Глубинное обучение
Лекция 6: Нейросети в задачах обработки текстов

Лектор: Антон Осокин

ФКН ВШЭ, 2019

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Виды задач для нейросетей

- Классификация
  - Примеры: определение темы, sentiment analysis
  - Учитывать последовательность или нет?
  - Bag-of-words, RNN
- Разметка последовательности
  - Примеры: определение частей речи, chunking
  - Локальный классификатор, RNN, CRF, RNN-CRF (BiLSTM-CRF)
- Последовательности в последовательность (разной длины!)
  - Примеры: машинный перевод, аннотация (summarization), диалоги
  - Авторегрессионные модели: seq2seq (+attention), ByteNet, и т.д.
- Синтез текста
  - Примеры: описание изображения (captioning), диалоги, искусство?
  - Авторегрессионные модели на основе обученных представлений

План лекции

- Непрерывные представления слов (embedding)
  - word2vec, FastText
- Обработка последовательностей
  - Seq2seq
  - Seq2seq + attention
  - Transformer
- Контекстно-зависимые представления (предобучение)
  - ELMo, BERT, GPT-2

Как вставить текст в нейросеть?

Непрерывные представления слов (word embeddings)

- Позволяют строить непрерывные представления дискретных объектов
- Непрерывные представления – это способ поместить текст в нейросеть
- Представление – вектор по индексу (слова, символы, n-граммы)
- Представления могут обучаться совместно с моделью

- Предобученные представления
  - Обучены на больших корпусах текстов
  - Обучены без ручной разметки (self-supervision)
  - Freeze, fine-tune, train from scratch?

Представления word2vec (skipgram)

[Mikolov et al., 2013]

- Обучение на предсказании контекста по слову
  - Обучение на корпусе текстов без разметки (self supervision)
- Вспомогательная задача:
  - Предсказываем каждое слово из контекста отдельно

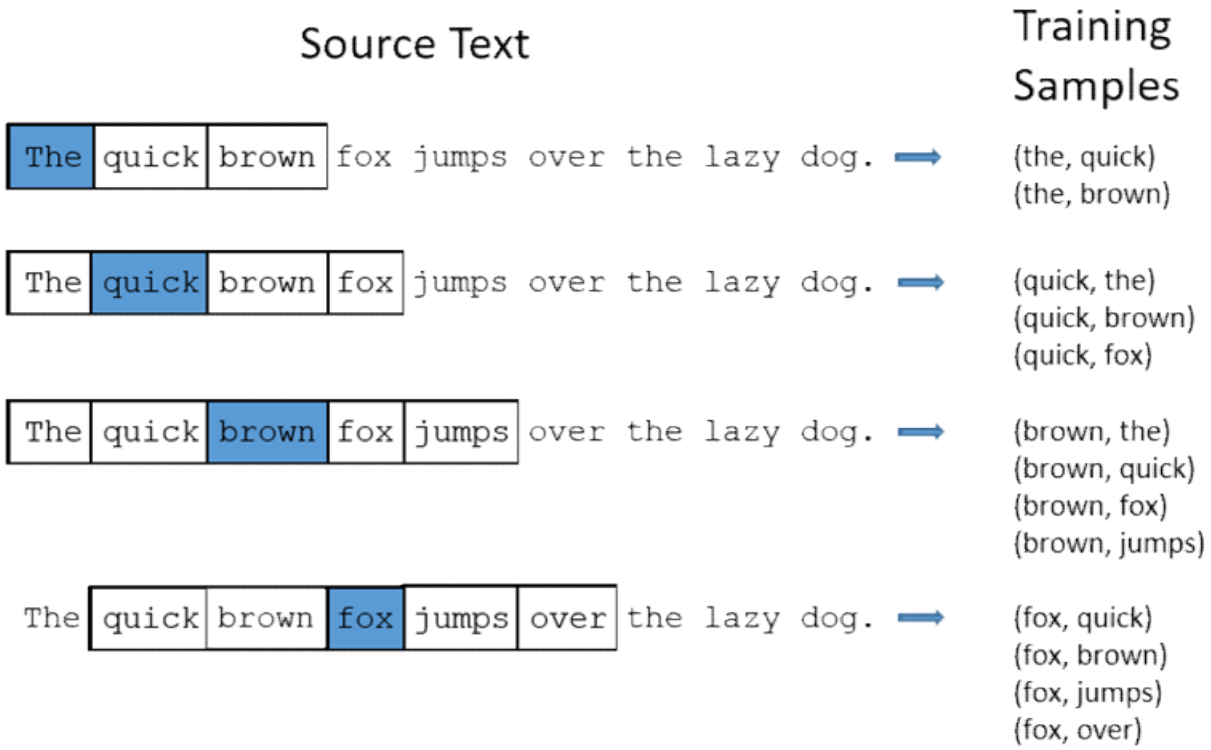


image credit:
Chris McCormick

Представления word2vec (skipgram)

[Mikolov et al., 2013]

- Обучение на предсказании контекста по слову
  - Обучение на корпусе текстов без разметки
- Предсказываем каждое слово из контекста отдельно
  - Текущее слово w; слово из контекста v
  - Для каждого слова – 2 представления (*in*, *out*)
  - Совместимость – скалярное произведение $\text{in}_w^T \text{out}_v$
  - Полезные – представления *in*
  - Модель с softmax $P(v \mid w, \theta) = \frac{\exp(\text{in}_w^T \text{out}_v)}{\sum_{v'} \exp(\text{in}_w^T \text{out}_{v'})}$
    - Медленная нормировка
  - Обычное решение – Noise Contrastive Estimation (NCC)

$$\text{loss}(w, v) = \log(1 + \exp(-\text{in}_w^T \text{out}_v)) + \sum_{\text{random } v'} \log(1 + \exp(\text{in}_w^T \text{out}_{v'}))$$

Представления word2vec (skipgram)

[Mikolov et al., 2013]

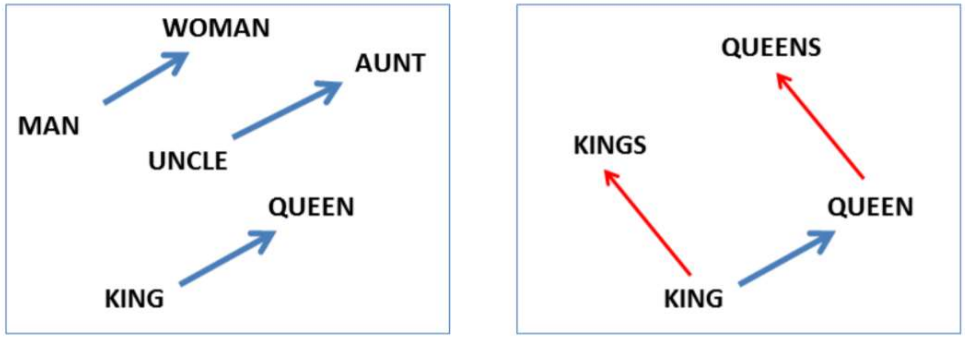
- Обучение на предсказании контекста по слову
  - Обучение на корпусе текстов без разметки
- Предсказываем каждое слово из контекста отдельно
  - Используются представления *in*
- Достоинства
  - Ближайшие соседи (cosine distance = норм. скал. произв., корпус GoogleNews)
    - university: student, teacher, teaching, students, schools
    - Putin: Medvedev, Vladimir_Putin, President_Vladimir_Putin, Prime_Minister_Vladimir_Putin, Kremlin
    - putin: lol, mr, don't, obama, Hahaha
    - obama: dems, americans, washington, america, libs

Source: http://bionlp-www.utu.fi/wv_demo/

Представления word2vec (skipgram)

[Mikolov et al., 2013]

- Обучение на предсказании контекста по слову
  - Обучение на корпусе текстов без разметки
- Предсказываем каждое слово из контекста отдельно
  - Используются представления *in*
- Достоинства
  - Ближайшие соседи (cosine distance = норм. скал. произв., корпус GoogleNews)
  - Арифметика над представлениями
    - king – man + woman = queen



Source: [Mikolov et al., 2013]

Представления fastText (skipgram)

[Bojanowski et al., 2017]

Как в word2vec:

- Обучение на предсказании контекста по слову
  - Обучение на корпусе текстов без разметки
- Предсказываем каждое слово из контекста отдельно

Новая идея:

- Добавить информацию о символах слова (через n-граммы)

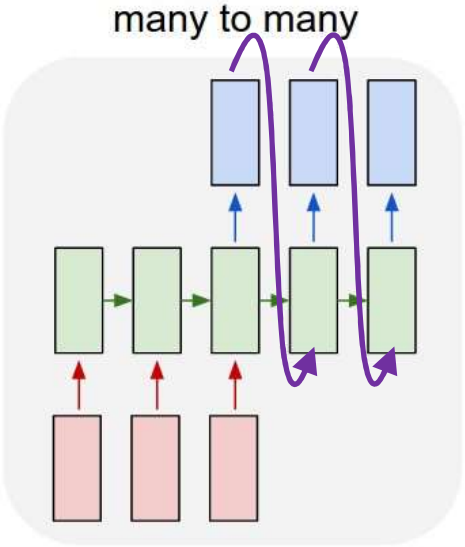
$$\text{in}_w = \text{word}_w + \sum_{p \in \text{n-grams(w)}} \text{part}_p \qquad \text{in}_w^T \text{out}_v \Rightarrow \text{word}_w^T \text{out}_v + \sum_{p \in \text{n-grams(w)}} \text{part}_p^T \text{out}_v$$

  - “where” = “<where>”, “<wh”, “whe”, “her”, “ere”, “re>”
- Важно использовать длинные n-граммы (n ≤ 6)
- Достоинства:
  - Близость по написанию
  - Слова вне словаря, опечатки и т.д.

Код и данные на fasttext.cc

Модель seq2seq [Sutskever et al. , 2014]

- Модели для предсказания последовательностей разной длины



Входы, память, выходы

Входы – представления входов

Память – слои RNN или CNN

Выходы – шансы слов из словаря (logits, идут в logsoftmax)

Авторегрессионные связи (→) передают решение о текущем слове

На входы → подаются представления выходного алфавита

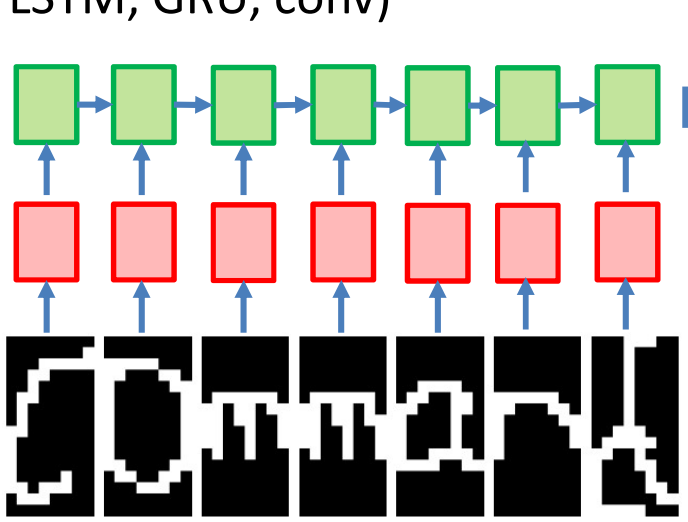
image credit: Andrej Karpathy

Последовательное предсказание

- Пример:  command

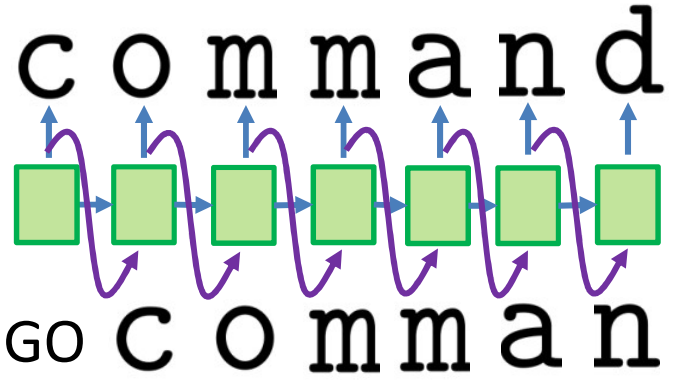
Кодировщик
(RNN, LSTM, GRU, conv)

CNN



Представление входа

Выход (предсказание)



Авторегрессионные связи

Если не фиксирована длина выхода, то используют символ EOS (с барьером)

Обучение авторегрессионных моделей

- Обычный способ – метод максимального правдоподобия на каждом шаге декодировщика

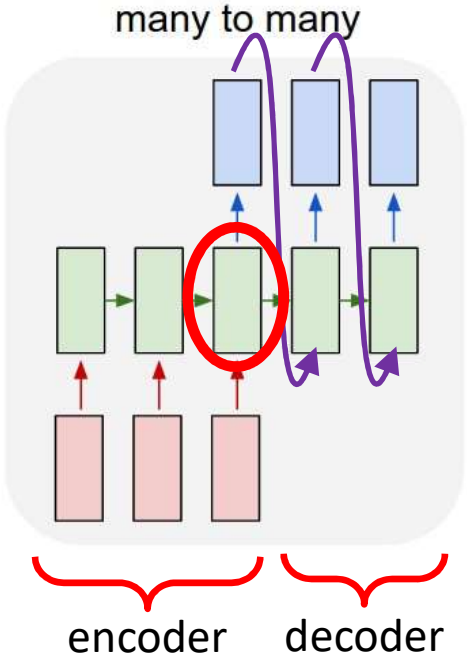
$$P(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = P(y_1 \mid \boldsymbol{x}, \boldsymbol{\theta}) P(y_2 \mid y_1, \boldsymbol{x}, \boldsymbol{\theta}) P(y_3 \mid y_2, y_1, \boldsymbol{x}, \boldsymbol{\theta}) \dots$$

- Teacher forcing – на вход декодировщику подаются правильные ответы
- Проблема:
  - Модель видит только правильные траектории
  - Не знает, что делать при ошибке
  - Как исследовать траектории (exploration)?
  - Связь с «Обучением с подкреплением»
- Можно использовать любые функции потерь
  - Информацию о качестве каждого предсказания!

[LeBlond et al., 2018]

Модель seq2seq [Sutskever et al. , 2014]

- Модели для предсказания последовательностей разной длины



Модель encoder-decoder

○ – представление всего входа

Модель плохо работает для длинных последовательностей

Причина: представление входа – вектор фиксированной размерности

(не может представить весь язык)

Решение: механизм внимания

(attention)

image credit: Andrej Karpathy

Модель seq2seq с вниманием

[Bahdanau et al. , 2015]

- Модели для предсказания последовательностей разной длины

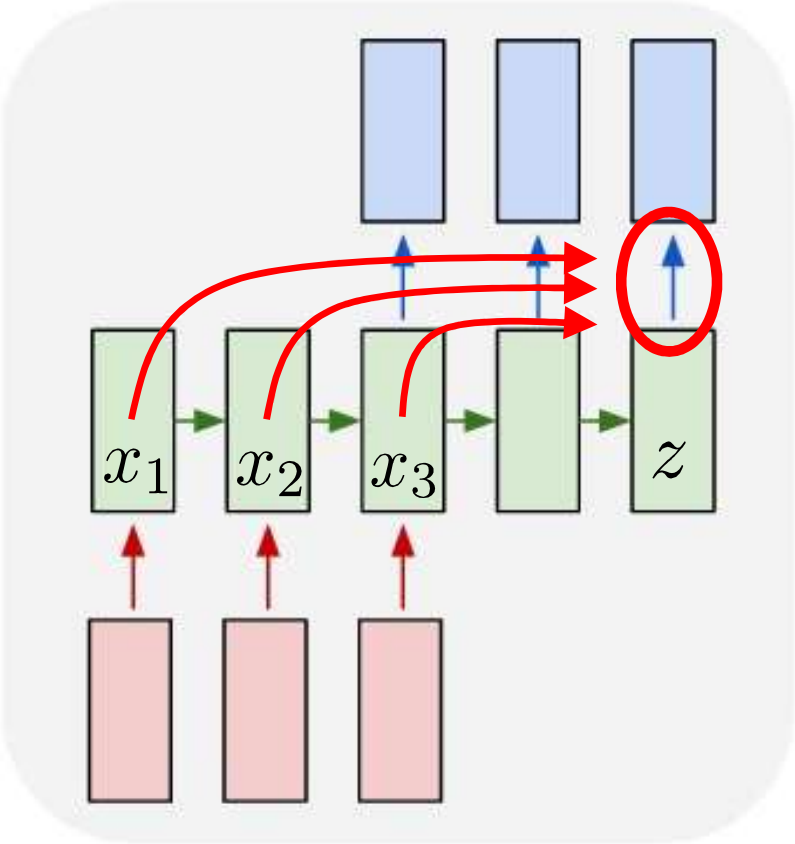


image credit: Andrej Karpathy

Внимание «выбирает релевантные элементы памяти»

Модель внимания:

- релевантность

$$s_i := \text{score}(x_i, z) = \begin{cases} x_i^T z \\ W[x_i; z] \end{cases}$$

- вероятности

$$a_1, a_2, \dots := \text{softmax}(s_1, s_2, \dots)$$

- контекст

$$c := \sum_i a_i x_i$$

- новые признаки $\tilde{z} := [c; z]$
- soft-argmax

Transformer

[Vaswani et al., 2017]

- Статья – Attention Is All You Need!
- Архитектура:
  - Scaled multi-head self attention

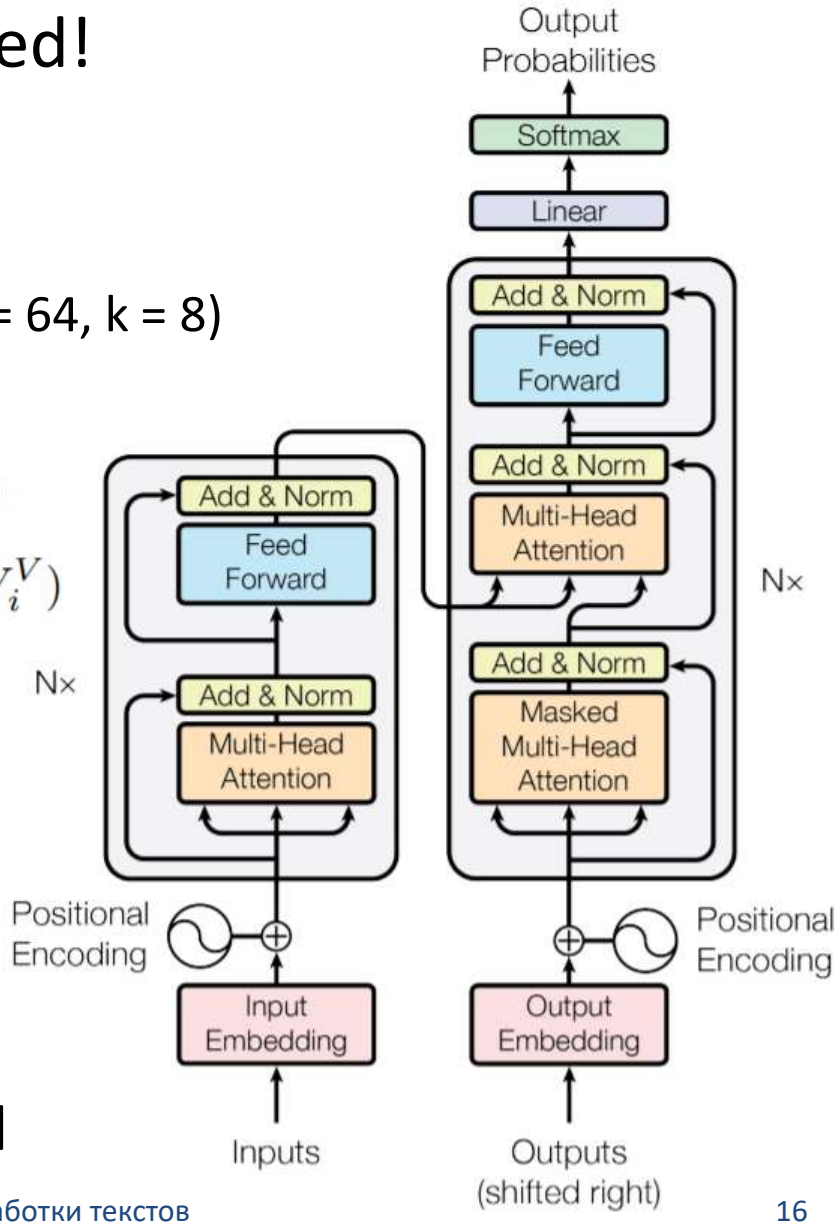
  (Q = query, K = key, V = value, d = dimension = 64, k = 8)

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

  - 2-layer NN with ReLu
  - Encoding: token and positional
    - Positional – sin и cos от длины
  - В декодере –маска будущего!
- SOTA в переводе и др.!
- Поиск архитектуры [So et al., 2019]



Словарь из Byte Pair Encoding (BPE)

[Sennrich et al., 2015]

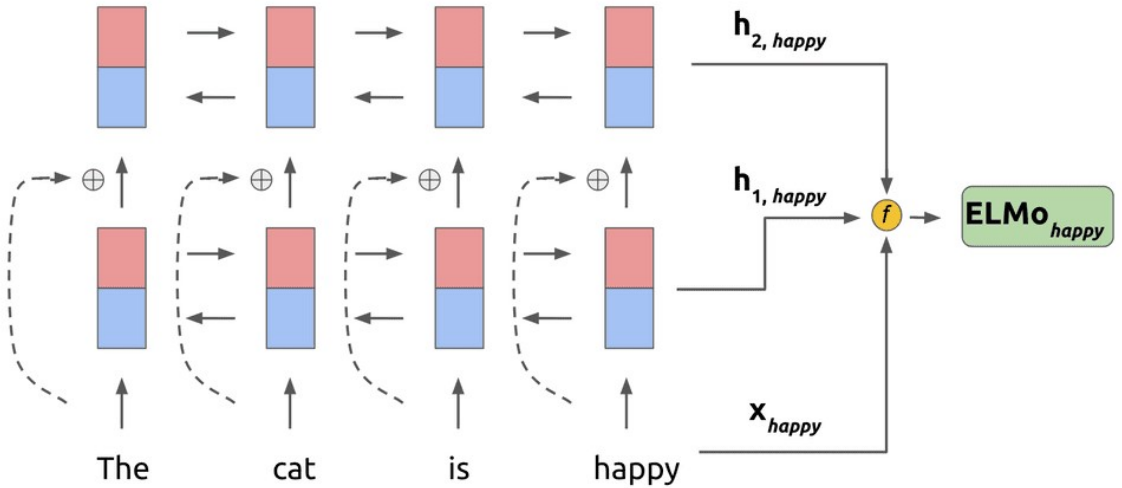
- Размер словаря – важный параметр!
  - Большой => мало слов на редкие позиции, медленно, память
  - Маленький => много слов вне словаря
- Токены – из наиболее частых пар токенов
  - Инициализация – из токенов символов (unicode - осторожно)
  - Итеративное склеивание самых частых пар
  - Пересечение границ слов?
  - Символы типа знаков препинания?
- Позволяет делать представления любого слова
- Размер словаря – контролируемый параметр

Предобученые представления слов – из языковых моделей!

ELMo

[Peters et al., 2018; AllenNLP]
https://github.com/allenai/allennlp

- ELMo = Embeddings from Language Models
- Контекстно зависимые представления!
- Архитектура:
  - Представления слов = свертки поверх представлений символов
    - Легко обрабатывать слова вне словаря
  - 2 модели поверх – forward и backward (2-layer LSTM + skip con.)
  - Итоговое представление – линейная комбинация слоев!
- Обучение:
  - Forward – след. слово
  - Backward – пред. слово



BERT

[Devlin et al., 2018; Google]

- BERT = Bidirectional Encoder Representations from Transformers
- Очень большой трансформер:
  - L – глубина, H – размерность внутри, A – число голов multi-head attention
  - $BERT_{BASE}$: L=12, H=768, A=12, Total Parameters=110M
  - $BERT_{LARGE}$: L=24, H=1024, A=16, Total Parameters=340M
- Обучение:
  - Masked LM: 15% tokens
    - 80% - [MASK]
    - 10% - исходное слово
    - 10% - случайное слово
  - Next sentence

Input = [CLS] the man went to [MASK] store [SEP]
he bought a gallon [MASK] milk [SEP]
Label = IsNext
Input = [CLS] the man [MASK] to the store [SEP]
penguin [MASK] are flight ##less birds [SEP]
Label = NotNext

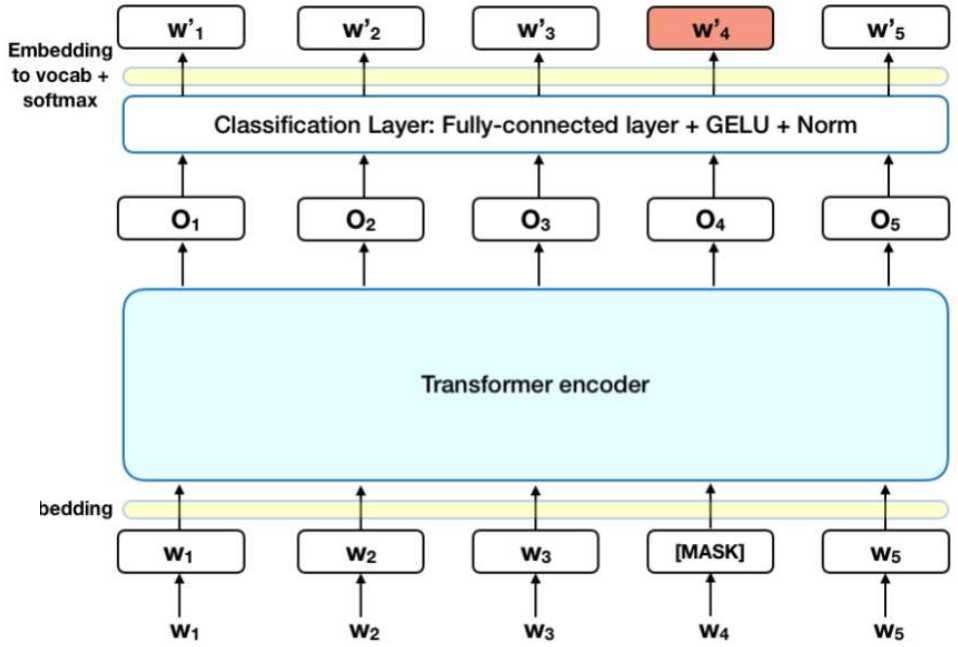


image credit: Rani Horev

GPT-2

[Radford et al., 2019; OpenAI]

- GPT = Generative Pre-trained Transformer
- Трансформер с 1,500,000,000 (1.5B) параметрами
- Обучение на 40GB с веб страниц (нет wiki)

SYSTEM PROMPT (HUMAN-WRITTEN):

Recycling is good for the world. NO! YOU COULD NOT BE MORE WRONG!!

MODEL COMPLETION (MACHINE-WRITTEN) (сокращено)

Recycling is NOT good for the world. It is bad for the environment, it is bad for our health, and it is bad for our economy. I'm not kidding. Recycling is not good for the environment. It is destructive to the earth and it is a major contributor to global warming. Recycling is not good for our health. It contributes to obesity and diseases like heart disease and cancer. Recycling is bad for our economy. It increases the cost of a product, and in turn, the price of everything that is made with that product. Recycling is not good for our nation. We pay a tremendous price for the privilege of having the world's most advanced and efficient recycling system. Recycling is a huge, colossal waste of time, energy, money, and resources. And THAT is why we need to get back to basics and get back to basics in our recycling efforts.

- OpenAI не выпустила модель + PR => нездоровый хайп

Заключение

- Обработка языка активно использует нейросети
- Очень большая область – много задач
  - Есть успехи!
- Представления, Seq2seq, внимание, transformer, etc.
- Понимание смысла – очень сложная задача!