# Теоретичні відомості

Текстовий запит можна розбити на слова та представити кожне у числовому форматі, тобто у вигляді вектору дійсних чисел, та, просумувавши ці вектори, поділити отриманий вектор на кількість початкових векторів, отримавши один вектор, координати якого будуть середніми арифметичними відповідних координат початкових векторів, за допомогою такого методу як вкладання слів [1]. Сутність цього методу полягає в тому, що кожному слову ставиться у відповідність вектор дійсних чисел, кожне число відповідає певній характеристиці цього слова, наприклад: семантичному відношенню та різним семантичним подібностям. Для цієї задачі використовується алгоритм *word2vec* [2]. Алгоритм *word2vec* використовує нейромережеву модель для навчання пов'язаностей слів із великого корпусу тексту. Вектори для слів ретельно підбираються таким чином, щоб проста математична функція (косинусна подібність векторів, тобто косинус кута між двома векторами) вказувала на рівень семантичної подібності між словами, представленими цими векторами. Щоб виробляти розподілене представлення слів, word2vec може використовувати будь-яку з двох архітектур моделей: неперервну торбу слів (НТС, continuous bag-of-words, CBOW) та неперервний пропуск-грам(continuous skip-gram). В даному випадку використовується саме неперервна торба слів. В архітектурі неперервної торби слів модель передбачує поточне слово з вікна слів навколишнього контексту. Порядок слів контексту не впливає на передбачення (припущення торби слів). На рисунку 2.1 зображена архітектура моделі CBOW.



|  |
| --- |
| Рис. 2.1 – модель CBOW |

Архітектура моделі CBOW намагається передбачити потрібне слово за словами, які знаходяться біля нього в контексті. Ця модель приймає на вхід розподілені представлення слів у контексті (наприклад, вектор  one-hot-representation, де координата з номером слова у словнику буде дорівнювати 1, а інші – 0), щоб передбачити потрібне слово. Таким чином, на вході моделі буде певна конкретна кількість векторів розмірності [4], де V – це кількість слів у словнику, який використовується для навчання моделі. У  прихованому шарі нейронної мережі кожен вектор множиться на відповідну матрицю розмірності , де E – це гіперпараметр, що відповідає розмірності результуючих векторів, тобто на виході утворюється певна кількість (дорівнює кількості вхідних векторів) векторів розмірності , що перетворюються в один вектор шляхом обчислення середнього значення для відповідних координат кожного вектору. Цей результуючий вектор передається вже в  шар softmax (нормованої експоненційної функції) [5] :

Після обчислення цієї функції активації для даного вектору результатом є вектор розмірності , кожна координата якого є певною характеристикою. Цей вектор буде використовуватись для обчислення косинусу подібності між цим вектором запиту та кожним вектором питання статті із множини відібраних статей.

Для передбачення найбільш відповідних тегів для даного запиту використовуються вентильні рекурентні вузли (GRU - Gated recurrent units) [6]. Повний рекурентний вузол працює наступним чином. На вхід подаються значення вектору входу та значення виходу (при , вектор виходу ). По ним обчислюється претендент на нове значення виходу — вектор вузла скидання (*reset gate vector*) , який обчислюється як функція активації (зазвичай сигмоїд) від матричного виразу по параметрамWb. Незалежно, подібним чином, обчислюється вектор вузла уточнення (*update gate vector*) . Цей вектор містить значення, які визначають, чи варто залишити значення зі старого вектору, чи взяти нове значення. Фактично, це набір «вентилів» (*gate*), які «пропускають» або старе, або нове значення. Далі обчислюється вектор виходу , в якому з ймовірністю береться старе значення з вектору , або з ймовірністю обчислюється нове значення. Формули для обчислень наступні:

, де – добуток Адамара (поелементий добуток матриць)

Змінні:

– вектор входу;

– вектор виходу;

– вектор вузла уточнення;

– вектор вузла скидання;

*–* матриці та вектор параметрів.

Функції активації:

– сигмоїдна функція[7]:

– гіперболічний тангенс[8]:

На рисунку 2.2 зображений повний рекурентний вузол.

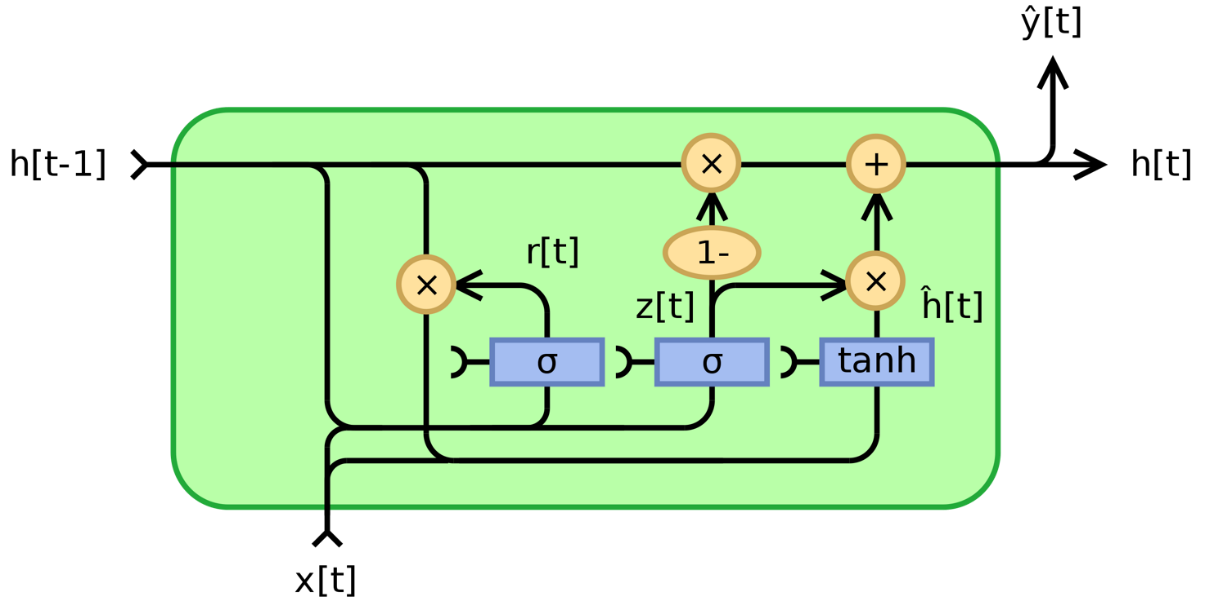


Рис. 2.2 – повний рекурентний вузол

Для передбачення категорії заданого запиту по отриманим тегам використовується модель мультиноміальної логістичної регресії []. У статистиці мультиноміальна логістична регресія — це метод класифікації, який узагальнює логістичну регресію на багатокласові проблеми, тобто з більш ніж двома можливими дискретними результатами. Мультиноміальна логістична регресія використовується, коли відповідна залежна змінна є номінальною (еквівалентно категоричною, що означає, що вона потрапляє в будь-яку з набору категорій, які не можуть бути впорядковані будь-яким значущим чином) і для якої існує більше двох категорій. Для нашої задачі номінальні значення це теги, які містить та чи інша стаття. Як приклад: одна стаття може містити теги python, django, web, а інша може містити теги c#, .net, string. Ми не можемо заздалегідь знати, які значення тегів буде містити вхідний у модель елемент.