9.2. Long Short Term Memory Units

Lecture based on "Dive into Deep Learning" **http://D2L.AI** (Zhang et al., 2020)

Prof. Dr. Christoph Lippert

Digital Health & Machine Learning

One of the earliest approaches to address long-term information preservation and short-term input skipping was the LSTM by Hochreiter and Schmidhuber, 1997.
It introduces **memory cells** that take the same shape as the hidden state.
To control a memory cell it uses a number of gates inspired by logic gates of a computer.

1. the *output* gate

   To read out the entries from the current cell.

2. the *input* gate

   To decide when to read data into the cell.

3. a *forget* gate

   To reset the contents of the cell.

The motivation for this design is again to be able to decide when to remember and when to ignore inputs into the latent state.

The data feeding into the LSTM gates is

- the input at the current time step $\mathbf{X}_t$
- the hidden state of the previous time step $\mathbf{H}_{t-1}$.

These inputs are processed by

- a fully connected layer
- a sigmoid activation function σ

to compute the values of input, forget and output gates.
As a result, the three gate elements all have a value range of $[0, 1]$.

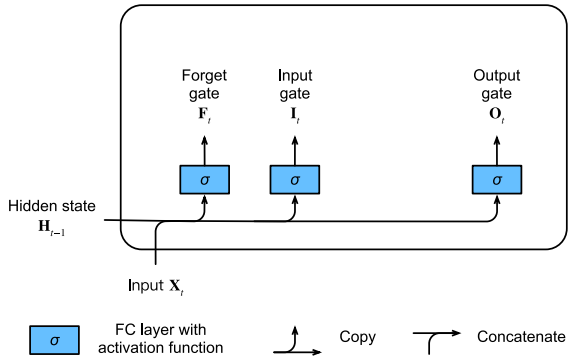

Figure: Calculation of input, forget, and output gates in an LSTM.

For h hidden units and a mini-batch of size n

- The input is $\mathbf{X}_t \in \mathbb{R}^{n \times d}$
  - number of examples: n
  - number of inputs: d
- hidden state of the last time step is $\mathbf{H}_{t-1} \in \mathbb{R}^{n \times h}$.

Correspondingly the gates are defined as follows:

- the input gate is $\mathbf{I}_t \in \mathbb{R}^{n \times h}$

$$\mathbf{I}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i)$$

- the forget gate is $\mathbf{F}_t \in \mathbb{R}^{n \times h}$

$$\mathbf{F}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f)$$

- the output gate is $\mathbf{O}_t \in \mathbb{R}^{n \times h}$

$$\mathbf{O}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o)$$

With weight and bias parameters:

- $\mathbf{W}_{xi}, \mathbf{W}_{xf}, \mathbf{W}_{xo} \in \mathbb{R}^{d \times h}$
- $\mathbf{W}_{hi}, \mathbf{W}_{hf}, \mathbf{W}_{ho} \in \mathbb{R}^{h \times h}$
- $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^{1 \times h}$

candidate memory cell $\tilde{\mathbf{C}}_t \in \mathbb{R}^{n \times h}$.
Computation is similar to the three gates described above, but using a $\tanh$ function with a value range for $[-1, 1]$ as activation function.
At time step t,

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c)$$

With

- weights $\mathbf{W}_{xc} \in \mathbb{R}^{d \times h}$ and $\mathbf{W}_{hc} \in \mathbb{R}^{h \times h}$
- bias $\mathbf{b}_c \in \mathbb{R}^{1 \times h}$

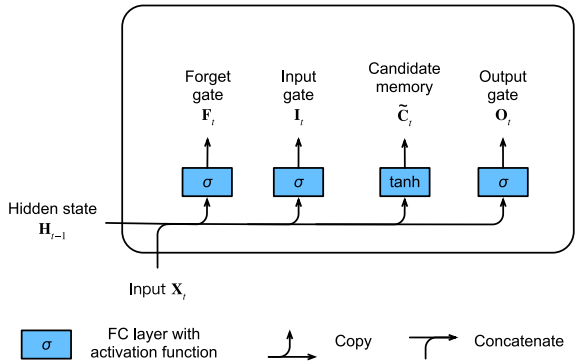

Figure: Computation of candidate memory cells in LSTM.

Memory cell

- input parameter $\mathbf{I}_t$

  governs how much we take new data into account via $\tilde{\mathbf{C}}_t$

- forget parameter $\mathbf{F}_t$

  addresses how much we of the old memory cell content $\mathbf{C}_{t-1} \in \mathbb{R}^{n \times h}$ we retain.

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t.$$

If $\mathbf{F}_t \approx 1$ and $\mathbf{I}_t \approx 0$, the past memory cells will be saved over time and passed to step t. This design allows to

- alleviate the vanishing gradient problem
- better capture dependencies for time series with long range dependencies.

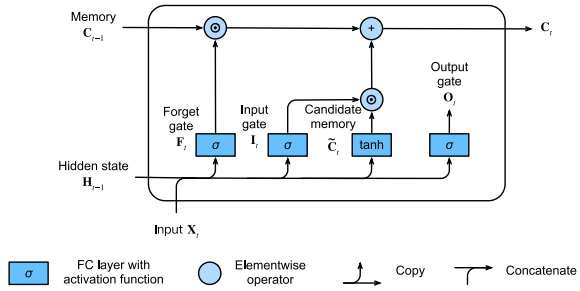

Figure: Computation of memory cells in an LSTM. Here, the multiplication is carried out element-wise.

Hidden state $\mathbf{H}_t \in \mathbb{R}^{n \times h}$.
The output gate is a gated version of the $\tanh$ of the memory cell.

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t).$$

Thus, the values of $\mathbf{H}_t$ lie in $[-1, 1]$.

- If $\mathbf{O}_t \approx 1$
  pass the memory information through to predictor
- If $\mathbf{O}_t \approx 1$
  retain all information only within the memory cell

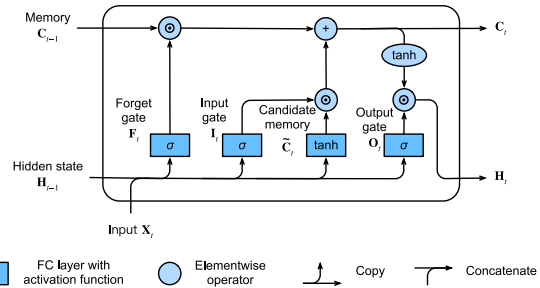

Figure: Computation of the hidden state. Multiplication is element-wise.

Summary

- LSTMs have three types of gates: input, forget and output gates which control the flow of information.
- The hidden layer output of LSTM includes hidden states and memory cells. Only hidden states are passed into the output layer. Memory cells are entirely internal.
- LSTMs can help cope with vanishing and exploding gradients due to long range dependencies and short-range irrelevant data.
- In many cases LSTMs perform slightly better than GRUs but they are more costly to train and execute due to the larger latent state size.
- LSTMs are the prototypical latent variable autoregressive model with nontrivial state control. Many variants thereof have been proposed over the years, e.g. multiple layers, residual connections, different types of regularization.
- Training LSTMs and other sequence models is quite costly due to the long dependency of the sequence. Later we will encounter alternative models such as transformers that can be used in some cases.

[1] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.