

## РАЗРАБОТКА АЛГОРИТМА ПЕРЕДАЧИ ИНФОРМАЦИИ МЕЖДУ СЛОЯМИ РЕКУРРЕНТНОЙ НЕЙРОННОЙ СЕТИ

*Пеганов Антон Олегович*

*Студент*

*Факультет нано-, био-, информационных и когнитивных технологий МФТИ,*

*Москва, Россия*

*E-mail: peganov@phystech.edu*

Рекуррентные нейронные сети (далее РНН) широко используются в задачах, связанных с анализом данных с временными зависимостями. В то же время остается открытой проблема создания РНН, которая также способна, подобно сверточным нейронным сетям, распределять процессы обработки абстракций разного уровня, между своими слоями.

Как правило, при использовании РНН (например, LSTM [1] или GRU [2]) с более чем одним слоем контроль над передачей информации между слоями осуществляется с помощью специальных элементов архитектуры, называемых вентилями (gates) и рассчитывающими входные сигналы следующего слоя. В данной работе используется альтернативный подход «всё или ничего», при котором специальный переключатель определяет будут ли производиться вычисления на очередном слое.

К достоинствам этого метода можно отнести экономию ресурсов. Кроме того, редкое использование части слоев сети уменьшает влияние проблемы исчезающих градиентов (vanishing gradient problem).

В данной работе в качестве базовой архитектуры используется простейшая РНН без вентилях с гиперболическим тангенсом в качестве функции активации. Переключатель определяет, когда будет произведена запись текущего состояния сети  $s_t$  в вектор памяти  $m_t$ . В основу механизма переключателя положен прием, описанный в статье [3]. Алгоритм обновления памяти выглядит следующим образом:

1. вычисление скаляра *trigger* согласно формуле

$$trigger = \sigma \left( w_{tr} \cdot [x_t, s_t, m_t]^T \right),$$

где  $\sigma(x) = \frac{1}{1+e^{-x}}$ , скобки  $[ ]$  обозначают конкатинацию векторов,  $x_t$  — входной вектор,  $w_{tr}$  — вектор весовых коэффициентов, используемый в переключателе;

2. если  $trigger > threshold$ , где  $threshold \in (0, 1)$ , в память  $m_t$  записывается вектор  $s_t$ , в противном случае  $m_t$  остается без изменений.

Второй особенностью предлагаемой архитектуры является ее способность выучивать, в какой момент следует задействовать слой РНН. На этапе обучения модели обновление памяти происходит случайно с некоторой фиксированной вероятностью, и при этом в специальных переменных сохраняются значения, хранившиеся в  $s_t$  и  $m_t$  до обновления. Старая пара  $s_t$  и  $m_t$  используется для работы модели параллельно с новой, а различие в значениях функции ошибки, полученных на обновленной и необновленной памяти, служит для обучения переключателя.

РНН с памятью была опробована на задаче моделирования естественного языка. Модель обучалась на 100 МБ текста статей английской википедии вместе XML разметкой, используемых в соревновании Hutter Prize [4]. Сеть последовательно получала по одному символу из текстового файла и каждый раз должна была предсказать следующий.

Было установлено, что переключатель, обученный с помощью описанного метода, срабатывает на первых буквах слов, то есть обучается сегментировать текст на слова. Кроме того, данный подход использования памяти показал лучшие результаты в сравнении с моделью, в которой память обновляется с фиксированной частотой.

Исследования и разработки выполнены при поддержке Фонда поддержки проектов Национальной технологической инициативы и ПАО «Сбербанк». Идентификатор проекта 0000000007417F630002.

### Литература

1. Hochreiter S., Schmidhuber J. Long short-term memory // Neural computation. 1997. Vol. 9, № 8, P. 1735-1780.
2. Cho K., van Merriënboer B., Gulcehre C., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation // In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, October 2014, P. 1724-1735.
3. Graves A. Adaptive Computation Time for Recurrent Neural Networks // <https://arxiv.org/abs/1603.08983>
4. Hutter Prize contest // <http://prize.hutter1.net/>