Toward a Tabular Foundation Model

October 2024

other examples:

- energy consumption prediction
- predictive maintenance

Classic Supervised Learning

specific models for each task and data set

feature engineering

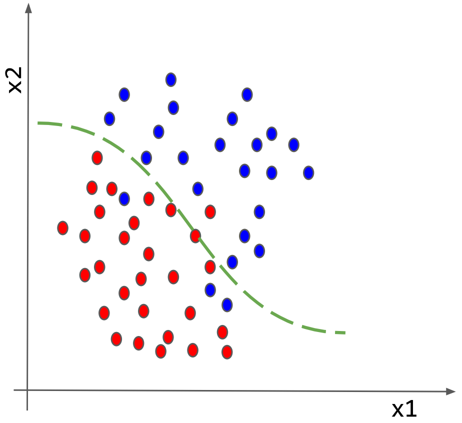
input-output mapping

Example: Spam Filtering

Classify emails as spam or no spam

use accordingly ***labeled emails as training set***

use information like **occurrence of specific words** or **email length** as **features**



features x1 and x2

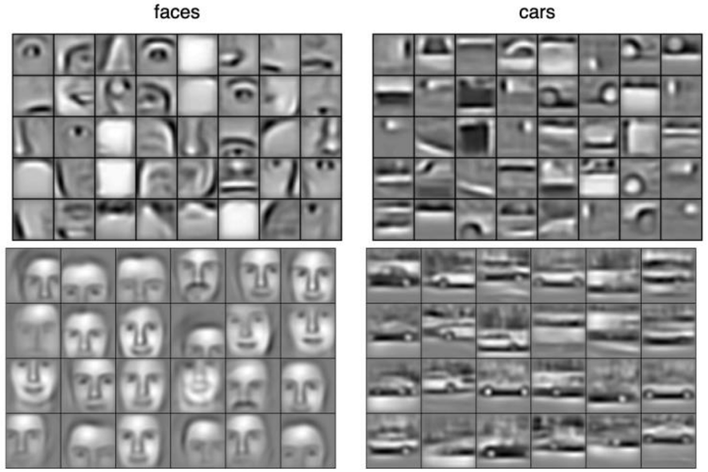
spam, no spam

Ladder of Generalization

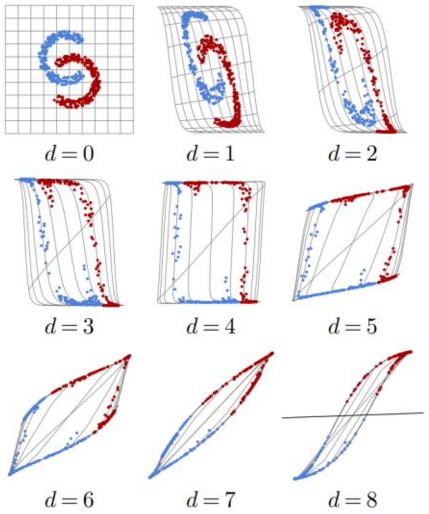
classic ML: feature engineering

deep learning: feature learning

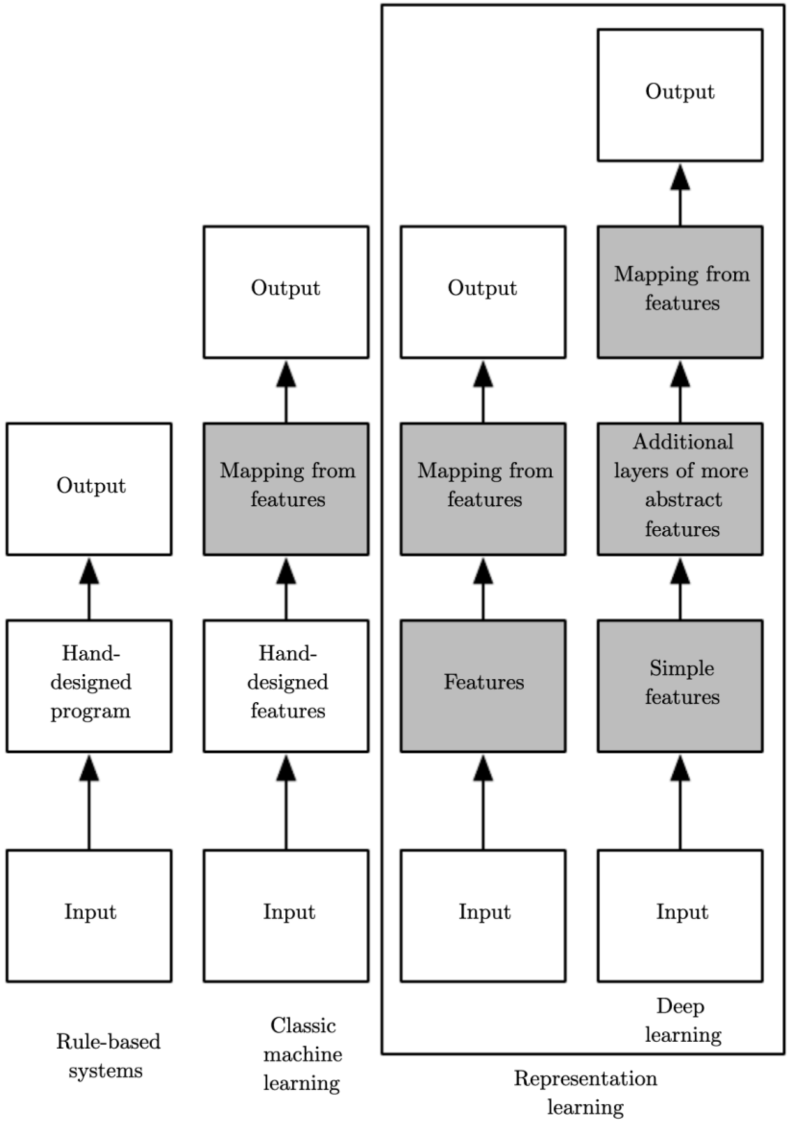
(hierarchy of concepts learned from raw data in deep graph with many layers)



source



source



source

Transfer Learning

idea:

- generic pre-training of foundation models on huge data sets
- subsequent finetuning for specific tasks on small(er) data sets

(usually done by means of deep learning methods, thanks to its compositional nature)

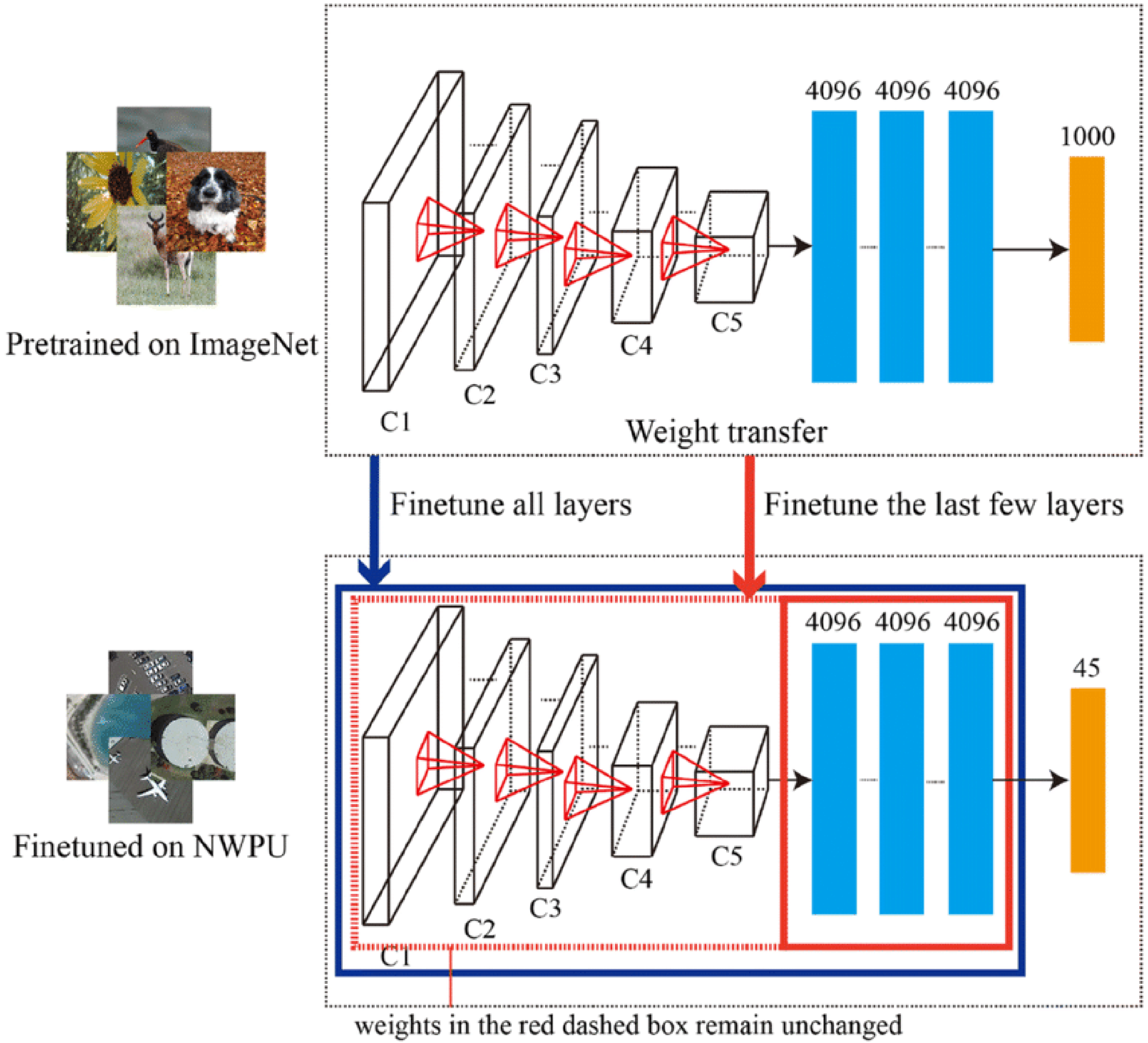
very successful for:

- computer vision (e.g., object classification)
- language models (e.g., BERT, GPT)

other examples:

- visual defect detection
- support chip design

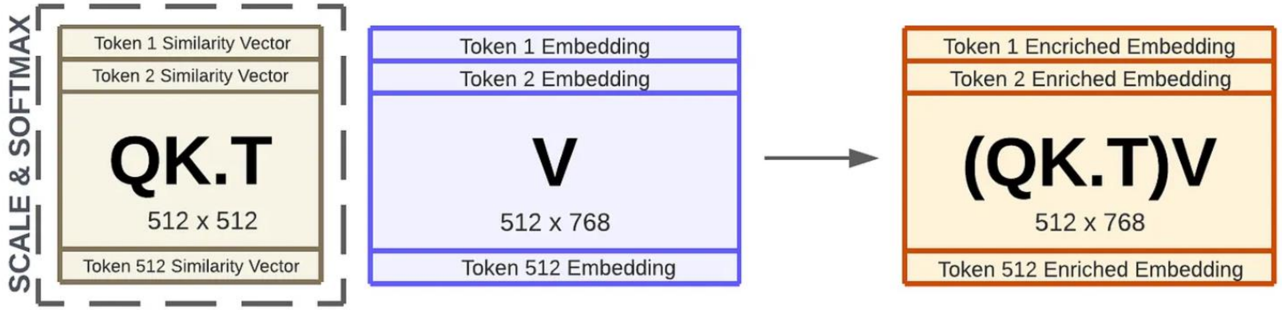
CNN Finetuning



source

Language Models: Contextual Semantics

- self-supervised learning: e.g., next/masked-word prediction
- tokenization: split text into chunks (e.g., words)
- semantics by means of vector embeddings: e.g., via bag-of-words (or end-to-end in transformer)
- positional encoding & embeddings: order of sequence
- contextual embeddings: (self-)attention (weighted averages: influence from other tokens)



source

example use case: small language models on edge devices

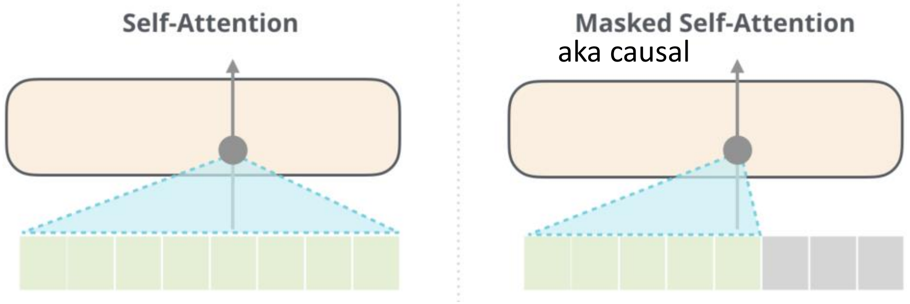
Encoder vs Decoder LLMs

encoder-only LLMs

- prime example: BERT
- self-supervised pre-training: masked-word prediction
- finetuning on downstream tasks (e.g., sequence classification)
- can’t generate text
- can’t be prompted

decoder-only LLMs

- prime example: GPT
- self-supervised pre-training: next-word prediction
- instruction tuning (e.g., RL from human feedback)
- generate text: chat bots
- prompt engineering (zero-/few-shot)



source

Structured/Tabular vs Unstructured Data

unstructured data: homogenous

→ deep learning rules

→ allows transfer learning (foundation models in CV and NLP)

ImageNet

The Lord of the Rings

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from Lord of the rings)

This article is about the book. For other uses, see The Lord of the Rings (disambiguation).
"War of the Ring" redirects here. For other uses, see War of the Ring (disambiguation).

The Lord of the Rings is an epic[1] high fantasy novel[a] by the English author and scholar J. R. R. Tolkien. Set in Middle-earth, the story began as a sequel to Tolkien's 1937 children's book *The Hobbit*, but eventually developed into a much larger work. Written in stages between 1937 and 1949, *The Lord of the Rings* is one of the best-selling books ever written, with over 150 million copies sold.[2]

The title refers to the story's main antagonist,[b] Sauron, the Dark Lord who in an earlier age created the One Ring to rule the other Rings of Power given to Men, Dwarves, and Elves, in his campaign to conquer all of Middle-earth. From homely beginnings in the Shire, a hobbit land reminiscent of the English countryside, the story ranges across Middle-earth, following the quest to destroy the One Ring, seen mainly through the eyes of the hobbits Frodo, Sam, Merry, and Pippin. Aiding Frodo are the Wizard Gandalf, the Men Aragorn and Boromir, the Elf Legolas, and the Dwarf Gimli, who unite in order to rally the Free Peoples of Middle-earth against Sauron's armies and give Frodo a chance to destroy the One Ring in the fire of Mount Doom.

Although often mistakenly called a trilogy, the work was intended by Tolkien to be one volume in a two-volume set along with *The Silmarillion*.[3][T 3] For economic reasons, *The Lord of the Rings* was first published over the course of a year from 29 July 1954 to 20 October 1955 in three volumes rather than one[3][4] under the titles *The Fellowship of the Ring*, *The Two Towers*, and *The Return of the King*; *The Silmarillion* appeared only after the author's death. The work is divided internally into six books, two per volume, with several appendices of background material.[c] These three volumes were later published as a boxed set, and even finally as a single volume, following the author's original intent.

structured data: heterogenous

→ feature engineering needed

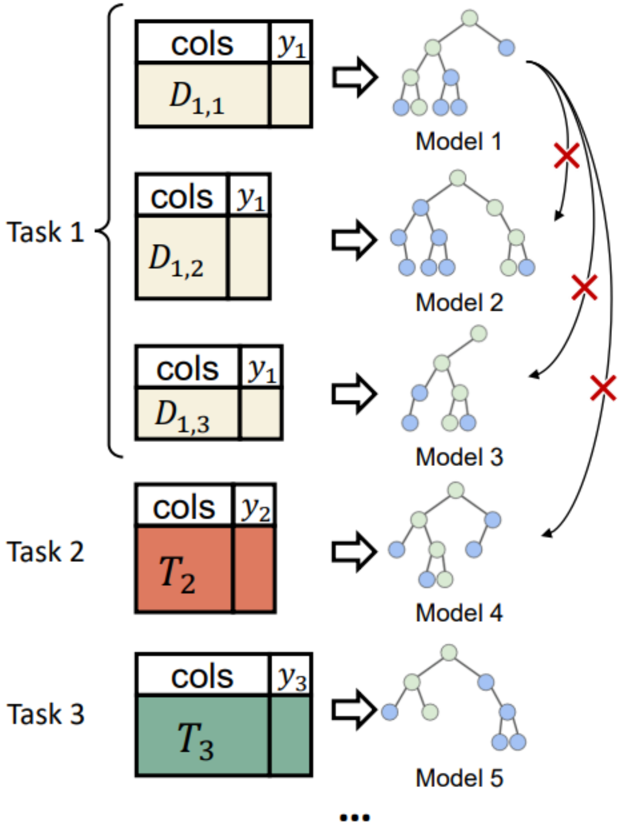
→ deep learning loses its advantage over shallow methods

→ e.g., gradient boosting still used a lot

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD	Normal	175000
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	Normal	210000
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	Normal	266500
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD	Normal	142125
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD	Normal	147500

[1460 rows x 81 columns]

Idea of Tabular Foundation Models



Existing works:
- **one** model, **one** dataset;
- not transferable across datasets
- if transferable, needs finetuning on each dataset

source

pre-training across data sets and even different tasks

finetuning on small data sets

benefit from world knowledge in LLMs, for example in terms of data imputation

Overcome the Data Integration Challenge

generate vector embeddings for each entry (column name & row value)

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD	Normal	175000
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	Normal	210000
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	Normal	266500
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD	Normal	142125
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD	Normal	147500

[1460 rows x 81 columns]

convert to prompts for LLM calls

extract embeddings
(average of last hidden state)

```
tensor([[[ 0.2406, -0.0340, -0.5141,  ...,  0.0476, -0.1114, -0.0198],
         [ 0.9594,  0.9598,  0.4653,  ...,  1.1557,  1.3493,  1.0192],
         [ 0.6332,  0.3364,  0.8191,  ...,  0.7699,  1.2227,  0.7292],
         ...,
         [ 1.1688,  0.4768,  0.5724,  ...,  0.7802,  0.9589,  1.1461],
         [ 0.8999,  0.5200,  0.7979,  ...,  0.9777,  0.7836,  0.8079],
         [ 0.9854,  0.2225,  0.9218,  ...,  0.9033,  0.9173,  0.8929]],
```

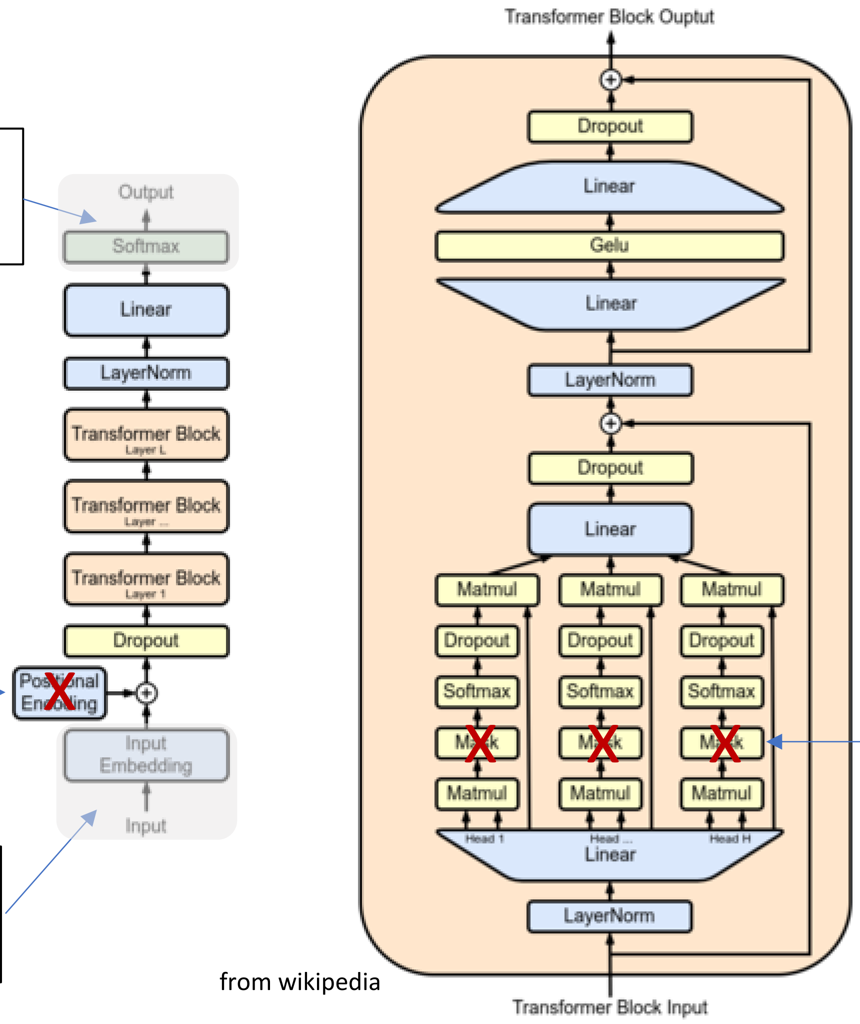
Transformer for Numerical Data

adaptions to GPT architecture:

replace language model head and loss function to reflect tabular prediction setup

drop positional encoding (permutation invariance)

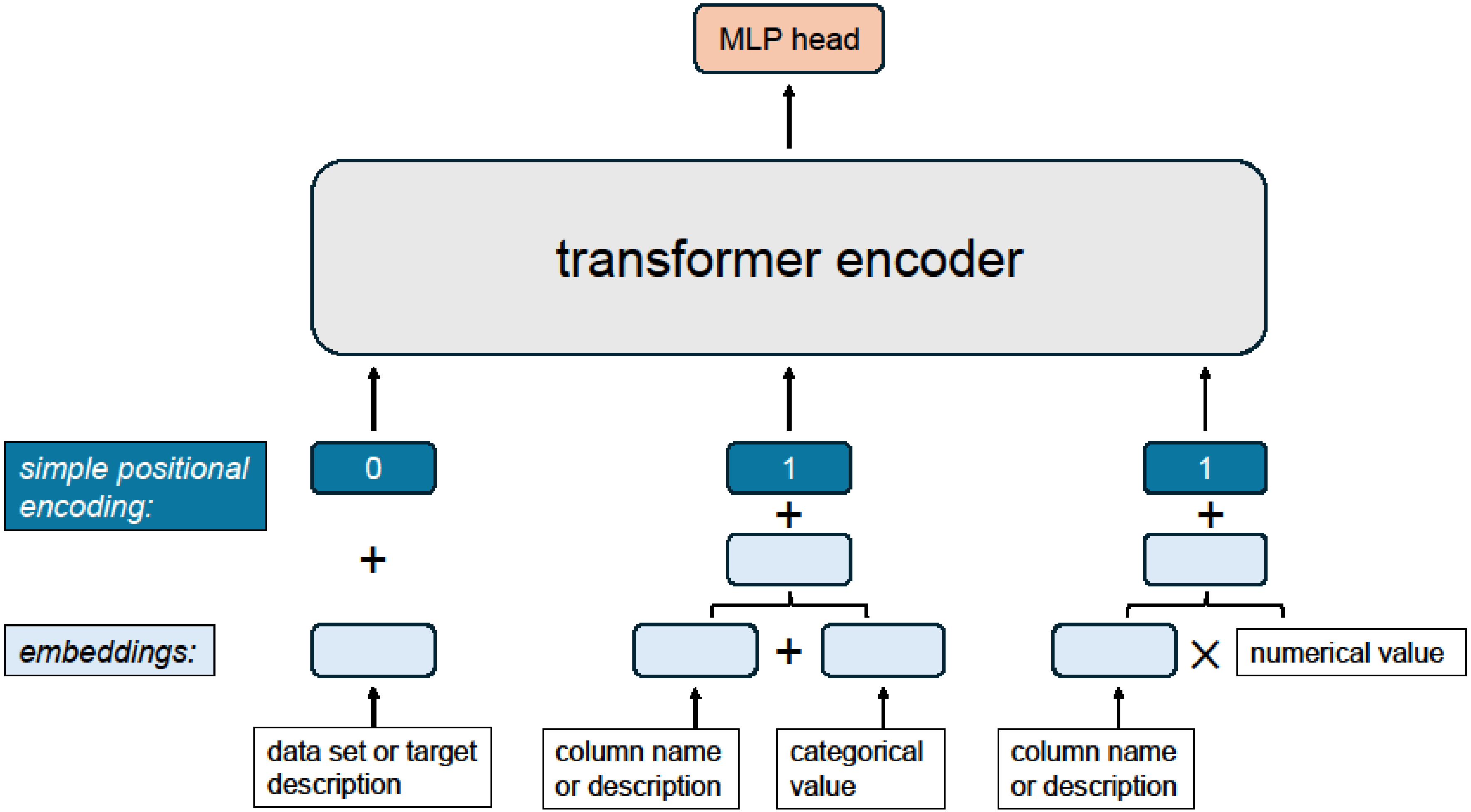
input previously generated column embeddings directly



drop causal masking (permutation invariance)

from wikipedia

Concept Model

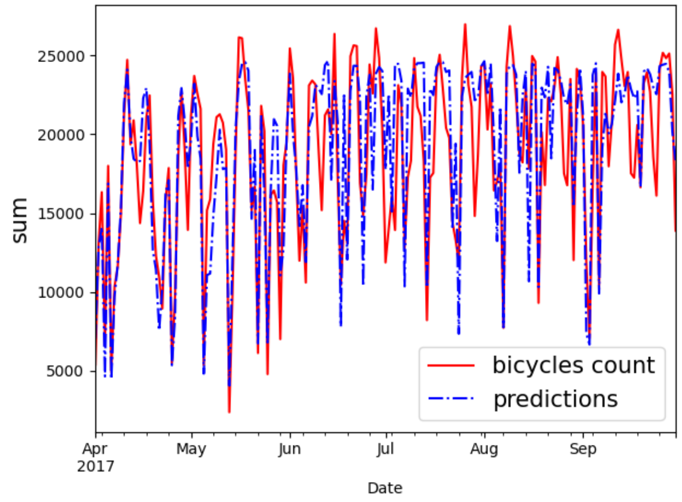


Early Results and Outlook

feasibility checks: method works

- prediction accuracy of individual models competitive to prevalent methods
- can be trained across data sets and tasks without significantly losing accuracy
- pre-train and finetune approach can be applied

	house prices	store sales	spaceship Titanic
TABGPT	0.138	0.450	79.9%
Kaggle leaderboard	0.113	0.379	85.3%



next step: proper pre-training of a foundation model
→ need to add many data sets and tasks (self-supervised with rotating target column)