University of Antwerp
| Adrem | Adrem Data Lab

Neural Networks

Toon Calders

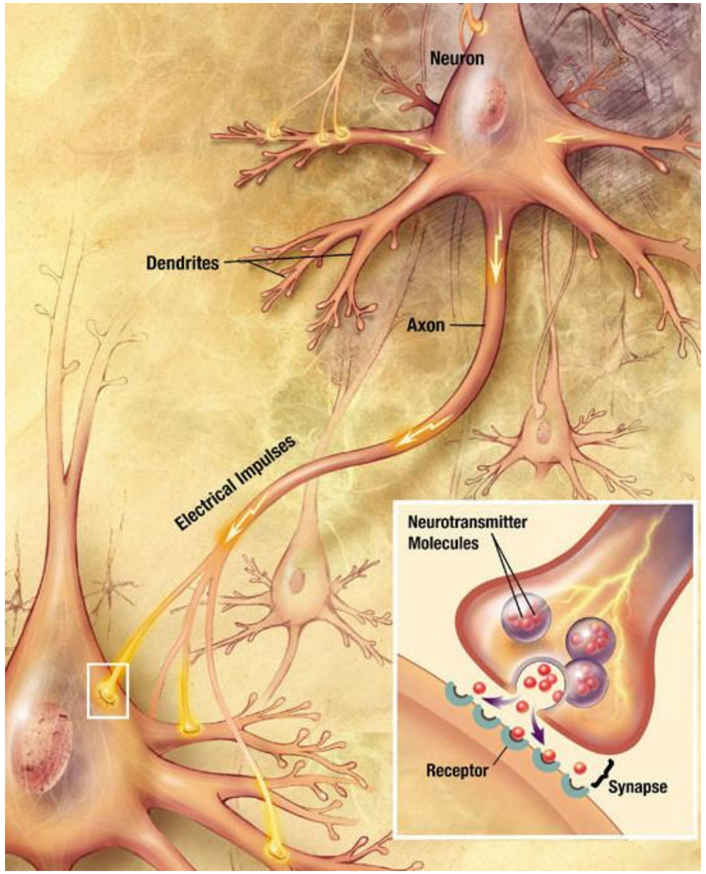
Dept. Computer Science

University of Antwerp

toon.calders@uantwerpen.be

Building Block: Perceptron

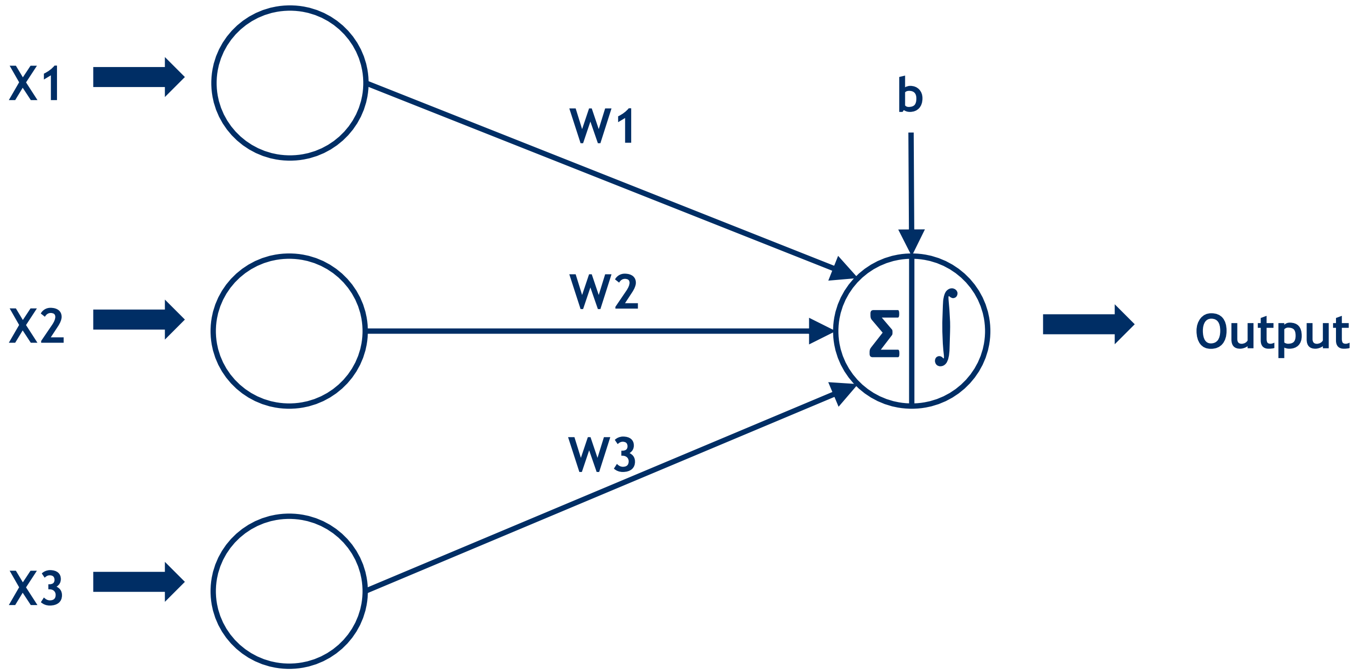
- **A perceptron is a simplified model of a biological neuron.**
  - A neuron receives "input" from different other cells (other neurons or "sensors")
  - If total input exceeds a threshold, the neuron will "spike" (i.e., emit a signal)
- **Human brain consists out of approximately 86 billion neurons**



Picture from Wikipedia (http://en.wikipedia.org/wiki/Chemical_synapse)

Perceptron

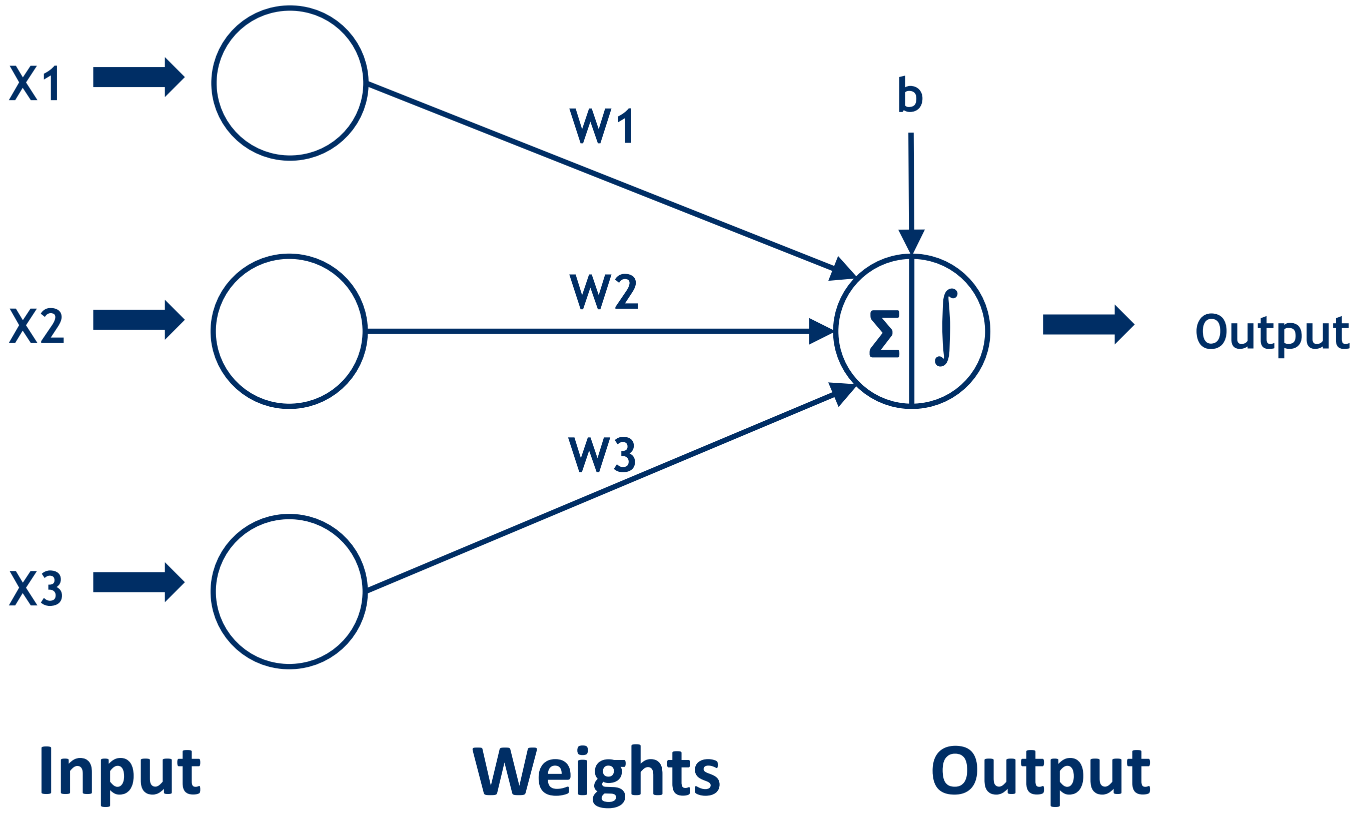
- A perceptron takes inputs and computes a function
- Output depends on *weights* and an *activation function*



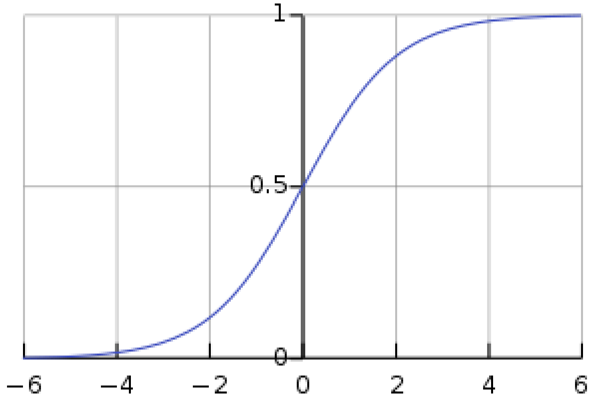
Input **Weights** **Output**

Perceptron

- A perceptron takes inputs and computes a function
- Output depends on *weights* and an *activation function*



Perceptron



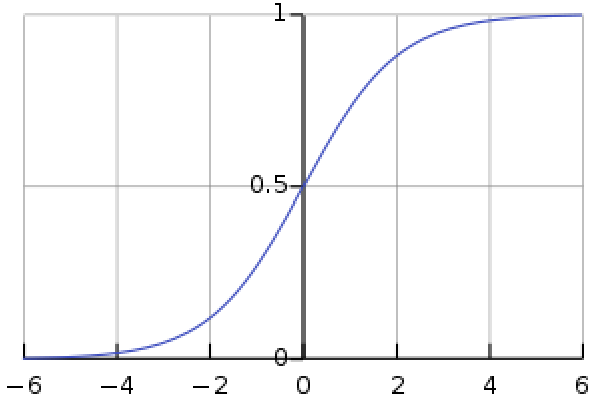
- **A perceptron takes inputs and computes a function**
- **Output depends on *weights* and an *activation function***

- **Example of an activation function:** $$\sigma(x) = \frac{1}{1 + e^{-}x}$$

- **The perceptron on last page computes the function:**

$$\sigma(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

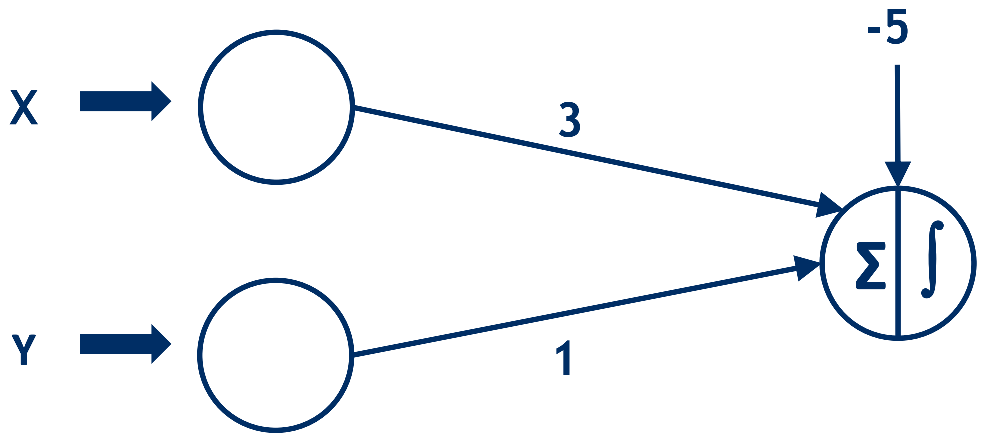
Perceptron



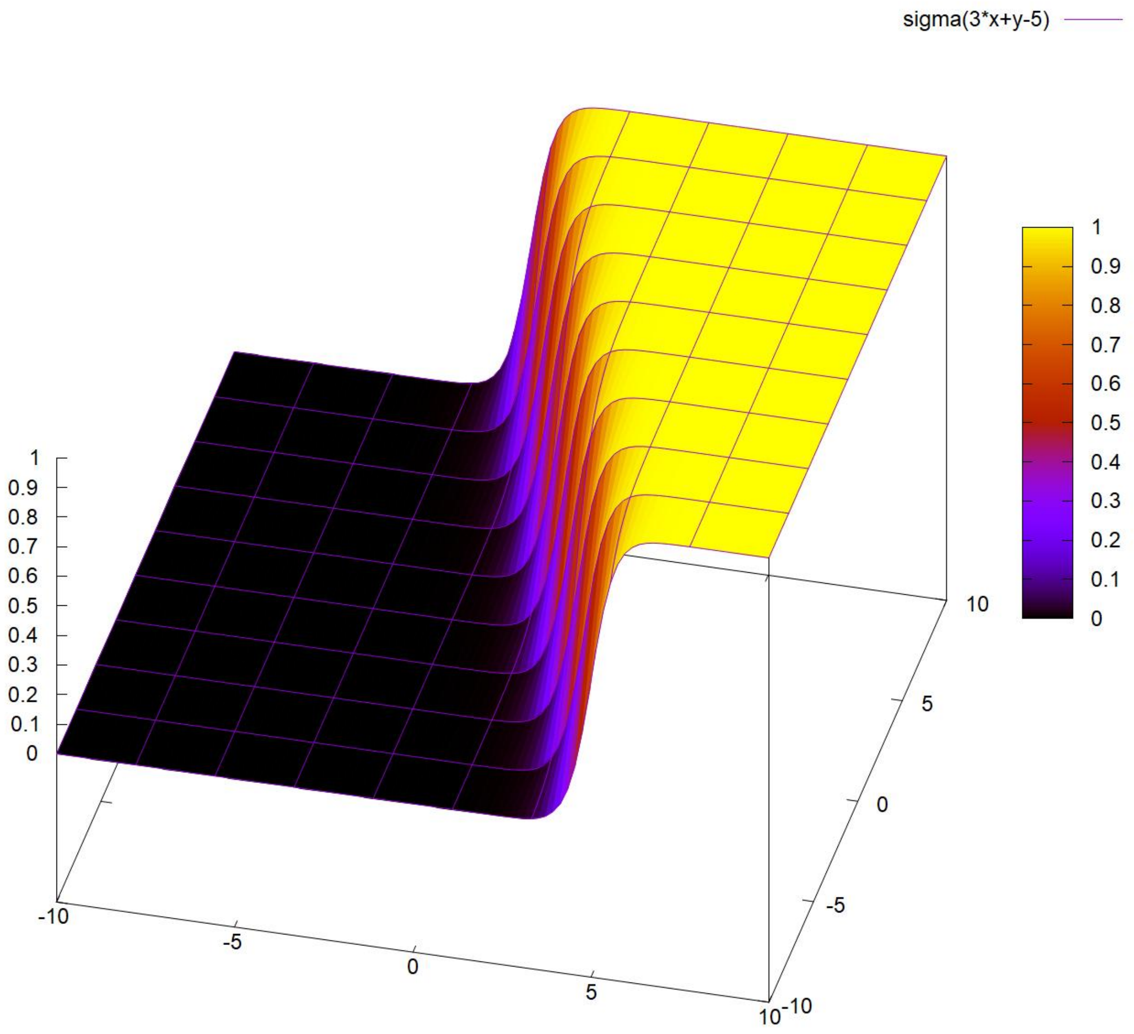
- **A perceptron takes inputs and computes a function**
- **Output depends on *weights* and an *activation function***
- **Example of an activation function:** $\sigma(x) = \dfrac{1}{1+e^{-x}}$
- **The perceptron on last page computes the function:**

$$\sigma(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

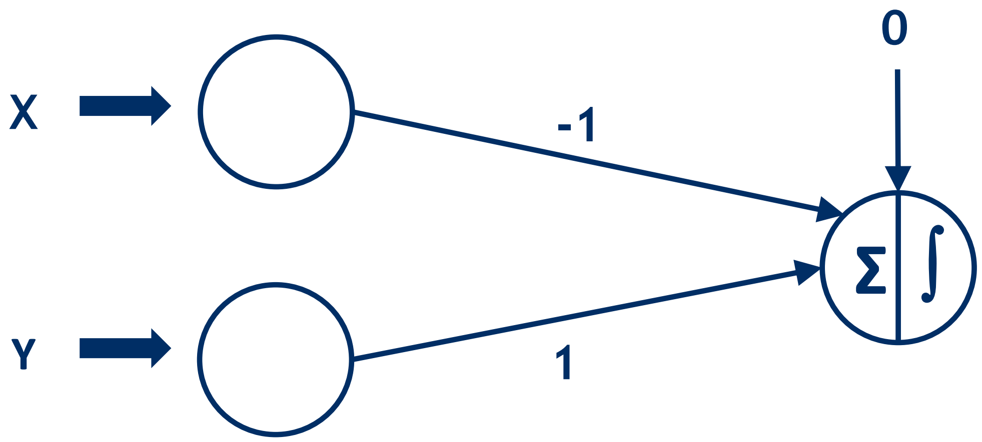
Examples - Perceptrons



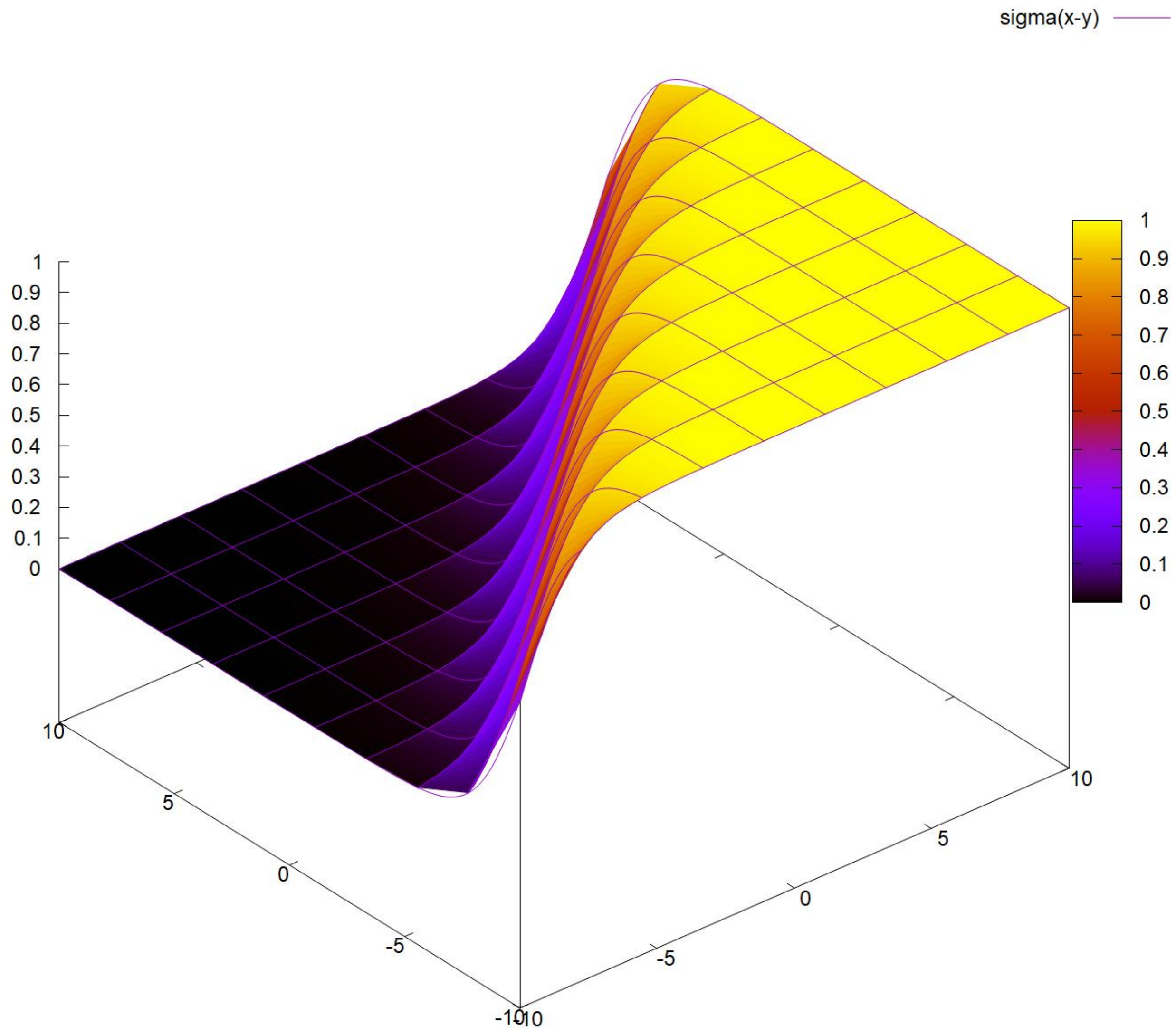
$$\sigma(3x + y - 5)$$



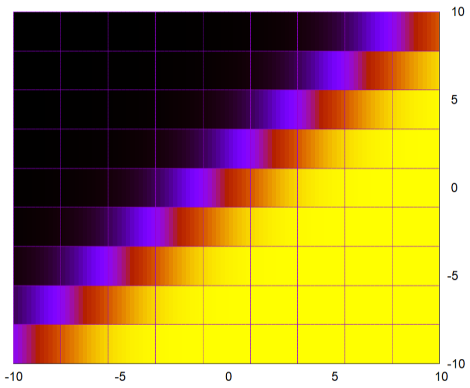
Examples - Perceptrons



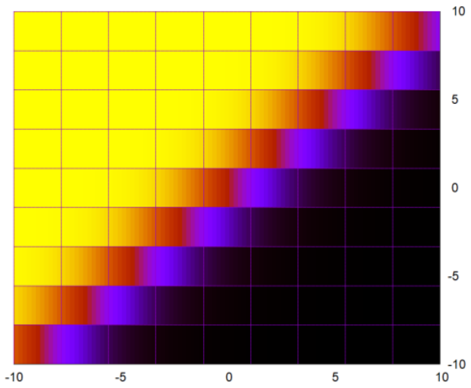
$$\sigma(x - y)$$



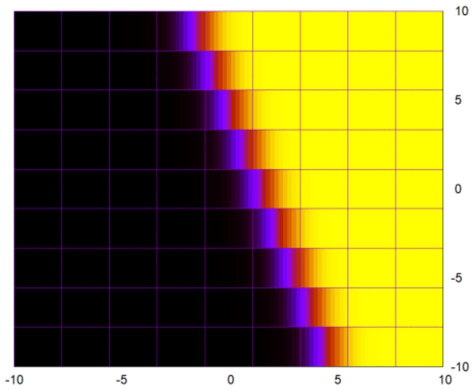
Examples - Perceptrons



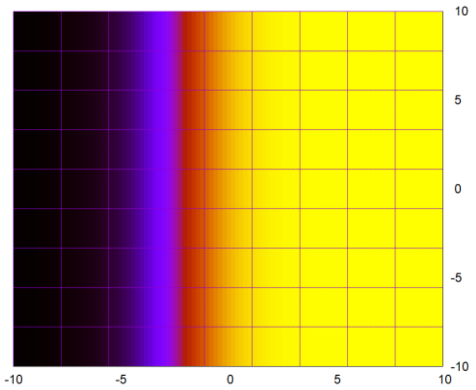
$$\sigma(x-y)$$



$$\sigma(y-x)$$



$$\sigma(3x+y-5)$$



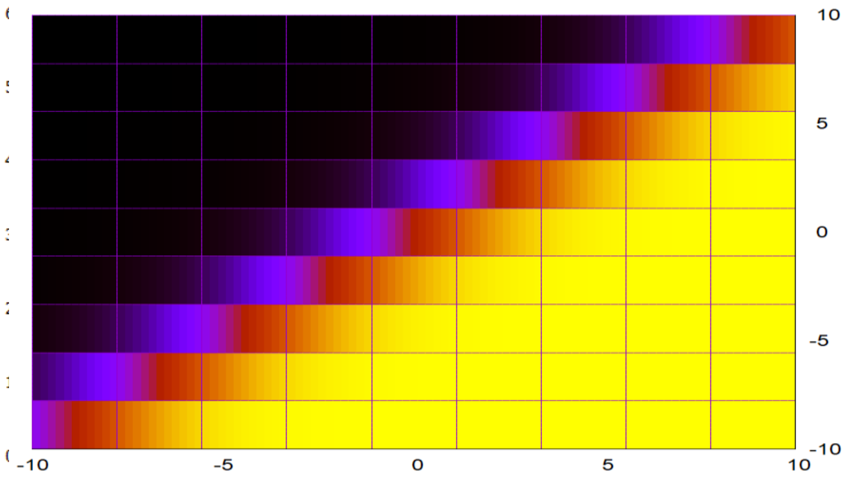
$$\sigma(x-2)$$

- **Each choice of W1, W2, B corresponds to another function, mapping (x,y) to a number**

Perceptron Learning - Algorithm

- **Given a dataset, learn the parameters of a perceptron that produces similar output**

x_1	x_2	y
1	3	0
1	4	0
2	4	0
4	2	1
5	1	1



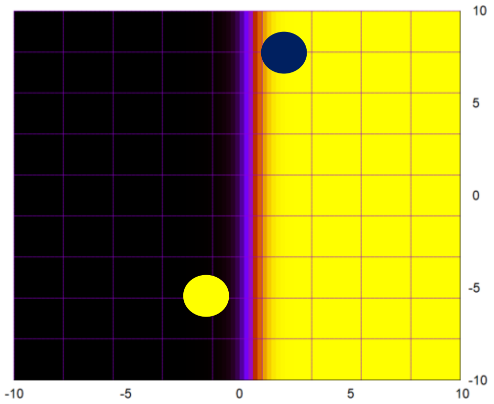
$$\sigma(x - y)$$

Perceptron Learning - Algorithm

- **Given a dataset, learn the parameters of a perceptron that produces similar output**
- **Perceptron learning = learning weights = training the network**
  - Iteratively refine the model until it fits the examples
- **Training proceeds as follows:**
  - Start with random weights
  - Repeat until "good enough":

    For each training example (x,y,label):

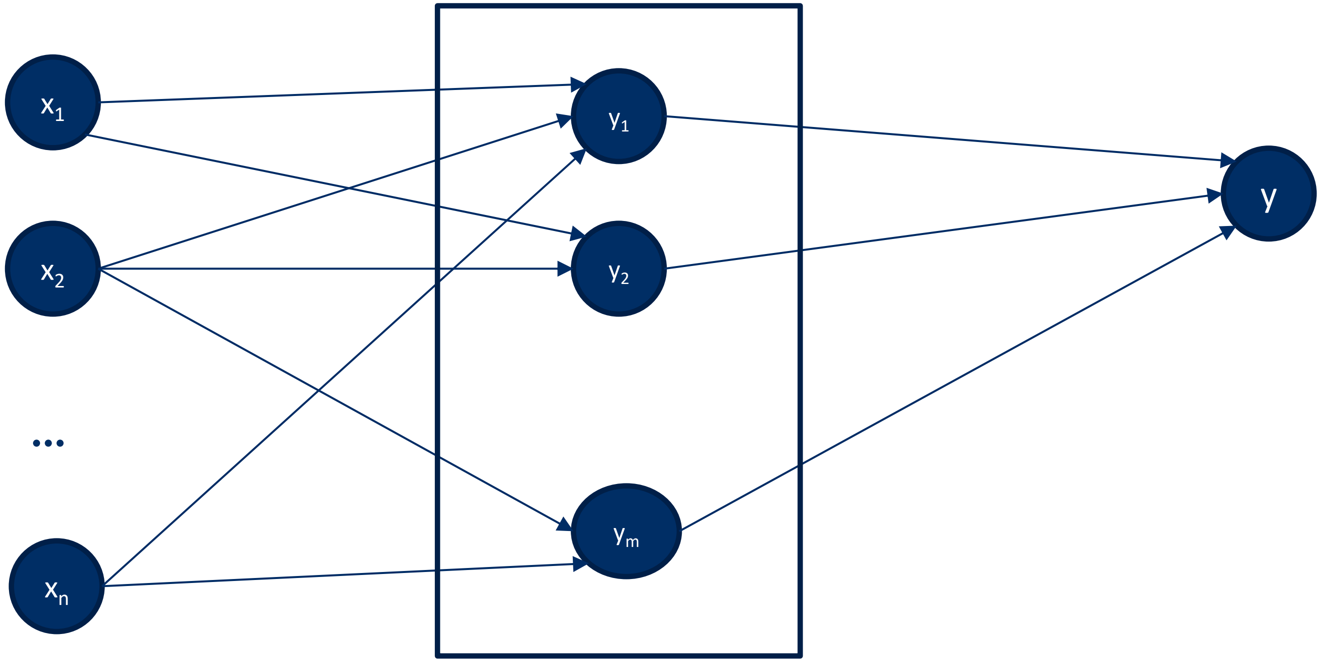
    slightly change the weights to improve prediction



example

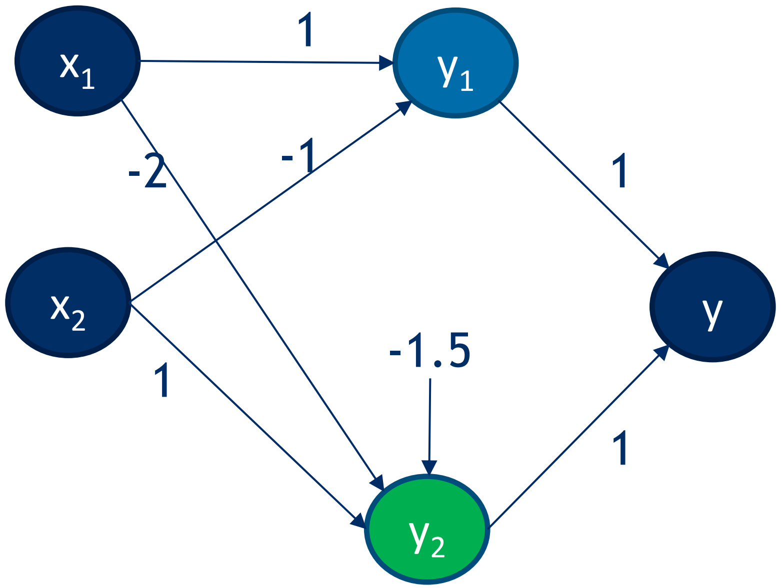
Extending to Multiple Perceptrons

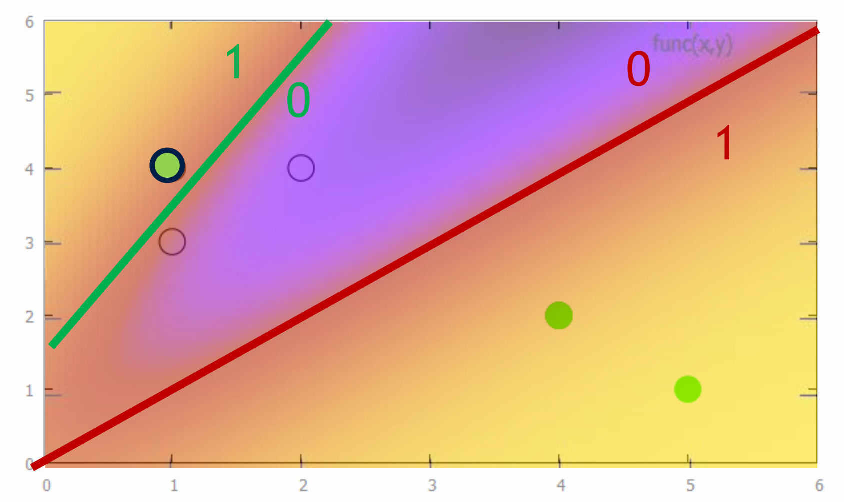
- One perceptron has limited *representational power*
- We can combine multiple perceptrons to create a more complex *neural network* that can express more complex functions



Hidden layer

Example : Multi-Layer Perceptron Network





Training a Neural Network

- **Exactly as for one perceptron:**
- **Neural Network learning = learning weights = training the network**
  - Iteratively refine the model until it fits the examples
- **Training proceeds as follows:**
  - Start with random weights
  - Repeat until "good enough":

    For each training example (x,y,label):

    slightly change the weights to improve prediction

One complete run through the dataset is called an "epoch"

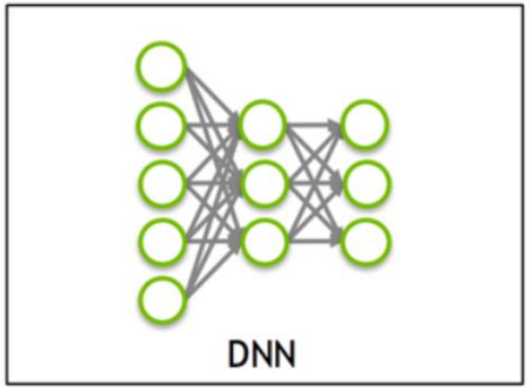
- **Training a NN can be time-consuming**

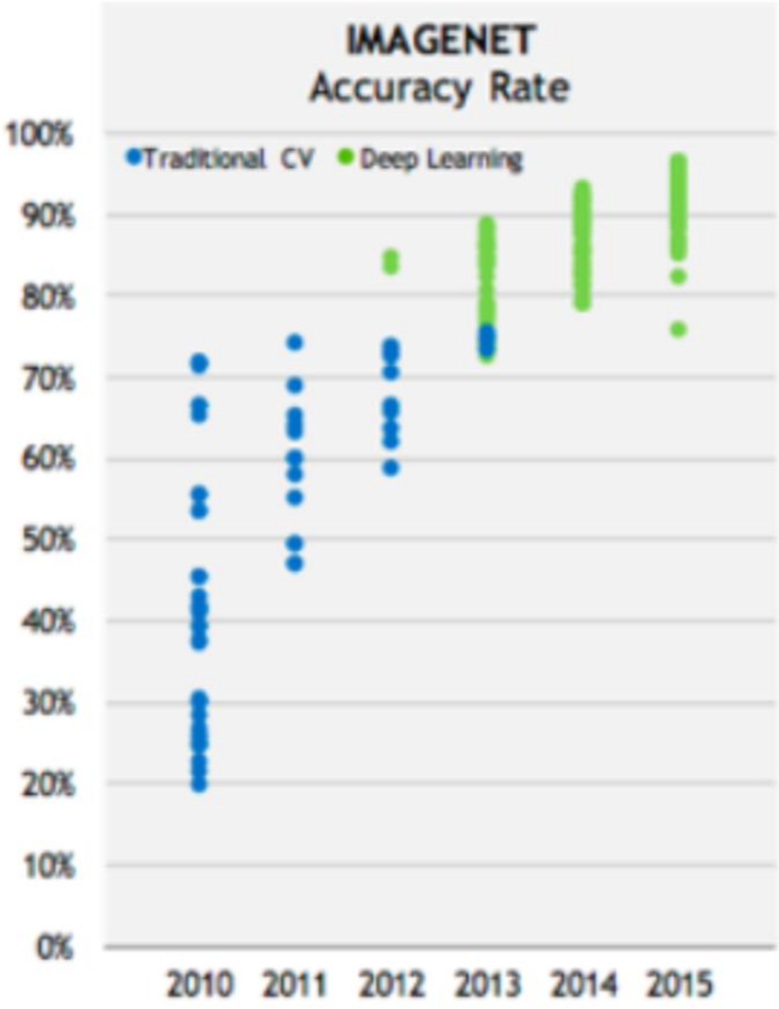
example 1

Deep Learning

- Deep learning = learning of large NNs
- NNs were already studied in the 60s

- What's new?





What is so great about Neural Networks ?

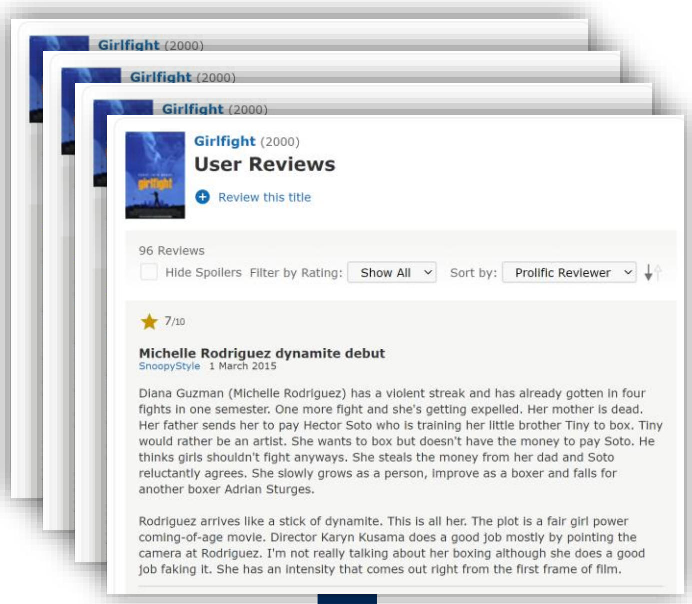
- **Little to no model bias**
  - They can present any function (if sufficiently large)
  - If there exists a mapping from features to labels, a neural network can in principle learn it
- **Built-in feature engineering**
  - No need for inventing complex features to capture meaningful patterns in the data

- **But, comes at a cost!**
  - Lot and lots of data required!
  - Huge computational demands
  - Expensive hardware

Overview

- “Traditional” NLP
- Word embeddings
- Transformers

Textual data


Girlfight (2000)
User Reviews
Review this title

96 Reviews
Hide Spoilers Filter by Rating: Show All Sort by: Prolific Reviewer

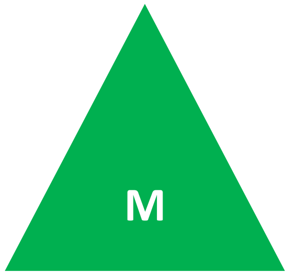
7/10

Michelle Rodriguez dynamite debut
SnoopyStyle 1 March 2015

Diana Guzman (Michelle Rodriguez) has a violent streak and has already gotten in four fights in one semester. One more fight and she's getting expelled. Her mother is dead. Her father sends her to pay Hector Soto who is training her little brother Tiny to box. Tiny would rather be an artist. She wants to box but doesn't have the money to pay Soto. He thinks girls shouldn't fight anyways. She steals the money from her dad and Soto reluctantly agrees. She slowly grows as a person, improve as a boxer and falls for another boxer Adrian Sturges.

Rodriguez arrives like a stick of dynamite. This is all her. The plot is a fair girl power coming-of-age movie. Director Karyn Kusama does a good job mostly by pointing the camera at Rodriguez. I'm not really talking about her boxing although she does a good job faking it. She has an intensity that comes out right from the first frame of film.

X1	X2	X3	X4	…	label
0.3	0.5	1.4	-0.2	…	POS
1.4	0.2	1.7	2.4	…	NEG
…	…	…	…	…	…




An average but enjoyable character piece that is vastly improved by a moody Rodriguez
bob the moo 10 December 2004

Being a regular troublemaker at school, Diana starts to entertain the idea of learning to box properly like her brother is allowed to do. Knowing her father will never let her do it she steals the money from him and starts to train with Hector. Quickly improving in the ring despite the hoots of derision aimed at her from her fellow boxers, Diana finds problems with a lack of female opposition, love in the shape of another up and coming amateur and a conflict on the horizon with her father bound to find out sooner or later.

I'm not sure where I got the idea but for years I had the impression that this was a foreign indie film that had made a big impression and was critically praised. Mostly for these reasons I did really want to see it but never got round to it until it came onto television recently. By this point I had realized that it was an American movie with some indie aspirations but not the grit or adult content I had expected ☐ anyway, this

8 out of 13 found this helpful. Was this review helpful? Sign in to vote.
Permalink

X1	X2	X3	X4	…	label
0.3	0.5	1.4	-0.2	…	???

https://www.imdb.com/title/tt0210075/reviews?ref_=tt_urv

Representation for Text

- **Preprocessing:**

Girlfight follows a project dwelling New York high school girl from a sense of futility …

Punctuation removal, Lower casing, Entity recognition

girlfight follows a project dwelling <u>newyork</u> high school girl from a sense of …

Stopword removal, Tokenization, Counting

girlfight	1
follows	3
project	2
dwelling	1
<u>newyork</u>	2
high	2
school	3
girl	7
sense	2
…	

Representation for Text

- **“Bag-of-words”**

girlfight 1
follows 3
project 2
dwelling 1
newyork 2
high 2
school 3
girl 7
sense 2
…

dwelling	newyork	way	high	good	…	label
1	2	0	2	0	…	neg
0	3	2	7	2	…	pos
					…	neg
…	…	…	…	…	…	…

Overview

- “Traditional” NLP
- Word embeddings
- Transformers

Neural Nets for Natural Language Processing

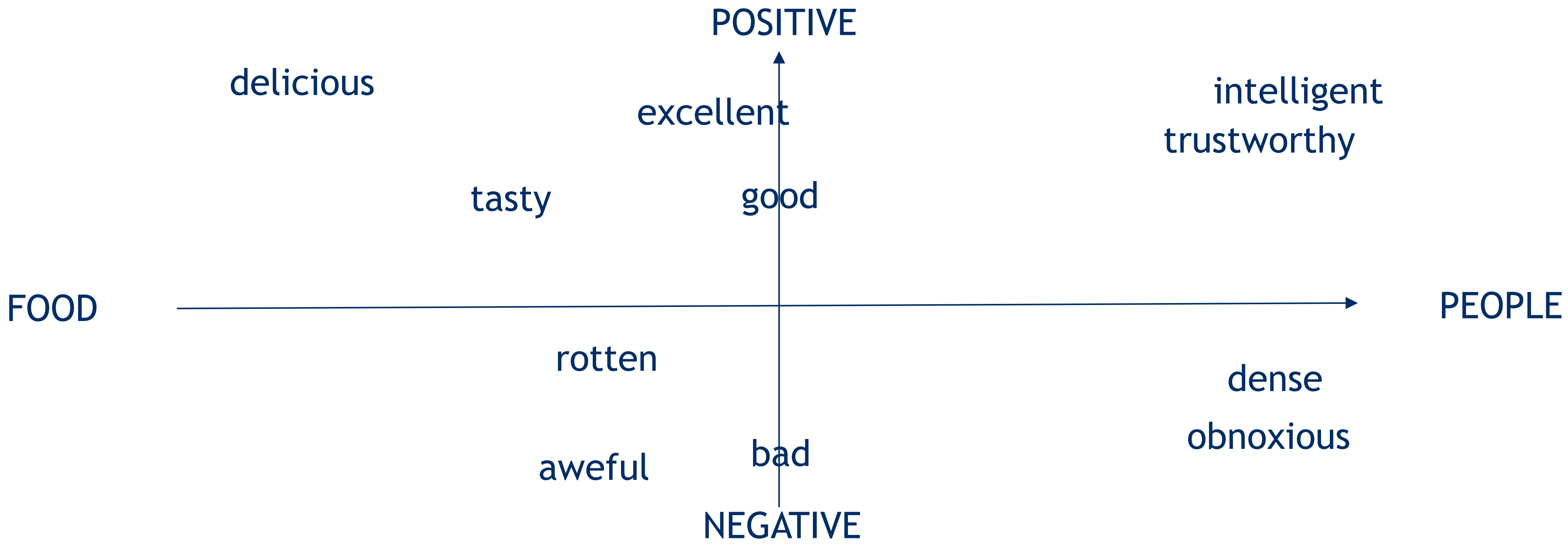
- Word embedding
  = representation of words/texts as a vector of numbers

- Banana → (0.3, 5.8, 7.3, 0.1)
- Father → (0.4, 0.7, 1.2, 0.4)
- Baby → (0.3, 0.6, 1.5, 3.0)
- …

- Why? Hundreds of algorithms work with numbers. Word2Vec is like an "adaptor"

Embeddings are Not Random

- **Words used in similar contexts should have similar vectors**



How are Embeddings Learned from Data ?

- **We turn the problem into a game**

Guess the word !

Florida man charged for assault with a deadly weapon after throwing ? into Wendy's drive thru

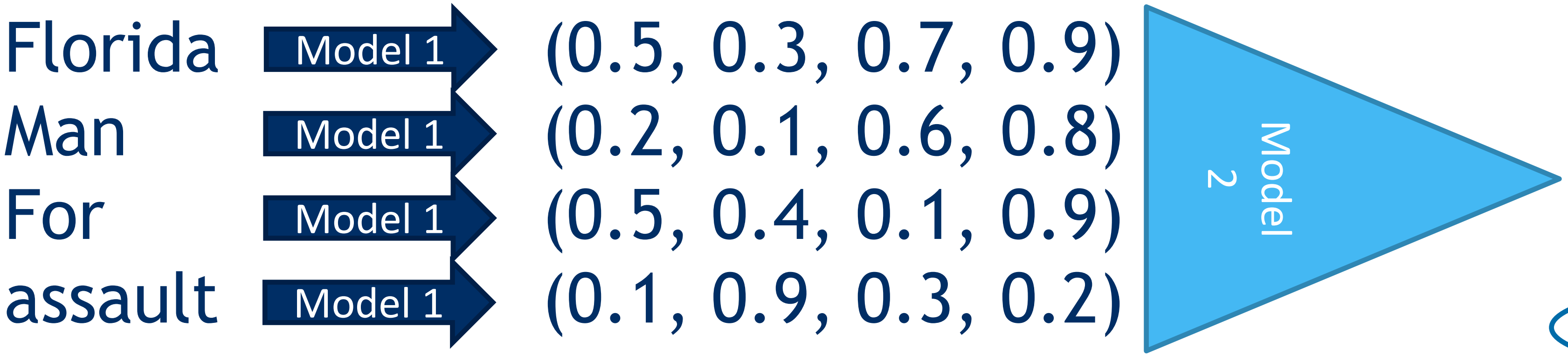
How are Embeddings Learned from Data ?

- **AI will learn a model to predict the word**
  - Easy to generate test data:

Florida man	?	for assault	→	charged
man charged	?	assault with	→	for
charged for	?	with a	→	assault
for assault	?	a deadly	→	with

How are Embeddings Learned from Data ?

- **AI will learn a model to predict the word**
  - The *structure* of the model is fixed:



P	
0.1	sought
0.1	blamed
0	for
0	a
0	an
0.1	convicted
0.7	charged
0	the
0	man
0	men
0	…
0	Florida

  - Becomes an optimization problem

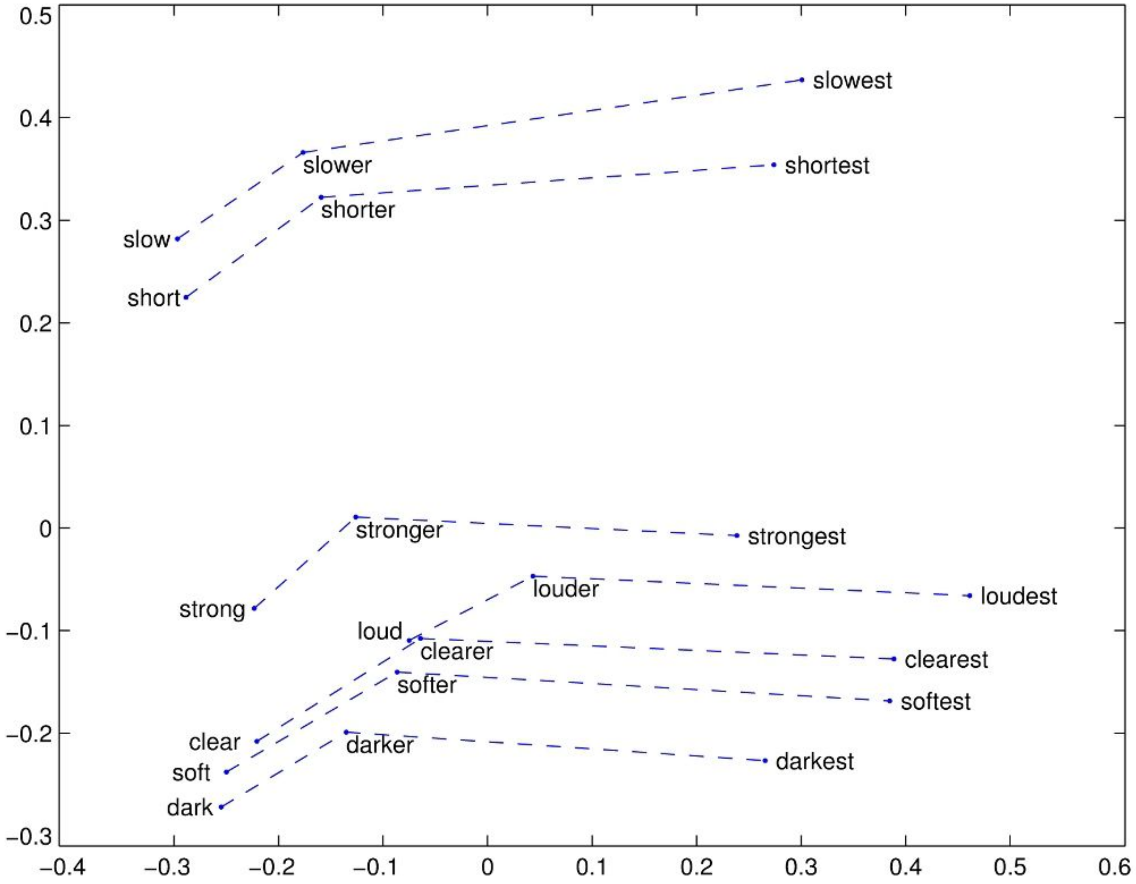
How are Embeddings Learned from Data ?

- **AI will learn a model to predict the word**
  - Easy to generate test data
  - The *structure* of the model is fixed
  - We are only interested in a part of the model:

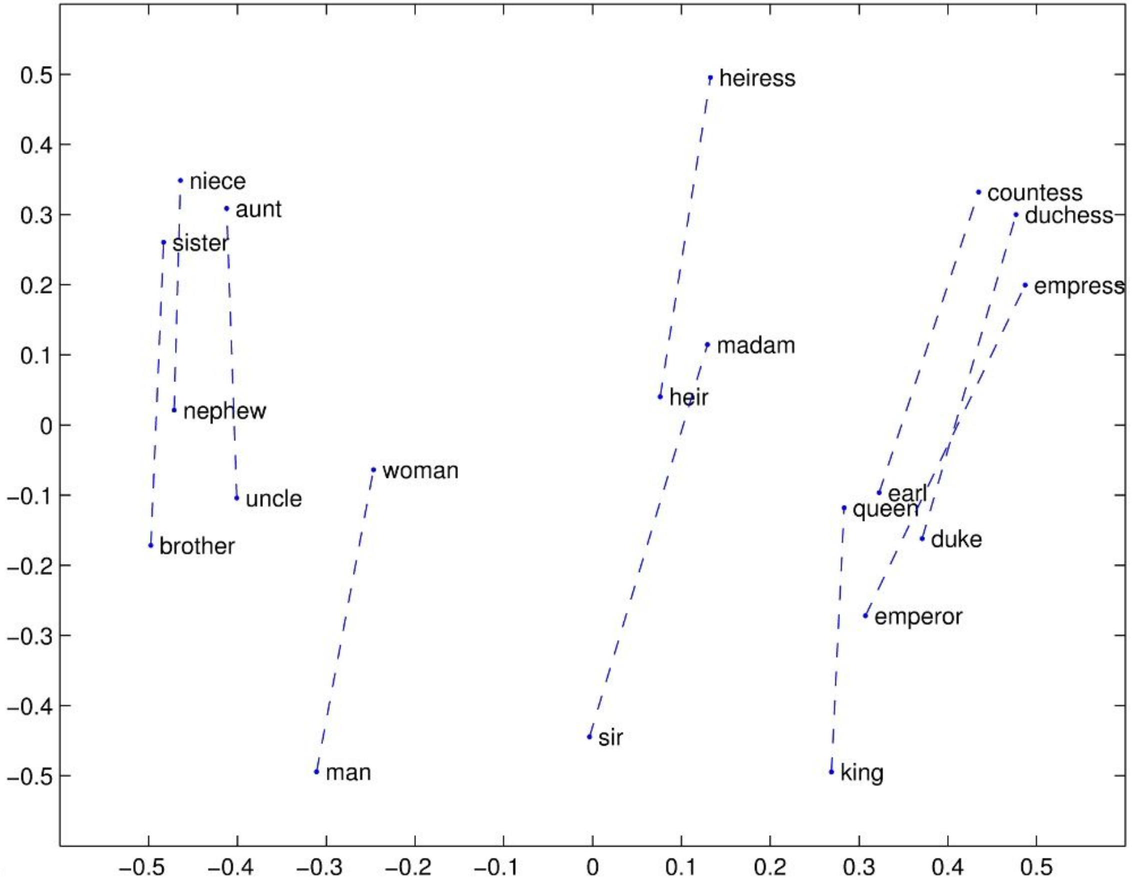
Florida $\xrightarrow{\text{Model 1}}$ (0.5, 0.3, 0.7, 0.9)

This part of the model captures *semantic information in a vector* that allows to play the “guessing game”

Word2vec Captures Semantic Information

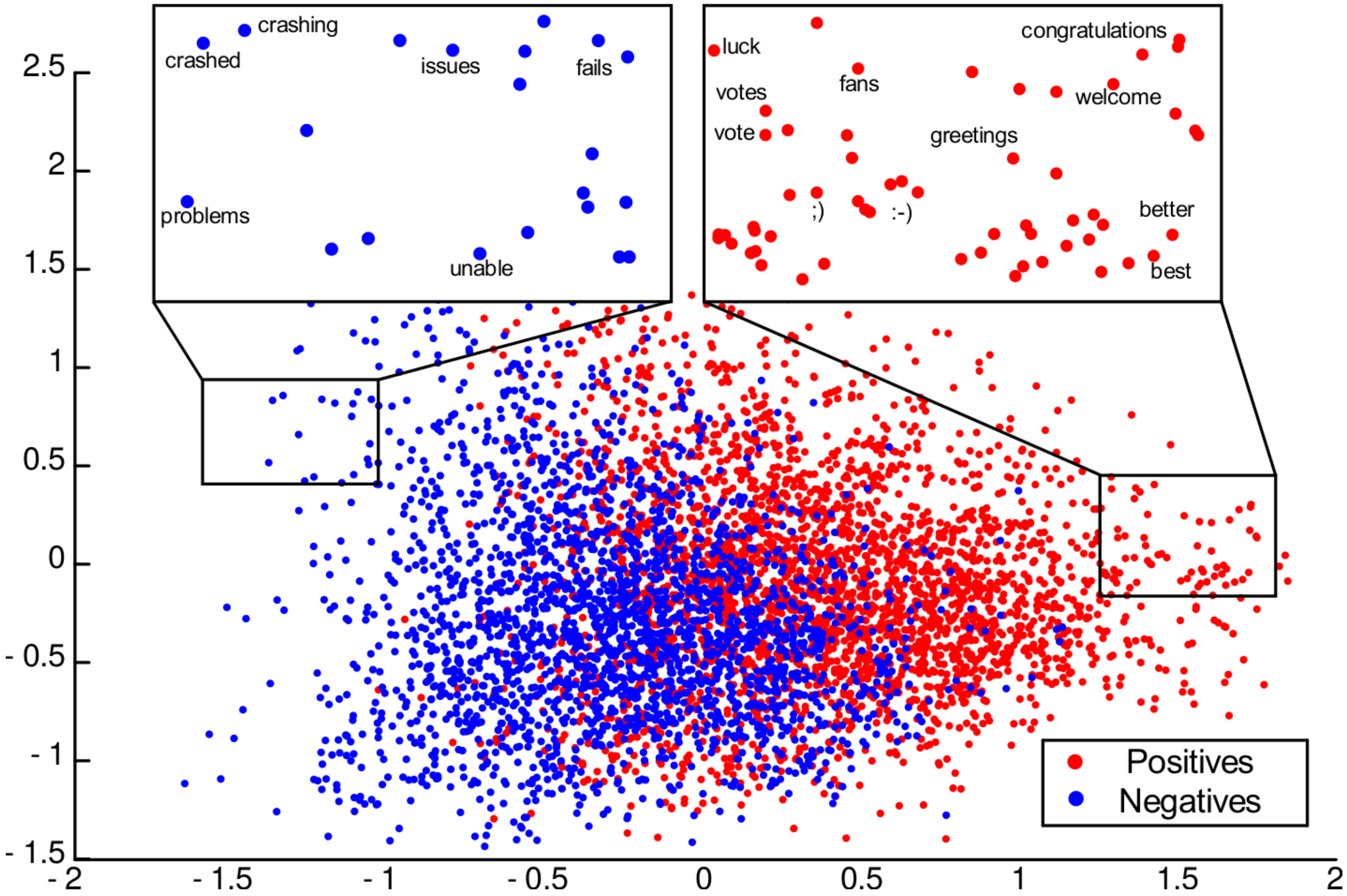


$$\overrightarrow{slower} - \overrightarrow{slow} \approx \overrightarrow{shorter} - \overrightarrow{short}$$



$$\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{king} - \overrightarrow{queen}$$

Amazing Applications: Sentiment Analysis



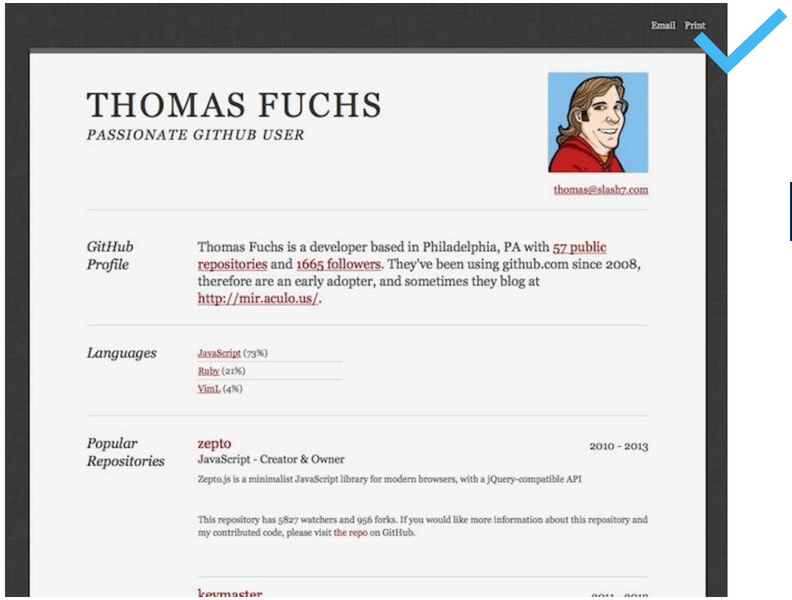
https://www.micc.unifi.it/projects/advanced-web-applications/sentiment-analysis-of-tweets-from-twitter/

Demo word embeddings

- https://projector.tensorflow.org/

Example: CV Screening ?

- **First transforms data into numbers via WordToVec:**



transform

C1	C2	C3	C4	…	Hire?
0.5	7.3	0.34	3.1	…	Yes

- **Then apply any of the classification methods we have seen**

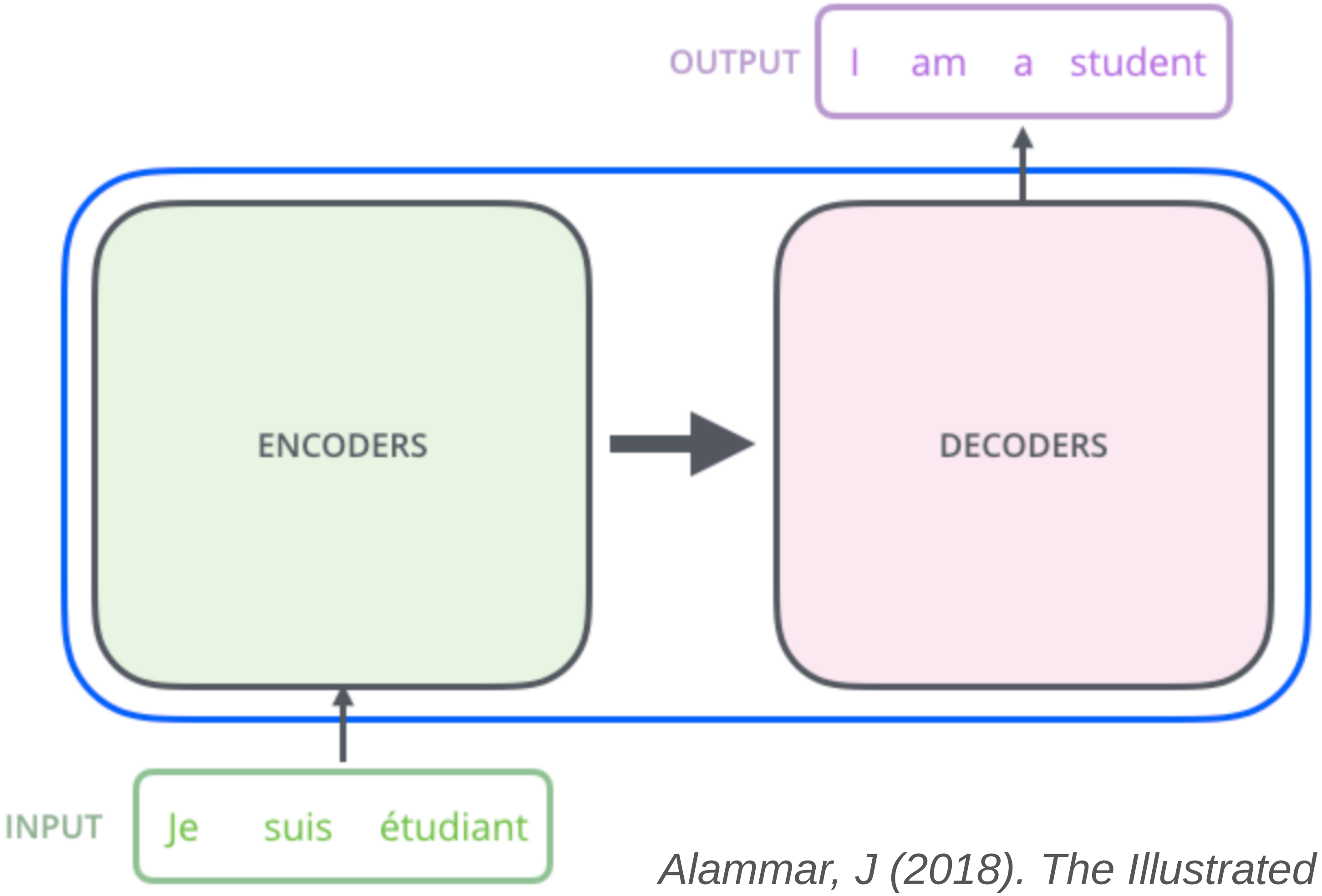
Overview

- “Traditional” NLP
- Word embeddings
- Transformers

Transformers

- **Until the advent of transformers, LSTM-based RNNs were state-of-the-art for sequence-to-sequence problems**
- **Problem with the RNN architecture:**
  - Hard to take long-term dependencies into account
- **The transformer architecture makes the path between the information needed to understand/translate a token shorter**

Transformer: High level view



Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

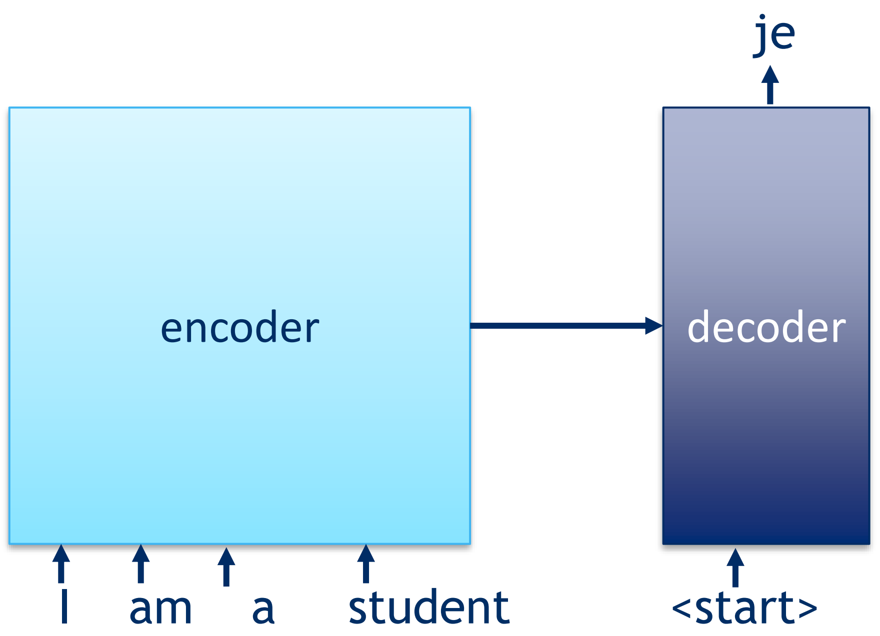
Attention Mechanism

- **The *attention mechanism* allows the network to concentrate on specific tokens to enrich the encoding of a token**

- **Example: The animal did not cross the road because it was too tired.**
  - Who or what is too tired?
- **Example: The animal did not cross the road because it was too wide.**
  - Who or what is too wide?

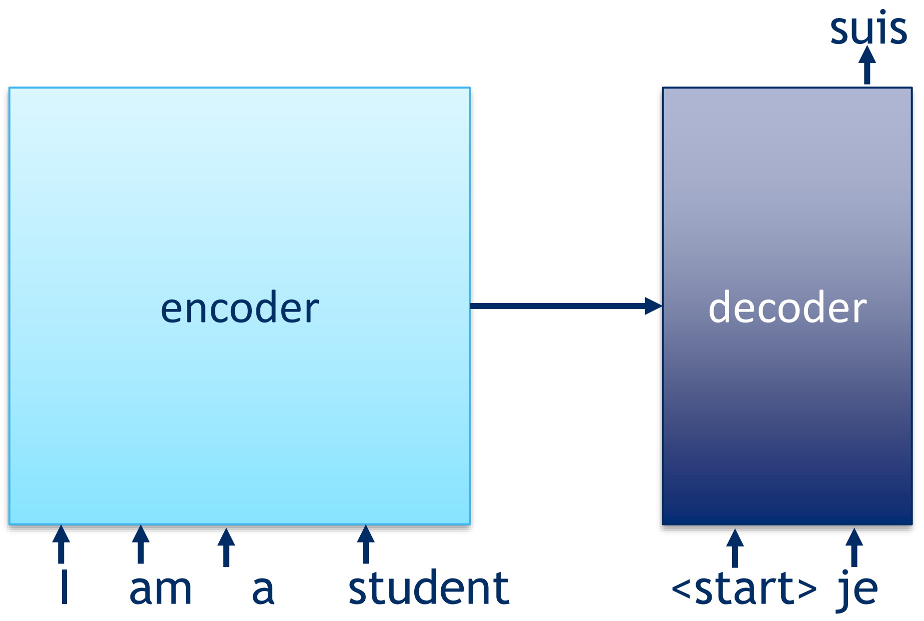
Decoder

- **The decoder produces the next word based on input and preceding words**
- **A similar architecture is used**



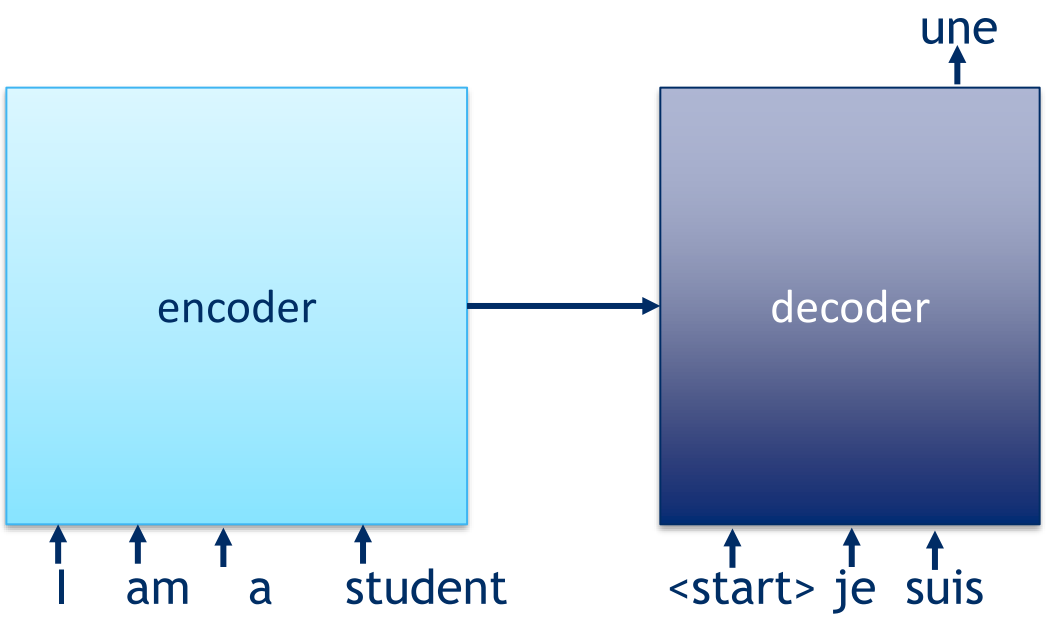
Decoder

- **The decoder produces the next word based on input and preceding words**
- **A similar architecture is used**



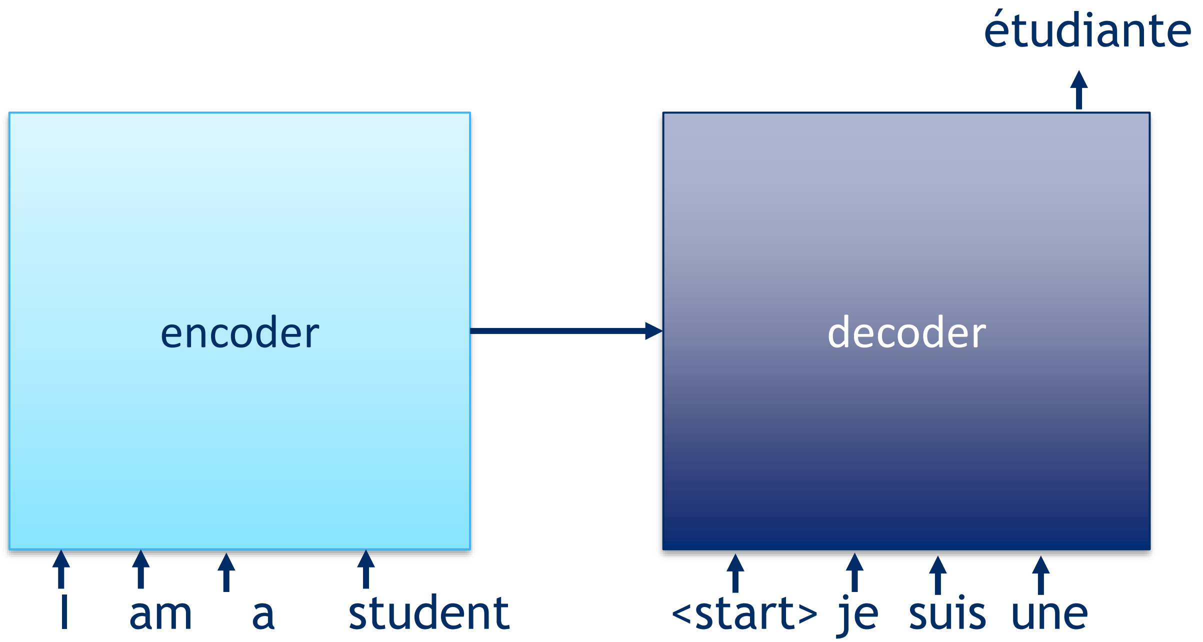
Decoder

- **The decoder produces the next word based on input and preceding words**
- **A similar architecture is used**



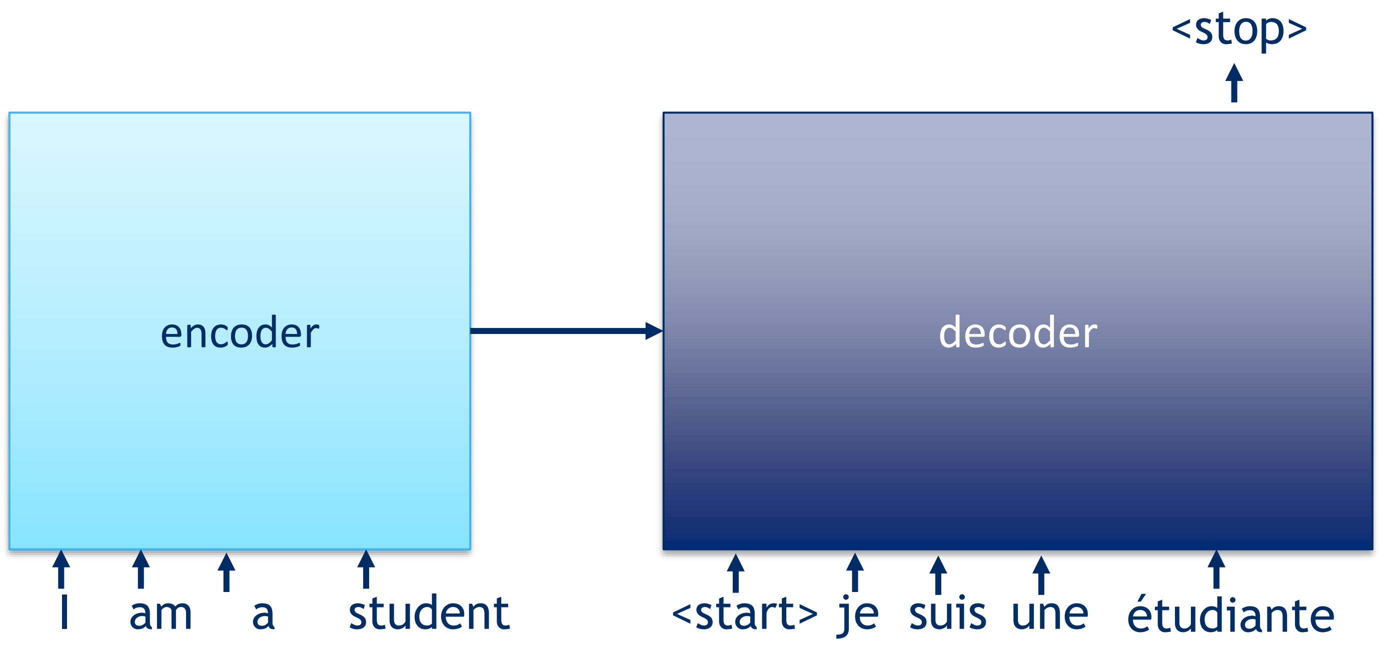
Decoder

- **The decoder produces the next word based on input and preceding words**
- **A similar architecture is used**



Decoder

- The decoder produces the next word based on input and preceding words
- A similar architecture is used



Training a transformer

- **We need a huge corpus of input-output pairs**

  I am a student – Je suis une étudiante

  I am Manuel from Barcelona - Je suis Manuel de Barcelone

  I learn English from a book - J'apprends l'anglais à partir d'un livre

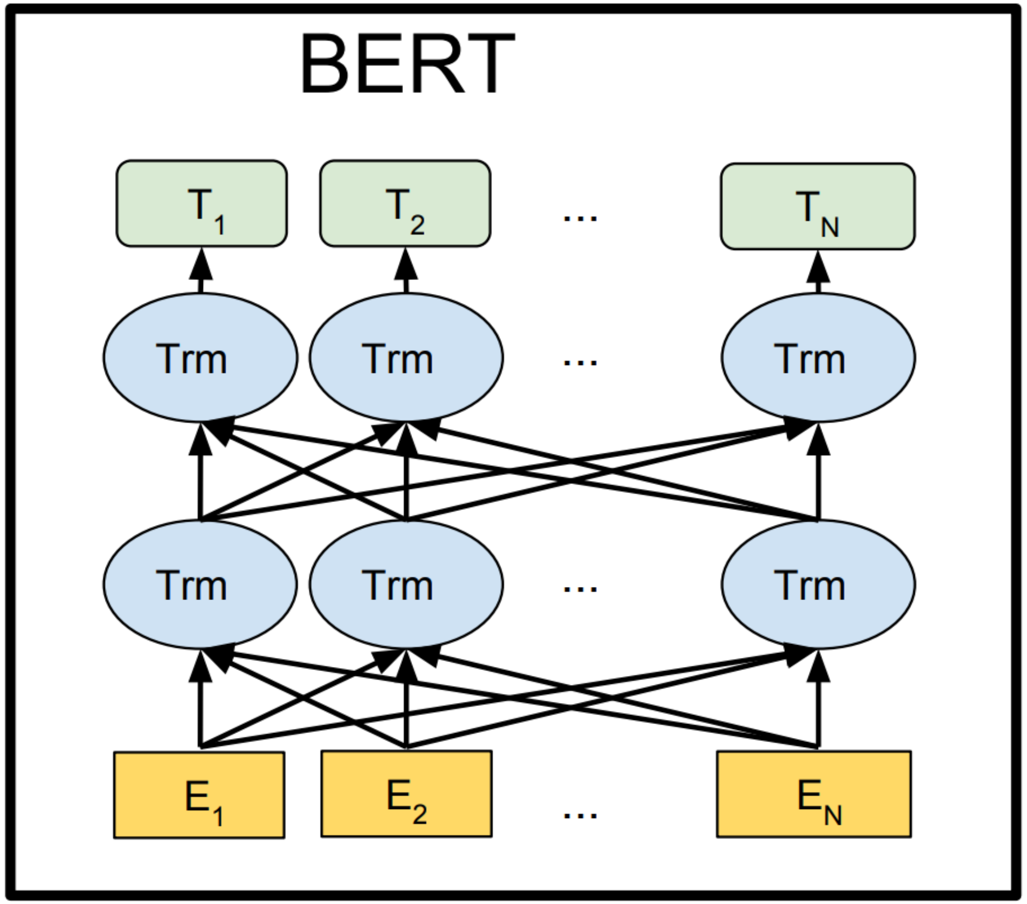
  …

- **For a chatbot: learn to predict the next word/sentence in a conversation**

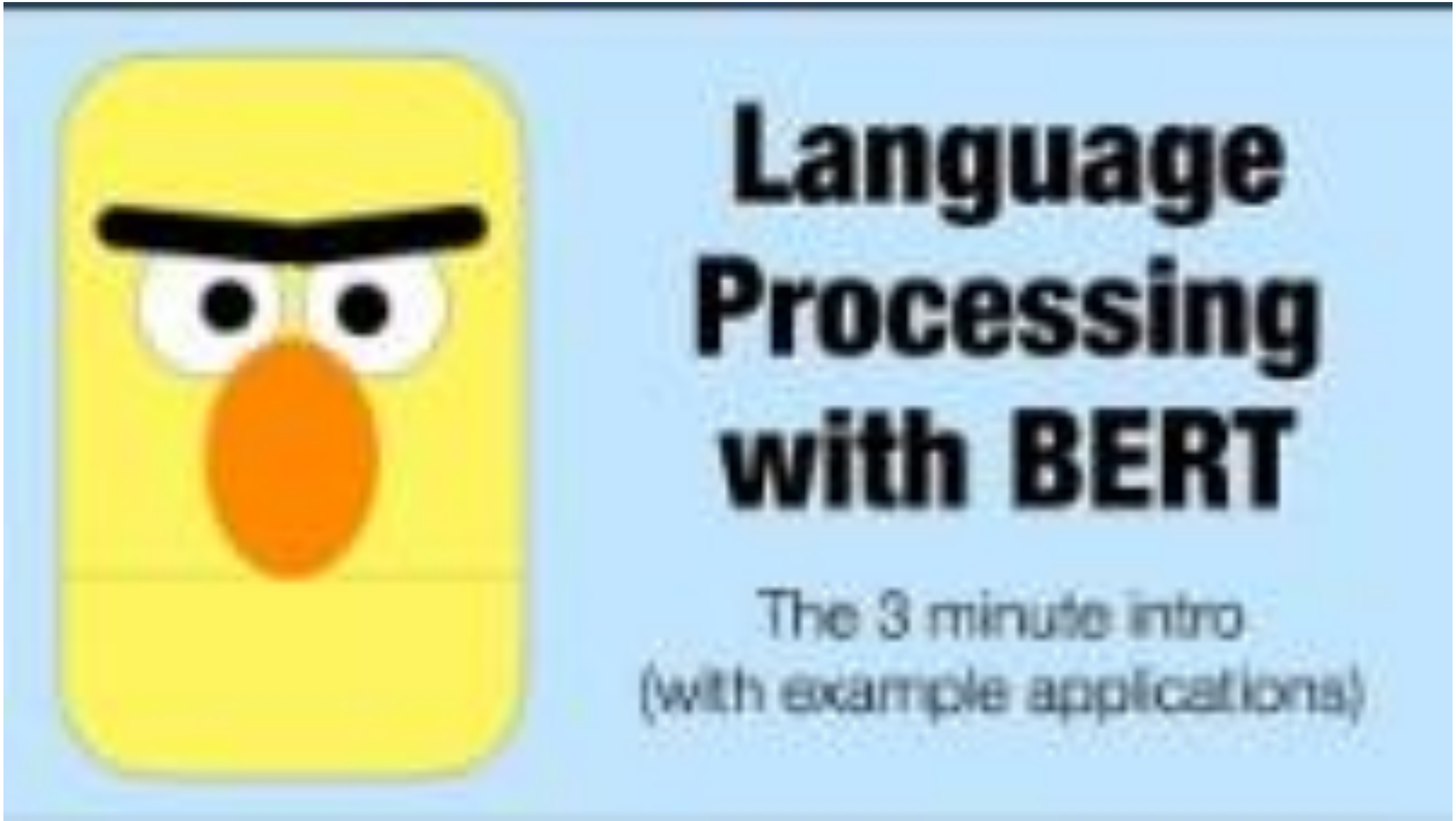
BERT

- **BERT stands for Bidirectional Encoder Representations from Transformers**

- **Trained in a very peculiar way such that the embedding vectors contain a lot of information**
  - missing words
  - context of a word
  - consistency between sentences



Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

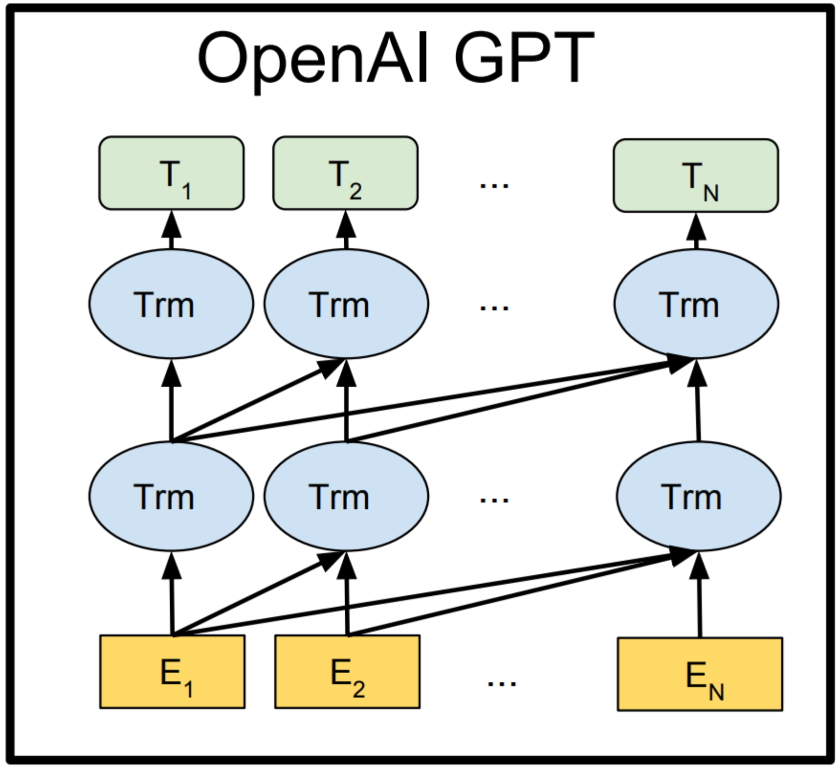


https://youtu.be/ioGry-89gqE

OpenAI GPT Model

- **The OpenAI GPT**
  - General-purpose
  - Pre-trained
  - Transformer

Embeddings themselves are of interest; e.g., build spam prediction on top



- **Trained with next-word prediction**
  - only attend to previous tokens
  - predict next word based on final embedding

Easy to obtain training data; no need for bilingual corpus

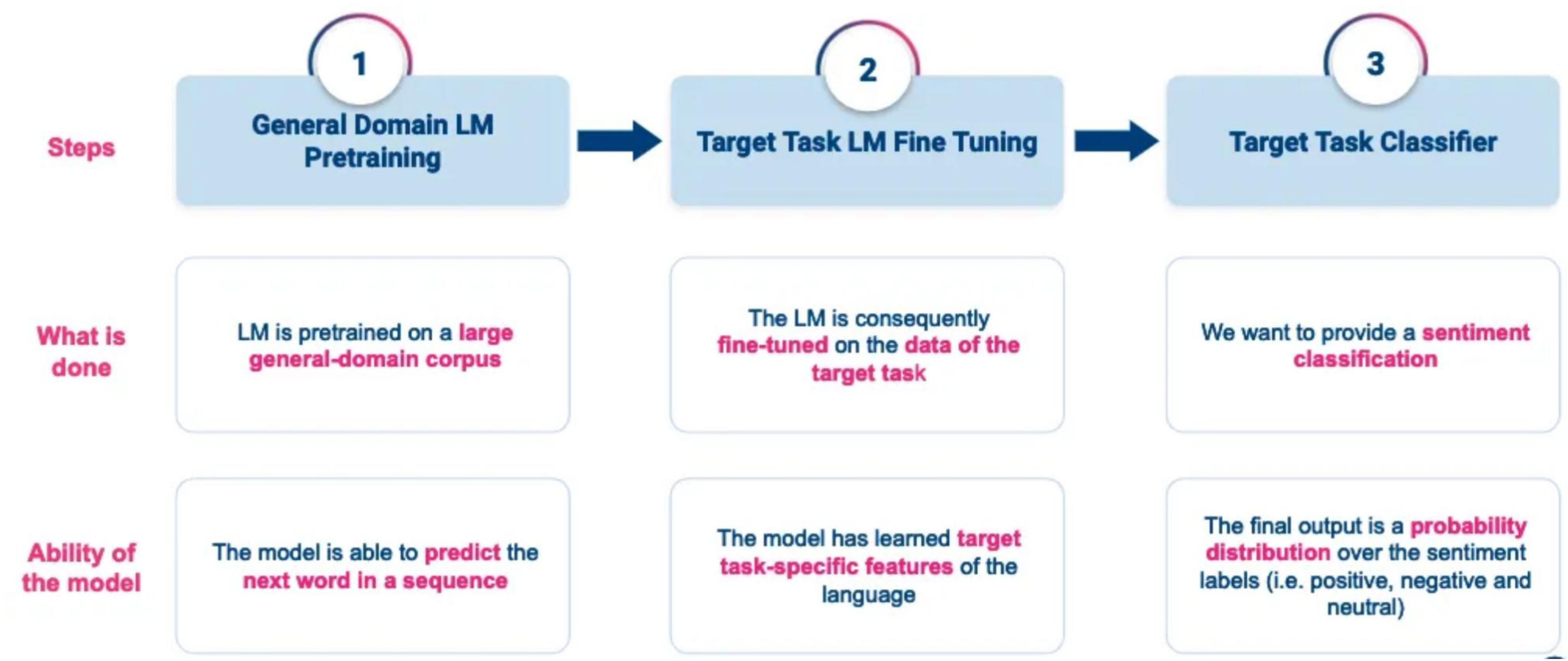
Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

OpenAI GPT-2's Famous Example

SYSTEM PROMPT (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.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	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.

https://openai.com/blog/better-language-mode

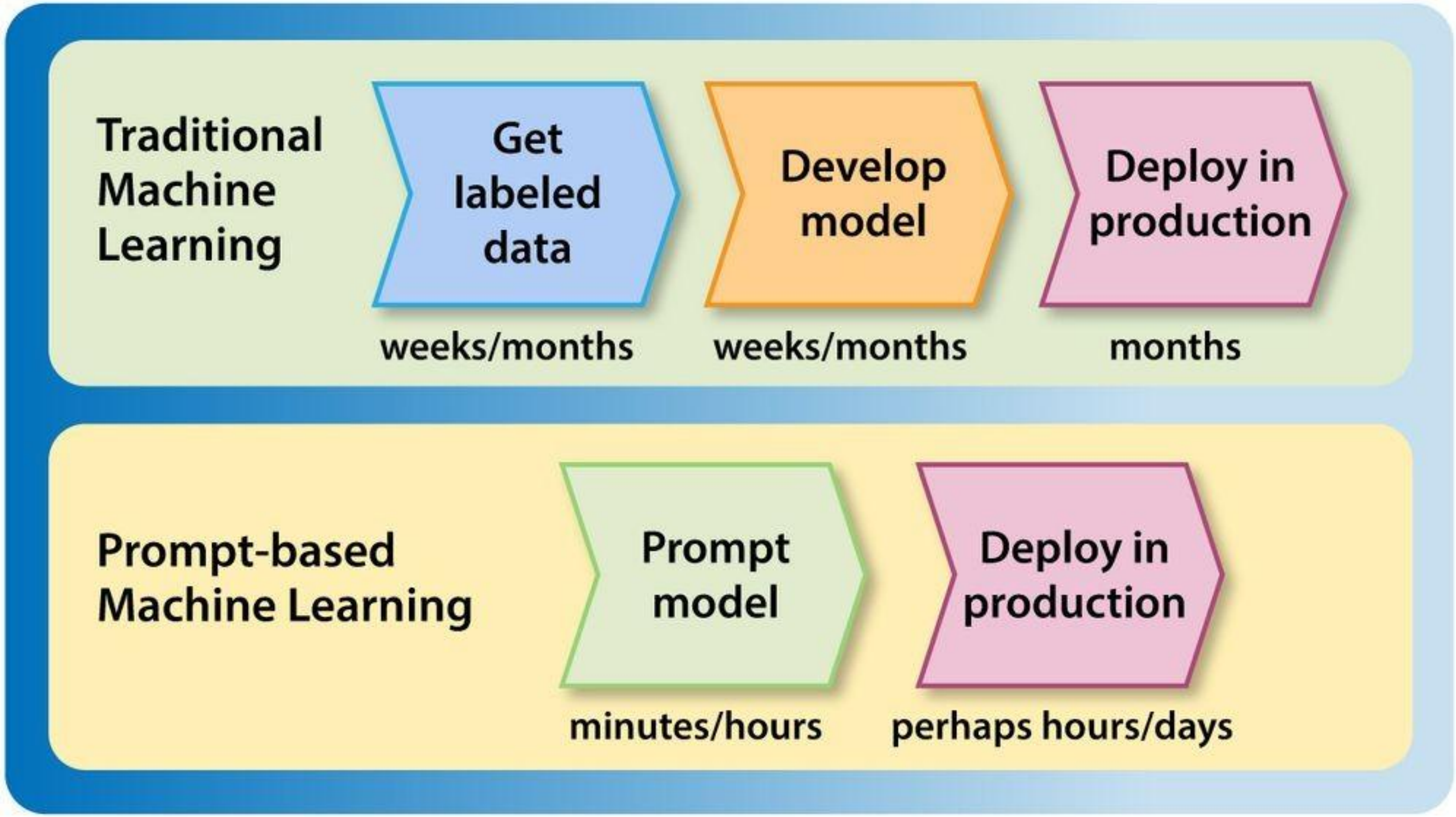
Building an application based on LLMs



https://medium.com/artefact-engineering-and-data-science/how-to-train-a-language-model-from-scratch-without-any-linguistic-knowledge-11acaa933e84

Zero-Shot Learning

- **LLMs are trained on huge amounts of data – surprising powerful**



Example Zero-Shot Learning

Instead of learning a model for a task, design a prompt:

"I bought a bike from you 3 years ago; could you retrieve the make?"

Annotate the request; add context:

"Take the following context into account:

User purchased:

[List of user transactions]

Product catalogue:

[List of products]

Answer the following question by mr X:

I bought a bike from you 3 years ago; could you retrieve the make?"

Send anotated request to a general-purpose LLM

Conclusion

- **Natural Language Processing dominated by Deep Neural Network Models**
  - Word embeddings → classification
  - Transformers → Sequence to sequence tasks
    - Translation, chat-bots, summarization, …
- **Similarly as for image recognition networks can pre-trained models be downloaded, often for free**
  - New standard : offered as a service ?
- **Some issues to take into account:**
  - Fine-tuning on company data may cause privacy issues
  - LLMs can show unpredictable behavior, biases, stereotypes