

## Описание нейронной сети, реализованной для проекта Ask me

Нейронная сеть состоит из трёх модулей: модуль ввода, модуль эпизодической памяти (я его также буду называть эпизодическим модулем) и модуль вывода.

### Модуль ввода

Нейронная сеть получает на входе 2 основные матрицы. Первая матрица кодирует текст: каждый вектор в ней является one-hot-кодированным словом. Предложения в тексте разделяются специальными словами-символами EOF. Вторая матрица кодирует вопрос таким же образом, как и первая, за исключением того, что вопрос состоит из одного предложения.

Вначале векторы из 1-й матрицы с помощью рекуррентной сети GRU преобразовываются в последовательность векторов такой же длины. Затем из этой последовательности берется подпоследовательность векторов, соответствующих символам EOF. Получается, что каждый из векторов этой подпоследовательности соответствует одному предложению и содержит информацию о всем тексте до него. Эти векторы записываются в новую матрицу, которая передается в следующий модуль и будет называться матрицей текста.

Аналогичным образом обрабатывается матрица вопроса. Как уже говорилось, вопрос состоит из одного предложения, поэтому после преобразований мы получаем на выходе один вектор, соответствующий вопросу. Он передается в следующие модули и будет называться вектором вопроса.

### Модуль эпизодической памяти

Обозначим  $GRU(x, y)$  однократное применение ячейки рекуррентной нейронной сети GRU с внутренним состоянием  $y$  к вектору  $x$ , эта функция на выходе выдает вектор - новое состояние рекуррентной сети.

Модуль эпизодической памяти является самым сложным из всех модулей, поскольку содержит в себе вложенные рекуррентные сети. В нем используется много матриц, но все матрицы, размеры которых не определяются входными, имеют одну и ту же размерность. Вычисления происходят по следующим формулам:

$$\begin{aligned} m^0 &= q; \\ m^i &= GRU(e^i, m^{i-1}), \end{aligned}$$

где  $q$  - это вектор вопроса,  $m$  - это вектор эпизодической памяти,  $e$  - это вектор эпизода. Значение  $i$  пробегает от 0 до NUM\_PASSES - константа, равная количеству проходов внутри сети по тексту. Каждый эпизод  $e$  образуется как раз за счет одного прохода. Если  $T$  - это размер текста, то

$$\begin{aligned} h_t^i &= g_t^i \cdot GRU(c_t, h_{t-1}^i) + (1 - g_t^i) \cdot h_{t-1}^i; \\ e^i &= h_T^i, \end{aligned}$$

где  $c_t$  - это  $t$ -е слово текста,  $g_t^i$  - это вектор, получаемый следующим образом:

$$g_t^i = G(c_t, m^{i-1}, q);$$

$$G(c, m, q) = \sigma(W^{(2)} \cdot \tanh(W^{(1)} z(c, m, q) + b^{(1)}) + b^{(2)}).$$

Здесь  $z(c, m, q)$  определяется как конкатенация векторов  $c, m, q, c \circ q, c \circ m, |c - q|, |c - m|$ , где  $\circ$  - это поэлементное произведение.

Стоит отметить, что размер текста в моей реализации должен быть задан как константа NUM\_SENTENCES. Это связано с некоторыми особенностями tensorflow. Тем не менее, на вход сети можно подавать тексты любой длины, не превышающей эту константу.

Выходом эпизодического модуля является последний вектор эпизодической памяти  $m^{NUM\_PASSES}$ .

### Модуль вывода

В моей реализации нейронной сети ответом всегда является ровно одно слово.

Выходом эпизодического модуля является один вектор  $m$ . Ответ  $A$  вычисляется по формуле

$$A = softmax(W^{(a)} \cdot GRU([\bar{0}, q], m)),$$

где  $q$  - это вектор вопроса;  $\bar{0}$  - это вектор из одних нулей, который по сути нужен только для простоты добавления кода в случае, если мы хотим генерировать ответ из нескольких слов;  $W^{(a)}$  - это просто матрица, которая обучается;  $[ ]$  - обозначение конкатенации.

Этот ответ обучается с помощью кросс-энтропии.