University of Nottingham
UK | CHINA | MALAYSIA

COMP3055 Machine Learning

Topic 18 – Others

Zheng Lu

2024 Autumn

Pre-training and Fine-tuning

- Practical situation
  - You have some specific task that you want to apply machine learning techniques, for example, classifying medical dialogues into certain medical categories.
  - You only have a tiny labelled dataset to train with.
  - There is a general dataset with large number of labelled data.

slide credit: M. Gormly, J. Hewitt, G. Durrett, A. Goldie

Pre-training and Fine-tuning

- Pre-training
  - Train the large network models on a large amount of general data
  - Save the network parameters
  - Learn general things

Pre-training and Fine-tuning

- Fine-tuning
  - Initialize the large network model using parameters trained during pre-training.
  - Continue training with additional data (usually small number and task-specific).
  - Adapt to the task.

Using Large Model for Your Own Task

- Large network models trained on a large general dataset.
- Usually target general tasks.
- Using large language models as encoder or feature extractor for your own problem.
  - Backbone for object detection, resnet, inception, etc.
  - Language model for natural language processing, BERT, GPT, etc.

Using Large Model for Your Own Task

- Fine-tune model for your own task.
  - Directly add a classifier after the large model.
  - Update the parameters during the fin
- Create a new network model
  - Add new networks after the large model.
  - During training
    - Freeze the large model and update the subsequent network only.
    - Update both the large model and the subsequent network.

Transformer

- Self-attention
  - Each word' representation as a query to access and incorporate information from a set of values.
  - How important a word is related to others.
- Position representations
  - Specify the sequence order, since self attention is an unordered function of its inputs.
- Nonlinearities
  - At the output of the self-attention block.
  - Frequently implemented as a simple feed-forward network.
- Masking
  - Keeps information about the future from "leaking" to the past.

Self-Attention

- Attention operates on **queries**, **keys**, and **values**.
  - We have some **queries** $q_1, q_2, \ldots, q_T$. Each query is $q_i \in \mathbb{R}^d$
  - We have some **keys** $k_1, k_2, \ldots, k_T$. Each query is $k_i \in \mathbb{R}^d$
  - We have some **values** $v_1, v_2, \ldots, v_T$. Each query is $v_i \in \mathbb{R}^d$
- In **self-attention**, the queries, keys, and values are drawn from the same source.
  - For example, if the output of the previous layer is $x_1, \ldots, x_T$, (one vec per word) we could let $v_i = k_i = q_i = x_i$ (that is, use the same vectors for all of them!)
- The (dot product) self-attention operation is as follows:

$$e_{ij} = q_i^\top k_j$$

Compute **key-query** affinities

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

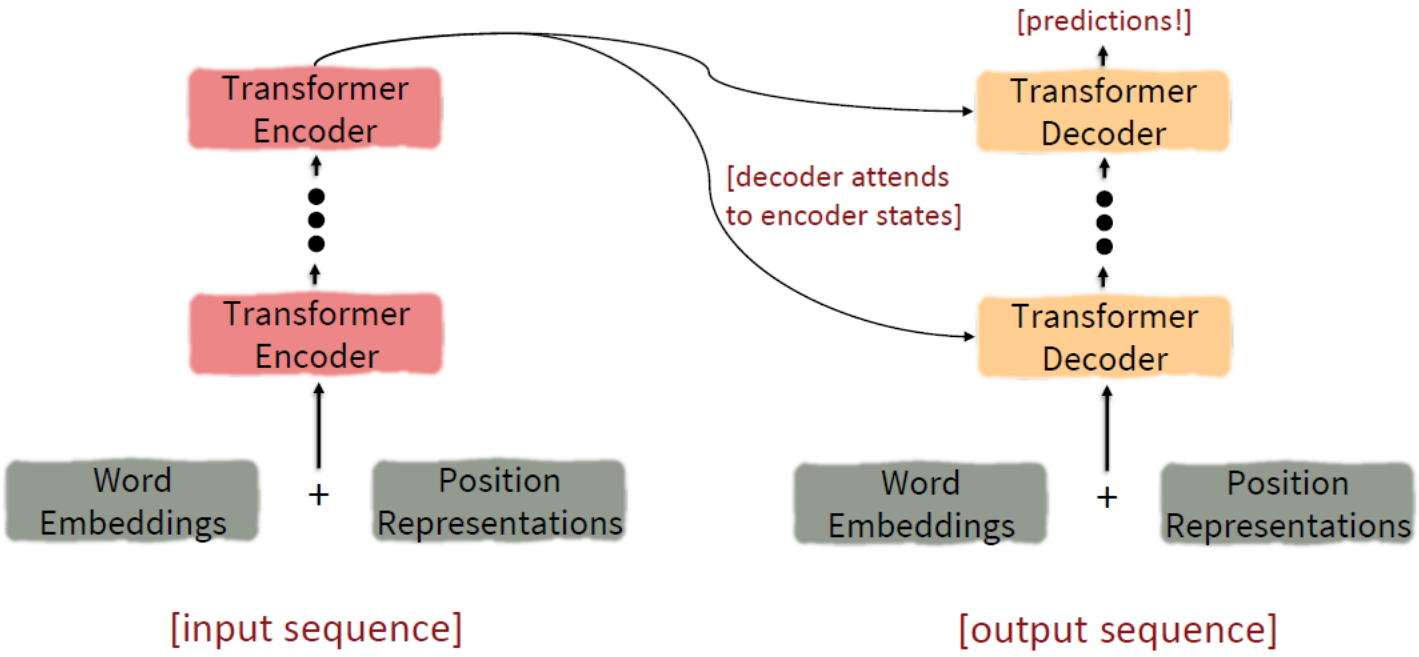
Compute attention weights from affinities (softmax)

$$\text{output}_i = \sum_j \alpha_{ij} v_j$$

Compute outputs as weighted sum of **values**

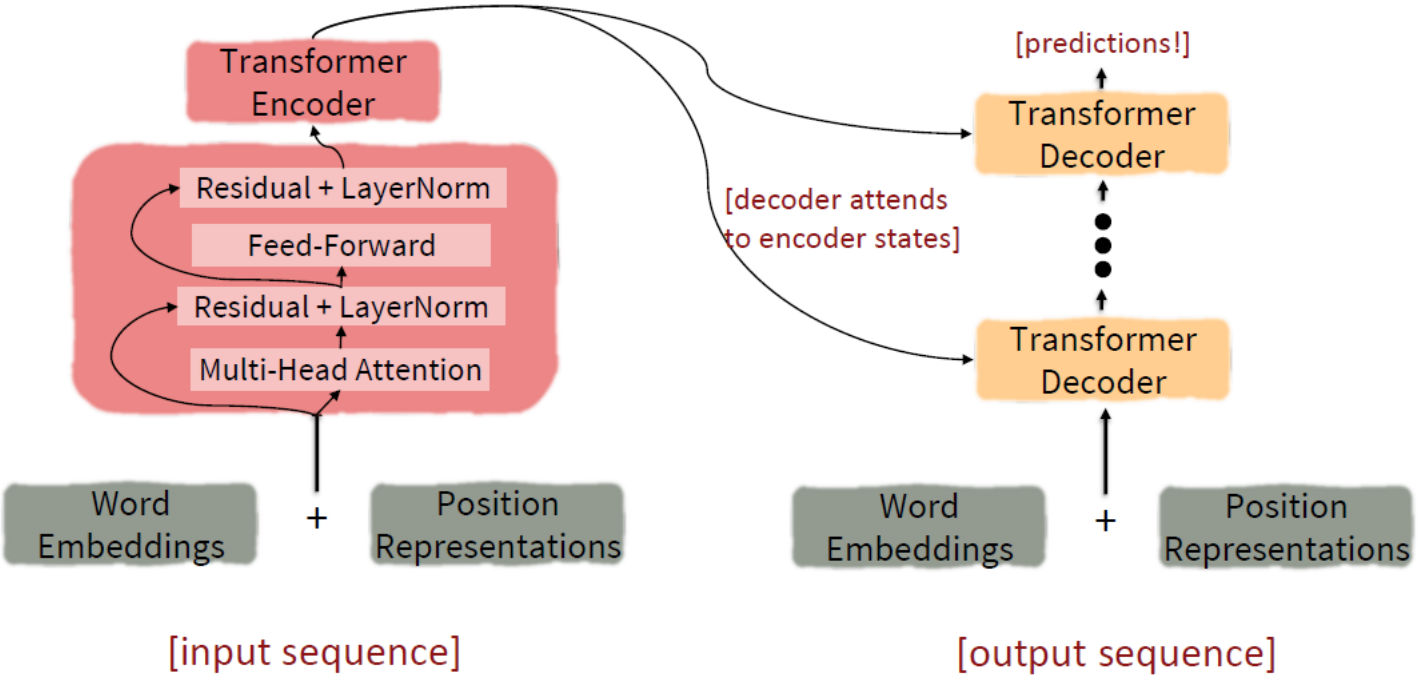
Transformer

- Encoder + decoder



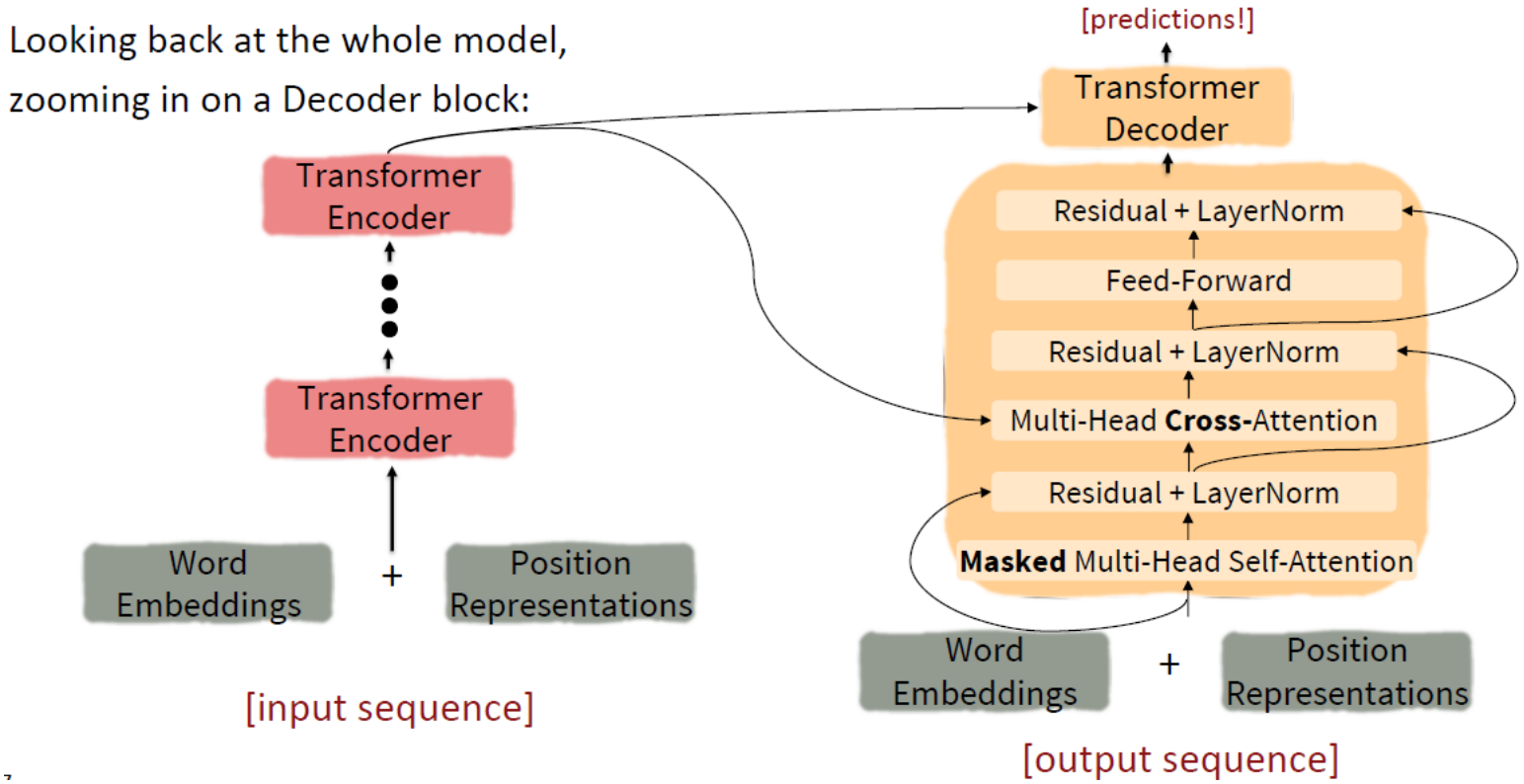
Transformer

- Zooming into an encoder block



Transformer

- Zooming into an encoder block

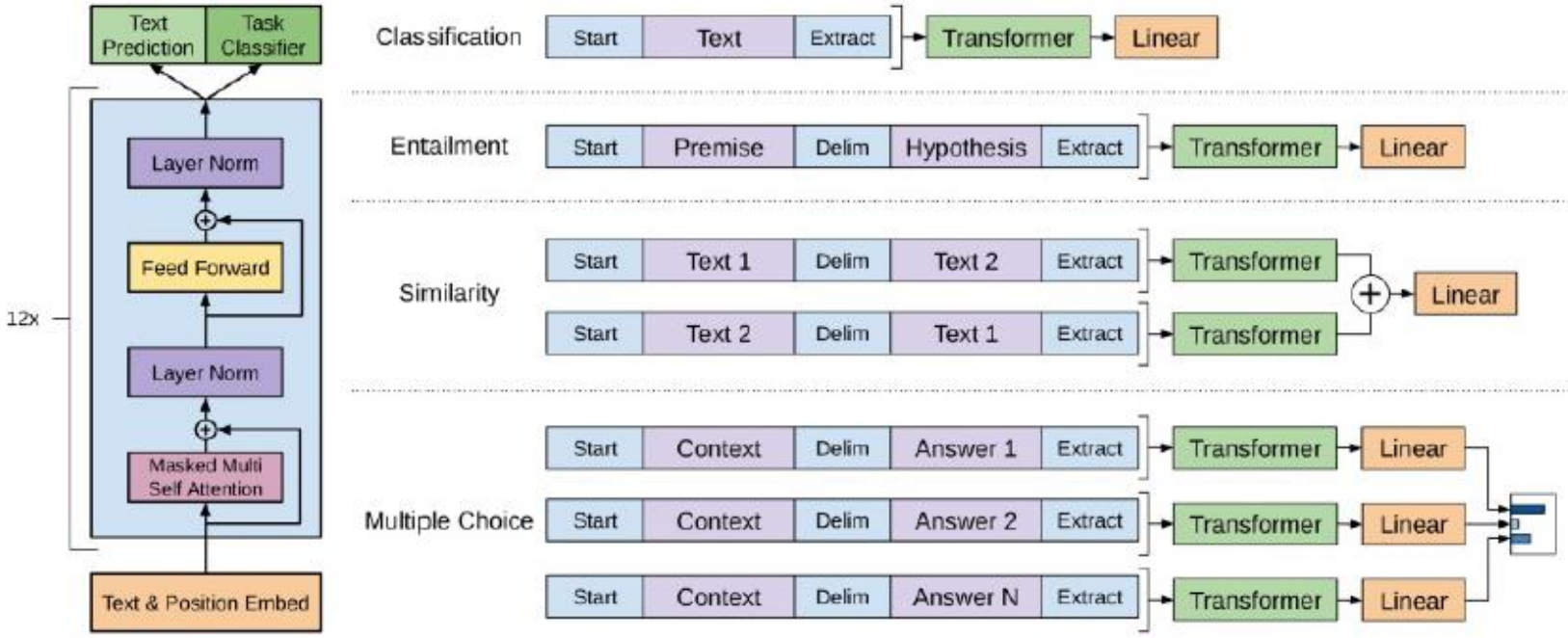


GPT

- Generative pre-trained transformers.
  - A type of large language model.
  - Based on the transformer architecture.
  - Pre-trained on large datasets of unlabeled text.

GPT

How do we format inputs to our decoder for **finetuning tasks?**



The linear classifier is applied to the representation of the [EXTRACT] token.

GPT-2

- A larger version of GPT trained on more data, was shown to produce relatively convincing samples of natural language.

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

GPT-2

- Trained on 40GB of text collected from upvoted links from reddit ->1.5B parameters.

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

GPT-3

- In-context learning, very large models
  - Very large language models seem to perform some kind of learning without gradient steps simply from examples you provide within their contexts.

- The in-context examples seem to specify the task to be performed, and the conditional distribution mocks performing the task to a certain extent.

  Input (prefix within a single Transformer decoder context):
  “ thanks -> merce
  hello -> bonjour
  mint -> menthe
  otter -> ”
  Output (conditional generations):
  loutre…”

GPT-n series

OpenAI's "GPT-n" series

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus:[27] 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, 2018[8]	30 days on 8 P600 GPUs, or 1 petaFLOP/s-day.[8]
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, 2019 (initial/limited version) and November 5, 2019 (full version)[28]	"tens of petaflop/s-day",[29] or 1.5e21 FLOP.[30]
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion[31]	499 billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, 2020[29]	3640 petaflop/s-day (Table D.1 [29]), or 3.1e23 FLOP.[30]
GPT-3.5	Undisclosed	175 billion[31]	Undisclosed	March 15, 2022	Undisclosed
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.[26]	Undisclosed. Estimated 1.7 trillion[32]	Undisclosed	March 14, 2023	Undisclosed. Estimated 2.1e25 FLOP.[30]