△ Generalized agents (Yuke Zhu)

△ Big data & RL agents

- web data → simulated data → real data

→ encoders ?

- vids, texts] → correlation model → give reward

Big models | RL

△ LLM → task planning

observation, history experiences, task goal } → tokenizer ? → pretrained language model → policy model → action

e.g.
input: get me a coffee
output: series of actions . . .

△ GPT

Generative
Pretrained
Transformer

△ token

text → little pieces of words
images → chunk of pixels
sound → patches of sound.

encode w/ meaning

tokens ↓ vectors

[I am a pig]

△ attention block operation

[I am a pig]

- update the values of each vec.

△ MLP [ ]
like asking lost list of question ↓ throw answer

△ Attention & MLP repeat

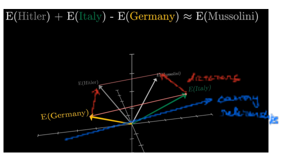
△ predict next objective

△ word embedding.

△ Embedding matrix { } various words

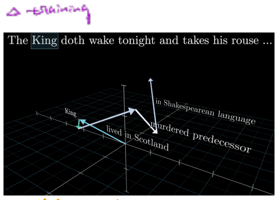
$W_E = [\ | \ | \ \cdots \ ]$ @ i=0, random ↓ then learn

GPT projects word to $R^{12,288}$ w 50257 token



dot products : correlation

△ training



w/ data the token "king" could start from 1 place; then w/ different data flowing in, we can then pull the token around, then establish connection, like above, perfect visualization.

then, we can do prediction, i.e., generative AI

also, we need "attention", the relation between words.

△ unembedding matrix

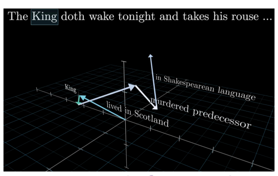
$W_u = [\ \ ] \in R^{50257 \cdot 12288}$

△ softmax

values → prob. distribution

$\frac{e^{x_i/T}}{\sum_{n=0}^{N-1} e^{x_n/T}}$ temperature. can add some randomness to the prediction

△ Transformer



- get attention ! between words
- allow words to encode other meanings from other words.

△ Query layer

e.g.
A fluffy blue creature roamed the verdant forest.
$\vec{E}_1$ $\vec{E}_2$ . . . $\vec{E}_8$

multiple w/ W_Q, query weights

Q_1 Q_2 . . . Q_8

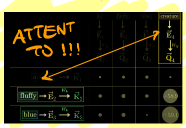
- these represent the query type. for instance, Q_4's query might be "searching for adjectives". & this is also "embed w/ a vector".
- GPT-3 is using R^{128} query space

△ Keys layer

$\vec{E}_1$ $\vec{E}_2$ . . . $\vec{E}_8$
W_K ↓
K_1 K_2 . . . K_8

- this "answers" the "query"
- $Q_i \cdot K_i$ → larger, align!

△ Construct a table of Q & K's dot product



sometimes w/ masking so that future data will not look to previous data...

△ softmax

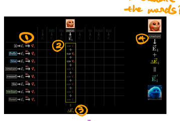
[Q·K] → normalized

Attention Pattern! (sometimes try to downsize)

△ Value layer

$\vec{E}_1, \vec{E}_2$ . . . $\vec{E}_8$
W_V ↓
V_1 V_2 . . . V_8 → what to add to relate the words?

same dim.

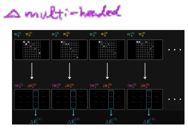


"one head of attention"

△ Update embedding

Original embedding $\vec{E}_i + \Delta\vec{E}_i^{(1)} + \Delta\vec{E}_i^{(2)} + \Delta\vec{E}_i^{(3)} + \Delta\vec{E}_i^{(4)} + \cdots$

△ multi-headed



△ multilayer

repeat!

for translation

E_1 E_2 ... E_n english e.g.
F_1 ... F_n french, e.g.