

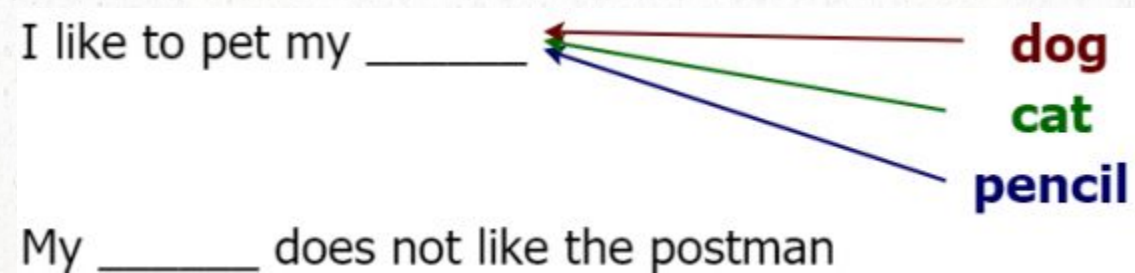
ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗУМІННЯ І ВИКОРИСТАННЯ ТЕКСТУ. МЕТОДИ WORD2VEC

**ВИКОНАВ:
СТУДЕНТ ГРУПИ МПМ-1
СТЕПАНЮК Д. А.**

**НАУКОВИЙ КЕРІВНИК:
БОГАТИРЬОВ О. О.**

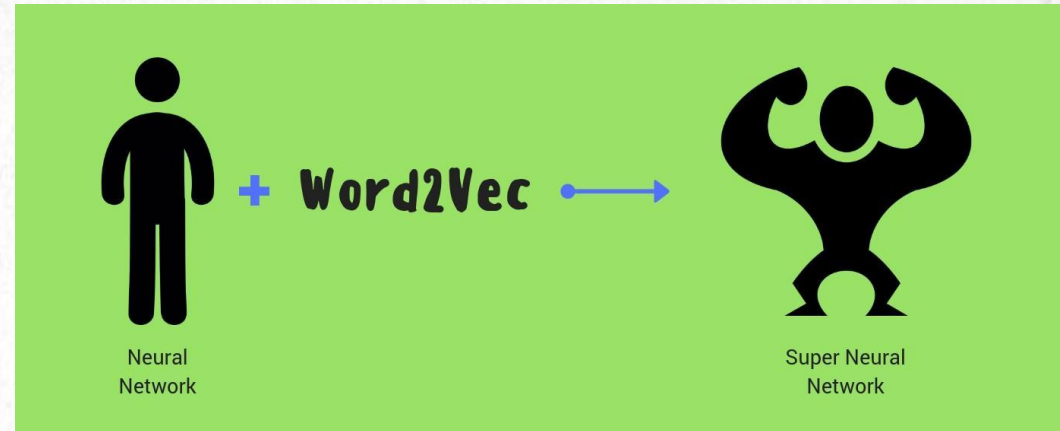
СЛОВА-ВЕКТОРИ

- Слова-вектори (word vectors) - це чисельні уявлення слів, що зберігають семантичний зв'язок між ними. Наприклад, для вектора cat (кішка) одним з найбільш близьких буде слово dog (собака). Однак векторне подання слова pencil (олівець) буде досить сильно відрізнятися від вектора cat.



ЗАСТОСУВАННЯ WORD2VEC

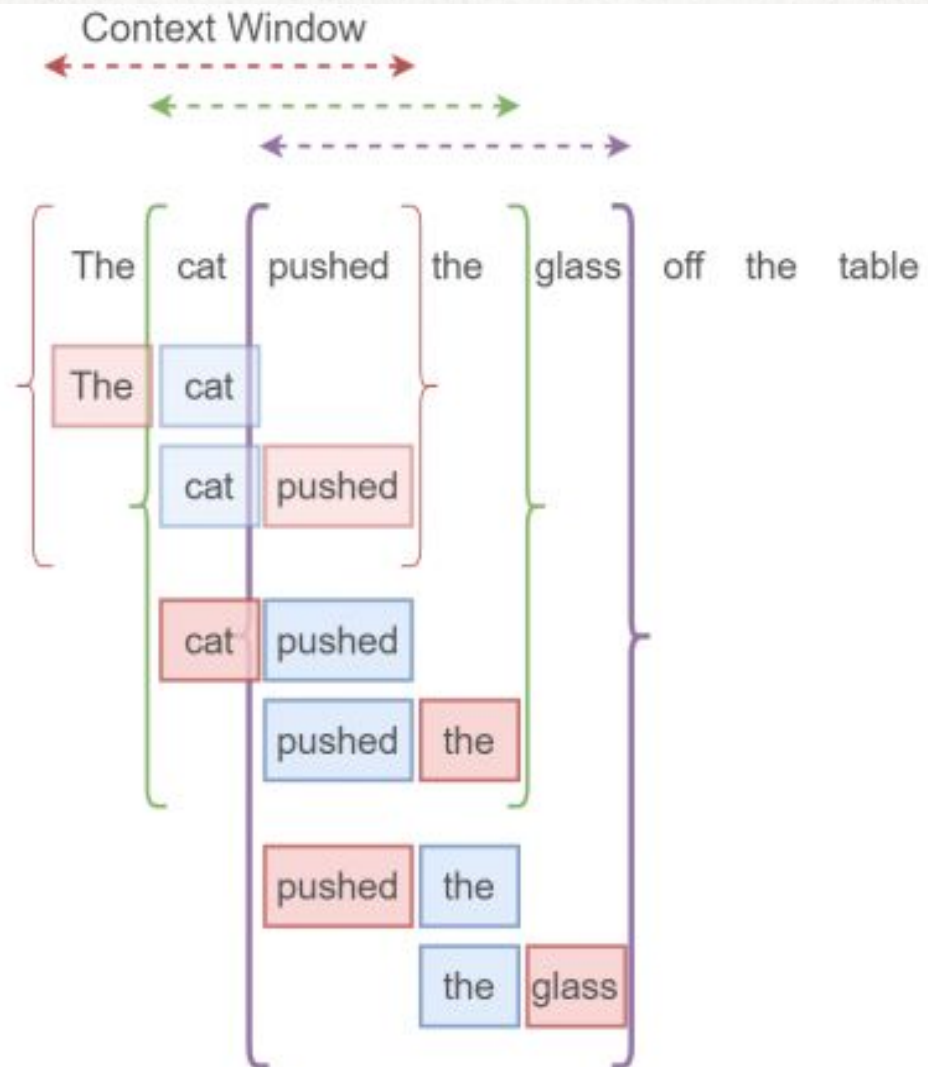
- Векторні уявлення слів використовуються в багатьох областях:
- Моделювання мов;
- Чат-боти;
- Машинний переклад;
- Питально-відповідні системи;



ПРОЦЕС СТОВРЕННЯ ВЕКТОРІВ СЛІВ

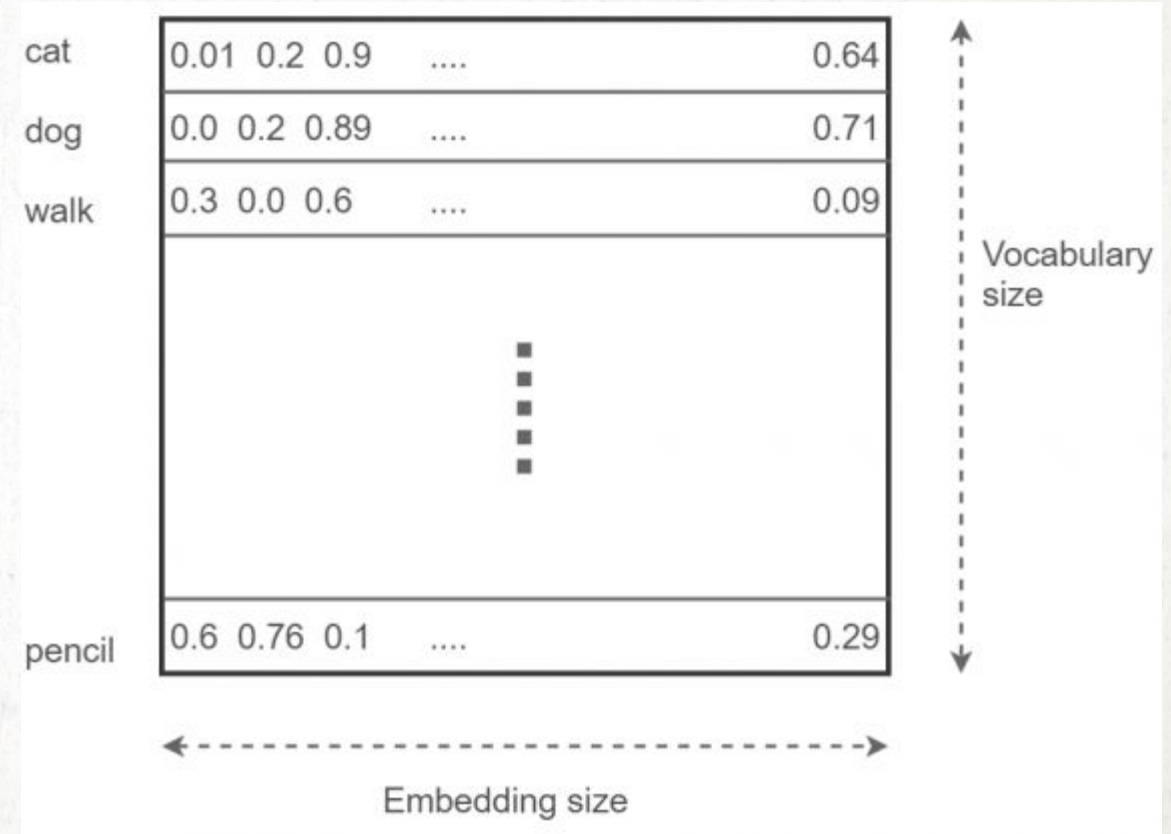
1. Створити кортежі даних в форматі [вхідне слово, вихідне слово]
2. Створити модель, яка на вхід і вихід отримує one-hot вектори;
3. Визначити функцію втрат, яка вгадує вірне слово, щоб оптимізувати модель;
4. Визначити якість моделі, переконавшись, що схожі слова мають схожі векторні уявлення.

ЗАГАЛЬНА МОДЕЛЬ

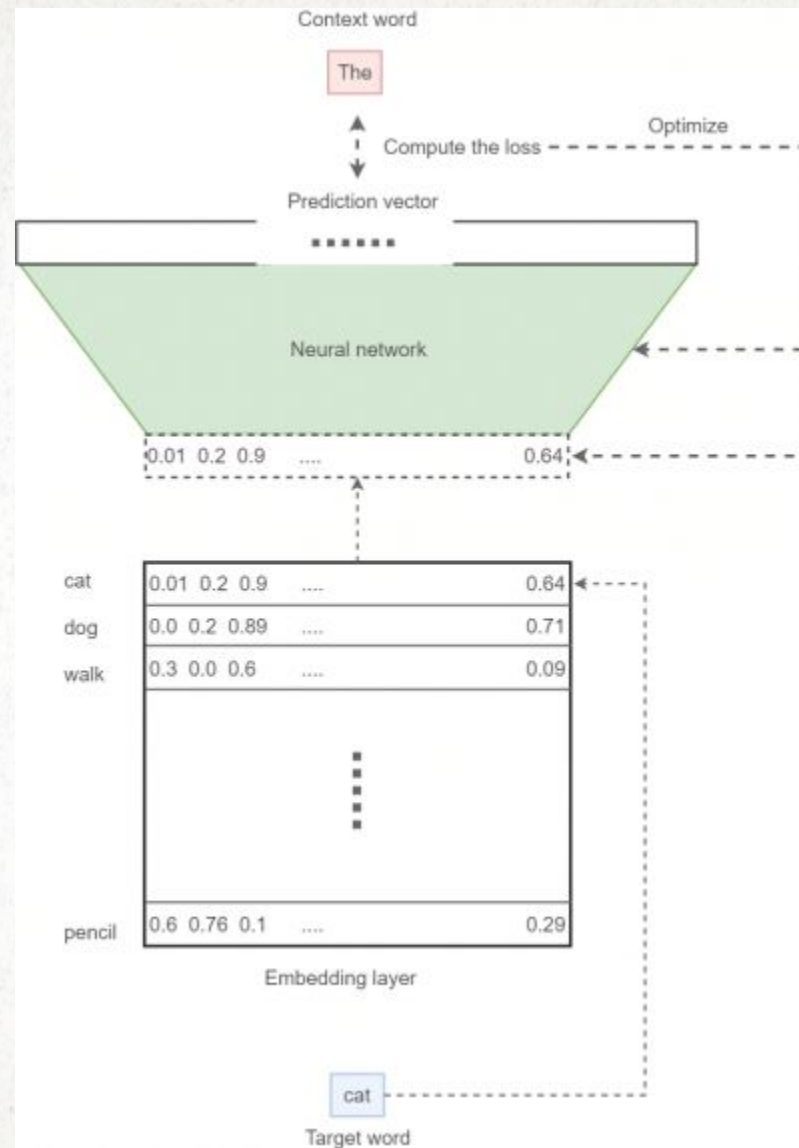


НЕОБХІДНІ ДАНІ ДЛЯ РОБОТИ АЛГОРИТМУ

- набір вхідних one-hot векторів;
- набір вихідних one-hot векторів (після навчання);
- embedding layer;
- нейромережа.



ЗАГАЛЬНА КАРТИНА АЛГОРИТМУ



ФУНКЦІЯ ВТРАТ

| | The cat pushed the glass off the table | | | | | | | | | |
|------------------|--|-----|--------|--|--|--|--|-------|--|--|
| Positive Samples | The | cat | | | | | | | | |
| | | cat | pushed | | | | | | | |
| Negative Samples | | cat | | | | | | glass | | |

ЗАПУСК АЛГОРИТМУ

```
num_steps = 250001

session = tf.InteractiveSession()

# Initialize the variables in the
graph

tf.global_variables_initializer().run()

print('Initialized')

average_loss = 0

for step in range(num_steps):

    batch_data, batch_labels = generate_batch( batch_size,
    window_size)
    feed_dict = {train_dataset : batch_data, train_labels :
    batch_labels}
    _, l = session.run([optimizer, loss], feed_dict=feed_dict)
    if (step+1) % 5000 == 0:
        if step > 0:
            average_loss = average_loss / 5000

            print('Average loss at step %d: %f' % (step+1,
            average_loss))
            average_loss = 0
```

РЕЗУЛЬТАТИ РОБОТИ АЛГОРИТМУ

Average loss at step 225000: 2.410598

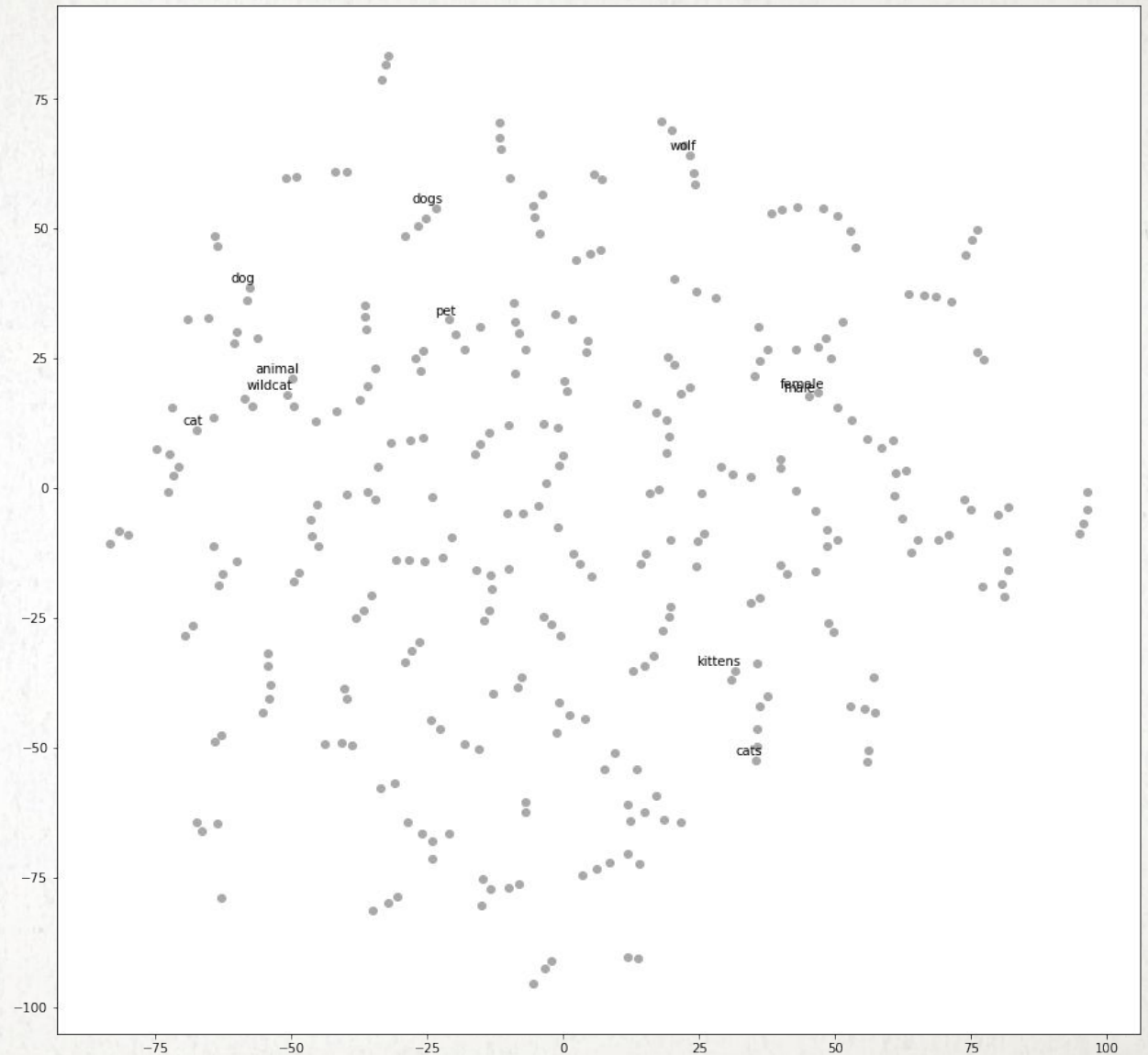
Average loss at step 230000: 2.410446

Average loss at step 235000: 2.414872

Average loss at step 240000: 2.412652

Average loss at step 245000: 2.420230

Average loss at step 250000: 2.408468



ДЯКУЮ ЗА УВАГУ!