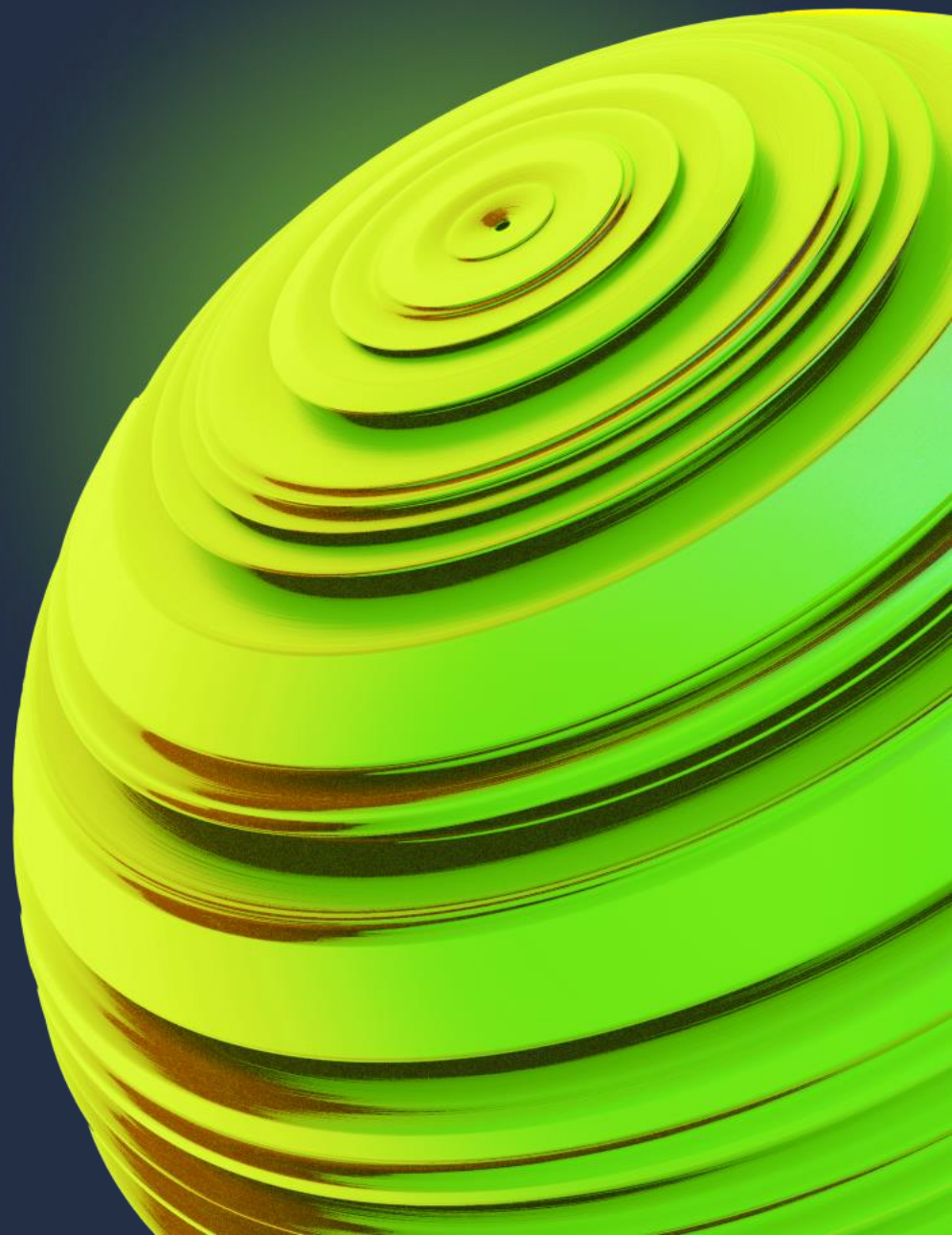
ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС

УНИВЕРСИТЕТ ИННОПОЛИС

Эмбединги. ELMO

Корнеева Елена
e.korneeva@innopolis.ru, https://t.me/Allyonzy

2024



План занятия (лекция + семинар)

1. Подходы к векторному представлению текста
2. LSTM и GRU
3. GloVe
4. ELMo

Векторизация текстов: 2 подхода

Статистические подходы

Векторизовать текст целиком, превращая его в один вектор.

Рассматривается текст как неупорядоченный набор («мешок») токенов (обычно токенами являются слова)

Алгоритмы:

- **Bag-of-Words** (только токены)
- **One-hot-encoding** (только токены),
- **TF-IDF** (токены + контекст)

Подходы с учётом контекста

Векторизовать отдельные структурные единицы, превращая текст в последовательность векторов.

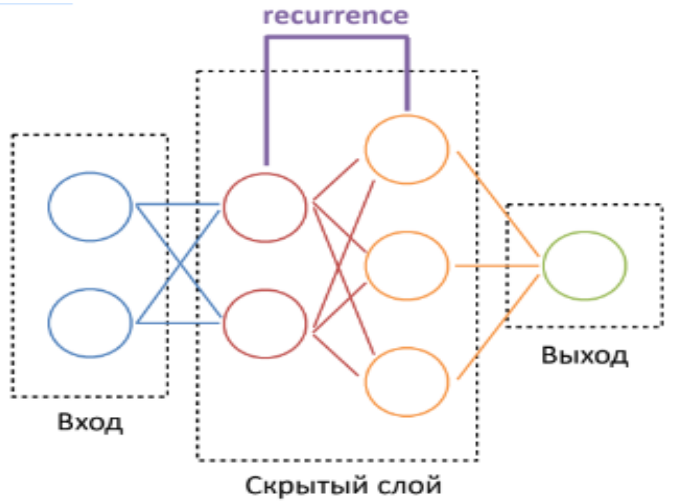
Используется контекст. Для обучения авторы предложили две стратегии: Skip-gram (модель учится по слову предсказывать контекст) и CBOW (учится предсказывать слово по контексту).

Алгоритмы:

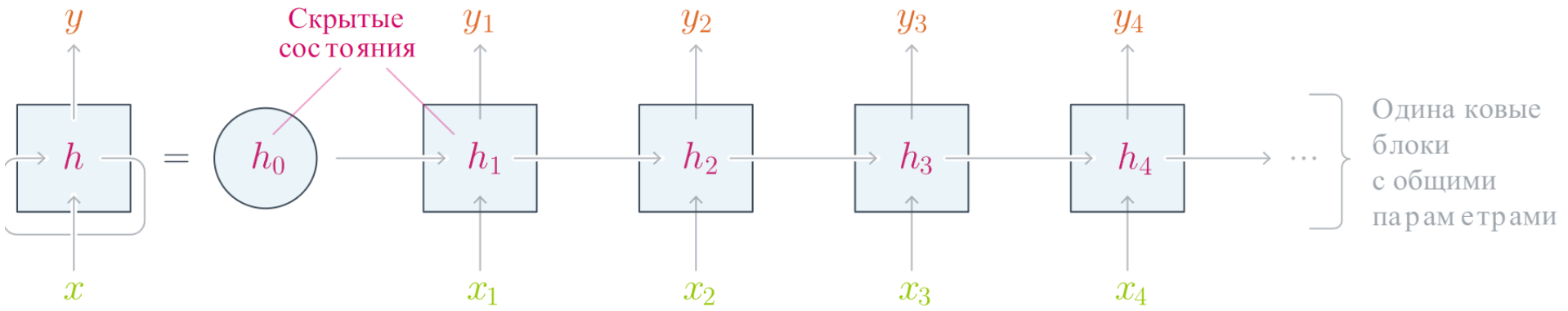
- **Word2Vec**, fastText, GloVe
- **ELMo**, GPT-2/GPT-3,
- **BERT**

Рекуррентная нейронная сеть

Рекуррентные нейронные сети (Recurrent neural network, RNN) — вид нейронных сетей, где связи между элементами образуют направленную последовательность.



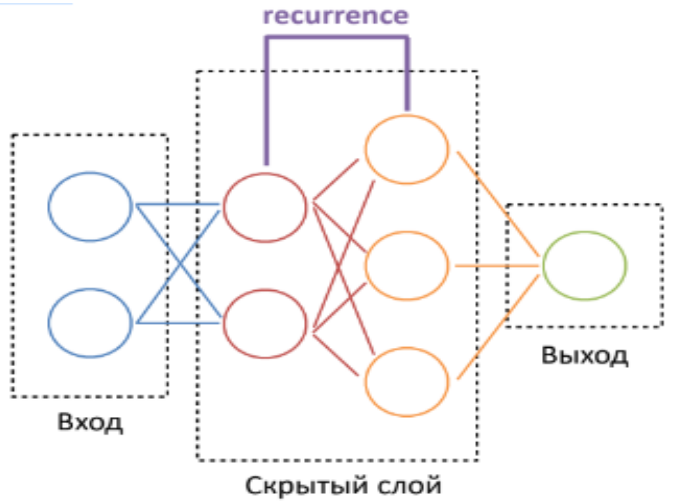
разобьём память на долгосрочной и краткосрочную и получим LSTM

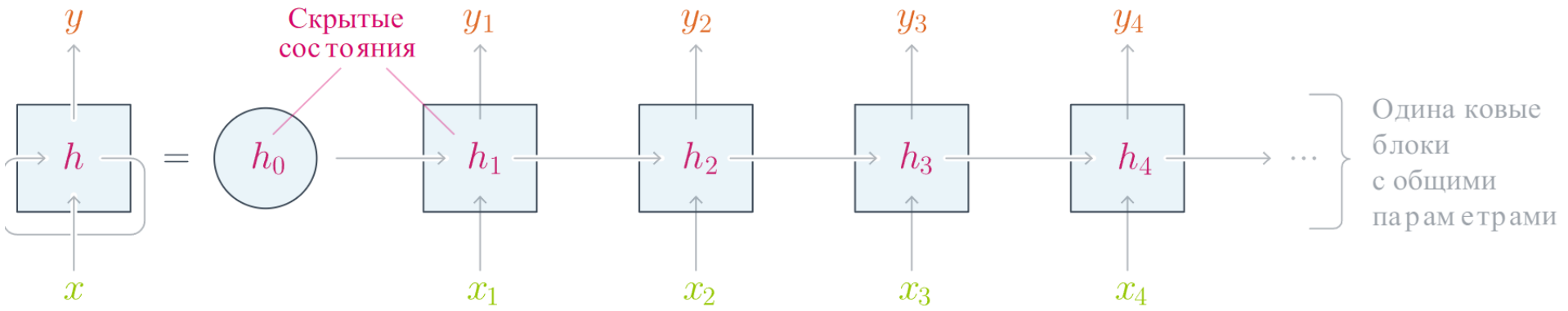


Рекуррентная нейронная сеть

Появляется возможность обрабатывать **серии событий во времени** или **последовательные пространственные цепочки.**

Рекуррентность. *Вычисляемость на основе значений предыдущих членов последовательности + Повторяемость*



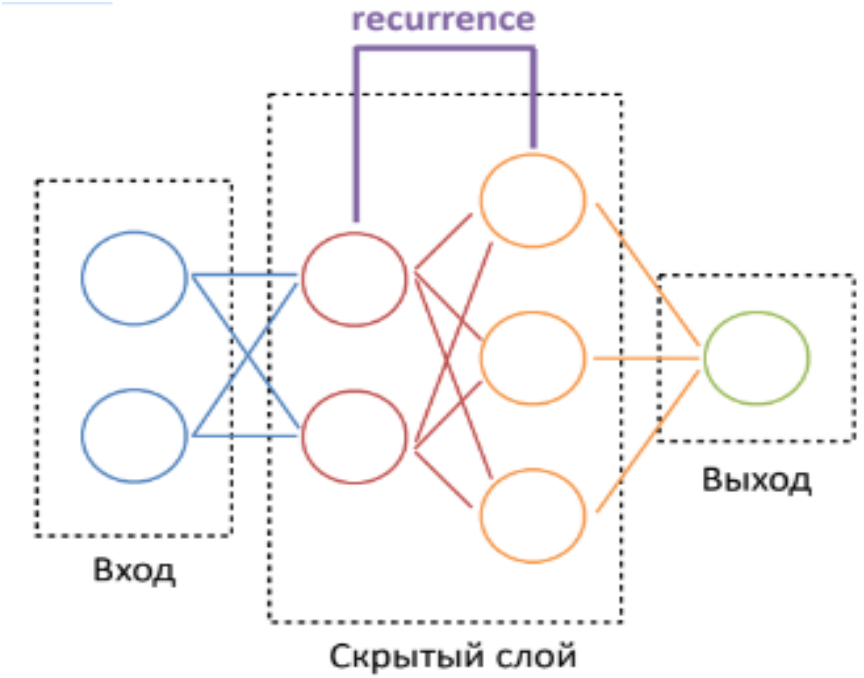


Рекуррентная нейронная сеть

Сети RNN применимы в таких задачах, где нечто целостное разбито на части

- распознавание рукописного текста
- распознавание речи
- тренды

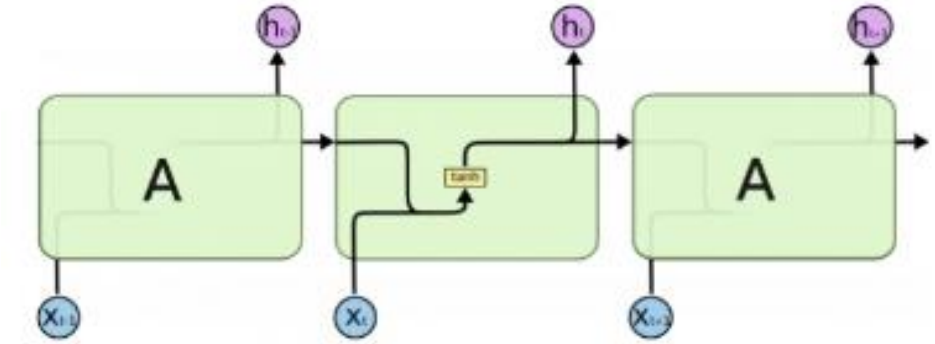
В последнее время наибольшее распространение получили сеть с долговременной и кратковременной памятью (**LSTM**) и управляемый рекуррентный блок (**GRU**).



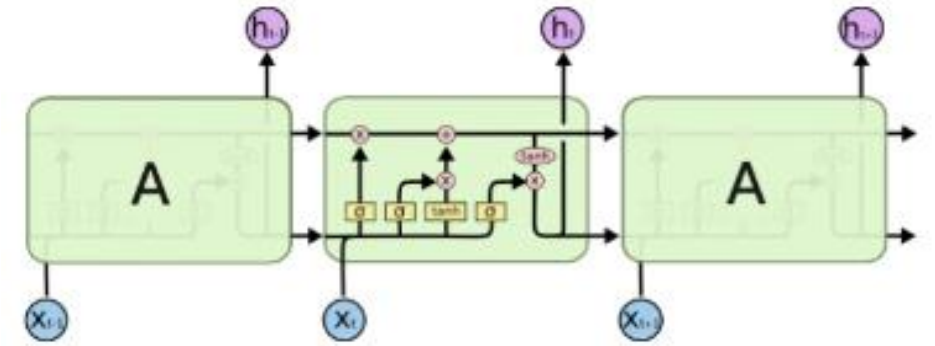
LSTM

Сеть с **долговременной и кратковременной памятью (Long short term memory, LSTM)** частично решает проблему исчезновения или зашкаливания градиентов в процессе обучения рекуррентных сетей методом обратного распространения ошибки

LSTM построена таким образом, чтобы учитывать долговременные зависимости



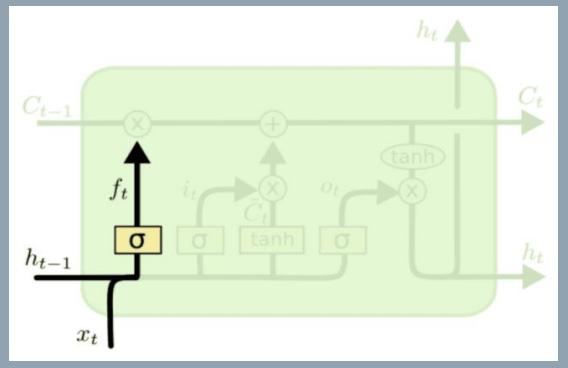
Повторяющийся модуль стандартной RNN состоит из одного слоя



Повторяющийся модуль LSTM состоит из четырех взаимодействующих слоев

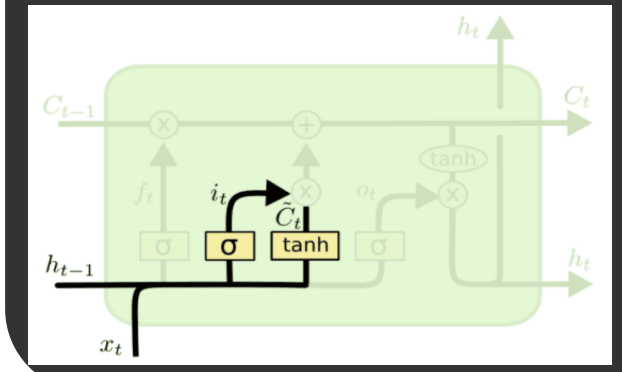
LSTM

Слой утраты



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

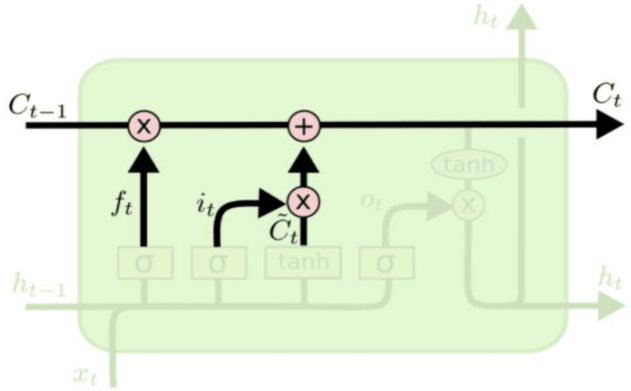
Слой сохранения



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

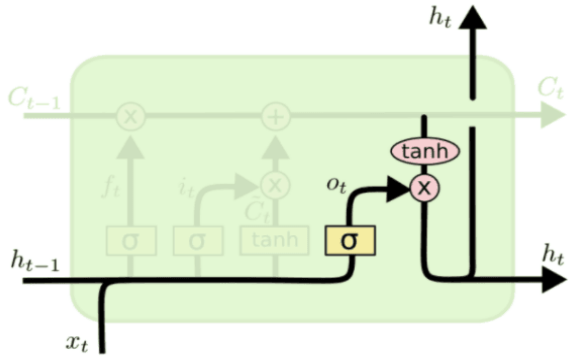
$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Новое состояние



$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t$$

Результат (выход)



$$o_t = \sigma(W_o[h_{t-1}, x_t + b_o]$$

$$h_t = o_t \cdot tanh(C_t)$$

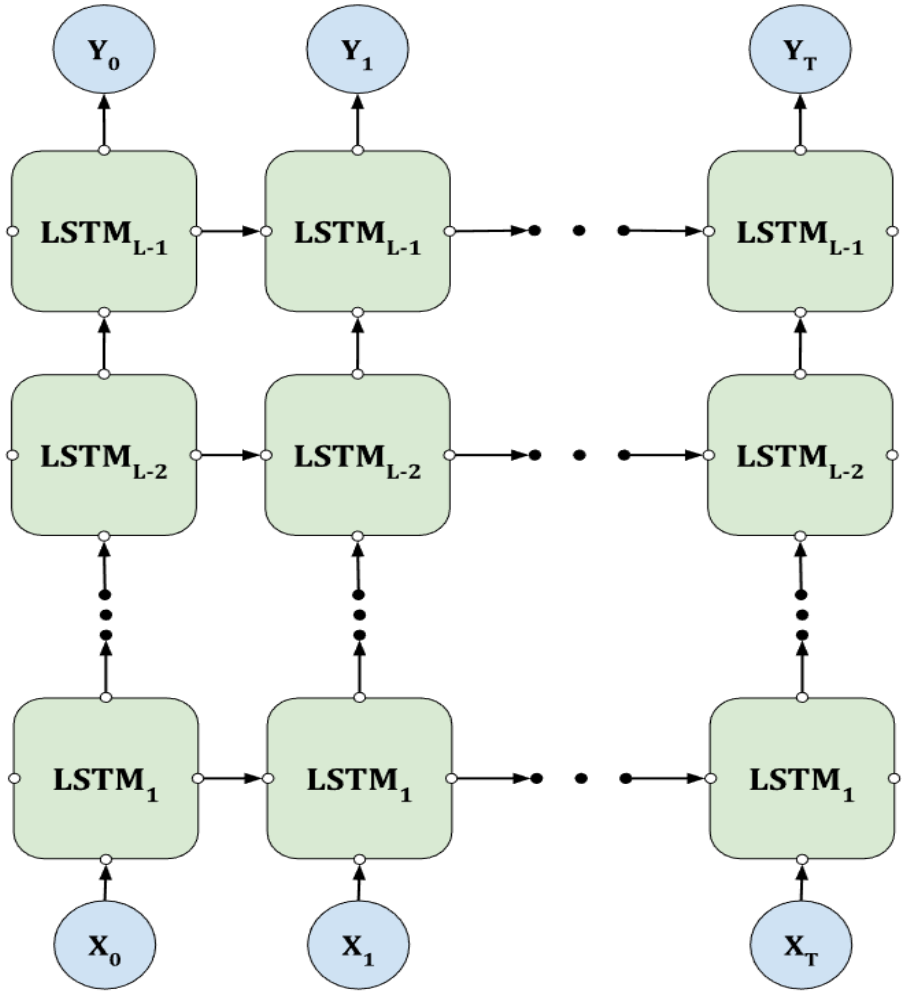
Многослойный LSTM

- RNN, LSTM и GRU ячейки можно использовать точно так же, как линейные слои.

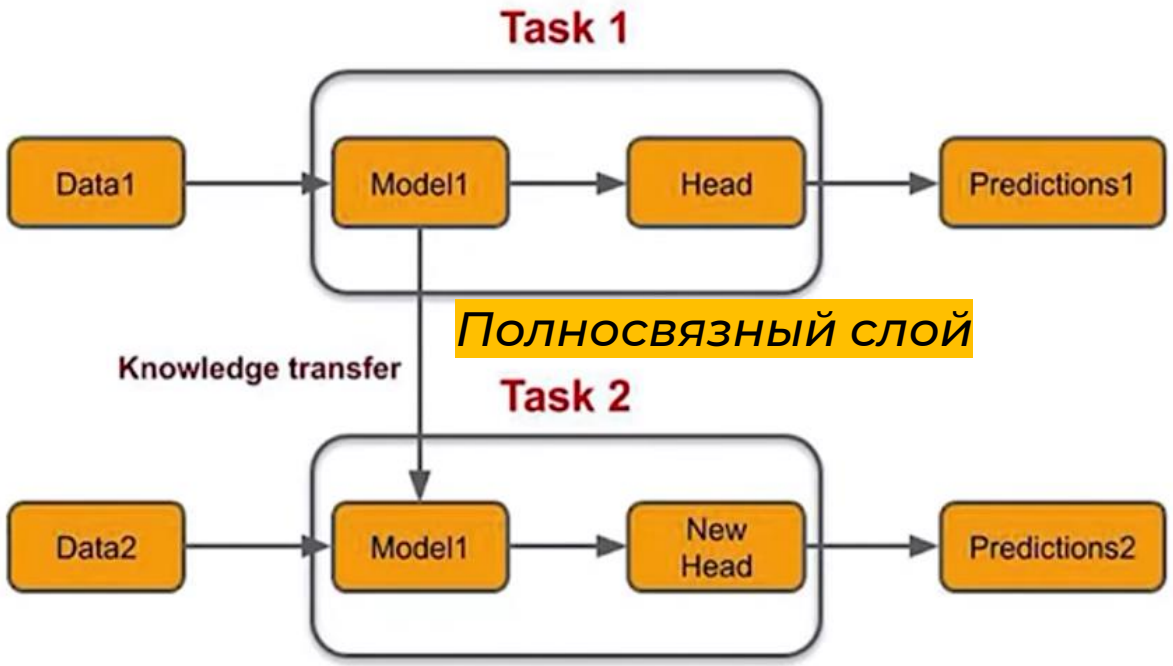
- Все ячейки можно делать двунаправленными. Можем запустить параллельную ветку, которая будет делать то же самое справа-налево.

- Это возможно не для всех задач, потому что иногда мы хотим генерировать (однонаправленная сеть).

Есть другие сети RNN... Для увеличения “приёмистости” сети, обычно набирают несколько слоёв LSTM

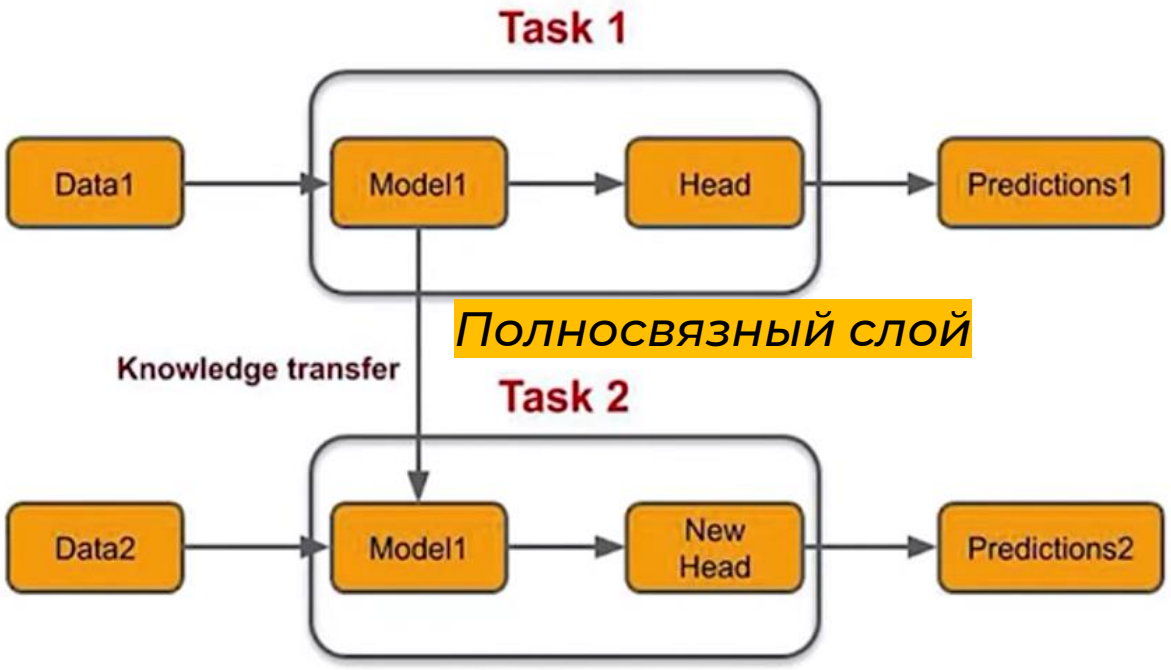


Эмбеддинги слов



- Общее название для различных подходов к моделированию языка, направленных на сопоставление словам из некоторого словаря векторов небольшой размерности
- Предобученная модель с векторным представлением (учитывается контекст)

Трансферное обучение Transfer Learning

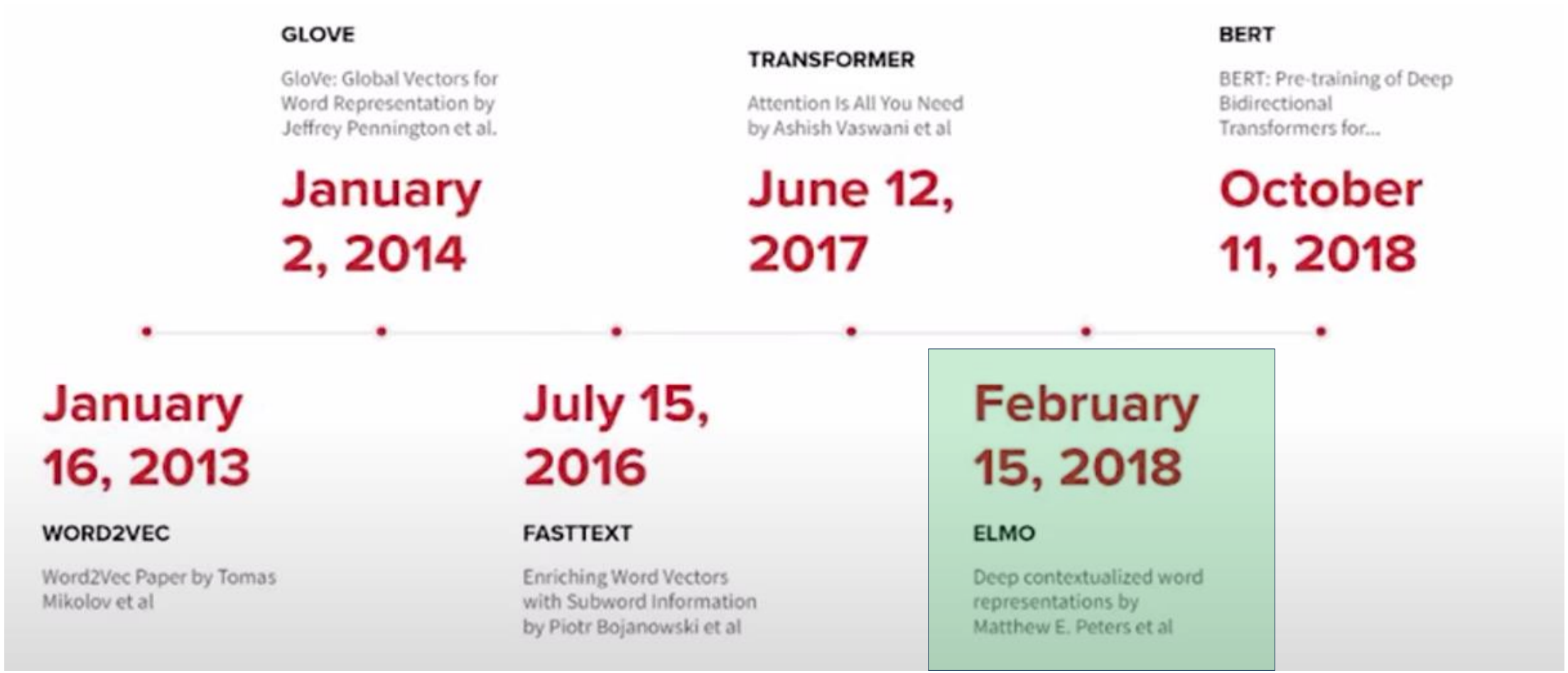


Копируем часть архитектуры, дообучаем веса и используем в другой задаче

Идея **трансферного обучения** строится на том, что знания, накопленные в модели нейронной сети могут быть перенесены на другую модель. Модель может помочь в построении прогнозов для другой, родственной задачи.

Трансферное обучение с предобученными моделями требует меньше данных (небольшие архитектурные модификации для адаптации модели к своему набору данных)

Основные проекты NLP для языковых моделей



GloVe

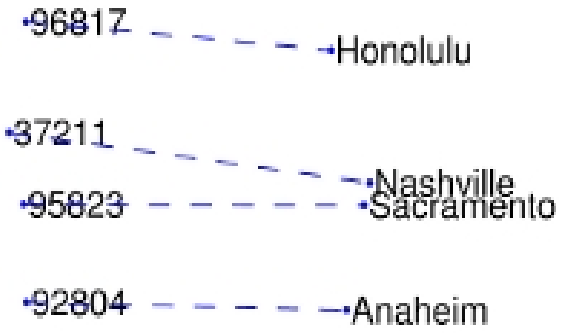
GloVe тесно ассоциируется с Word2Vec. Но GloVe учитывает совместную встречаемость слов, а не полагается только на контекстную статистику.

Модель GloVe *пытается решить проблему эффективного использования статистики совпадений*. Векторы слов группируются вместе на основе их глобальной схожести. GloVe минимизирует разницу между произведением векторов слов и логарифмом вероятности их совместного появления с помощью стохастического градиентного спуска.

```
import numpy as np
vocab = {}
with open('glove.6B.100d.txt', 'r') as f:
    for line in f:
        values = line.split()
        word = values[0]
        vec = np.asarray(values[1:], dtype='float32')
        vocab[word] = vec
print(f'Loaded {len(vocab)} word vectors')
```

GloVe

Полученные представления отражают важные линейные подструктуры векторного пространства слов: *получается связать вместе разные спутники одной планеты или почтовый код города с его названием.*



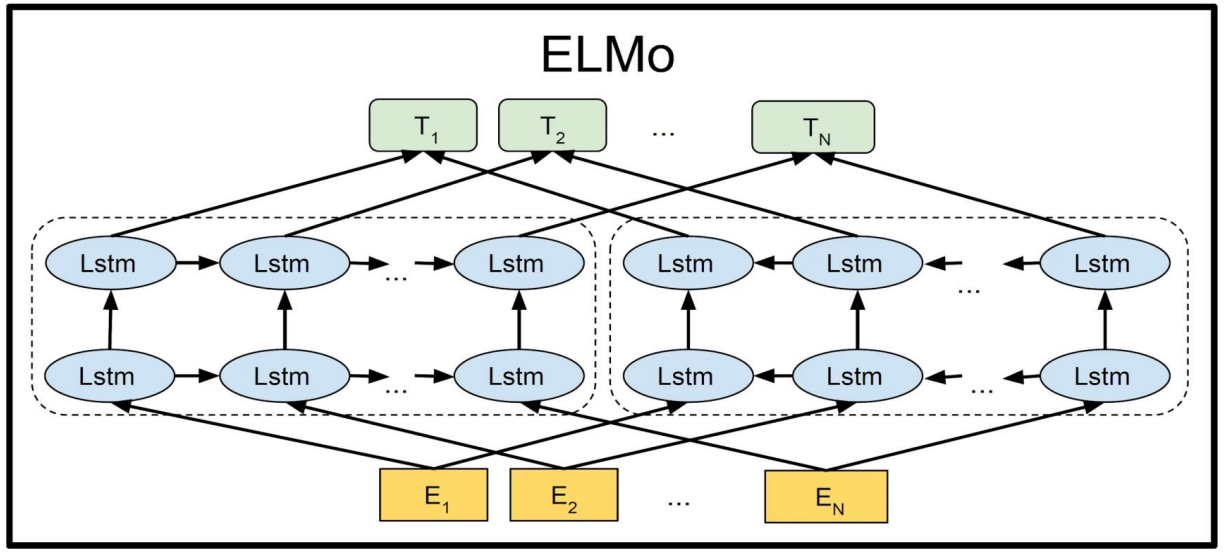

```python
import numpy as np
vocab = {}
with open('glove.6B.100d.txt', 'r') as f:
    for line in f:
        values = line.split()
        word = values[0]
        vec = np.asarray(values[1:], dtype='float32')
        vocab[word] = vec
print(f'Loaded {len(vocab)} word vectors')
```

ELMo (Embeddings from Language Models)

Метод эмбединга, который учитывает **контекст** предложения, **снимая** тем самым **семантическую неоднозначность**, присущую обычному эмбедингу.

Модель: обучение двунаправленных рекуррентных слоёв. Обученная сеть используется как "поставщик" контекстно зависимых векторов эмбединга слов. Эти векторы являются суммой скрытых состояний всех слоёв с некоторыми коэффициентами, которые служат параметрами обучения уже в конкретной задаче.

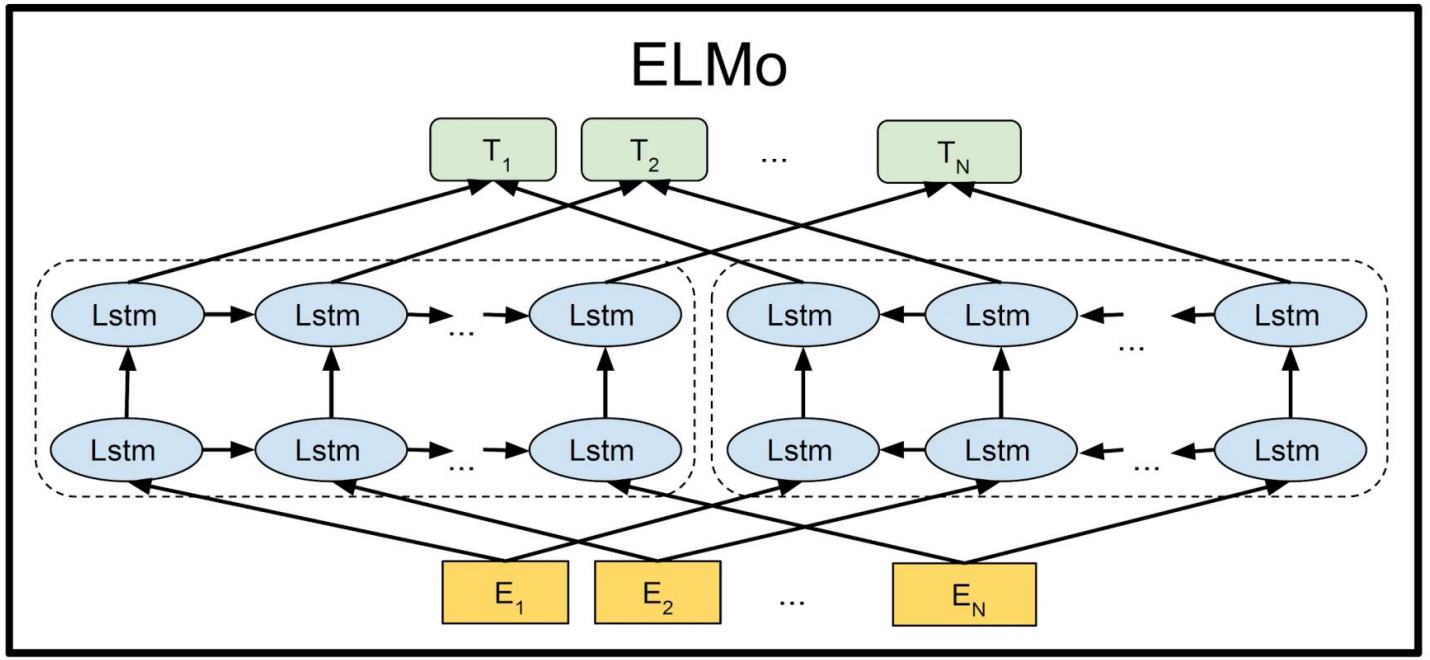
Разработана исследовательской группой Allen Institute for Artificial Intelligence (AI2).



ELMo (Embeddings from Language Models)

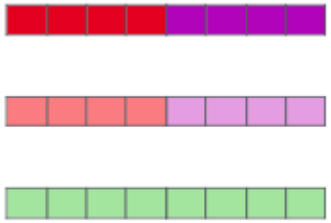
ELMo улучшает обычные методы векторного представления слов, такие как Word2Vec и GloVe, тем что представления слов учитывают контекст.

Для каждого слова ELMo генерирует различные векторы в зависимости от его контекста

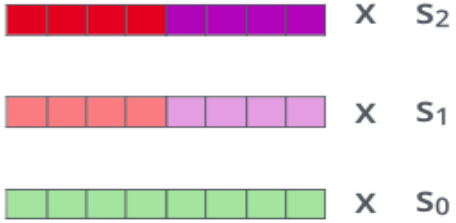


ELMo (Embeddings from Language Models)

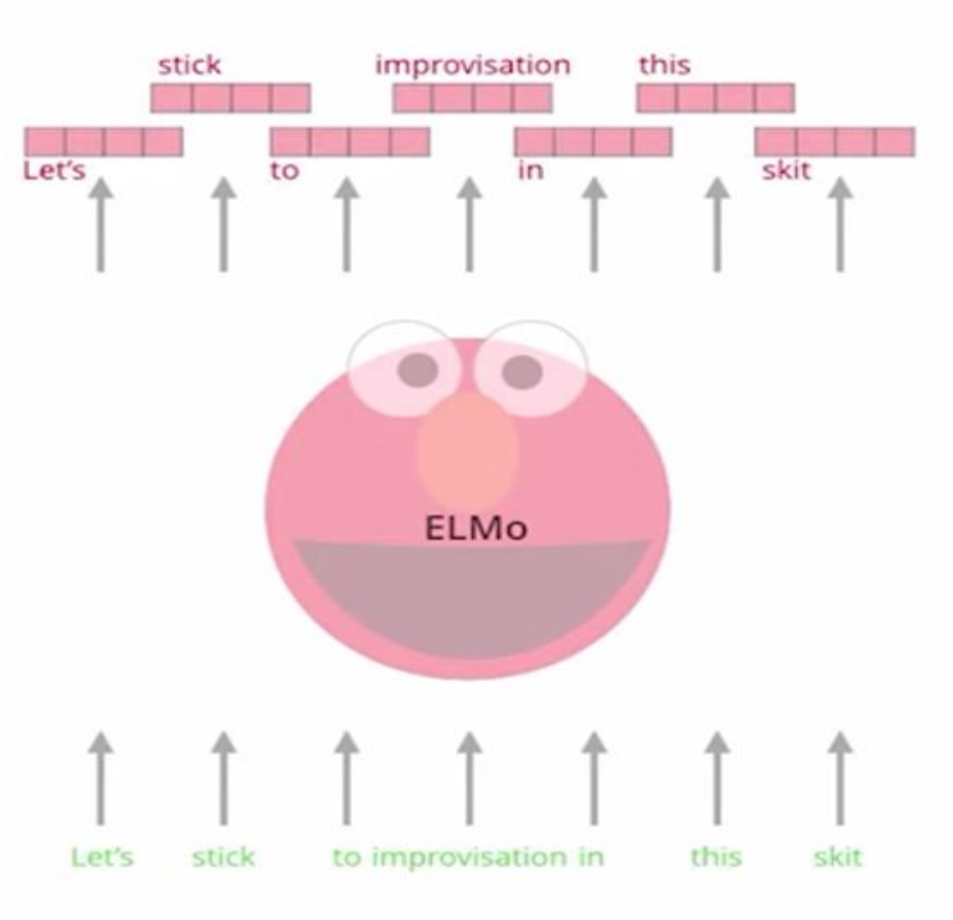
1- Concatenate hidden layer



2- Multiply each vector by a weight based on the task



3- Sum the (now weighted) vectors



ELMo эмбеддинги

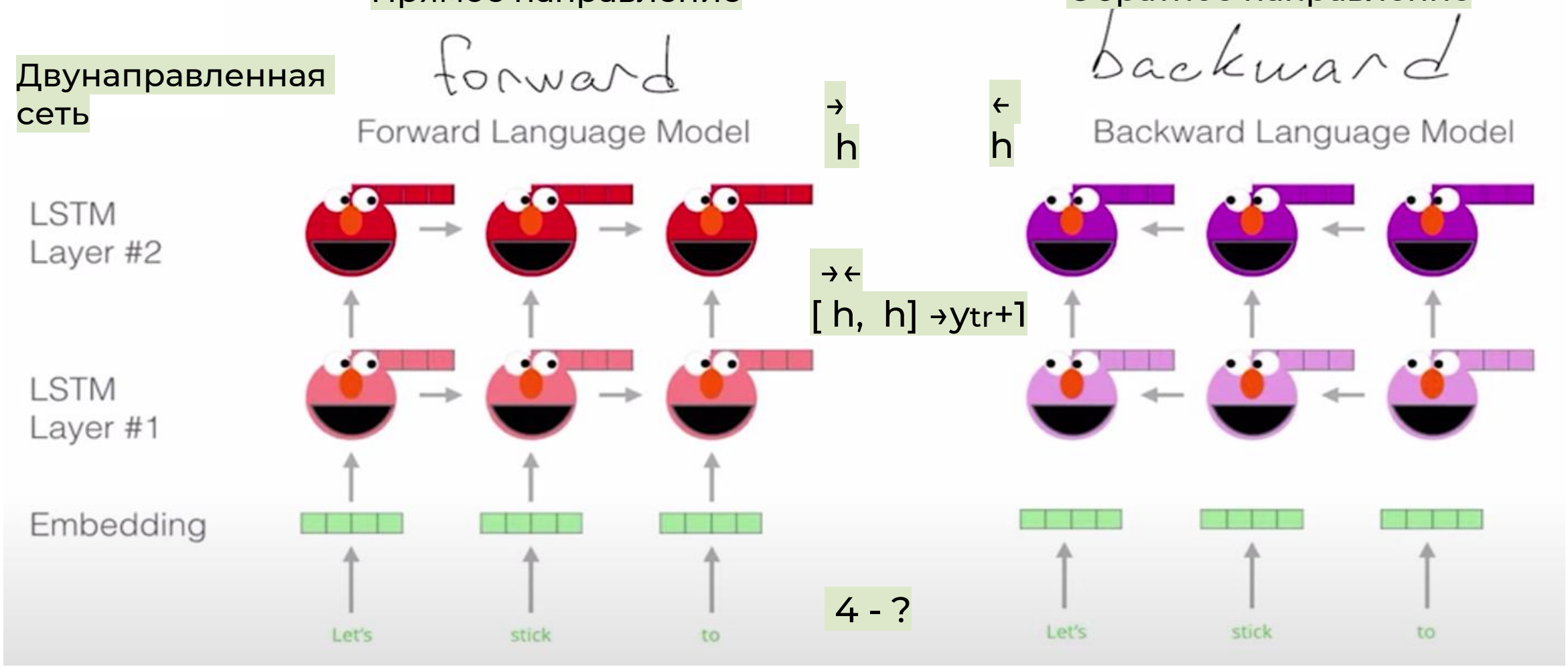
Слова для векторизации

ELMo (Embeddings from Language Models)

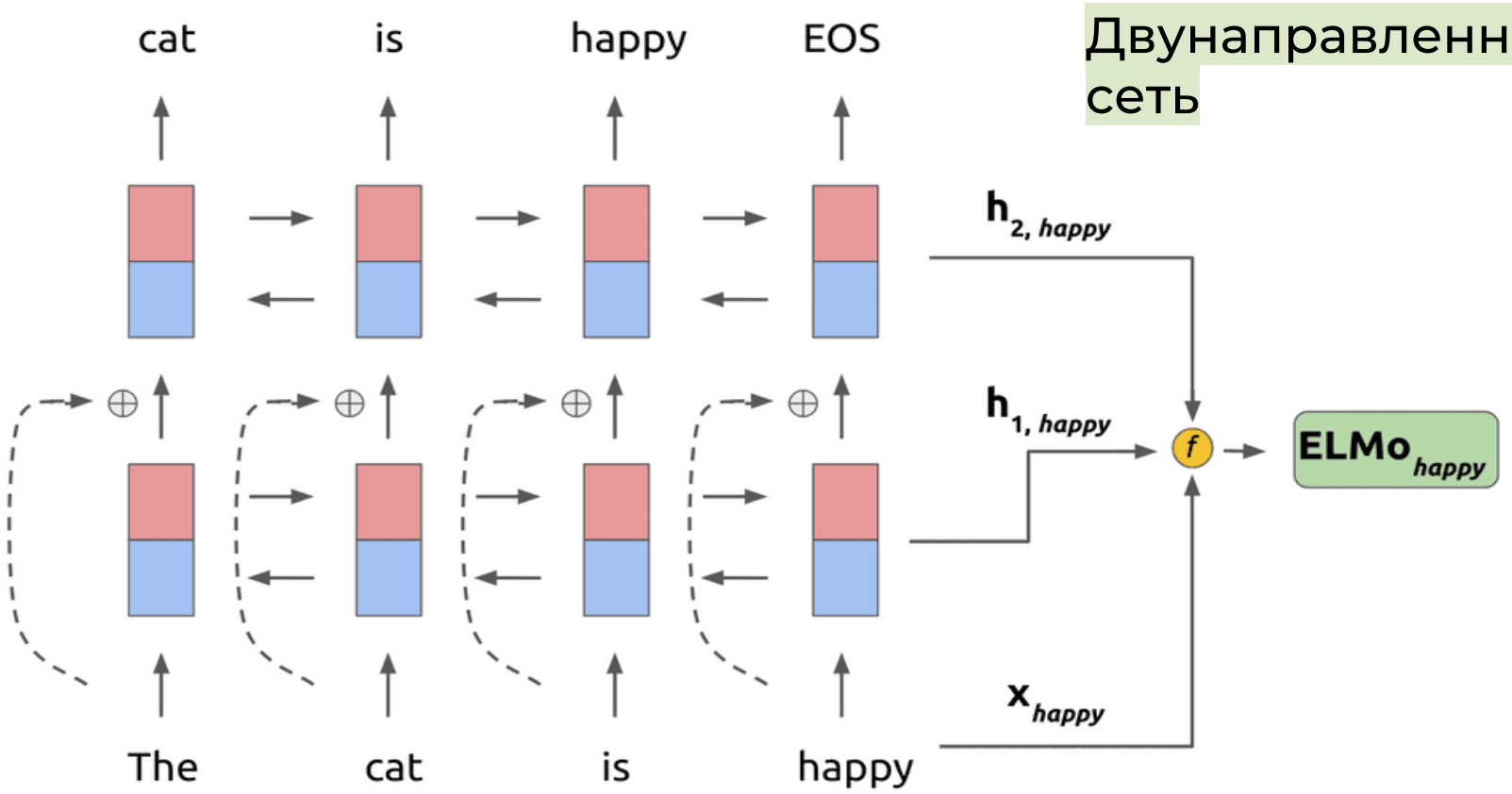
Прямое направление

Обратное направление

Двунаправленная сеть



ELMo (Embeddings from Language Models)



Двунаправленная сеть

→ h ← h

→←
[h, h] →ytr+1

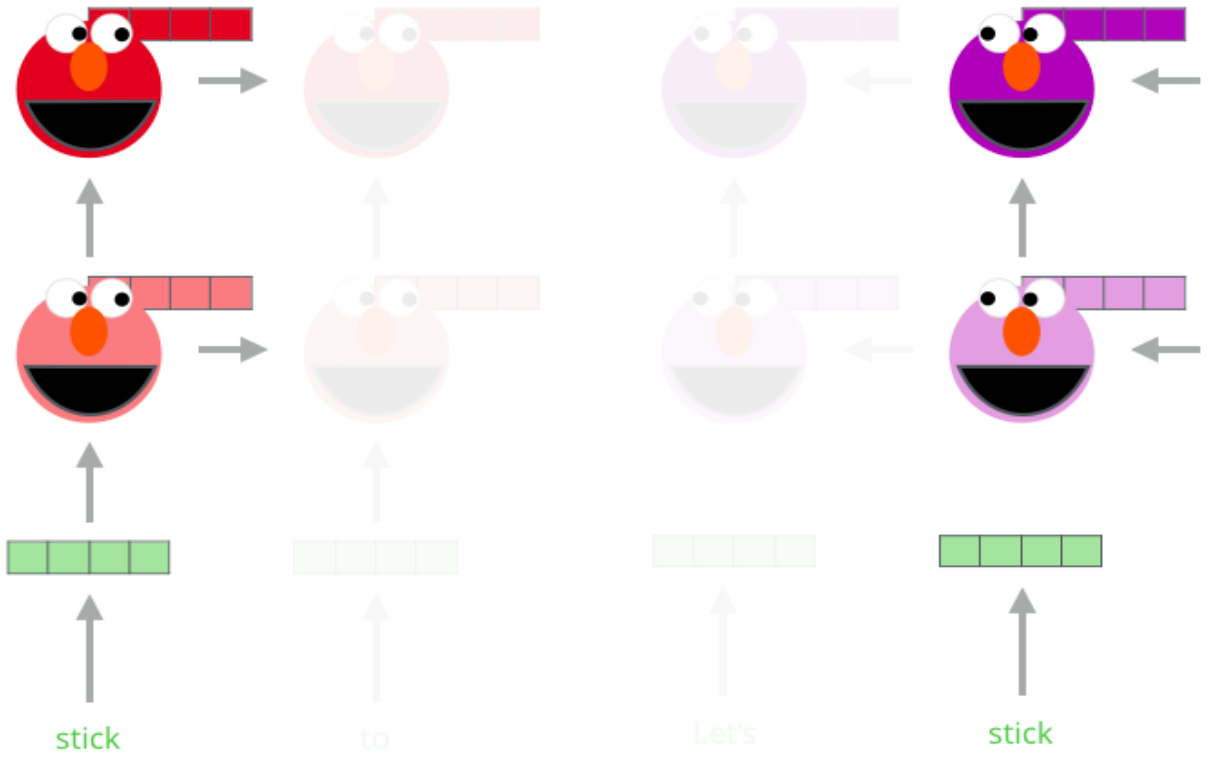
Пример объединения двунаправленных скрытых слоев и словесного представления для "happy", чтобы получить представление, специфичное для ELMo.
Примечание: упрощенное представление

ELMo после обучения (веса рассчитаны)

Для аналитики важно ***не предсказание сетки, а векторные представления слова*** (описания слова)

Описание слова зависит от контекста

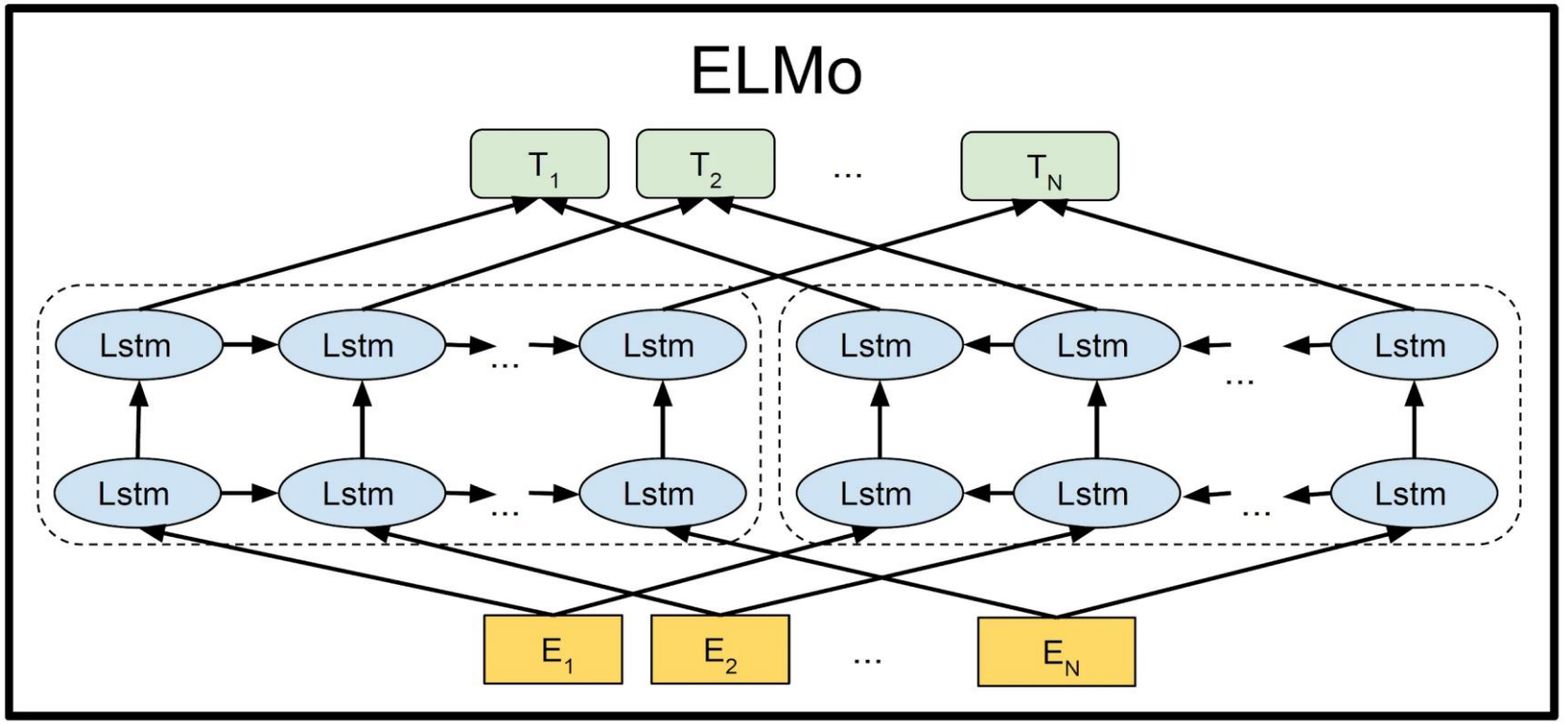
Храним модель, а не словарь



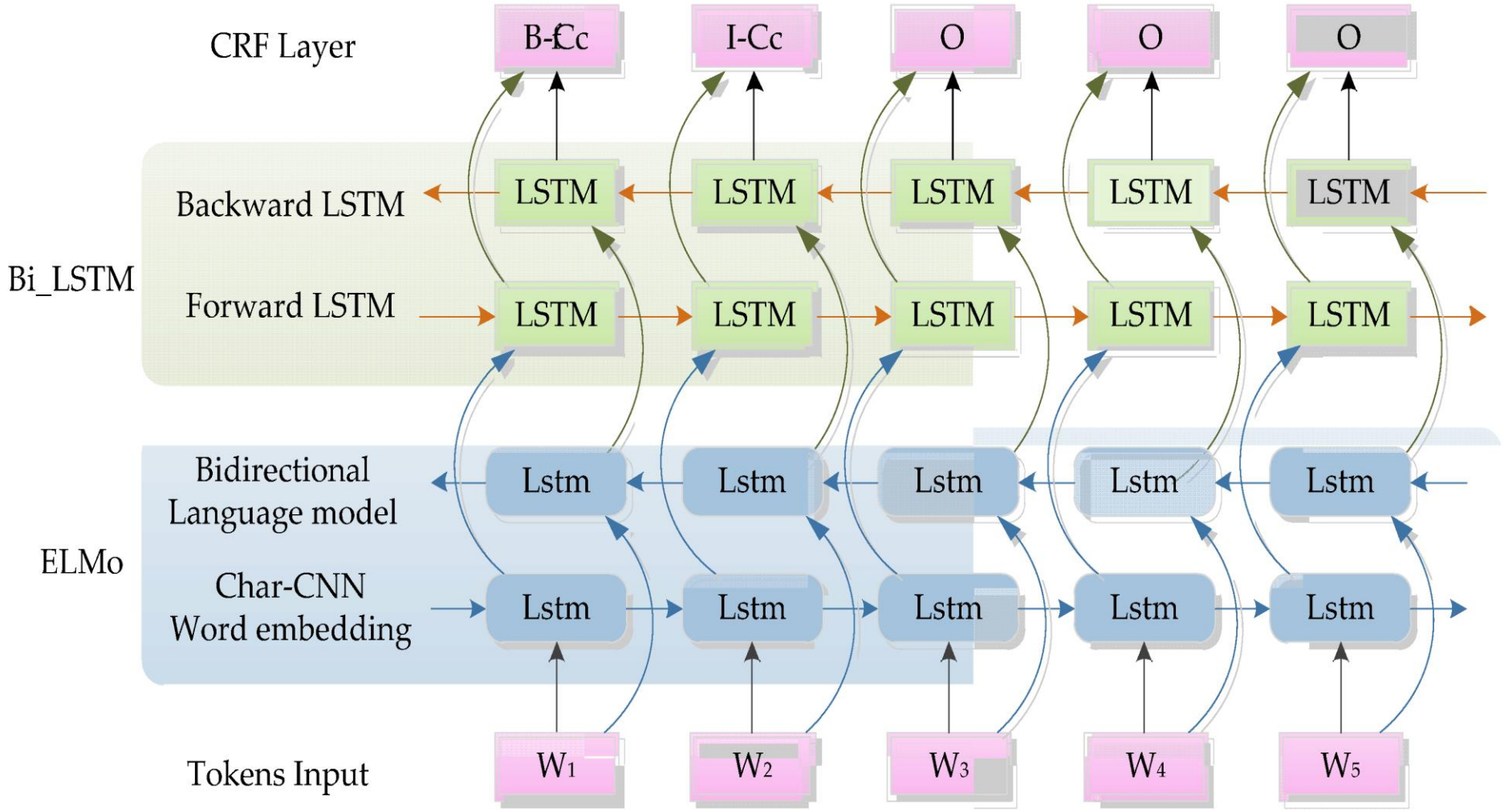
$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.$$

ELMo

Задача предсказания следующего слова с помощью двунаправленной нейронной сети. Интересует следующее представление слова. Описание зависит от контекста. Мы храним модель, а не словарь



ELMo и двунаправленная LSTM



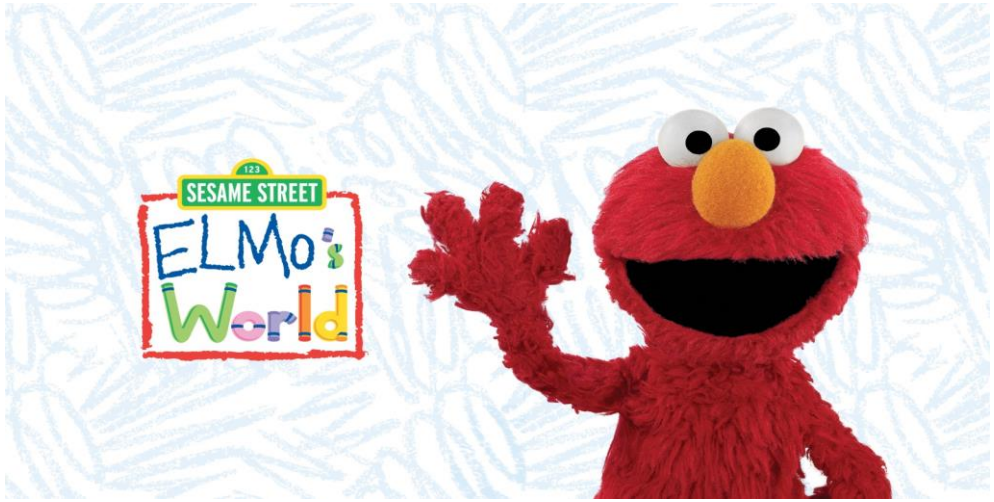
Реализация ELMo

```
from allennlp.modules.elmo import Elmo, batch_to_ids

# параметры, которые должны быть такими же, как у предобученной модели
options_file = "https://allennlp.s3.amazonaws.com/models/elmo/
2x4096_512_2048cnn_2xhighway/elmo_2x4096_512_2048cnn_2xhighway_options.json"

weight_file = "https://allennlp.s3.amazonaws.com/models/elmo/
2x4096_512_2048cnn_2xhighway/elmo_2x4096_512_2048cnn_2xhighway_weights.hdf5"

elmo = Elmo(options_file, weight_file, 2, dropout=0)

# используйте batch_to_ids, чтобы преобразовать предложения в индексы символов
sentences = [['First', 'sentence', '.'], ['Another', '.']]
character_ids = batch_to_ids(sentences)

embeddings = elmo(character_ids)

# embeddings['elmo_representations'] – это список из 2 тензоров (по одному для каждого
слоя в ELMo)
# каждый тензор имеет форму (2, 3, 1024), где 2 – это количество предложений, 3 –
количество слов в каждом предложении,
# а 1024 – размер векторного представления каждого слова.
```

ДЕМОНСТРАЦИЯ

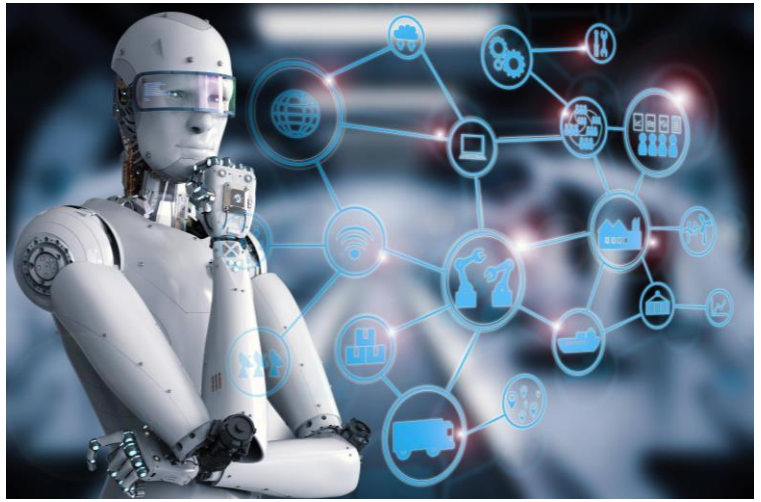
Повторение. Работа с GloVe для LTSM, GRU

Работа с ELMO



Полезные ссылки

1. Deep contextualized word representations. URL: https://arxiv.org/abs/1802.05365
2. The Illustrated BERT, ELMo, and co. URL: https://jalammar.github.io/illustrated-bert/
3. Нейросети для работы с последовательностями. URL: https://academy.yandex.ru/handbook/ml/article/nejroseti-dlya-raboty-s-posledovatelnostyami
4. BERT, ELMO и Ко в картинках (как в NLP пришло трансферное обучение). URL: https://habr.com/ru/articles/487358/
5. Deep Contextualized Word Representations with ELMo. URL: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/
6. Word Embeddings. URL: https://lena-voita.github.io/nlp_course/word_embeddings.html
7. ML: Буквенный и семантический эмбединг. URL: https://qudata.com/ml/ru/NN_Embedding_Elmo.html#ELMo
8. GloVe: Global Vectors for Word Representation. URL: https://nlp.stanford.edu/projects/glove/
9. GLoVE: теория и реализация на Python. URL: https://evogeek.ru/articles/268562/
10. Большой русский датасет. Russian text datasets + vanila GloVe + quantization. URL: https://github.com/natasha/navec
11. Russian Glove. URL: https://www.kaggle.com/datasets/tunguz/russian-glove
12. Deep Pavlov. Pre-trained embeddings. ELMo. URL: https://deeppavlov.readthedocs.io/en/master/features/pretrained_vectors.html



ИНСТИТУТ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС

Спасибо за внимание!

Telegram

E-mail

Контакты

Корнеева Елена

https://t.me/Allyonzy

e.korneeva@innopolis.ru