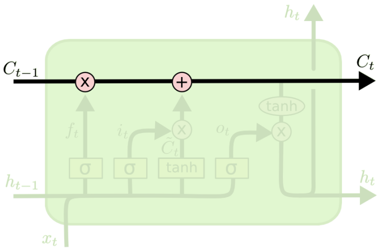
For a really nice explanation of LSTM see:
`colah.github.io/posts/2015-08-Understanding-LSTMs`



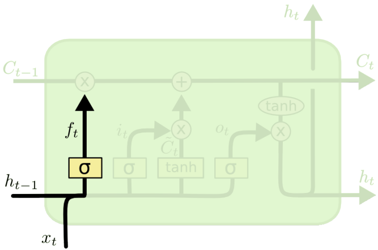
Gate=sigmoid + multiplication



0='nothing thru'
1='all thru'

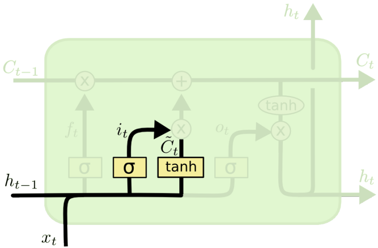


Information flows along cell state



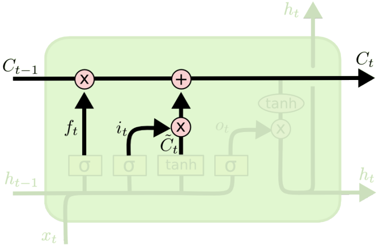
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Forget gate controls what to keep from previous



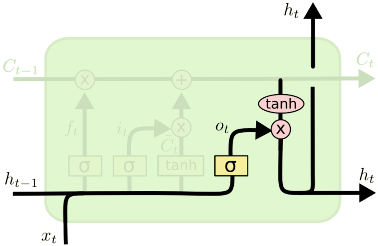
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Input gate controls what to add from input



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Update cell state from forget and input gates



$$o_t = \sigma \left(W_o \left[h_{t-1}, x_t \right] + b_o \right)$$

$$h_t = o_t * \tanh \left(C_t \right)$$

Output gate controls what to output from cell state

Backpropagation thru time

- Train with "backpropagation thru time"
  - Turn head sideways and do backpropagation
  - i.e. reverse-mode automatic differentiation
- Truncated backpropagation thru time
  - Improve numerical stability by truncating gradients if they get too big as we go backwards thru the sequence

Reference other papers?

Table of Contents

Text Prediction

Basic framework for text prediction
$y_t = \mathbb{P}(x_{t+1})$
Per-character vs per-word prediction

Penn Treebank Test Set

Penn Treebank
Perplexity? BPC?
Regularization schemes

Wikipedia Experiments

- Wikipedia experiments
- `karpathy.github.io/2015/05/21/rnn-effectiveness` has some nice visualizations for understanding what's going on, e.g. what the inner neurons represent

Table of Contents

Handwriting experiments

- Handwriting experiments
- Mixture density outputs
  - Basic framework for real valued outputs
- Figure 10 is cool

Handwriting Synthesis

- Handwriting synthesis
- Input, output sequences have different lengths
- Biased vs unbiased vs primed sampling