Lecture 9. Recurrent Neural Networks

Lecture 9. Recurrent Neural Networks

Alex Avdyushenko

Kazakh-British Technical University

November 5, 2022

Good afternoon, dear students. Today we move on to a discussion of recurrent neural networks. This is a very interesting approach, also similar to human information processing.

Imagine yourself reading a book, left to right, top to bottom, sentence by sentence. It turns out that neural networks now can also do something like this.

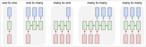
Great, let's start with traditional five-minutes questions. Please, write answers or send photos with them directly to me in private messages here in Teams or may be in Telegram, but choose only one option please :)

Disadvantages of Convolutional Neural Networks

Or why we need recurrent ones :)

- the input is only fixed-dimensional vectors (e.g. 28×28 images)
- the output is also a fixed dimension (for example, probabilities of 1000 classes)
- fixed number of computational steps (i.e. network architecture)

A. Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks



So, how to eliminate all these disadvantages? There are several different architectures. Let's look at them one by one.

Sequential Processing of Fixed Input

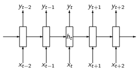
J. Ba, V. Mnih, K. Kavukcuoglu. Multiple Object Recognition with Visual Attention

The interesting fact is, that different recurrent architectures have also been successfully applied to problem with fixed input, such as object recognition for example.

And even image generation too!

K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra. DRAW: A Recurrent Neural Network For Image Generation

Recurrent Neural Network scheme



Ok, so what is inside the recurrent network, how this model works? You have some iteration proccess, where on each step one and the same function changes the hidden vector of model.

Recurrent Neural Network

We process the sequence of vectors x with **one and the same** function with parameters:

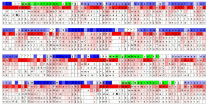
$$h_t = f_W(h_{t-1}, x_t)$$

f_W is a function parameterized by W
x_t — next input vector
h_t — hidden state

Question
What function can we take as f_W?

Which function, one of the simplest, you know from machine learning?

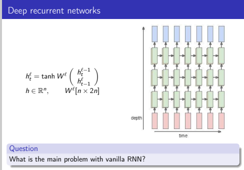
How does it work?



The neuron highlighted in this image seems to get very excited about URLs and turns off outside of the URLs. The LSTM is likely using this neuron to remember if it is inside a URL or not.

In his work on the analysis of recurrent networks, Andrey Karpaty tried to understand the mechanisms of this model.

You see, he discovered really interesting facts about RNN mechanics and showed it well with specific examples.



Let's go on and talk about deep RNN with several layers. So you simply add new layers in depth.

It is difficult for RNN to remember a long context.

Long short-term memory (LSTM)

$$W^\ell[4n \times 2n]$$

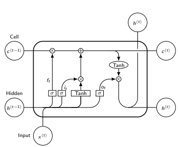
$$\begin{pmatrix} i \\ f \\ o \\ c_t' \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^\ell \begin{pmatrix} h_t^{\ell-1} \\ h_{t-1}^{\ell} \end{pmatrix}$$

$$c_t^\ell = f \odot c_{t-1}^\ell + i \odot c_t'$$

$$h_t^\ell = o \odot \tanh(c_t^\ell)$$

$\odot$ — component-wise product

LSTM modules come to the rescue! Now you have three gates and new vector in the model — cell state.



Look at this beatiful scheme of LSTM module. Cell state is flowing through the module, somewhat reminiscent of the skip-connection from Resnet.

Next formulas are quite simple, since they essentially perform linear transformations.

GRU: Gated Recurrent Unit

$$
\begin{aligned}
u_t &= \sigma(W_{xu} x_t + W_{hu} h_{t-1} + b_u) \\
r_t &= \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r) \\
h'_t &= \tanh(W_{xh'} x_t + W_{hh'}(r_t \odot h_{t-1})) \\
h_t &= (1 - u_t) \odot h'_t + u_t \odot h_{t-1}
\end{aligned}
$$

Only h_t is used, vector c_t is not introduced.
Update-gate instead of input and forget.
The reset gate determines how much memory to move forward from the previous step.

And finally another one module is GRU — Gated Recurrent Unit. In fact it is more or less equivalent to LSTM, only it uses fewer gates and doesn't introduce the cell state vector c_t.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. A
was some unprocessed data that should have been added to the
this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page
away, because LaTeX now knows how many pages to expect for t
document.