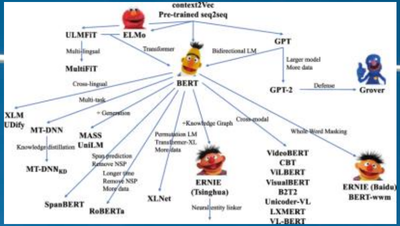
Introduction to the family of BERT transformer models

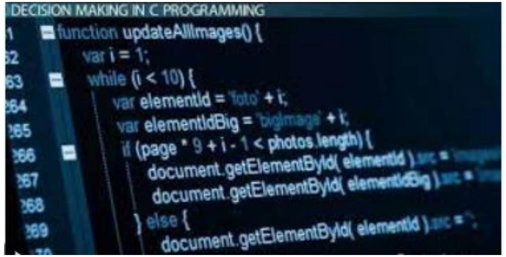


Thierry Desot

- Machine learning – deep learning
- BERT transformer model
- Sentiment analysis – zero shot learning

Machine learning

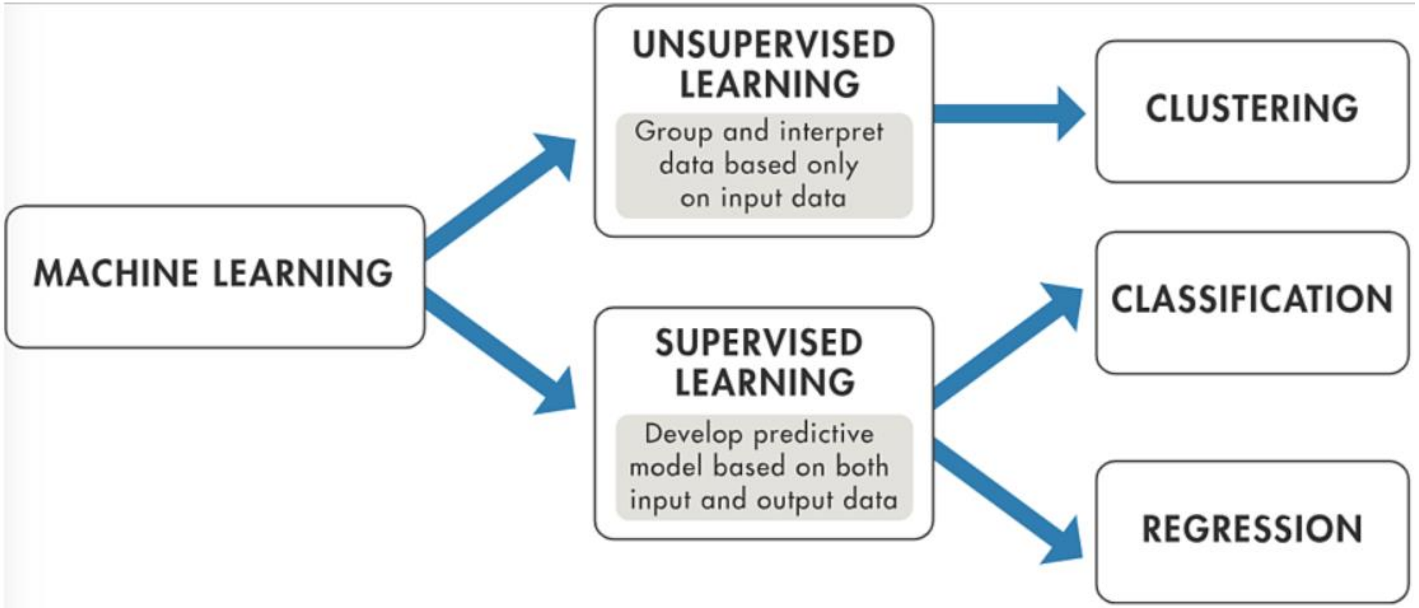
- **Computer software / rules**
  - Developer writes code, rules
  - Instructs system to react to a situation
  - Cannot deal with new situations:
    - Rules or data to be updated
  - Like a language course focusing on learning grammar rules and vocabulary
- **Machine learning**
  - Trained on a large number of data, examples
  - Data-driven
  - Learns based on experience
  - **Infers rules itself : functions**
  - Can deal with new situations
  - Like a language course focusing on practice
  - **For deep learning : open source frameworksTensorFlow, Keras, PyTorch**



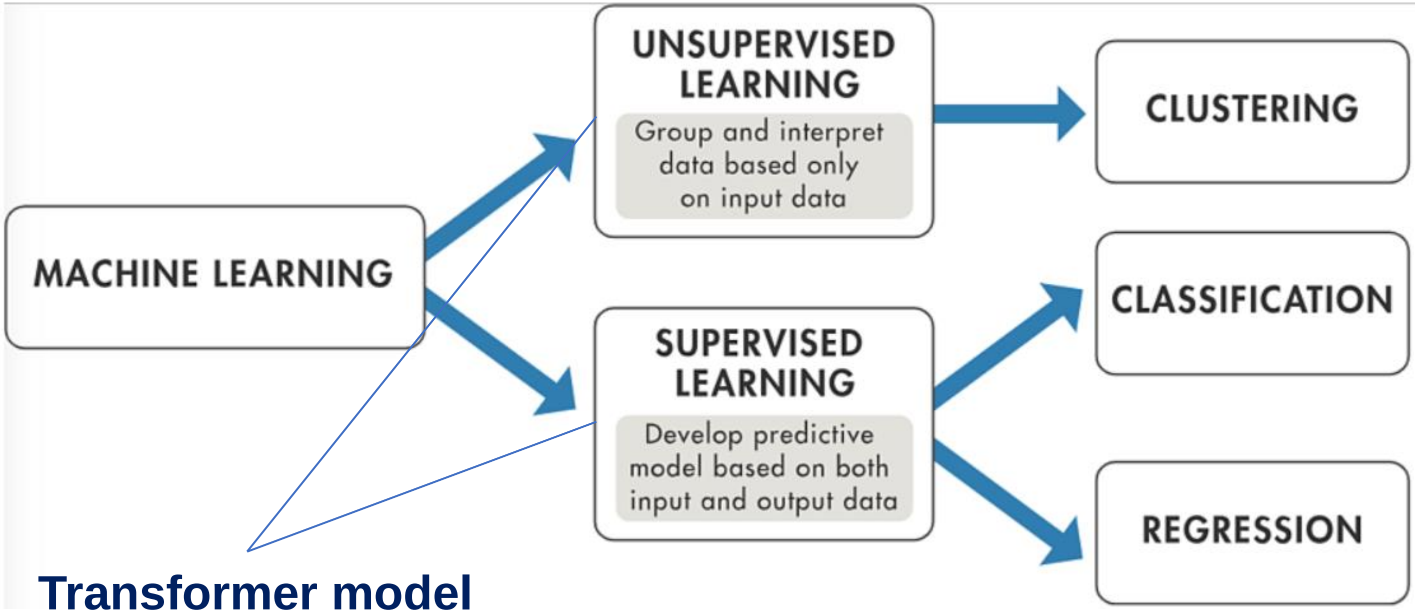
1+1=	2
4:2=	2
6x4=	24
10-3=	7

$3x^2 + 2x + 4$

Machine learning



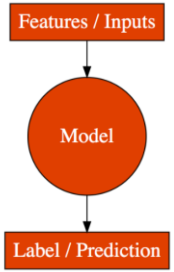
Machine learning



Transformer model
For downstream NLP task : semi-supervised

Machine learning : supervised learning

- **Supervised learning**
  - **Features and label**
    - For example: color, shape feature for image classification
    - **Feature engineering** : manual selection and processing of features
    - Features automatically learnt (**deep learning**)
    - Labeled data
  - **Linear regression**
    - Label is a numeric value
    - Prediction of rent



Input Label

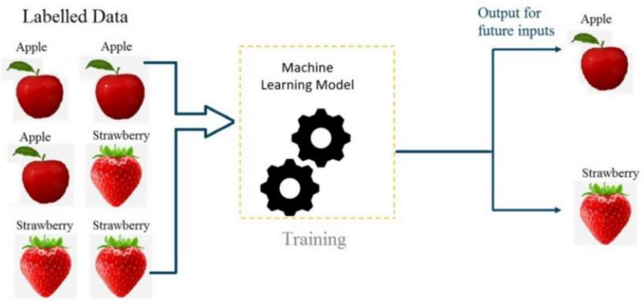
This apartment has a **surface of 200 m^2** and is located in the **city center of Brussels. -> 2000 (Euro rent / month)**

Features

surface : 200 m^2
location : city center

Machine learning : supervised learning

- Supervised learning
  - Classification – logistic regression
    - Category label
    - Image classification
    - Sentiment analysis



Input

Label

I feel **depressed** because the weather is so **bad. -> Negative sentiment**

Adjective features

Machine learning

- **Unsupervised learning**
  - Clustering : group similar entities
  - Cluster data into groups
  - Discover unknown patterns

  in unlabeled data sets

Machine learning

- **Unsupervised learning**
  - Clustering : group similar entities
  - Cluster data into groups
  - Discover unknown patterns

    in unlabeled data sets

- **Reinforcement learning**
  - Rewarding desired behaviors and/or punishing undesired ones
  - A reinforcement learning agent perceives and interprets its environment
  - Takes actions and learns through trial and error

Machine learning - deep learning

- **Data set**
  - Training set — to create the machine learning model
  - Validation set — to fine-tune model and improve performances
  - Held-out test set — informs us about the final performance of the model after completing the training phase

Data set: 10K instances

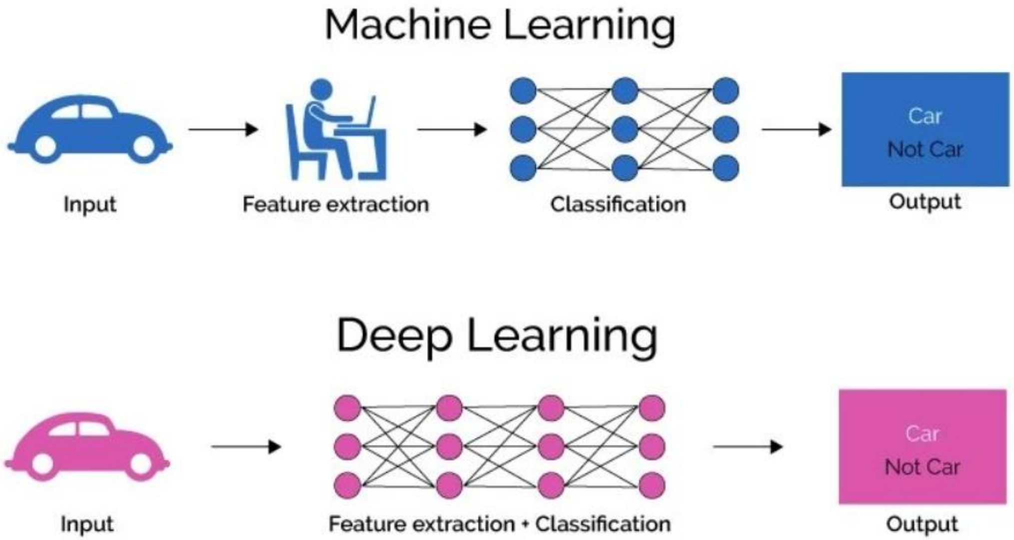
80%	**TRAINING SET**
10%	VALIDATION SET
10%	HELD-OUT TEST SET

Machine learning - deep learning

- Supervised learning : deep learning – deep neural networks
  - Model based on functioning of neurons, neuron layers in the brain

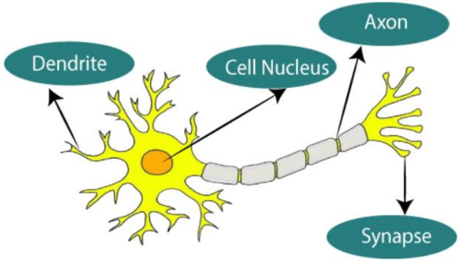
Feature engineering	Deep learning
Manual feature selection and engineering	**Infers features itself**
Data set of 10K instances	Data set of 100K instances Larger data sets needed
CPU sufficient	GPU required
< 1 hour of training time	Hours, days of training time

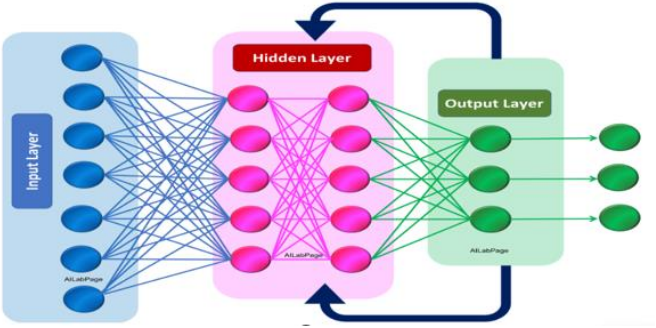
Machine learning - deep learning

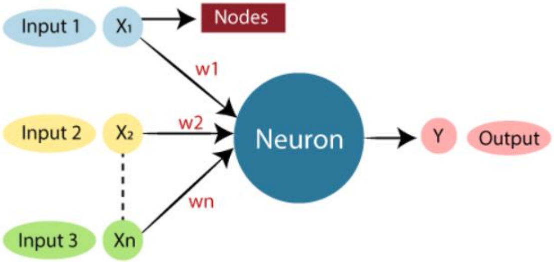


Machine learning - deep learning

Supervised learning : deep learning – deep neural networks







Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

Machine learning - deep learning

- Supervised learning : deep learning – deep neural networks
  - Applied to sentiment analysis / classification
    - 3 possible labels
      - ✓ Negative
      - ✓ Positive
      - ✓ Neutral

  - The weather is rainy, which depresses me -> Negative
  - I feel great today -> Positive
  - It is 10 degrees outside -> Neutral

Machine learning - deep learning

Applied to sentiment analysis

The

weather

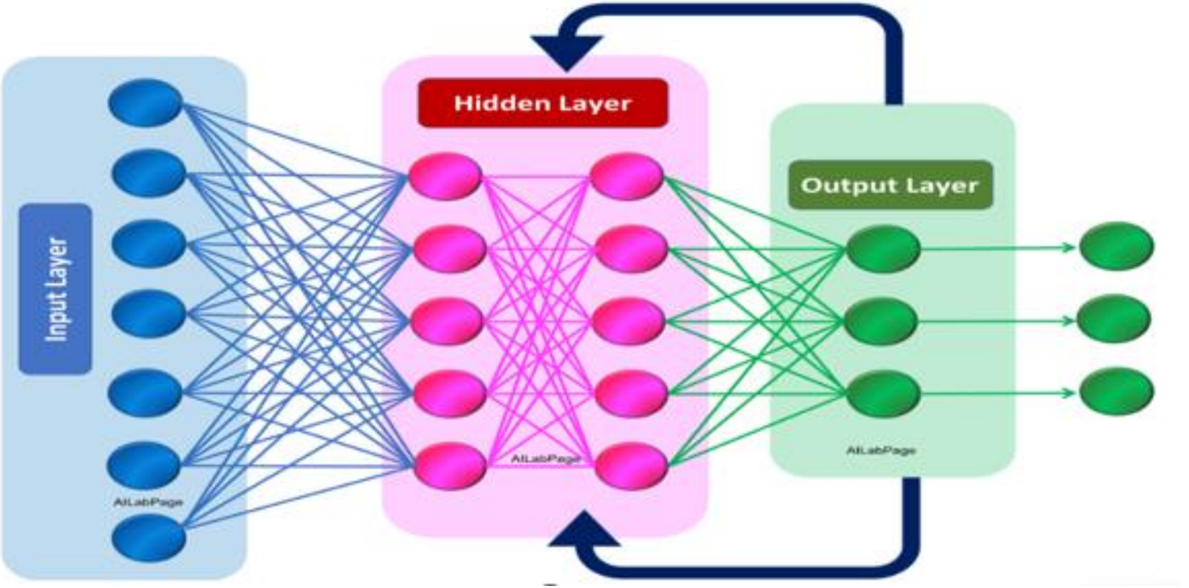
is

rainy,

which

depresses

me



encoder

Machine learning - deep learning

Applied to sentiment analysis

The

weather

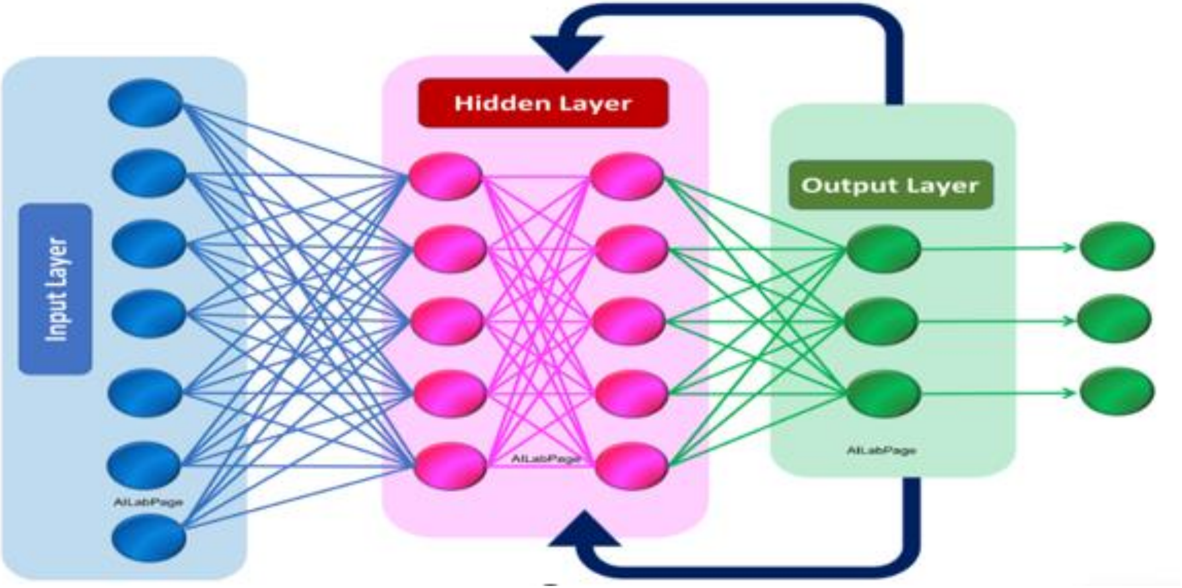
is

rainy,

which

depresses

me



encoder

Machine learning - deep learning

Applied to sentiment analysis

The

weather

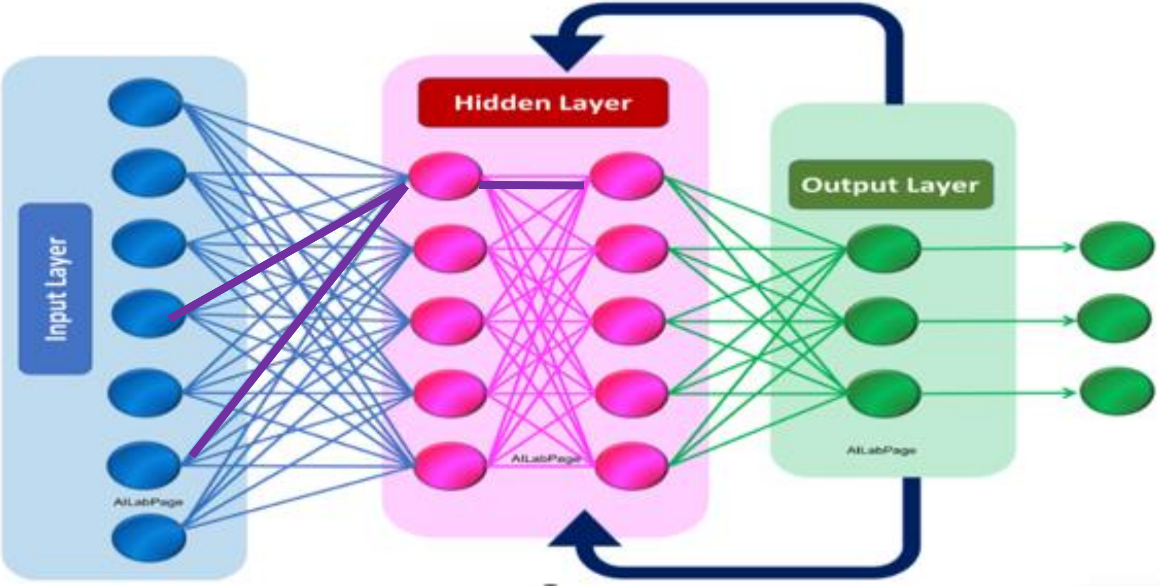
is

rainy,

which

depresses

me



encoder

Machine learning - deep learning

Applied to sentiment analysis

The

weather

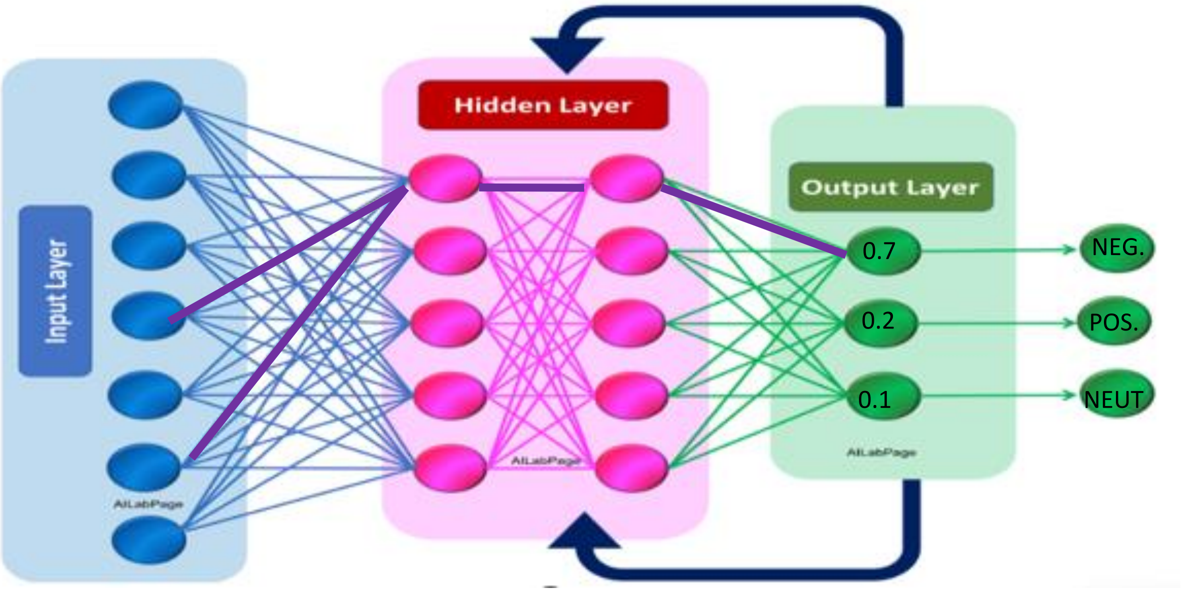
is

rainy,

which

depresses

me



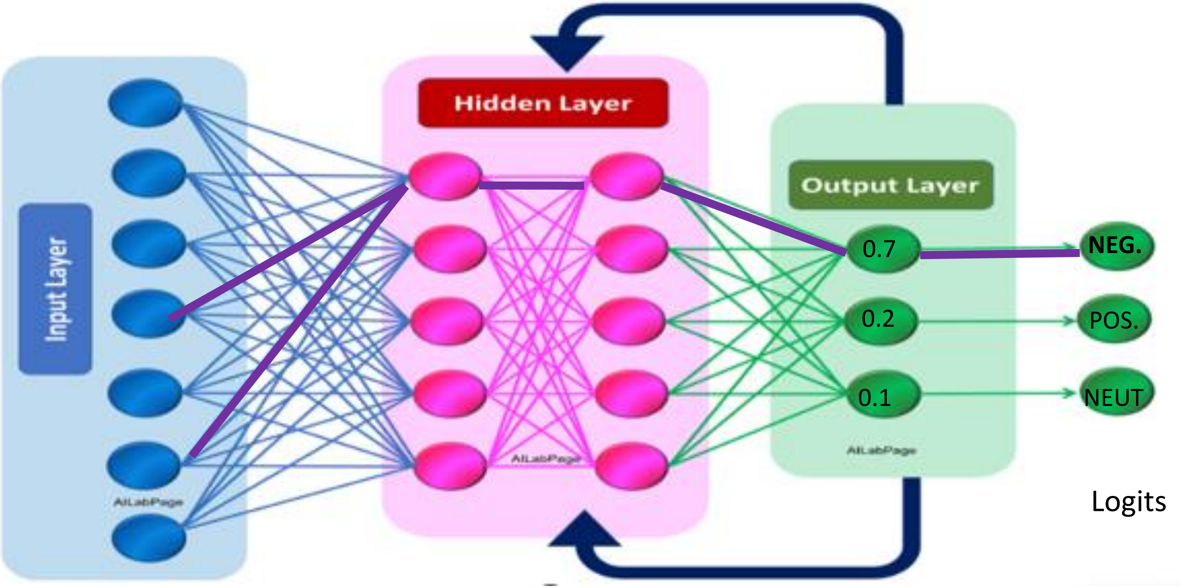
encoder

(Softmax)
probabilities

Machine learning - deep learning

Applied to sentiment analysis

The
weather
is
rainy,
which
depresses
me



Negative

Logits

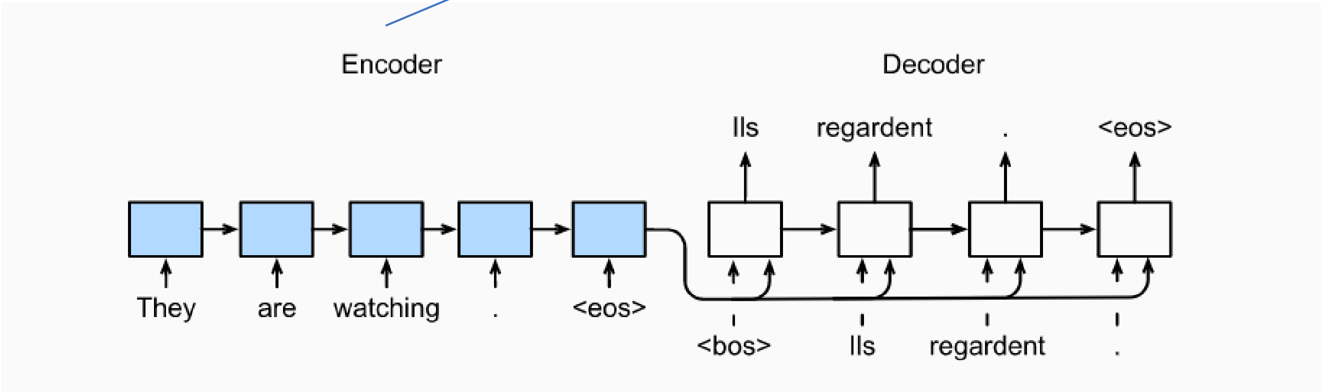
encoder (blackbox)

(Softmax)
probabilities

Deep learning : encoder - decoder

- Translation English-French : sequence-to-sequence
- Decoder predicts each successive target token given input sequence, and 1 **preceding** output token

Hidden layers



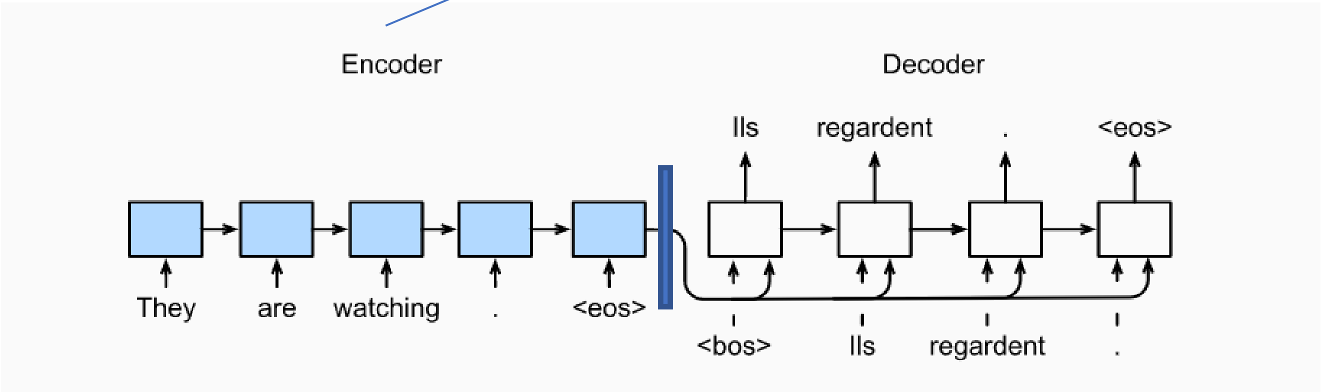
Encoder:
Word -> number

Decoder:
Number -> word

Deep learning : encoder - decoder

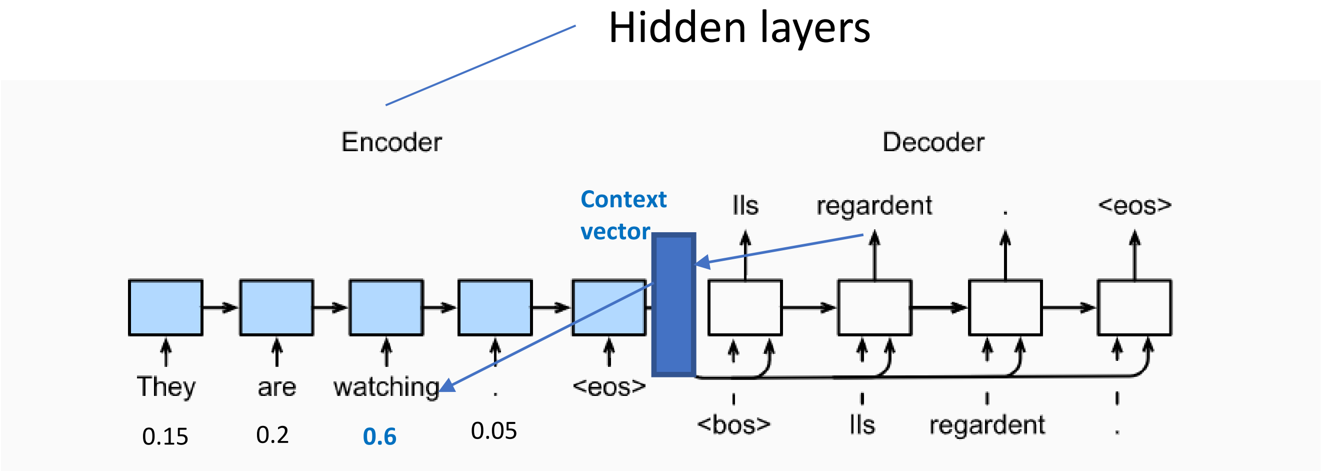
- Performing well for short sentences
- Issues with longer sentences : much information in last hidden state that has to ‘remember’ the whole input

Hidden layers



Deep learning : encoder - decoder

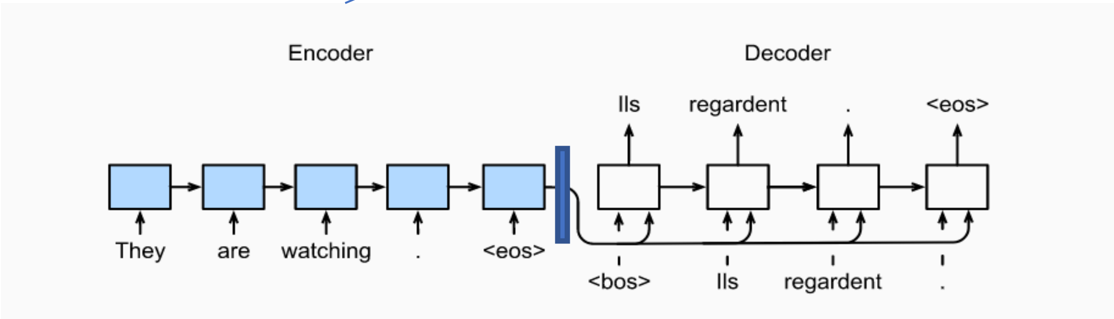
- Context vector / attention mechanism indicates decoder at each step, to which part of the input sequence to pay most attention to (probability values)



Deep learning : encoder - decoder

- Issues with longer sentences : much information in last hidden state that has to ‘remember’ the whole input
- Is this how we translate text as humans? Now, we rather focus on a few words at a time.

Hidden layers



BERT transformer : model definition

- BERT **LM for context modeling**
  - Transformer with a ***fully*** (self-)attention-based approach **[Vaswani et al., 2017]**,
    - Learns **long-range** dependencies in a sequence
    - **Self-attention** or intra-attention relates different positions in a sequence
    - Similar to seq2seq model : **encoder** – decoder

BERT transformer : model definition

- BERT **LM for context modeling**
  - Transformer with a ***fully*** (self-)attention-based approach **[Vaswani et al., 2017]**,
    - Learns **long-range** dependencies in a sequence
    - **Self-attention** or intra-attention relates different positions in a sequence
    - Similar to seq2seq model : **encoder** – decoder

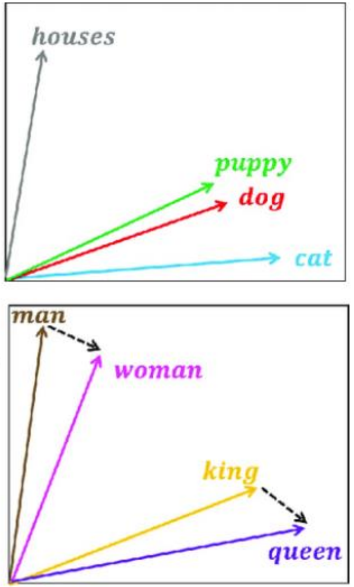
  - **B**idirectional **E**ncoder **R**epresentations from **T**ransformers **[Devlin et al., 2018]**
    - Only **encoder**
    - **Unsupservised** pre-training
    - **Supervised** fine-tuning for specific NLP tasks
    - **Deep learning :** Hidden feature representations learned from data

      word embeddings

BERT transformer : word embeddings

- Word embeddings contain meaning : represented by vectors of numbers

	d1	d2	d3	d4	d5	d6	d7
dog →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
puppy →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
cat →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9



Word

Word embedding

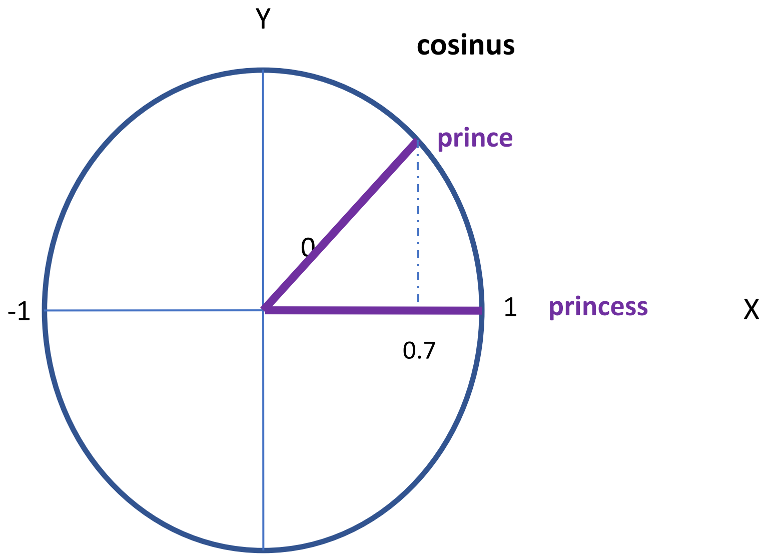
Dimensionality reduction

Visualization of word embeddings in 2D

Semantically similar words,
Closer in vector space:
-puppy - dog
-king - queen
-man - woman

BERT transformer : word embeddings

- Distance measured with cosine similarity



BERT transformer : self-attention

- **Self-attention**
    - You have to spot e-mail addresses in a document as fast as possible
        - What do you attend to?

BERT transformer : self-attention

- **Self-attention**
  - You have to spot e-mail addresses in a document as fast as possible
    - What do you attend to? **@**
  - While reading a book you encounter a sentence with missing words:
    - The cat ….

BERT transformer : self-attention

- **Self-attention**
  - You have to spot e-mail addresses in a document as fast as possible
    - What do you attend to? **@**
  - While reading a book you encounter a sentence with missing words:
    - The cat ....

What type of word do you expect?

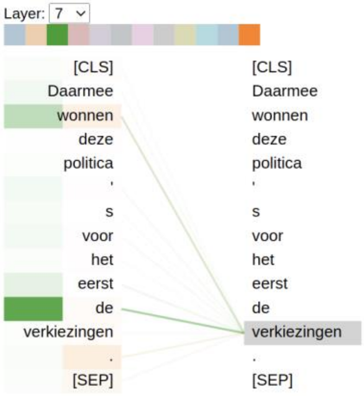
- The cat eats

The cat sleeps

The cat on the mat sleeps

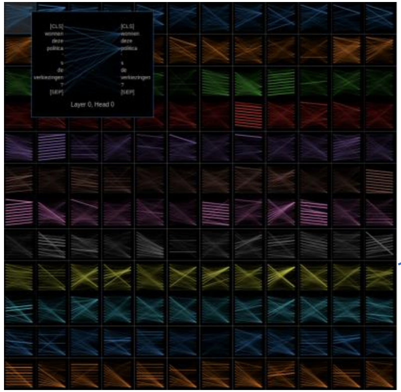
BERT transformer : self-attention

- Self-attention (example Dutch BERT model 'BERTje')
  - Allows each word (query) : 'verkiezingen'
  - to attend to other relevant words of the same sequence (keys) : 'wonnen, de'
  - This allows the model to learn the context of a word based on its surroundings (left and right) -> **bidirectional**
- Transformers are big and slow
  - However, computations are done in parallel
    - Which makes it faster



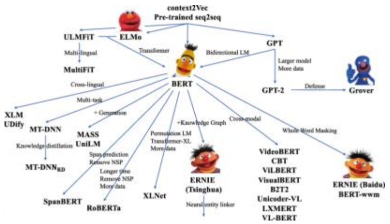
BERT transformer : self-attention

- **BERT is a language model : language understanding**
  - Bert Removes the decoder
  - Multi-layers of the **encoder**
  - Typically 12 layers x 12 heads
    - Each cell represents a linguistic process
      - For instance coreference
      - Dutch BERT model (BERTje)



BERT transformer : general and domain specific language model

- The LLM family:
  - **General** : BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., (2018)], RoBERTa [Liu et al., (2019)]



  - **Multi-lingual vs. mono-lingual** (possibly underresourced language)
    - Dutch BERTje [De Vries et al., (2019)], RobBERT [Delobelle et al
    - Multi-lingual : mBERT
    - French : CamemBERT

  - **Domain specific medical – scientific**:
    - BioBERT [Lee et al., (2020)] , SciBERT [*Beltagy et al., (2019)*]

BERT transformer : language model

- NLP
  - Neural machine translation
  - Question answering
  - Sentiment analysis
  - Event extraction
  - Text generation

Needs language understanding

BERT transformer : language model

- NLP
  - Neural machine translation
  - Question answering
  - Sentiment analysis
  - Event extraction
  - Text generation

Needs language understanding

- **BERT pre-training + fine-tuning**
  - **Pre-train** a BERT model to understand language : **unsupervised learning**
    - Typically trained on millions of sentences
    - Produces word embeddings

BERT transformer : language model

- NLP
  - Neural machine translation
  - Question answering
  - Sentiment analysis
  - Event extraction
  - Text generation -> ChatGPT

Needs language understanding

- **BERT pre-training + fine-tuning**
  - **Pre-train** a BERT model to understand language : **unsupervised learning**
    - Typically trained on millions of sentences
    - Produces word embeddings
  - **Fine-tune** BERT for a specific NLP task : **supervised learning**
    - Transfer learning training with smaller data set for target NLP task
    - Using pre-trained word embeddings

BERT transformer : language model

ChatGPT

- NLP application : text generation
- Under the hood : transformer model
- Encoder – decoder structure
  - Encoder processes the input – user query
  - Decoder generates the output – response
- Self-attention
- Allows the model to capture contextual relationships and dependencies between words more effectively, significantly enhancing its ability to generate coherent and contextually relevant responses
- **Possible to fine-tune**
- Why exploring other models than ChatGPT

BERT transformer : language model

- **Degrees of openness in language models**
- [Liesenfeld et al., (2023)] evaluating LLMs in 15 text generation systems on accessibility of:
  - Data on which they are pre-trained, fine-tuned
  - Documentation
  - Code, architecture
- ChatGPT model
  - In the cloud
- Open source models (Hugging face)

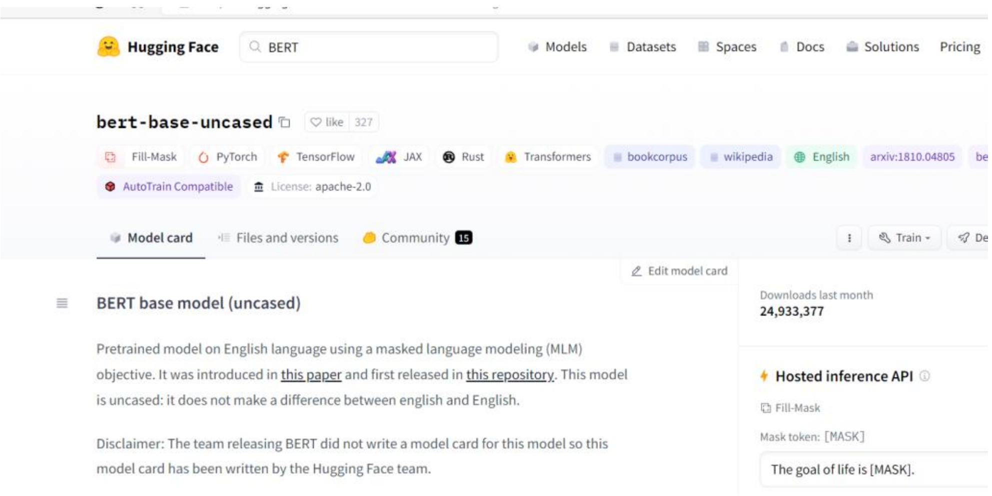
Project	Availability						Documentation					Access methods	
(maker, bases, URL)	Open code	LLM data	LLM weights	RLHF data	RLHF weights	License	Code	Architecture	Preprint	Paper	Data sheet	Package	API
chatGPT	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	~
OpenAI	LLM base: GPT3.5, GPT4			RLHF base: Instruct-GPT									https://chat.openai.com

Degrees of openness colour coded

- ✗
- ~
- ✓

BERT transformer : huggingface

- Pretrained models:
- https://huggingface.co



BERT transformer : pre-training and fine-tuning

BERT pre-train

unsupervised learning

Transfer learning

BERT fine-une

Supervised learning

For example sentiment analysis

Number of *deaths* for leading causes of *death*: Heart *disease*: 696,962; *Cancer*: 602,350; COVID-19: 350,831…

Sentiment label :

Negative

Billions of tokens

Thousands of tokens

Unfiltered data

BERT transformer : pre-training

- **Pre-training: language and context :**

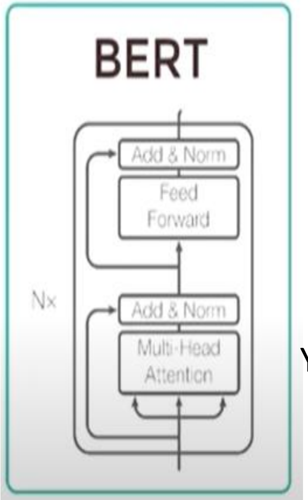
Unsupervised training

Simul-
taneous

- Masked language Model (MLM)

-> context within a sentence

The [MASK 1] brown
Fox [MASK 2] over
the lazy dog.



[MASK 1] = 'quick'
[MASK 2] = 'jumped'

Yes, sentence B. follows sentence A.

- Next sentence prediction (NSP)

-> context across sentences

A. His name is Bert
B. He lives in Sesame street

BERT transformer : pre-training

- Masked language Model (MLM)

  - 15% of words are replaced by a mask before feeding it to BERT
  - The model attempts to predict the masked words

Input: The quick brown fox jumped over the lazy dog.
New input: The [MASK] brown fox [MASK] over the lazy dog.

BERT

Predicted: quick sleeps

BERT transformer : fine tuning

- Fine tuning: downstream specific NLP task
  - Using pre-trained word features with transfer learning
  - For example : **Single sentence classification**

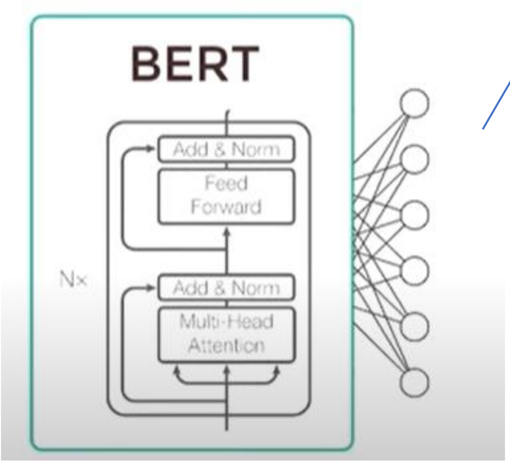
    Supervised training

  -> Fine tune

Fine tuning = fast

Sentence

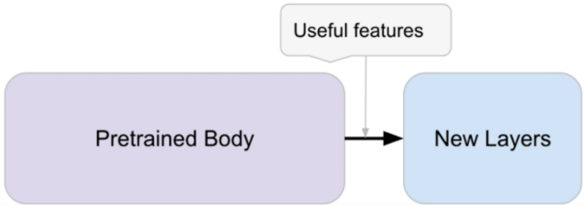
Class label



Predicted Class label

BERT transformer : fine-tuning

- **Pre-trained:**
  - Just use an already trained model for your purpose
  - We use them in 'predict mode'
- **Fine-tuning:**



  - Fine-tune for downstream NLP task -> Transfer learning
    - Add your target data to the model
    - Idea : chop off one (or more) layers, add one (or more) new layers (with randomly initialized weights) for downstream NLP task
    - freeze pre-trained model weights and only train last layer

BERT transformer : tokenization

- Tokenize

  - String split, split words + punctuation, characters, subwords
  - Map tokens to integers
    - E.g. I like cata -> [420, 650, 103]
  - Subword tokenization :
    - Split words into multiple tokens
    - running = run + suffix = runn ing

  - Different transformer models use different tokenization schemes
    - Each model has a model specific tokenizer

BERT transformer : 3 usage levels

- 1 Apply a pre-trained model
  - **Sentiment-analysis**
  - **Zero-shot learning**
  - Text generation
  - Masked language model
  - Named entity recognition
  - Text summary
  - Translation
  - Q&A
- 2 Fine-tune a pre-trained version for your proper use:
  - For instance you want chatgpt being able to answer to specific questions about your own company

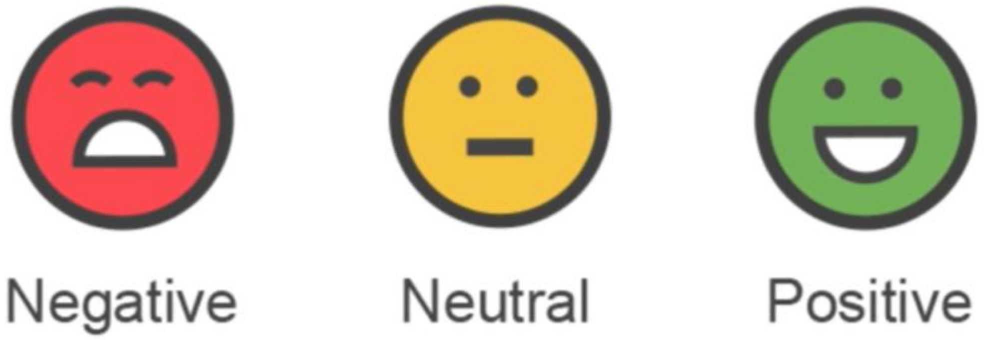
- 3 Pre-train a new model from scratch
  - Lots of computing power, GPUs needed

Sentiment analysis

I really wasted my time watching that movie.

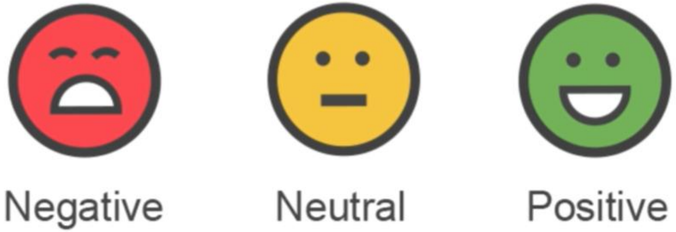
I watched a movie yesterday.

I really enjoyed that movie.



Sentiment analysis

- Binary classification (positive - negative)
- Multi-class (positive – negative - neutral)



Sentiment analysis

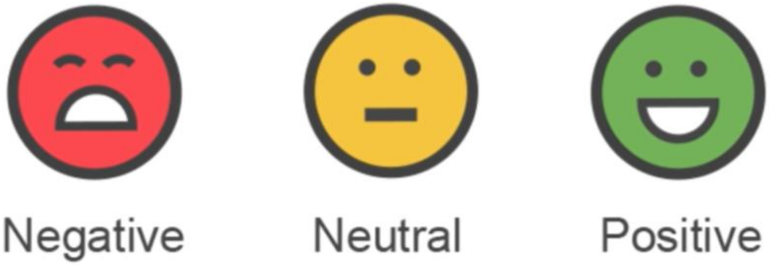
- Binary classification (positive - negative)
- Multi-class (positive – neutral - negative)
- Fine-grain (very- positive – positive – neutral – negative – very negative)
- Depending on the data set and how it is labelled

Sentiment analysis

- **Binary classification (positive - negative)**
- **Multi-class (positive – neutral - negative)**

Pre-trained + fine-tuned model

- Fine-grain (very- positive – positive – neutral – negative – very negative)
- Depending on the data set and how it is labeled



Usefulness of sentiment analysis

- Analyse sentiment in twitter, facebook messages, social media
- For business :
    - predict financial trends, buy or sell actions
    - Analyse your competitors
    - Customer satisfaction analysis
- Reports analysing sentiment over time

Why transformer models for sentiment analysis

- Compared to bag of words (BOW):
  - BOW : ordering and relationships are lost
  - Working with BOW :
    - This was a good movie
    - This was a bad movie
  - More challenging for BOW
    - This was a good movie
    - I cannot say this was a good movie
- Better to use deep learning models, or transformers **for long dependency relations in a sentence**.

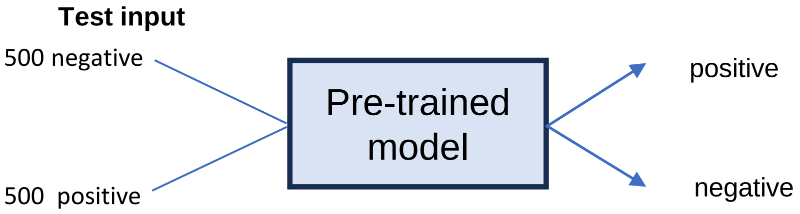
Sentiment analysis assignment

- Sentiment classifier with pre-trained **DistilBERT model : pipeline sentiment-analysis**
- 1500 test instances (from a data set of 14k instances):

500 negative

500 positive

~~500 neutral~~



- Data set preparation/analysis -> prediction -> evaluation
  - Twitter data set .csv file and read into Pandas data frame
  - Read into Pandas data frame

Zero-shot learning

- Normally a model is trained using a labelled data set (supervised learning)
  - For instance binary classification task : Spam or no spam ?
  - Training data have been labelled with classes spam – no spam
  - When applying your trained model, it is the only task it is capable of
- Ideal is a model that is so powerful that it can do *any* classification task
  - That would be a model to which you can ask :
  - is this news paper article about business, culture, health, entertainment & arts etc.?

Zero-shot learning

- Who is Leonardo Da Vinci?
- Classes ['farmer','poet','scientist','painter','politician','architect','pope','notary']

```
classifier("Who is Leonardo Da Vinci?", candidate_labels=['farmer','poet','scientist','painter','politician','architect','pope','notary'])

{'sequence': 'Who is Leonardo Da Vinci?',
 'labels': ['painter',
  'scientist',
  'architect',
  'poet',
  'pope',
  'notary',
  'farmer',
  'politician'],
 'scores': [0.6919945478439331,
  0.18152059614658356,
  0.03554949536919594,
  0.026862822473049164,
  0.019481830298900604,
  0.015883496031165123,
  0.01468390692025423,
  0.014023348689079285]}
```

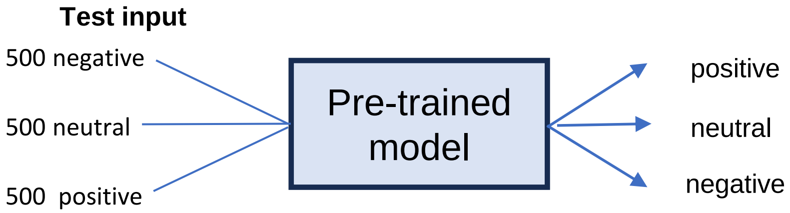
Softmax probabilities sum to 1

Zero-shot learning

- Why is it interesting?
  - We don't have several brains for several tasks
  - We only have one brain to do all tasks
  - You can classify a sentence given a list of categories, without first needing to see many examples of each
  - However, that might not always be the case for totally new categories

Zero shot learning assignment

- Classifier with pre-trained **DistilBERT model : pipeline zero-shot-classification**
- 1500 test instances (from a data set of 14k instances):



- Maybe the model is capable of classifying neutral sentiments, if zero-shot learning can do any classification task?
- Data set preparation/analysis -> prediction -> evaluation
    - Twitter data set .csv file and read into Pandas data frame

Assignments extra

- Cosine similarity
- Read and test small script for cosine similarity
- Adapt 2 scripts based on 1st script about cosine similarity
- Write a variation

References

- Vaswani et al., (2017), *Attention is all you need*
- Devlin et al., (2018), *Bert: Pretraining of deep bidirectional transformers for language understanding*
- Clark et al., (2019), *What does Bert look at? An analysis of Bert's attention*
- De Vries et al., (2019), *Bertje: A Dutch Bert model*
- De Vries et al., (2020), *What's so special about BERT's layers? A closer look at the NLP pipeline in monolingual and multilingual models*
- Rogers et al., (2020), *A Primer in BERTology, What we know about how BERT works?*
- Vig, (2019), *A multiscale visualization of attention in the transformer model*
- Vig et al., (2019), *Analyzing the structure of attention in a transformer language model*
- Liu et al., (2019), *Roberta: A robustly optimized bert pretraining approach*
- Beltagy et al., (2019), *SciBERT: A pretrained language model for scientific text*
- Lee et al., (2020), BioBERT: *a pre-trained biomedical language representation model for biomedical text mining*
- Liesenfeld et al., *(2023), Opening up ChatGPT: Tracking openness, transparency, and accountability in instruction-tuned text generators*

Thanks!