Introduction to Machine Learning

VU VRIJE UNIVERSITEIT AMSTERDAM

Definition

“Machine Learning is the study of the algorithms that improve their performance *P* at some task *T* with experience *E*.”

Tom Mitchell (1998)

”Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data”

Wikipedia

Machine Learning and Deep Learning

Deep Learning (DL) is a class of Machine Learning (ML) which relies on deep neural networks for learning and inference

$$DL \subset ML$$

OpenAI

Outline

It is impossible to cover everything in one lecture

- Unsupervised ML
- Deep Learning

I'll discuss what you need to know for this course

Supervised ML

- **Task:** Given the size, color, and shape, we want to predict the category (positive or negative)
- **Evidence:**

Example	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negative
4	large	blue	circle	negative

Content inspired by the slides at https://www.cs.utexas.edu/~mooney/cs343/ and http://people.csail.mit.edu/dsontag/courses/ml14/slides/lecture2.pdf

Supervised ML

- Goal is to find a *model* that **generalizes** to unseen data
  - Always seek to find a simple explanation (Occam's razor)

- There are many supervised methods that can compute such a model
  - Support-vector machines
  - Linear regression
  - Naïve Bayes
  - Decision Trees
  - Neural Networks
  - …

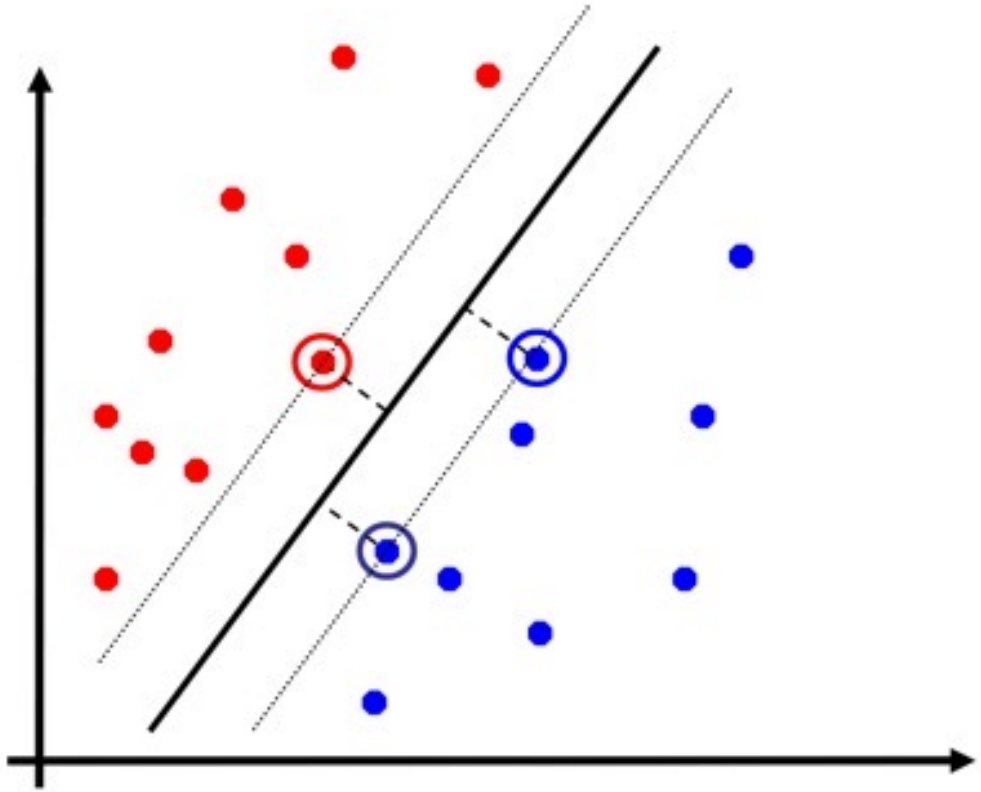
Example: Support vector machines (SVM)

- Transform all features into a numerical vector

Example	Size	Color	Shape	Category
1	0	0	0	positive
2	1	0	0	positive
3	0	0	1	negative
4	1	1	0	negative

Example: Support vector machines (SVM)

- We can represent every example in a n–dimensional space
- **Goal:** Find the linear separator with the largest margin



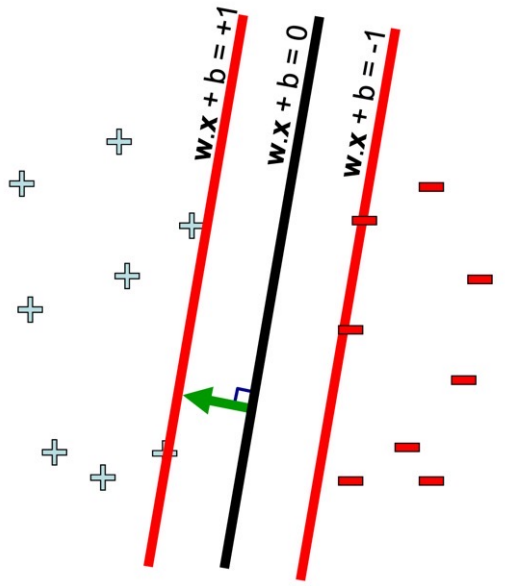
Red: positive
Blue: negative

VU VRIJE UNIVERSITEIT AMSTERDAM

Example: Support vector machines (SVM)

- First, we can start with a random hyperplane
- Then, every time that the hyperplane makes a mistake with some training data, we change it so that it makes fewer errors
- To improve generalization, seeks to find a large separator

Example



VU VRIJE UNIVERSITEIT AMSTERDAM

Gradient descent

How can we change a model so that it makes fewer mistakes?

1. Compute the **loss** using the training data, for instance using MSE:

$$\text{Mean Squared Error (MSE)} = \frac{\sum_{(\vec{x},y)\in T}(f(\vec{x})-y)^2}{|T|}$$

where T is the set of training data and $f(\cdot)$ is the model's output

Gradient: the gradient of $f(x)$ is the vector containing the partial derivatives over all variables of f at x

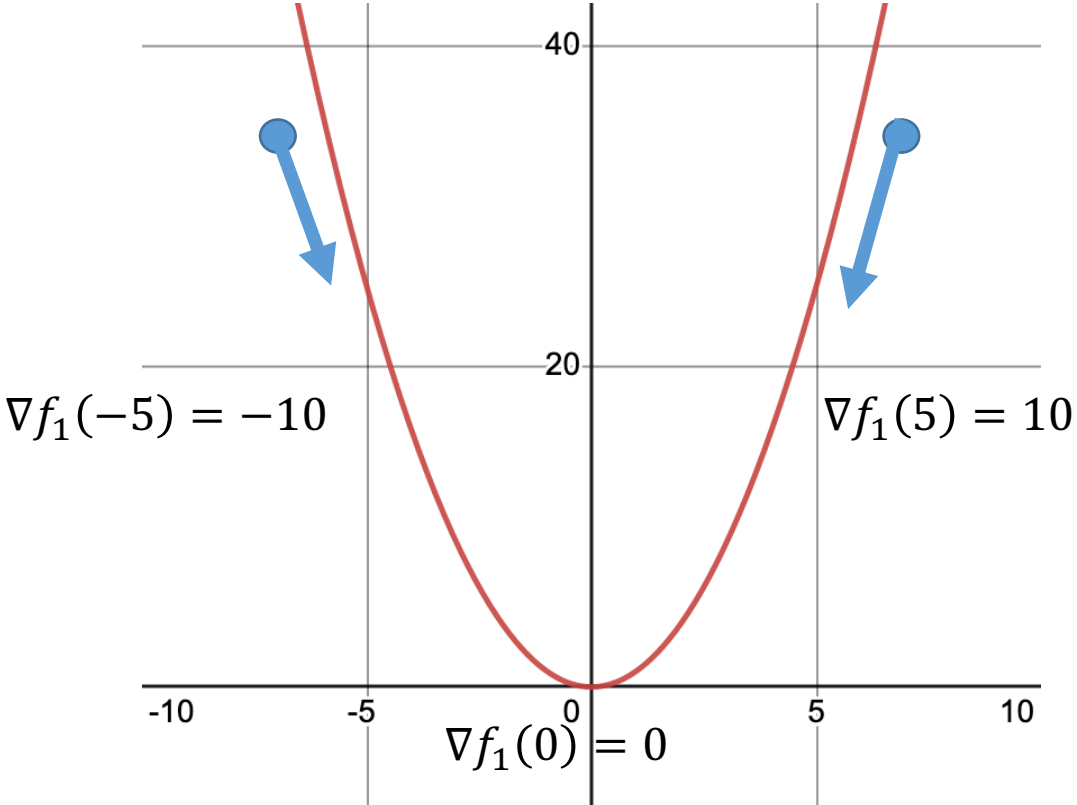
Gradient descent (example)

x is a data point, w is the parameter of the model, $f_x(\cdot)$ is the function that computes the loss at x

$$f_x(w) = xw^2 \qquad \frac{\partial f_x(w)}{\partial w} = 2xw$$

$$\text{Gradient } \nabla f_x(w) = [2xw]$$

How the loss changes if we change w given $x = 1$



Gradient descent

At the bottom of convex functions, the gradient is zero → change w as:

$$w_{t+1} = w_t - \alpha \nabla f(w)$$

Example: if $x = 1$ and $w_t = 5$, and $\alpha = 0.01$ then

$$w_{t+1} = 5 - 0.01 * 10 = 4.9$$

VU VRIJE UNIVERSITEIT AMSTERDAM

Semi-supervised Learning

Supervised learning

Semi-supervised learning

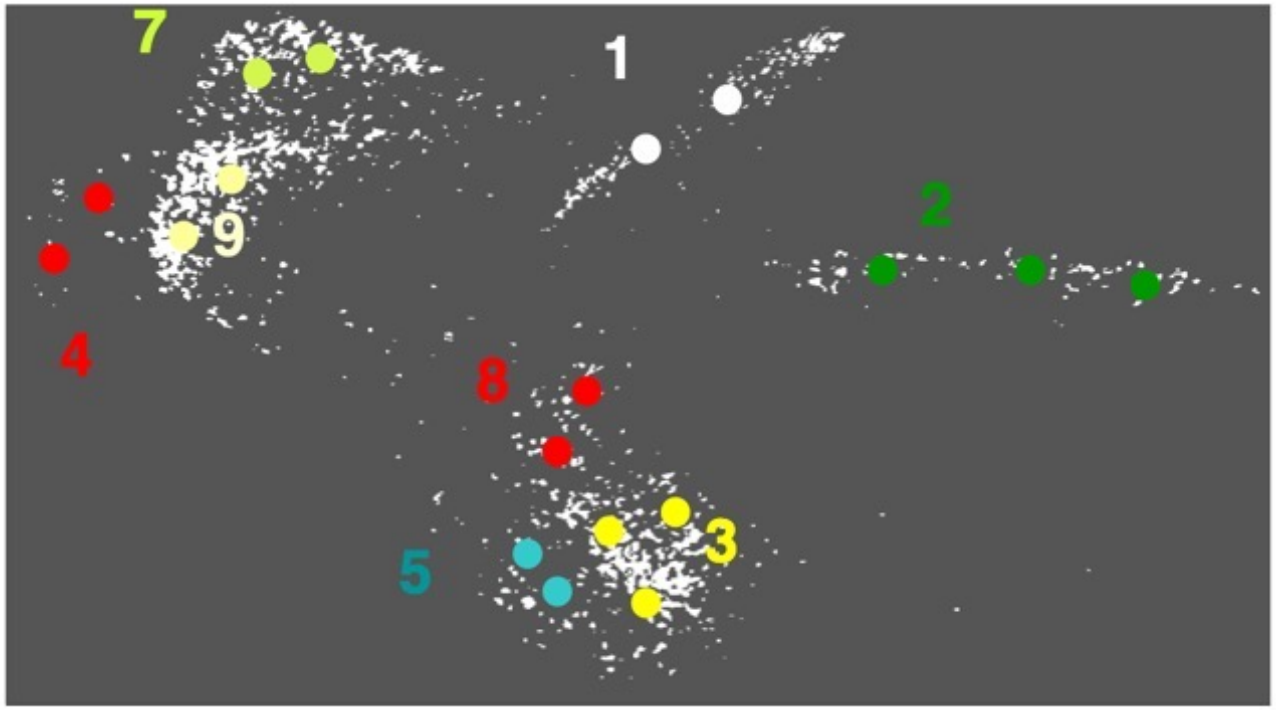
VU VRIJE UNIVERSITEIT AMSTERDAM

Some content is taken from http://www.cs.cmu.edu/~10701/slides/17_SSL.pdf

Semi-supervised Learning

Can unlabeled data help?

Unlabeled data



Similar data points have similar labels!

Semi-supervised Learning

A first approach (bootstrapping)

1. Train a model using the labeled data
2. Used the trained model to label the remaining unlabeled data
3. Re-training the model

Differences

- *(Warning) In the literature, people are not very consistent with the terminology*

- **Semi-supervised learning:** When learning includes both annotated and unannotated data

- **Weak supervision:** When learning is done using labels (or features) that are noisy

- **Distant supervision:** Essentially a synonym of weak supervision (at least in NLP)

Unsupervised Learning

- What do we do if we have no labels?
  - Clustering
  - Autoencoders
  - …

Clustering: group data points so that

- distance between members in the same group is as small as possible
- distance between members in different groups is as high as possible

K-means

Consider $X = \{x_i, \dots, x_n\}$ data points in R^d

Problem: Find *k* points $c_1, \dots, c_k$ (called centers or means) so that

$$\sum_{i=1}^{n} \min_j (dist(x_i - c_j))$$

is minimized

Demo

VU VRIJE UNIVERSITEIT AMSTERDAM

Deep Learning

Some images are taken from

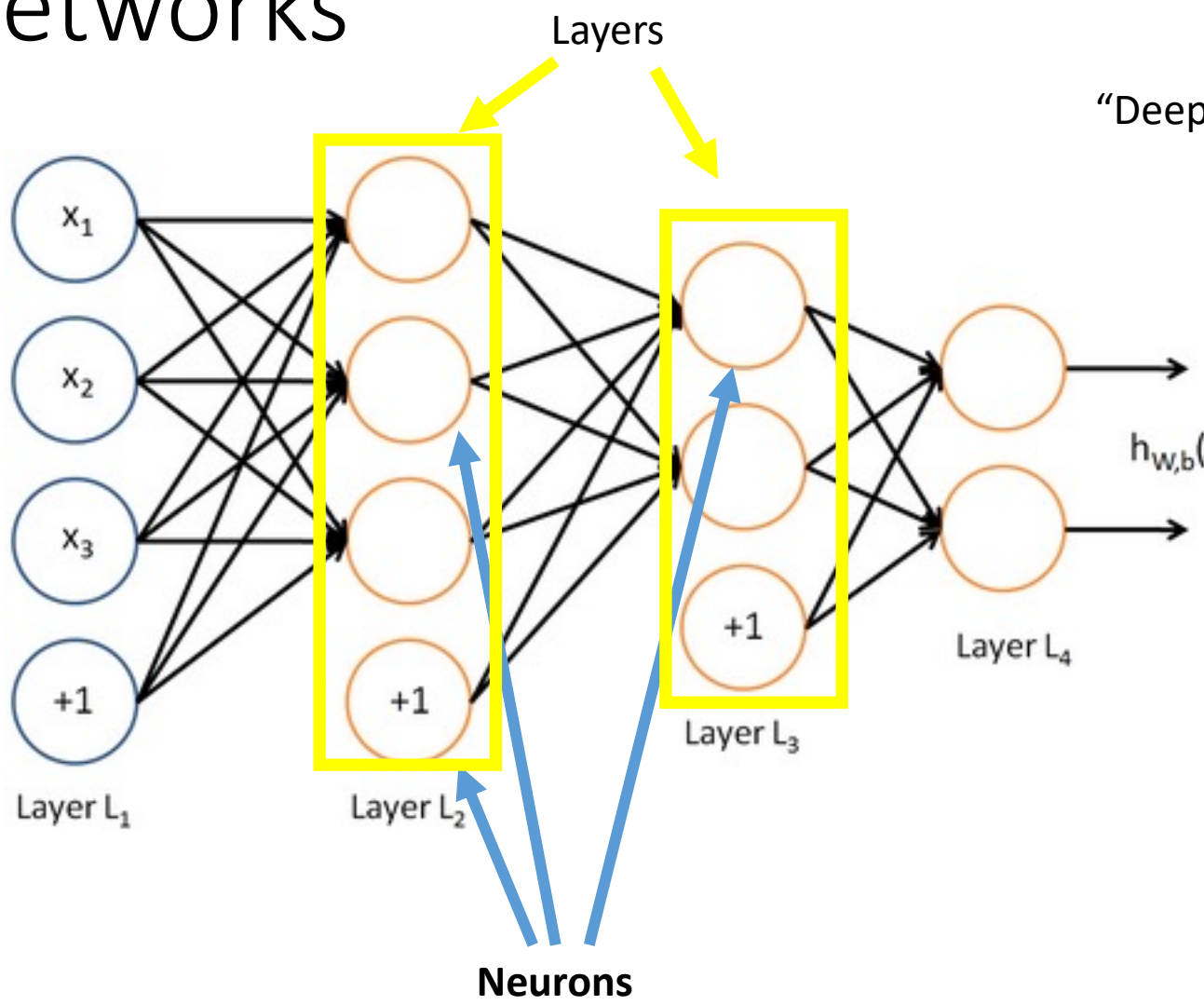
https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf

and

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

VU VRIJE UNIVERSITEIT AMSTERDAM

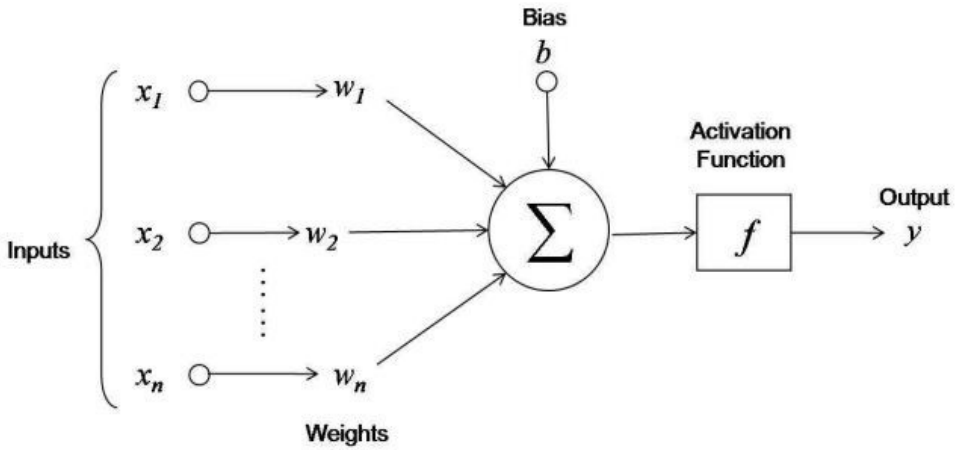
Neural Networks



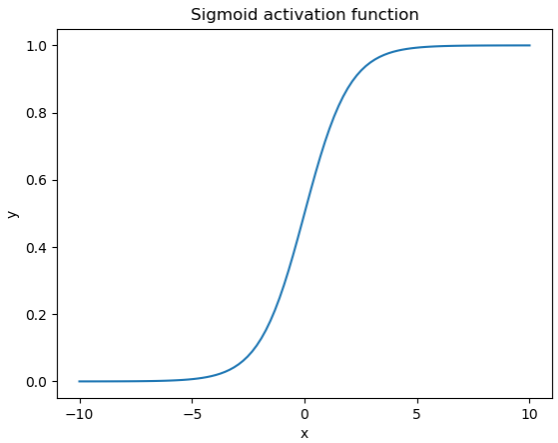
VU VRIJE UNIVERSITEIT AMSTERDAM

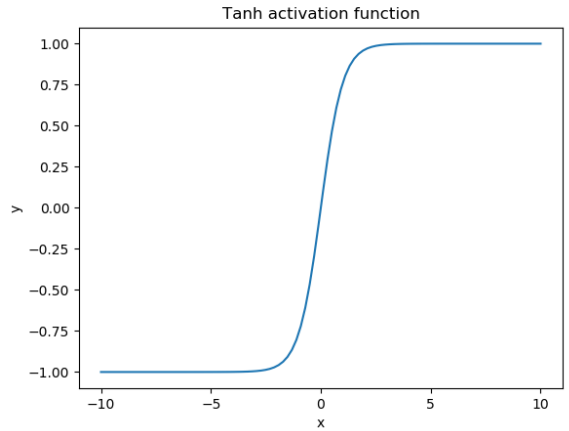
What does a neuron do?

Anatomy of a neuron



Example activation functions





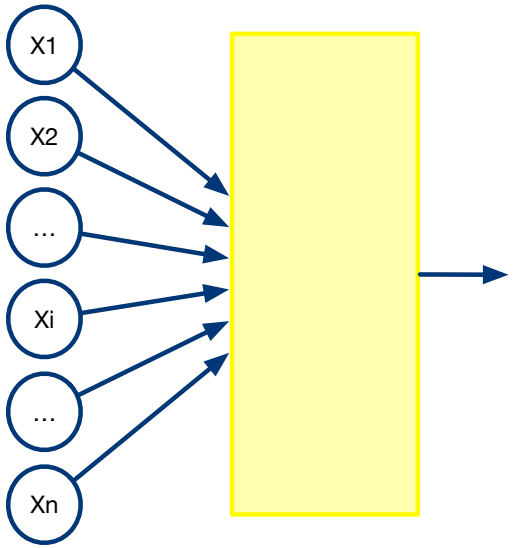
Training: Learn good values for the weights $(w_1, \ldots, w_n)$ so that the returned output is as close as the one observed in training data

Inference: Compute the output for previously unseen data

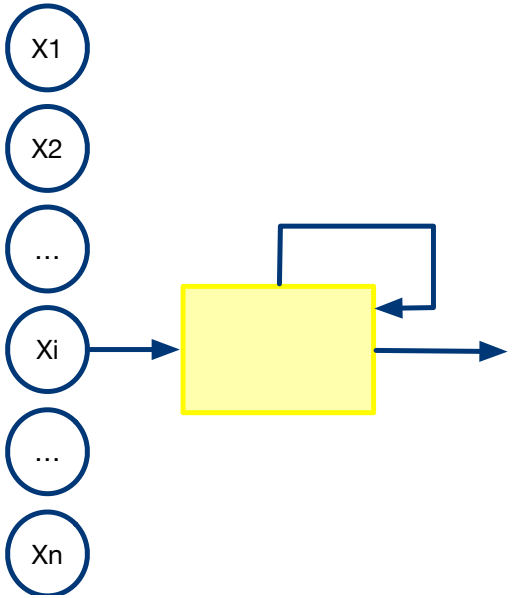
VU VRIJE UNIVERSITEIT AMSTERDAM

Feed-forward vs. Recurrent

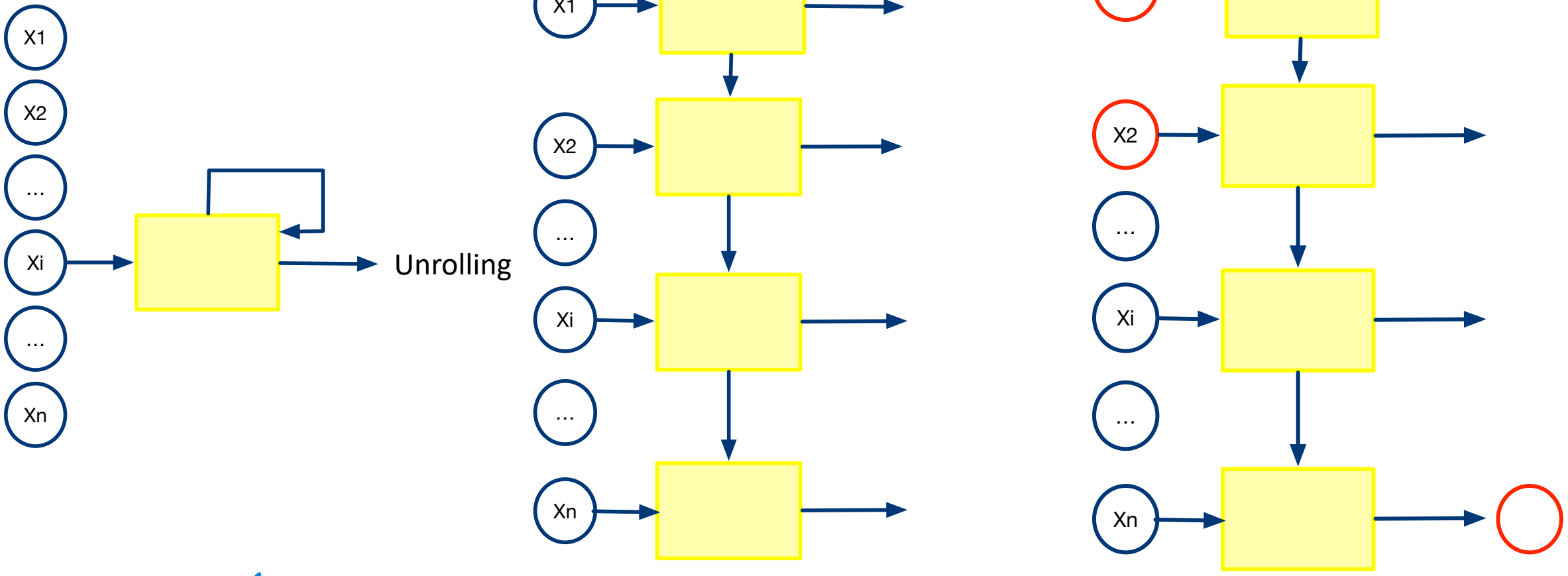
Feed-forward networks do not have memory



Recurrent neural networks do so by adding cycles



The problem of Long-term Dependencies

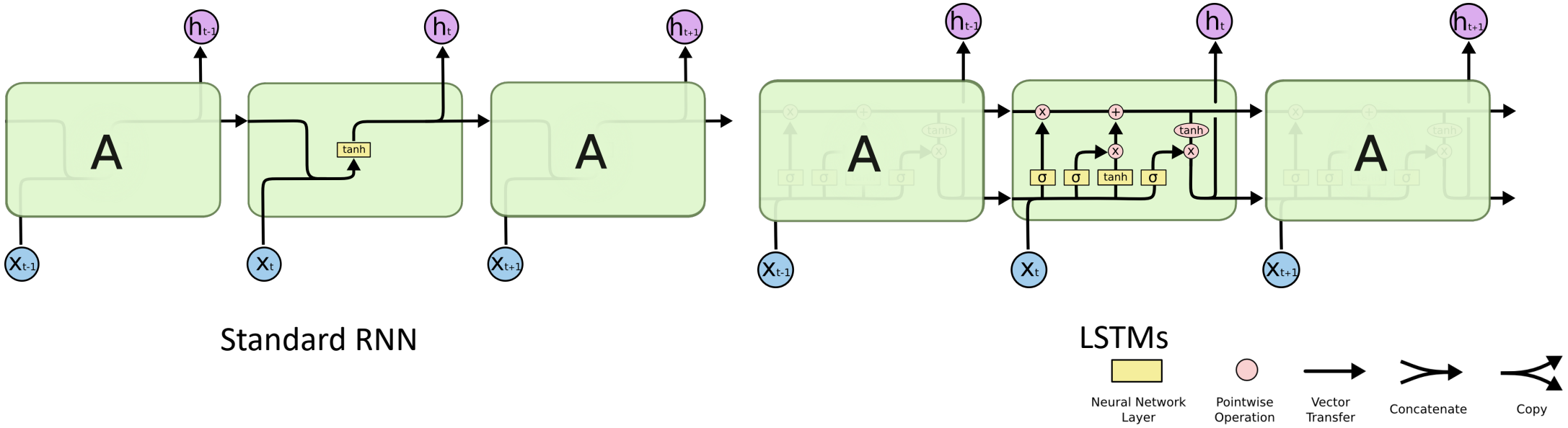


VU VRIJE UNIVERSITEIT AMSTERDAM

Problem: How can we remember long sequences?

Long Short Term Memory

Long Short Term Memory networks (LSTMs) [1] are a special type of recurrent neural networks for learning long-term dependencies



Basic idea: Instead of a single layer, have four interactive ones

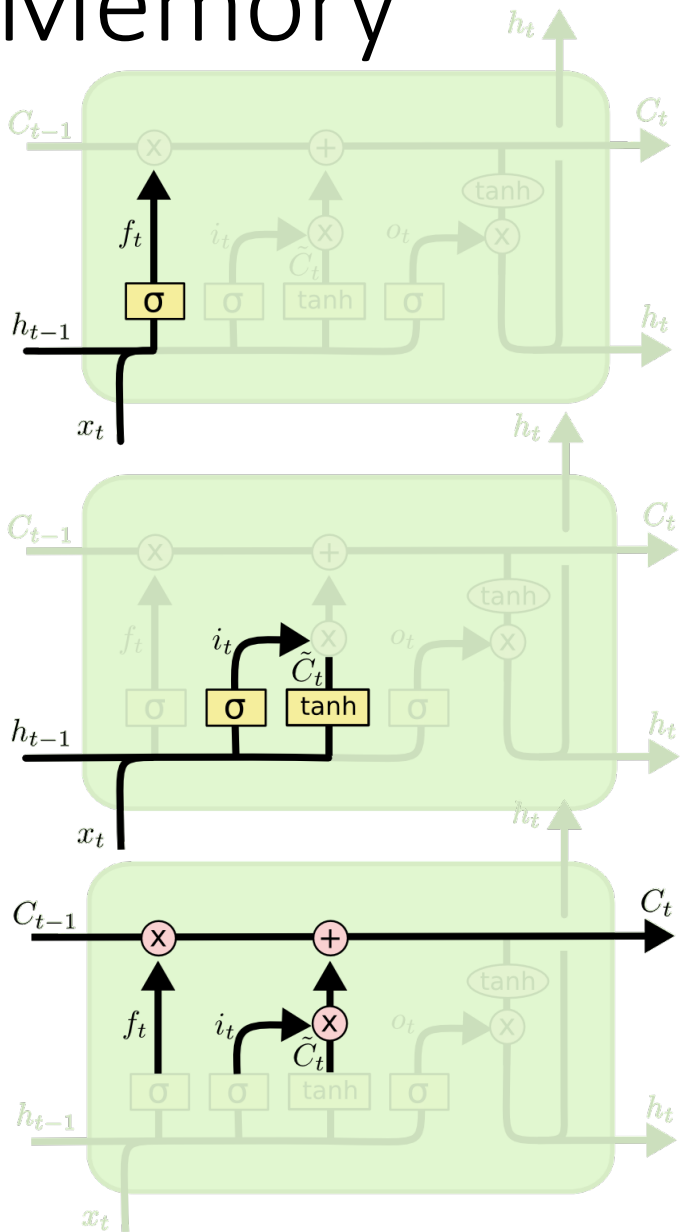
VU VRIJE UNIVERSITEIT AMSTERDAM

[1] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. *Neural Comput* 1997; 9 (8): 1735–1780

Long Short Term Memory

1st: Forget layer (should we forget what we have learned so far?)

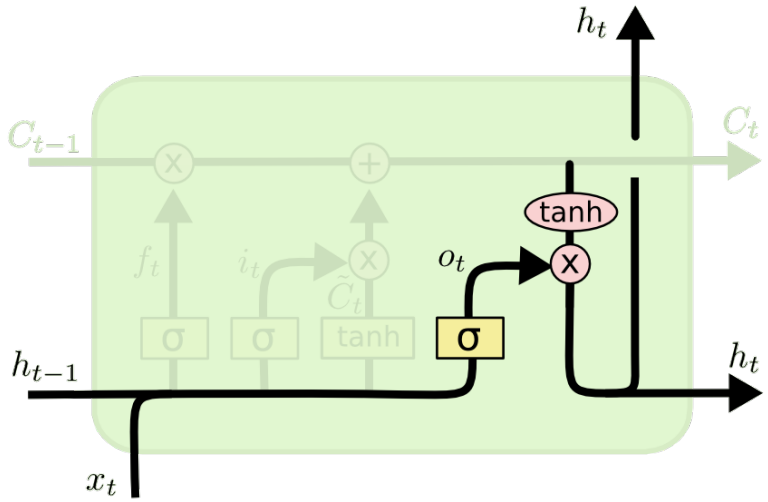
2nd and 3rd layers: Input layers (what should we remember from the current input?)



VU VRIJE UNIVERSITEIT AMSTERDAM

Long Short Term Memory

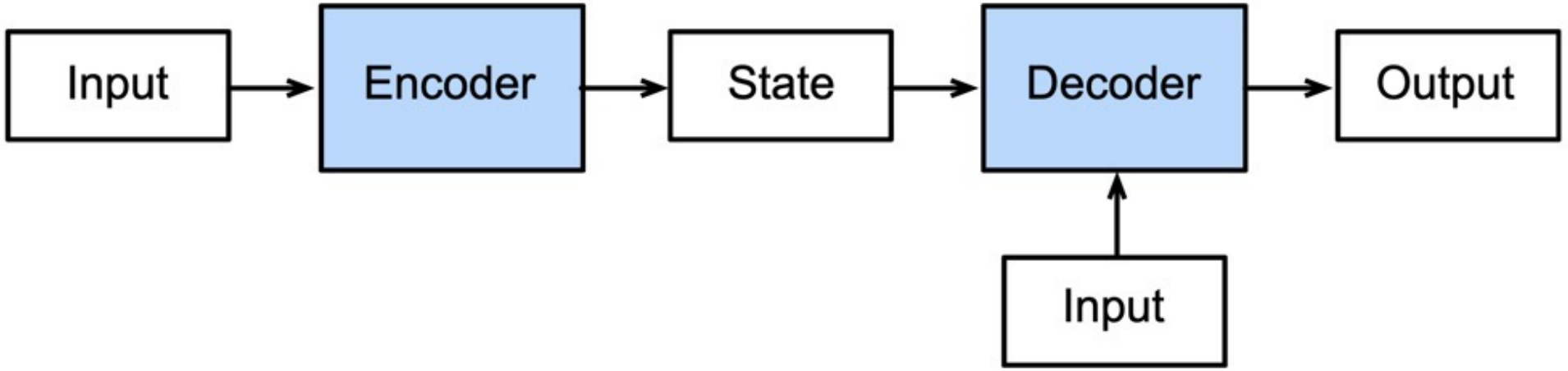
4st: output layer (should we return as output?)



More info: Please read http://colah.github.io/posts/2015-08-Understanding-LSTMs/ Excellent tutorial!

Encoder-Decoder

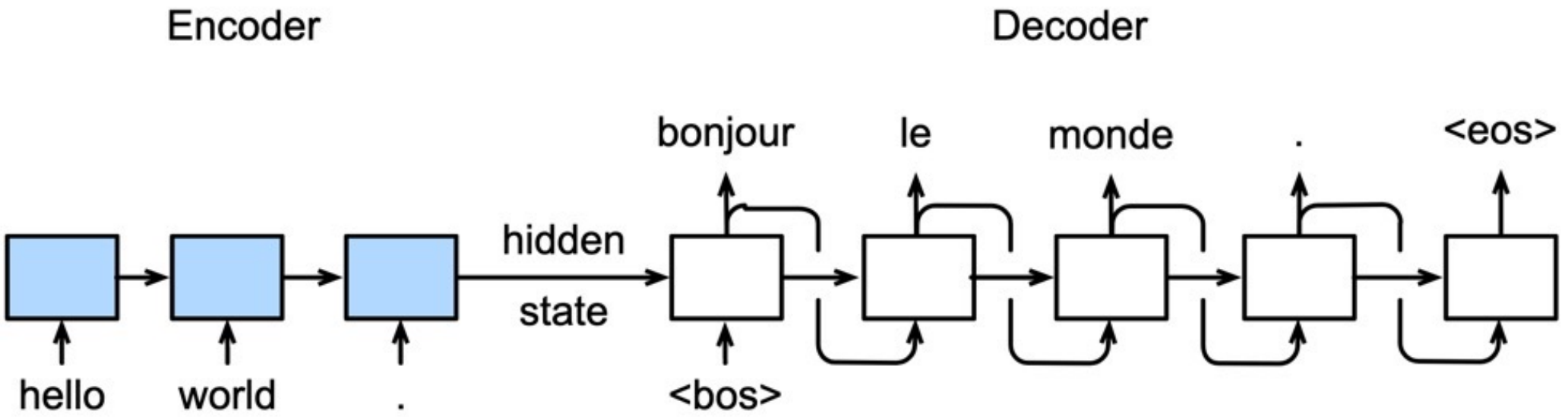
- Popular DL architecture used for tasks like translation, etc.
- **Encoder:** encode the input into an intermediate representation
- **Decoder:** decode the intermediate representation into the output



Original slides at https://courses.d2l.ai/berkeley-stat-157/slides/4_23/23-Encoder-decoder.pdf

seq2seq

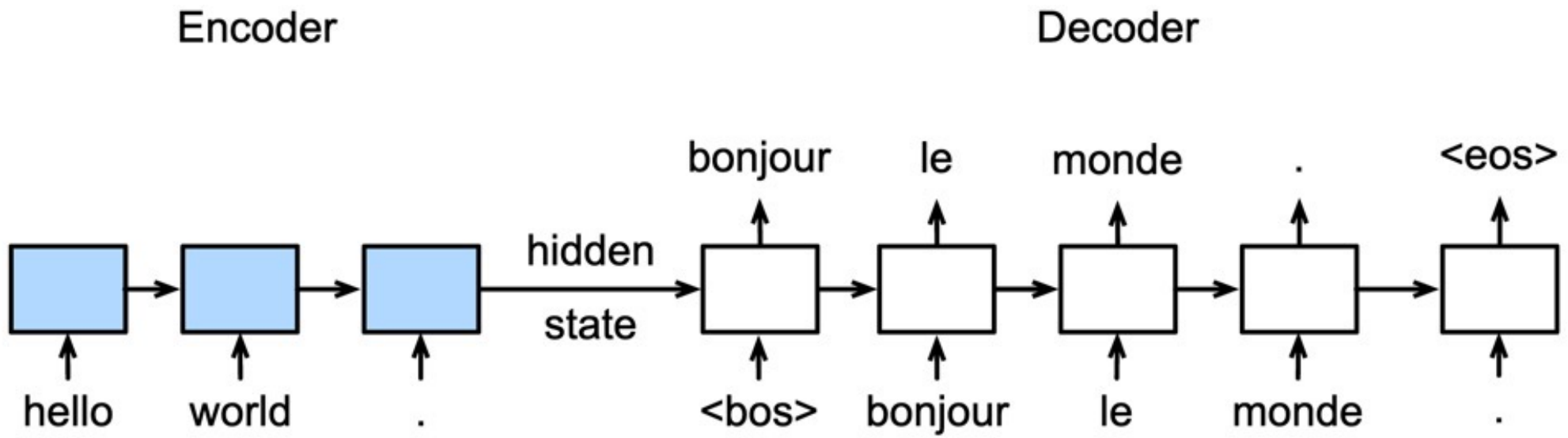
- Encoder-Decoder application to sequences of tokens



VU VRIJE UNIVERSITEIT AMSTERDAM

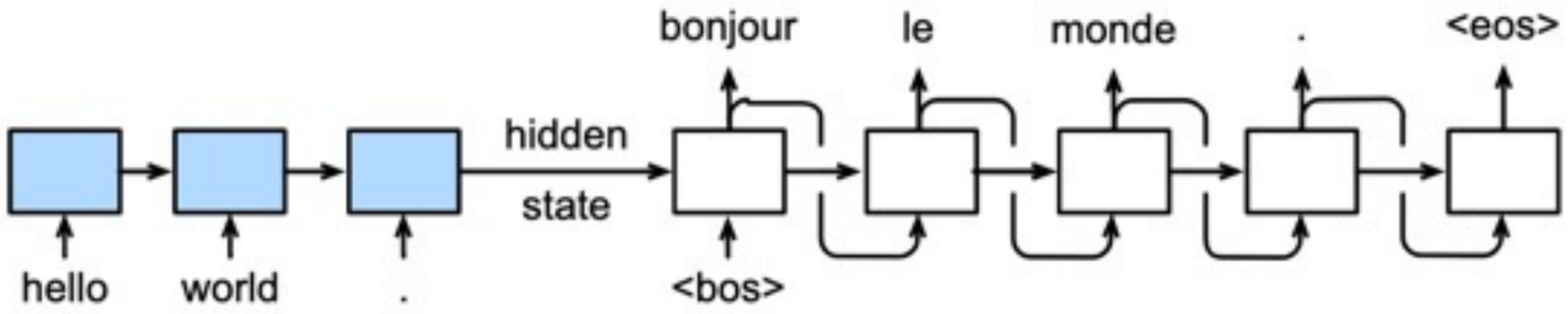
seq2seq

- During training we feed the targeted sentence during training



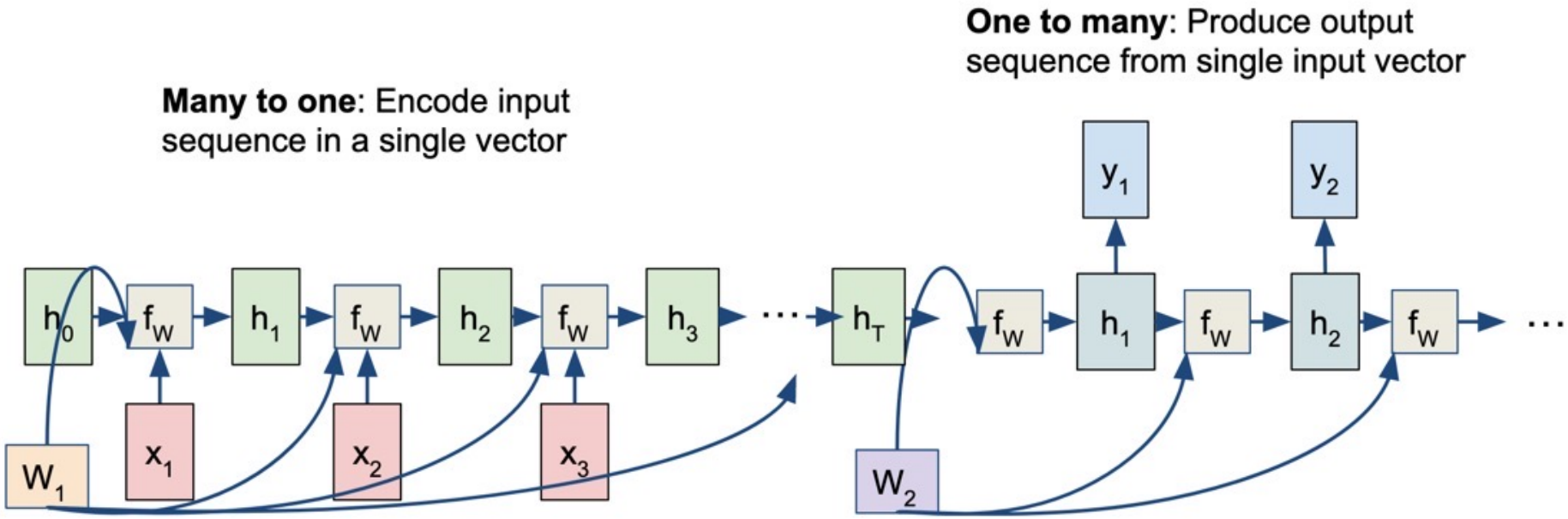
seq2seq

- During prediction we pass only the input



Problem

- Seq2seq: many-to-one + one-to-many

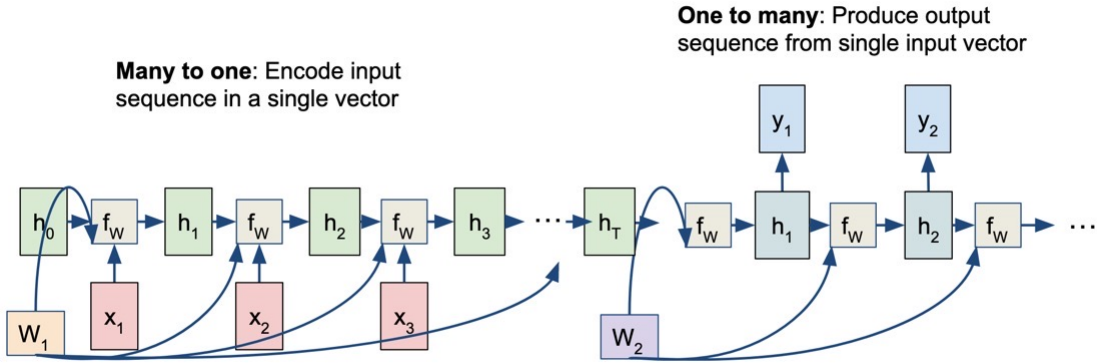


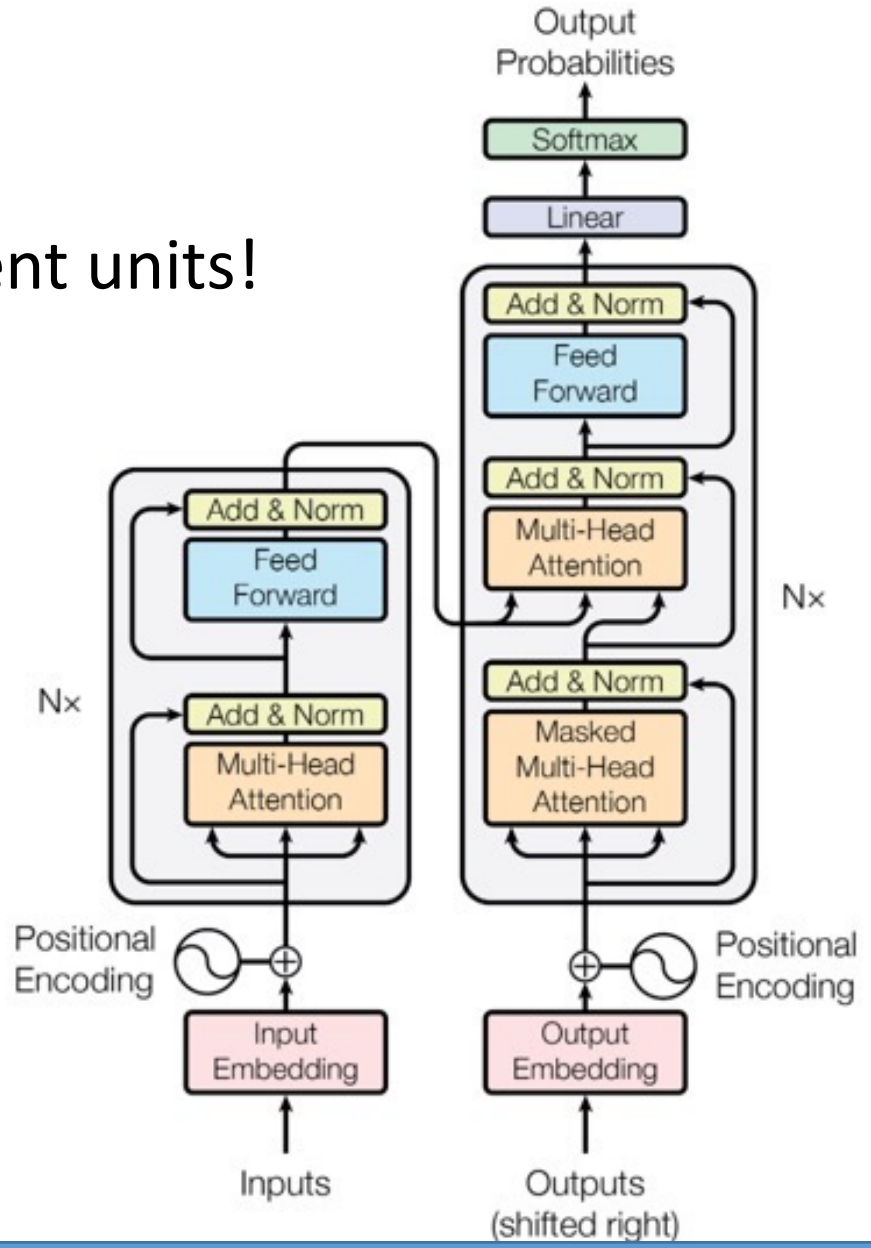
VU VRIJE UNIVERSITEIT AMSTERDAM

Original slides at https://www.cc.gatech.edu/classes/AY2020/cs7643_fall/slides/L16_attention_transformers.pdf and http://cs231n.stanford.edu/slides/2021/lecture_11.pdf

Transformer

No recurrent units!



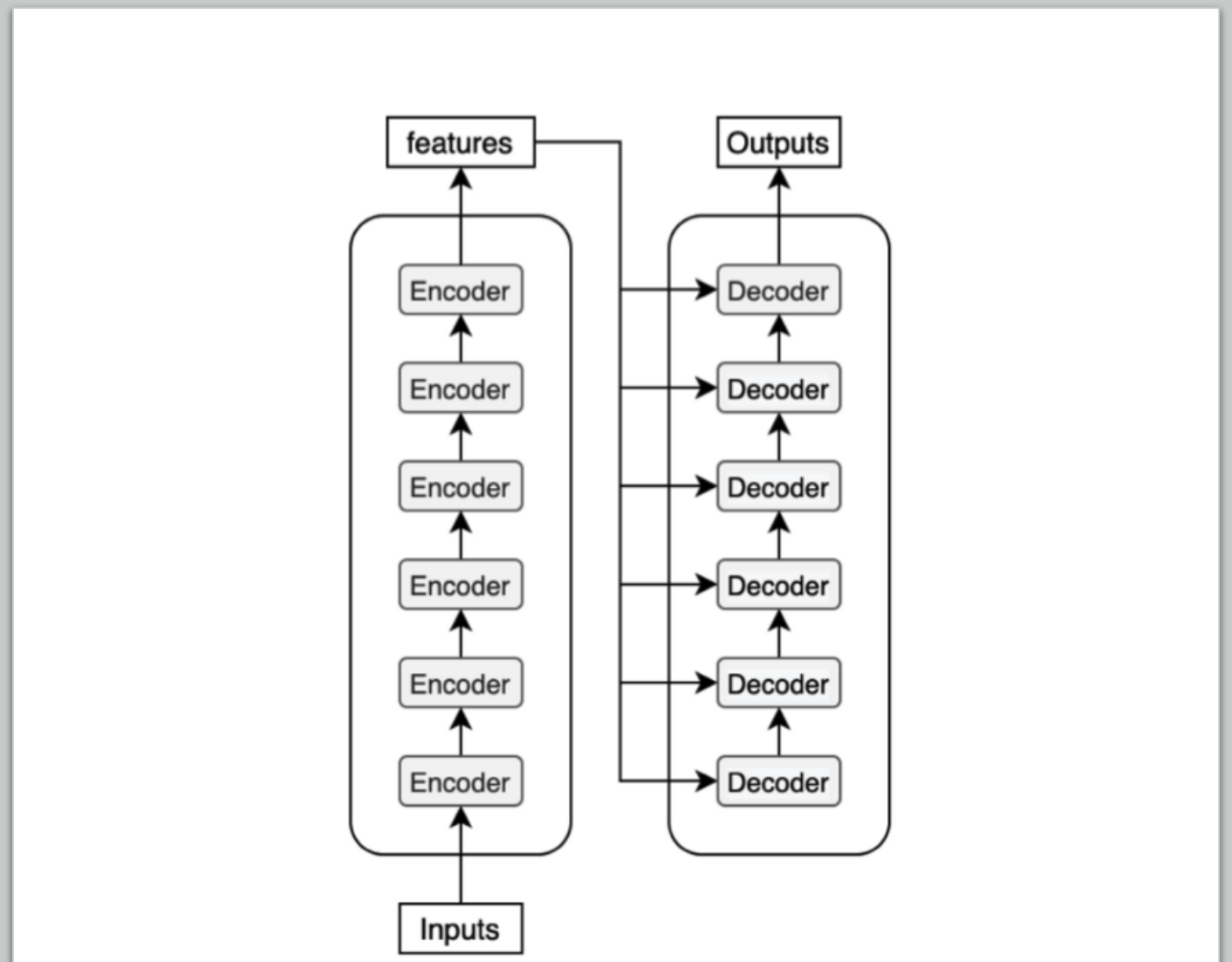


VU VRIJE UNIVERSITEIT AMSTERDAM

Original slides at
https://www.cc.gatech.edu/classes/AY2020/cs7643_fall/slides/L16_attention_transformers.pdf
and http://cs231n.stanford.edu/slides/2021/lecture_11.pdf

Transformer [1]

- In essence a transformer is a encoder/decoder architecture
- Multiple encoders/decoders are “stacked” together



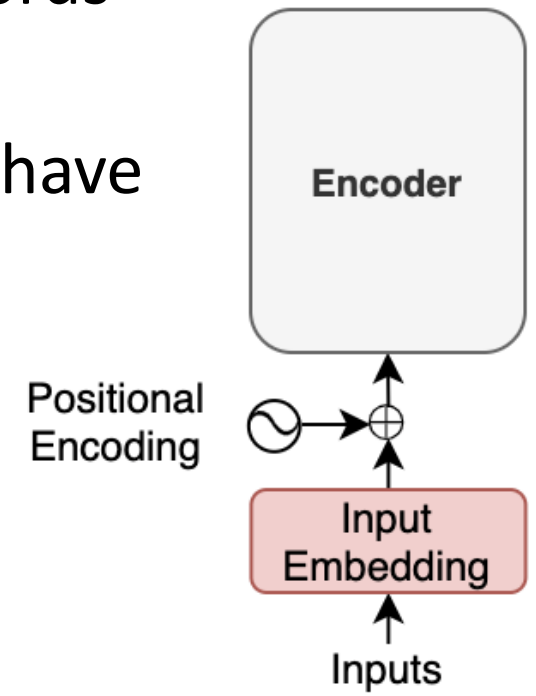
VU VRIJE UNIVERSITEIT AMSTERDAM

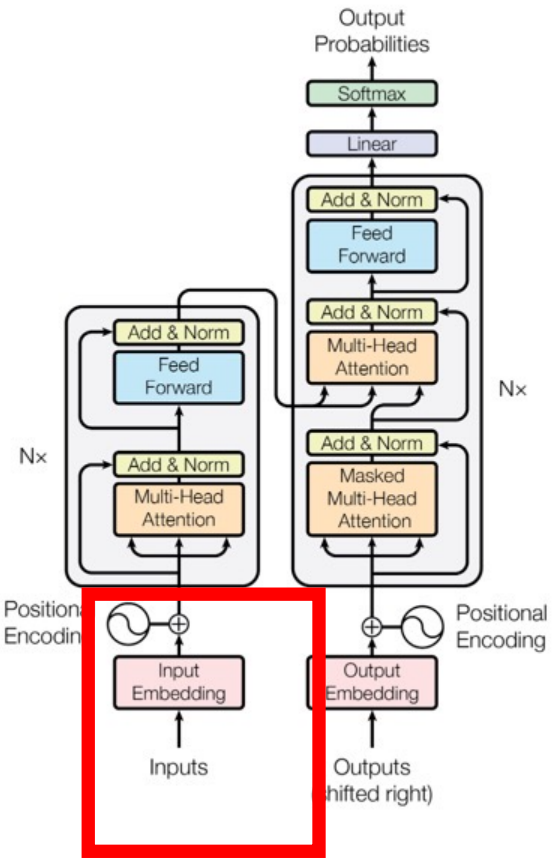
[1] *Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Lukasz; Polosukhin, Illia (2017-06-12). "Attention Is All You Need".*
See also https://kikaben.com/transformers-encoder-decoder/

Transformer: Input

- Words/tokens are representeby embeddings
- Input contains also **positional encodings** so that the model knows the position of words
- Without them, the model may believe the following senteces have the same meaning

1. Tom bit a dog
2. A dog bit Tom



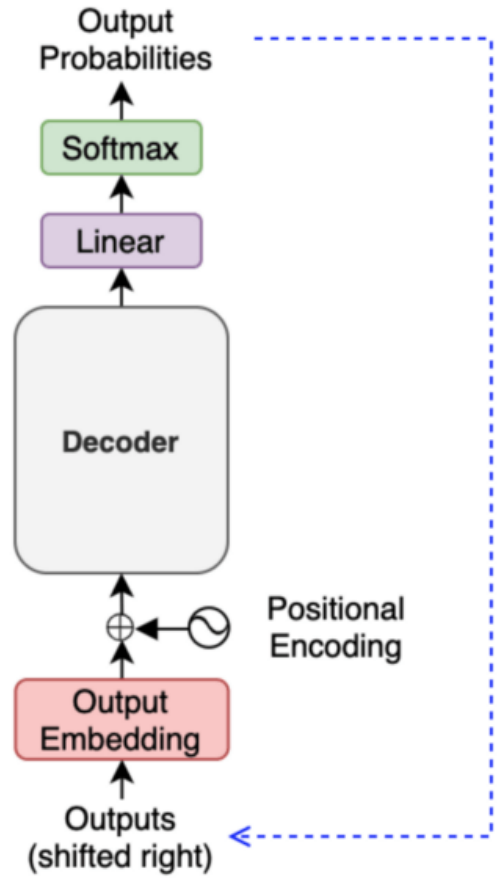


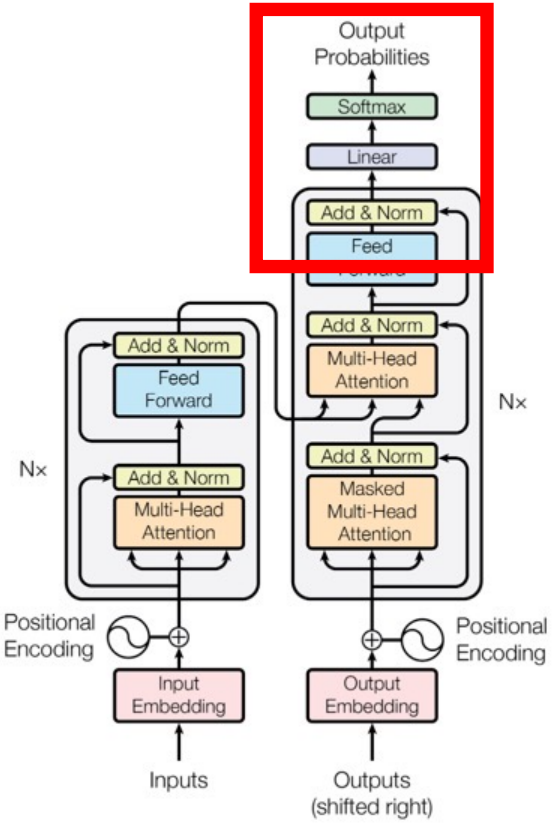
VU VRIJE UNIVERSITEIT AMSTERDAM

See also https://kikaben.com/transformers-positional-encoding/

Transformer: Output

- Like in a enc/dec architectures the output sequence is returned one at the time
- **For instance:**
- **Input:** (‘Hello’, ’World’, ‘!’)
- **Outputs:**
  1. (<SOS>,’Bonjour’)
  2. (<SOS>,’Bonjour’,’le’)
  3. (<SOS>,’Bonjour’,’le’,’monde’)
  4. …



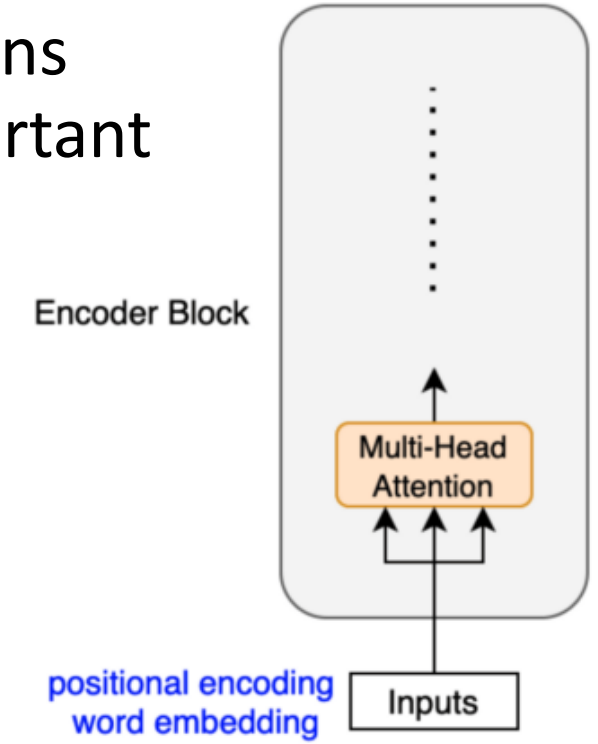


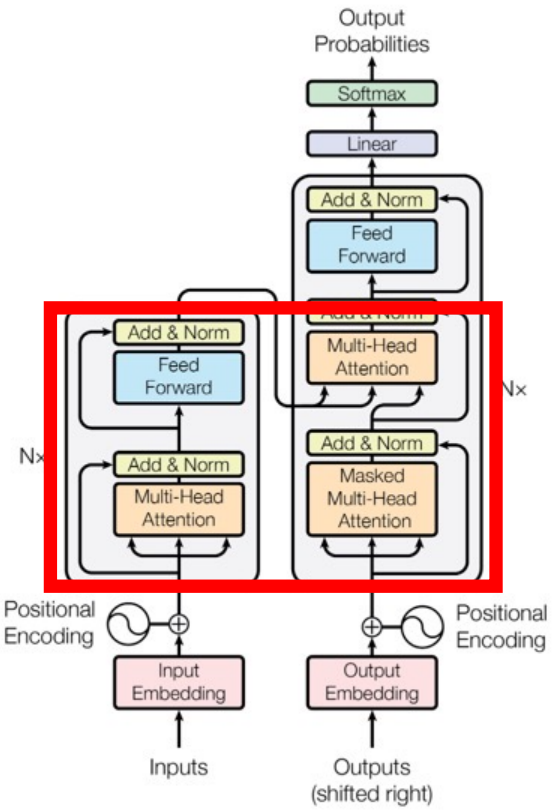
VU VRIJE UNIVERSITEIT AMSTERDAM

Attention is all you need

- Transformers replace recurrence with **self-attention**

In essence, the model learns to focus on the most important parts of the input





VU VRIJE UNIVERSITEIT AMSTERDAM

See also https://kikaben.com/transformers-self-attention/

Transformer: Impact

- Transformers are used to implement the current state-of-the-art for language modeling (BERT, GPT)

- They replaced earlier architectures like LSTM and are widely popular for NLP tasks

Final remarks

- The books below are excellent for start learning ML
  - **Pattern Recognition and Machine Learning (very good for basic of ML)**
  - **Artificial Intelligence – A Modern Approach (a more general overview)**
  - **Deep learning (a book that covers basic concepts in Deep Learning)**
  - **The Hundred-Page Machine Learning Book (not read but it was recommended to me)**

- Good library for implementing ML models is scikit-learn. Most DL methods are implemented either in Tensorflow or in PyTorch.
- Countless tutorials, guides, videos on ML and DL on the Web