MUNI FI

Understanding LLM

PA154 Language Modeling ()

Pavel Rychlý

pary@fi.muni.cz

28 February 2024

LLM = Large Language Models

- new term, big hype
- big expectations (AGI)
- little understanding

Language

- not normal distribution (Zipf's)
  - many rare events
- ambiguous words
- variable - changing, sublanguages

Symbolic processing

ambiguous, variable, rare events leads to:

- hard for symbolic manipulations
- multiple words for same meaning
- same word for multiple meanings

Word embeddings

- words represented as vectors
  - one word = vector of 500 numbers
- similar words - closer vectors
  - not important
- dimensions can represents different features

Word features

- gramatical
  - part of speech, nuber, gender
- syntactic
  - used with “in”/“at”, always with a particle
- semantic
  - positive sentiment, movement meaning, fruits
- style
  - formal, colloquial
- domain
  - math, biology
- form

Word features

- features are not independent
  - math – scientific
  - used with “in” – noun
  - in capital form – proper noun
- features are not discrete
- each feature corespond to a (set of) dimension
- most features are valid for only small set of words
  - most words have (almost) 0 for most features
- multiple meanings = union of features

Phrase/sentece embeddings

- vector space is very big
  - just 2 values (0,1) in each dimension = $2^5 00$ combinations, $10^5 0$
- same vector space for phrases
  - average of words
  - different words with same meanings – same embedding Czech president, president of the Czech Republic
- sentences, paragraphs, documents, ...

Neurons

- input: vector
- output: number in <0,1>
- $\sigma(x * w + b)$
- linear classifier
- hyperplane cutting the vector space
- selects one feature

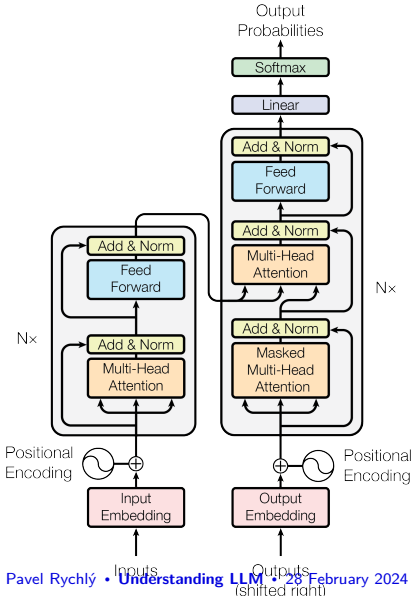
Neuron networks

- second layer can implement and/or operators
- grouping of features
  - all 10 features
  - 5 out of 10 features
- cutting the (original) vector space by several hyperplanes
  - regions in the vector space

Transformers

- *Attention Is All You Need* paper
  - attentions are important, but there other important components
- encoder/decoder
- layers of same structure with different parameters
- input: list of tokens (words)
- output depends on model

Transformers



BERT - encoder only

- output: vector (embedding) for each token
- each layer:
  - attention
  - feed forward network (2 layers)
  - direct links

Direct links

- $LayerNorm(x + Sublayer(x))$
- adding some information into original embedding
- for each token separately
- changing the original token embedding by small steps
- some part of the original information is preserved

Analogy

- for each token we have stream
- sending information/signal down the stream
- each layer can make a small transformation
  - adding information from context
  - decreasing/increasing importance of a feature

Feed forward

- first layer
  - single neuron selects a feature
  - result: vector of features
- second layer
  - select feature combinations
  - transforms them into word vector space

Attention

- highlites some features from context
- multi head = more features in single step
- transforms them into word vector space

GPT - decoder only

- auto-regressive decoder
- the last token is generating new token

Summary

- transormer is a flow of information
- starts with word embeddings
- adding/changing fatures – moving in the vector space
- all vectors are in the same vector space