**Paper: Long Short-Term Memory (LSTM)**

Name: Milon Hossain

ID: 1610497042

**Abstract:** Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. A novel, efficient and gradient-based method called “Long Short-Term Memory” used in this project. Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing constant error flow through “constant error carrousels” within special units. In comparison with RTRL, BPTT, Recurrent Cascade-Correlation, Elman Nets and Neural Sequence Chunking, LSTM leads to many more successful runs and learn much faster. LSTM also solves complex, artificial long-time lag task that have never been solved by previous recurrent network algorithms.

**Introduction:** Recurrent networks can in principle use their feedback connections to store representations of recent input events in form of activations. This is potentially significant for many applications, including speech processing, non-Markovian control and music composition. The most widely used algorithms for learning what to put in Short-Term memory, however, take too much time or do not work well at all, especially when minimal time lags between inputs and corresponding teacher signals are long.

**Kalman filters:** Puskorius and Feldkamp in 1994 first use Kalman filter techniques to improve recurrent net performance. Since they use “a derivative discount factor imposed to decay exponentially the effects of past derivatives”, there is no reason to believe that their Kalman Filter trained recurrent networks will be useful for very long minimal time lags.

**Long Sort-Term Memory (LSTM):** It is also called memory cells and gate units. To construct an architecture that allows for constant error flow through special, self- connected units without the disadvantage of the naïve approach, here extend the constant error carrousel CEC embodied by the self-connected, linear unit by introducing additional features. A multiplicative input gate unit is introduced to protect the memory contents stored in linear unit from perturbation by irrelevant inputs. Likewise, a multiplicative output gate unit is introduced which protects other units from perturbation by currently irrelevant memory contents stored in linear unit.

**Experimental Approaches:** Which tasks are appropriate to demonstrate the quality of a novel long-time lag algorithm is discussed below. Various approaches have been discussed here. Among them important are:

1. **EMBEDDED REBER GRAMMAR:** Since it allows training sequence with short time lags, it is not a long-time lag problem. By using RTRL and BPTT it shows nicely output gates can be beneficial.
2. **NOISE-FREE AND NOISY SEQUENCE:** A successful run is one that fulfills the following criteria: after training, during 10,000 successive, randomly chosen input sequences, the maximal absolute error of all output units is always low.
3. **NOISE AND SIGNAL OR SAME CHANNEL:** This section serves to illustrate that LSTM does not encounter fundamental problem if noise and signal are mixed on the same input line.
4. **ADDING PROBLEM:** The difficult task in this section is of a type that has never been solved by other recurrent algorithms. It shows that LSTM can solve long time lag problems involving distributed, continuous valued representation.
5. **MULTIPLICATION PROBLEM:** One may argue that LSTM is a bit biased towards tasks such as the adding problem from the previous topic. Solutions to the adding problem may exploit the CES’s built-in integration capabilities.
6. **TEMPORAL ORDER:** In this part, LSTM solves other difficult tasks that have been never been solved by previous recurrent net algorithms. Here we can see that LSTM is able to extract information convoyed by the temporal order of widely separated inputs.

**Limitations of LSTM:**

* The particularly efficient truncated backprop version of the LSTM algorithm will not easily solve problems similar to “Strongly delayed XOR problems”. Constant error flow through CECs can be shown only for truncated LSTM.
* Each Memory cell block unit needs two additional units.
* Generally speaking, due to its constant error flow through CECs within memory cells, LSTM runs into problems similar to those of feedforward nets seeing the entire input string at once.
* LSTM does not have any problem with the notion of “recency” that go beyond those of other approaches.

**Advantages of LSTM:**

* For long time lag problems such as those discussed here, LSTM can handle noise, distribute representations and continuous values.
* LSTM generalizes well- even if the positions of widely separated, relevant inputs in the input sequence do not matter.
* The constant error backpropagation within the memory cells result in LSTM ‘s ability to bridge very long-time lags case of problems similar to those discussion.
* The LSTM algorithms update complexity per weight and time step is essentially that of BPTT, namely O (1).

**Conclusion:** Each memory cell’s internal architecture guarantees constant error flow within its constant error carrousel CEC, provided that truncated backprop cuts off error flow trying to leak out of memory cells. This represents the basis of bridging very long-time lags. Two gate units learn to open and close access to error flow within each memory cell’s CEC. The Multiplicative input gate output gate affords protection of the CEC from perturbation by irrelevant inputs. Likewise, the multiplicative output gate protects other units from perturbation by currently irrelevant memory contents.

**Reference:**

1. https://www.researchgate.net/publication/13853244\_Long\_Short-term\_Memory/link/5700e75608aea6b7746a0624/download

https://www.researchgate.net/publication/13853244\_Long\_Short-term\_Memory/link/5700e75608aea6b7746a0624/download