Announcements

- Project Milestone 2 due **Wednesday, November 9 at 8pm**
  - Will open GradeScope submission tonight
  - **GPU option:** AWS SageMaker Studio
- Quiz 9 is due **Thursday, November 10 at 8pm**
- HW 5 due **Wednesday, November 16**
  - Please start early!

Word Embeddings, Ctd

https://tinyurl.com/cis5190-11-7-2022

Osbert Bastani and Zachary G. Ives

CIS 4190/5190 – Fall 2022

Recall: Similar Words Are Used in Similar Contexts

"I buy food for my **cat** at the pet store"

vs

"I buy food for my **dog** at the pet store"

vs

"My **car** guzzles gas"

Intuition: we can "semantically cluster" words based on vectors describing the contexts of their occurrences

Capturing Context in a Vector

Term-Frequency model:

- For each term, count # occurrences in each document in a corpus
- Vector is |terms| by |documents|

Wikipedia Article / Words	**Cat**	**Dog**	Apple Inc.	Apple (fruit)	Microsoft Inc.	…
a	377	370	842	231	286	…
the	929	787	1690	503	872	…
apple	0	0	1091	166	14	…

A "windows" term-term model:

- For each term, count # co-occurrences with other words *within an n-word window*
- Vector is |terms| by |terms| but sparse (*n* non-zero entries)

Words / Words	**pet**	**play**	tire	engine	run	…
dog	872	649	1	7	378	…
cat	789	831	5	0	285	…
car	12	4	290	927	562	…

These are huge vectors, likely with lots of zeros.

Can we get a more compact representation?

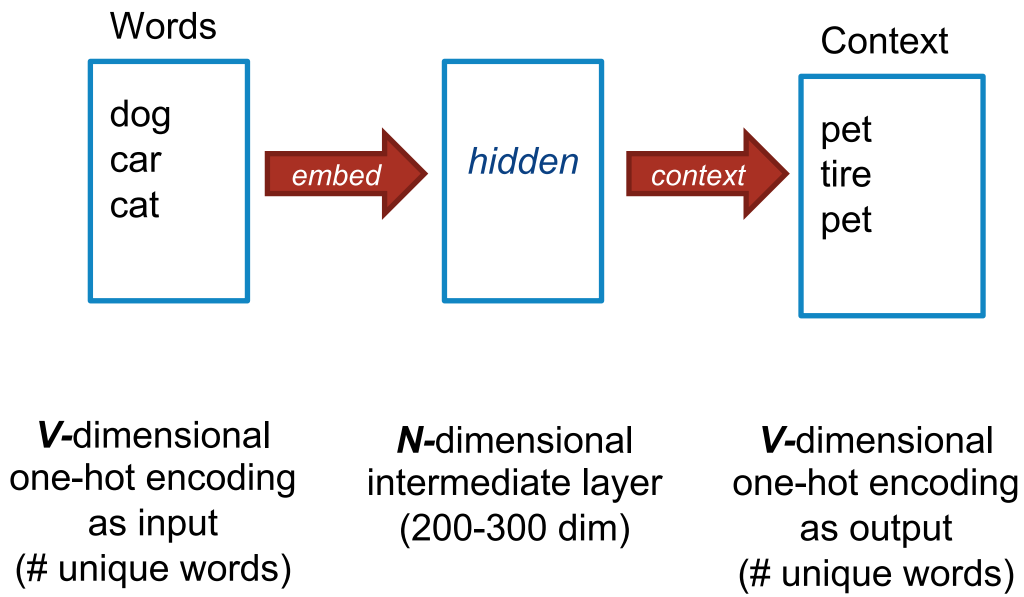
Intuition

Why not *learn* a way of mapping to a reduced number of dimensions?

We'll do this in a surprising (?) way:

- **Train a NN classifier** to predict words that will co-occur in the context by mapping them through a hidden layer with fewer dimensions
- Take the ***learned weights*** **as a compact vector space representation**!

Word2Vec Neural Network, Sketched



The hidden layer has a smaller number of dimensions – we'll learn N features useful in predicting context

Word2Vec Training Data

“The quick brown fox jumped over the lazy dog” (n=2)

Word	Context
quick	[the, brown]
brown	[quick, fox]
fox	[brown, jumped]
jumped	[fox, over]
…	…

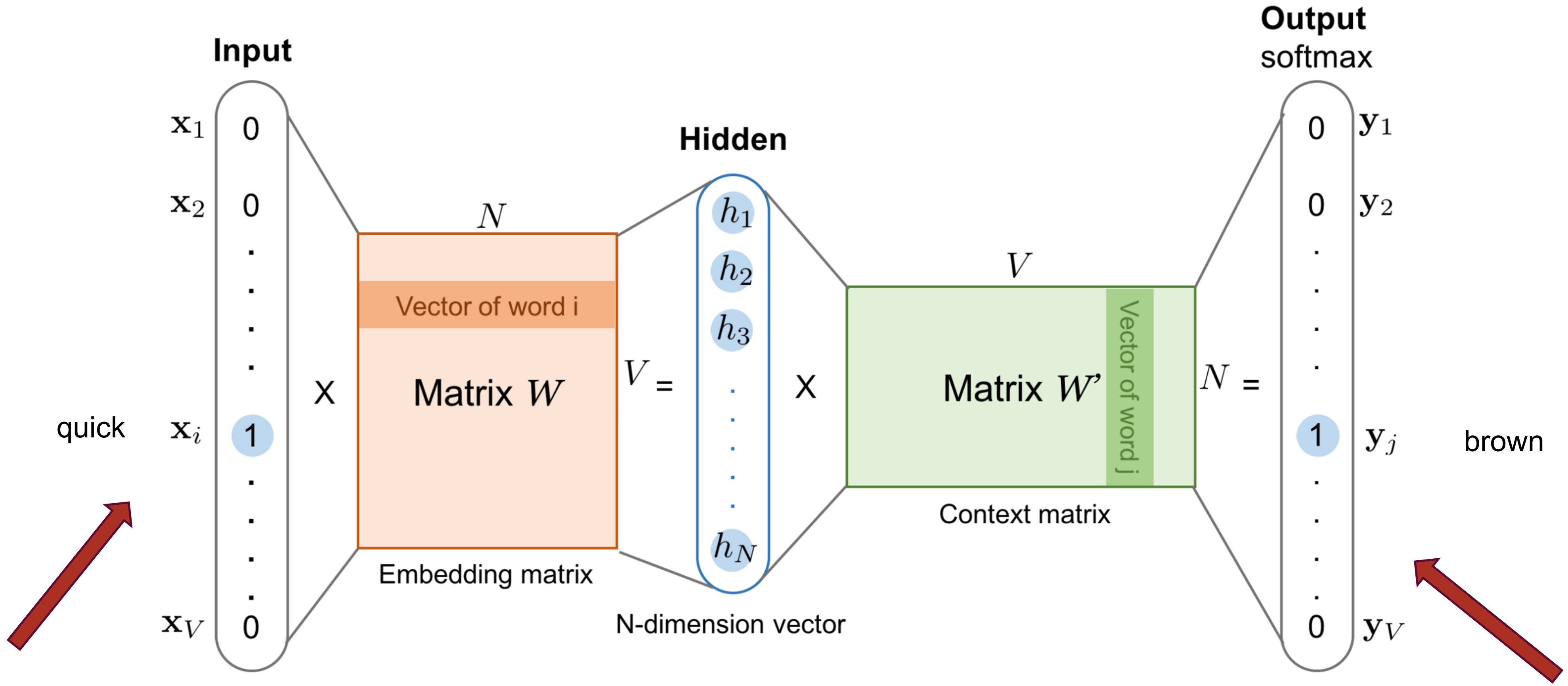
Word2Vec Training Data as Input/Output Pairs for Prediction

“The quick brown fox jumped over the lazy dog.”

Input	Output
quick	the
quick	brown
brown	quick
brown	fox
fox	brown
…	…

Millions of training input-output pairs, from parsing huge, unlabeled datasets (e.g., all of Wikipedia)

Word2Vec Classifier

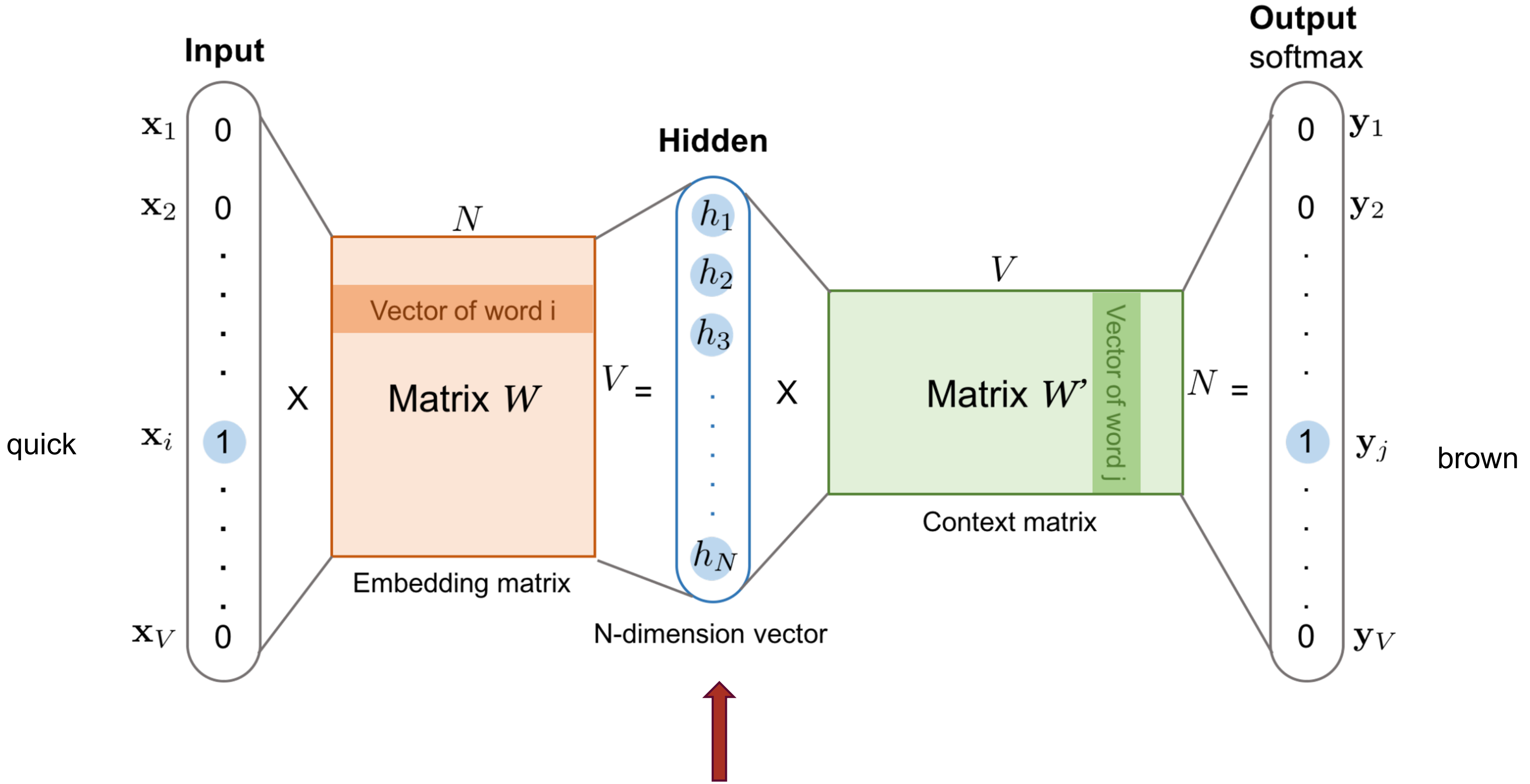


One-Hot Encoding
for
the Input Word

One-Hot Encoding
for
the Output Word

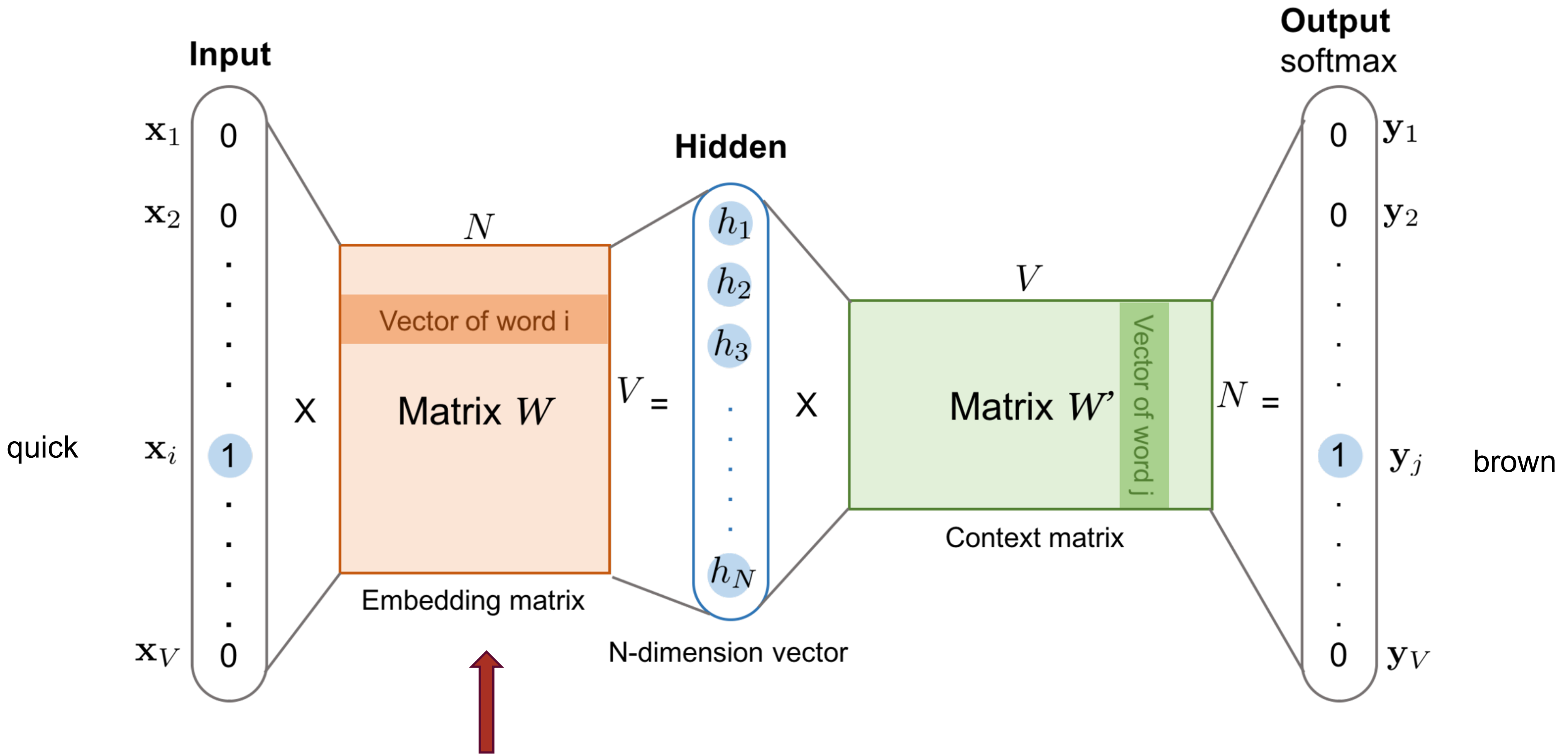
Source: https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

Word2Vec Classifier



N hidden units
where $N \ll V$

Word2Vec Classifier



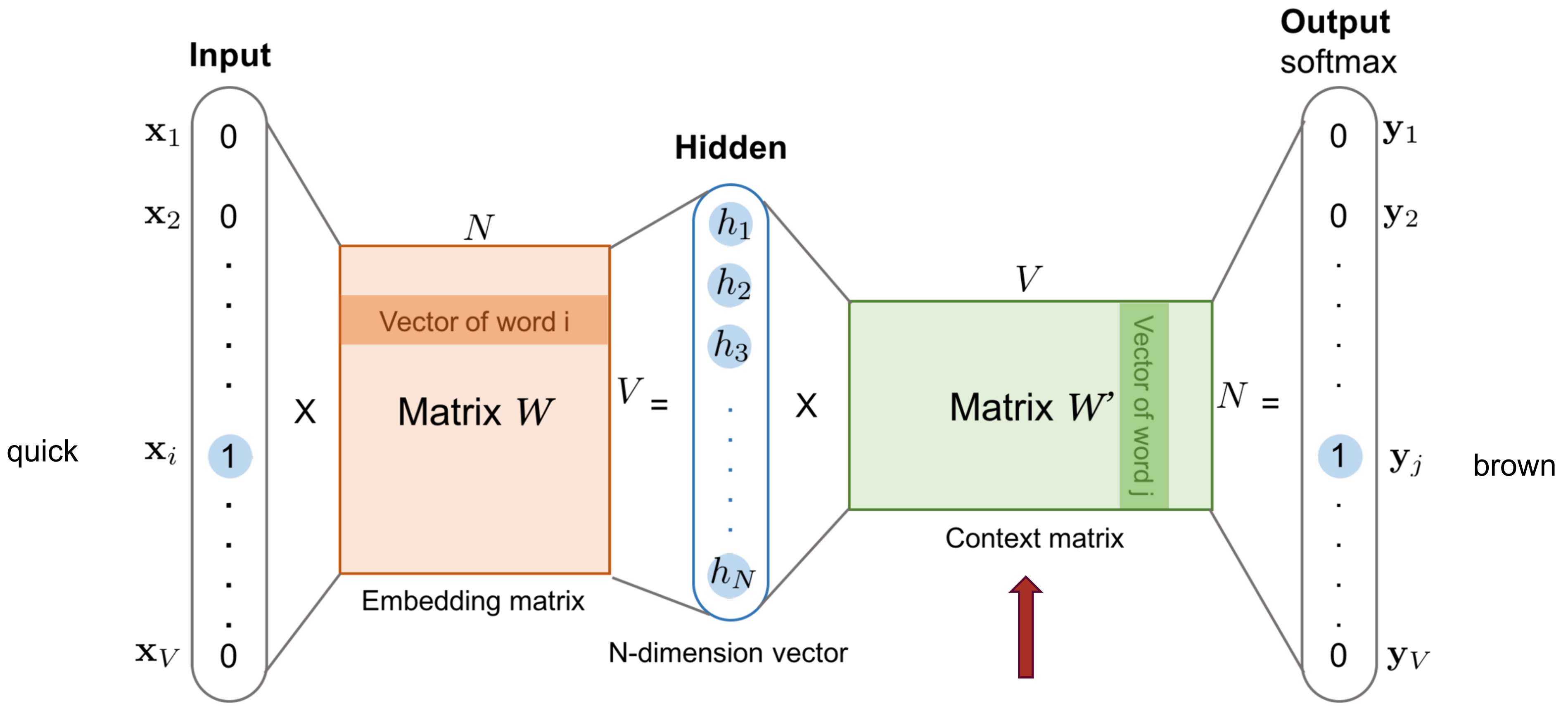
Has N columns, V (vocabulary size) rows.

Each row corresponds to a word.

i^{th} row = a vector representation for word #i

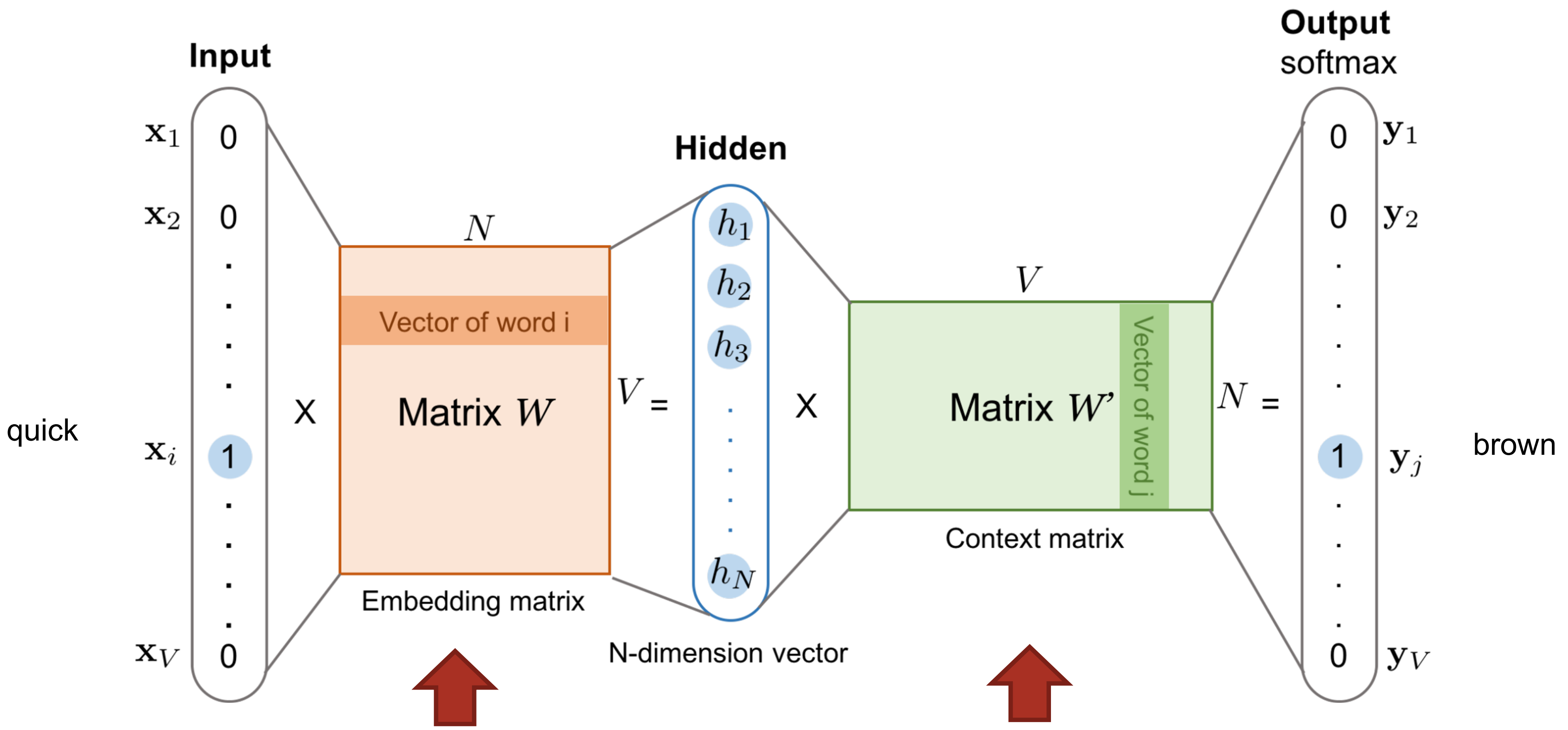
"Target Embedding"

Word2Vec Classifier



Has V (vocabulary size) columns, N rows.
Each column corresponds to a word.
i^{th} column = another vector representation for word #i
"Context Embedding"

Word2Vec Classifier

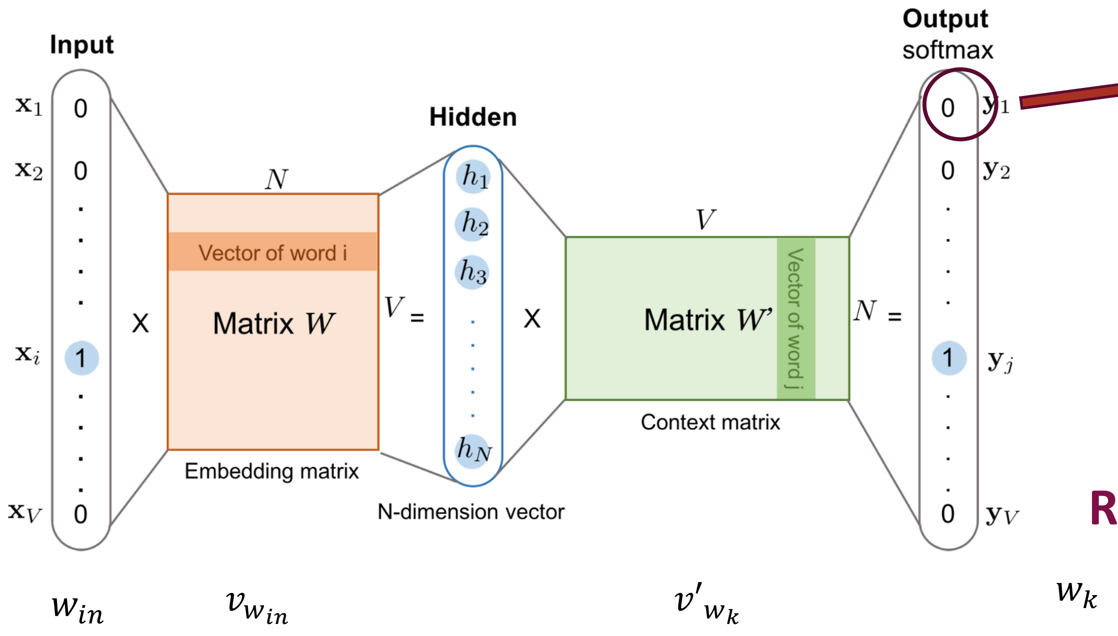


After training, we can make our final word vector a *concatenation* of the two embeddings, or just use *W*.

Word2Vec Training v1

Mikolov et al NeurIPS 2013

Standard softmax loss, then train the neural network.

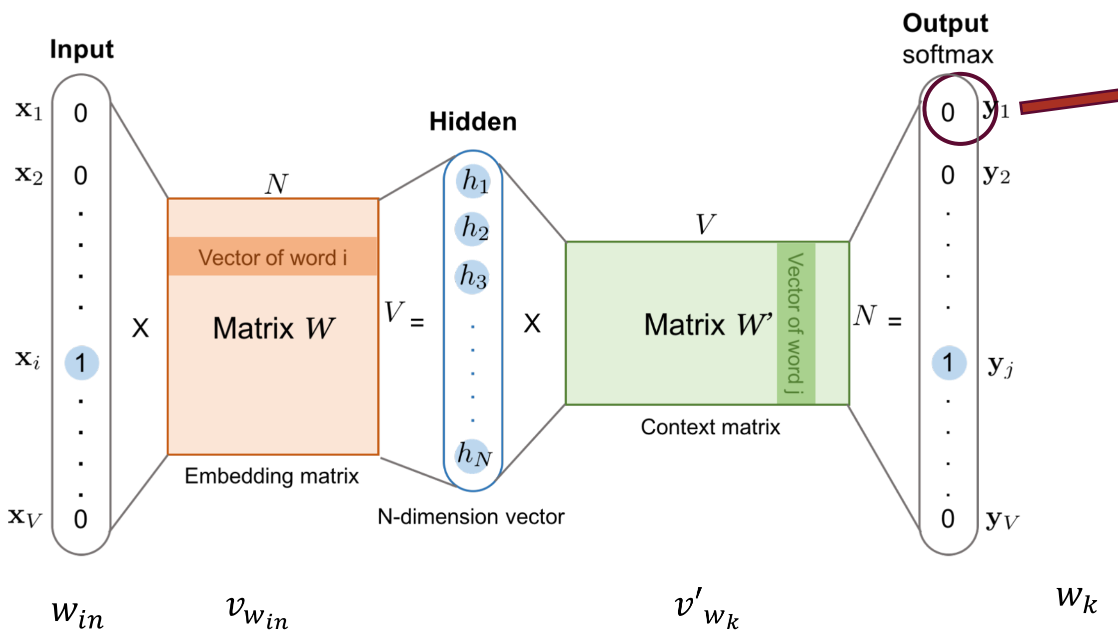


$$p(w_o|w_{in}) = \frac{\exp({v'_{w_o}}^T . v_{w_{in}})}{\sum_{k=1}^{V} \exp({v'_{w_k}}^T . v_{w_{in}})}$$

Computing this denominator will be expensive. Remember that the vocabulary size V is of the order of millions of words!

Scaling Word2Vec Training

Mikolov et al NeurIPS 2013

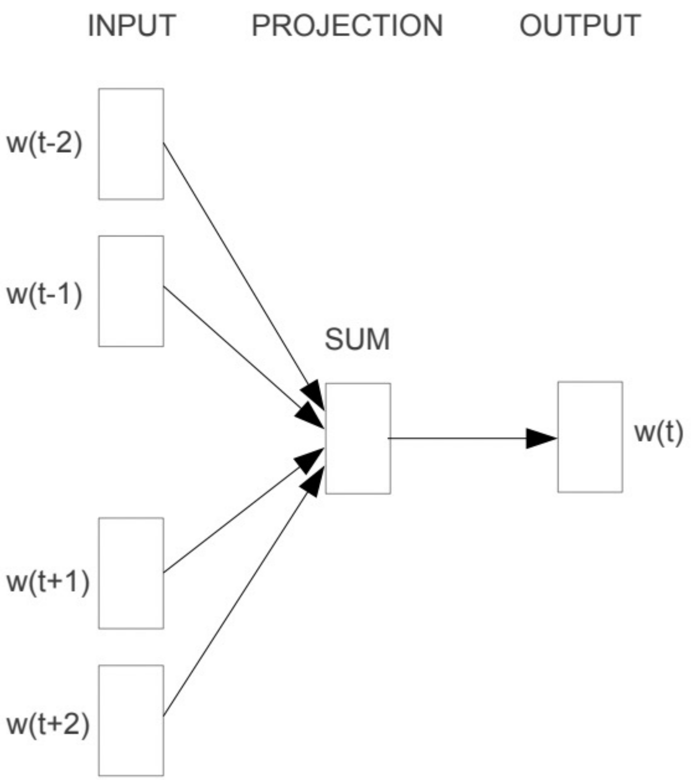


$$p(w_o|w_{in}) \approx \frac{\exp({v'_{w_o}}^T . v_{w_{in}})}{\sum_{k=1}^{K} \exp({v'_{w_k}}^T . v_{w_{in}})}$$

Simple Trick: Sample some random K-1<<V negative example words for each sample. e.g., K=2, 5, 20 etc. ["Negative sampling"]

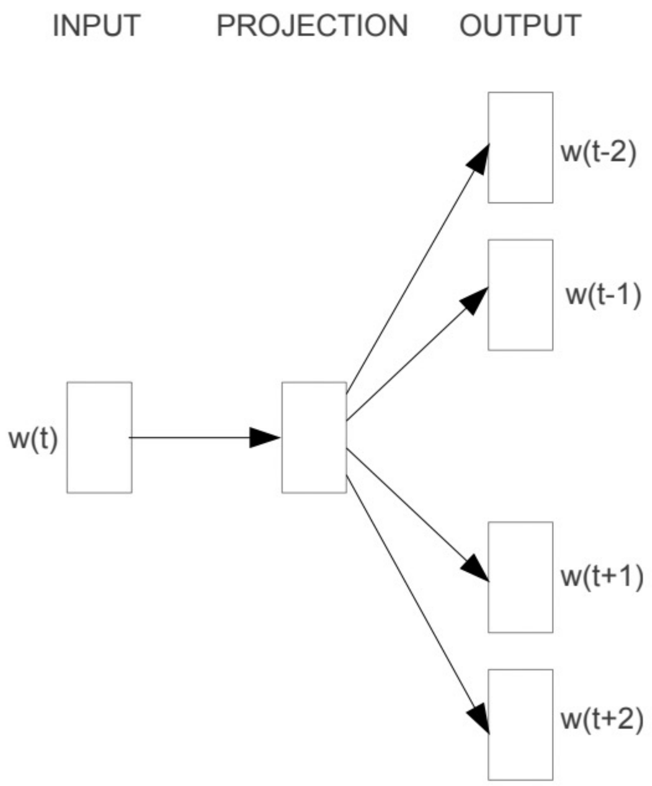
Also means we need to update many fewer weights during each iteration of gradient descent.

Using Word2Vec Predictions



CBOW

Predict word from bag-of-words context



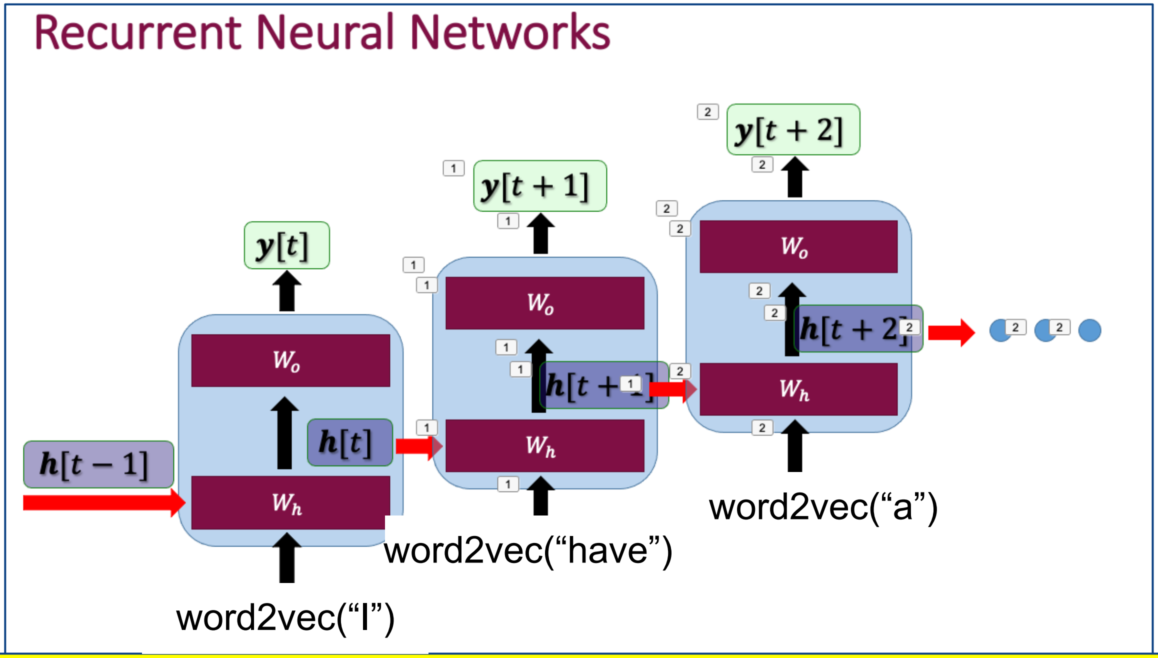
Skip-gram

Predict context from word

From Words to Documents

- Sentence2Vec, Paragraph2Vec scale these Word2Vec ideas to learn direct embeddings for sentences / paragraphs.
- However, much more common to treat as a sequence of words, and represent each word by its word2vec-style representation:

“j'ai un chien”



“I have a dog”

Simple “sequence-to-sequence” models like these produced huge advances in machine translation in 2014.

Properties of Word2Vec

Words that co-occur have vector representations that are close together (Euclidean distance).

"sofa" and "couch" (synonyms) will be close together, but also things like "hot" and "cold" (antonyms)

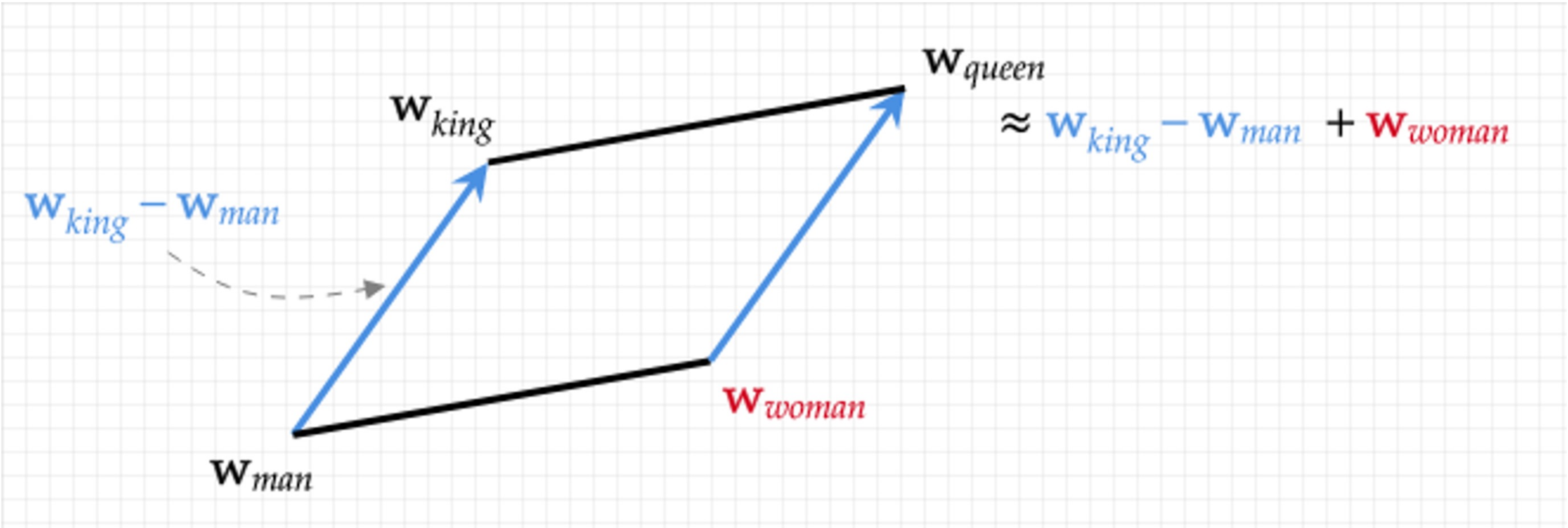
People say "It's ____ outside today" for both

- "hot" and "cold" co-occur with the same words often in sentences.

Properties of Word2Vec

Vector operations (vector addition and vector subtraction) on word vectors often capture the semantic relationships of their words.

man : king :: woman: ?



Source: https://www.ed.ac.uk/informatics/news-events/stories/2019/king-man-woman-queen-the-hidden-algebraic-struct

Use in Practice

GLoVe is an alternative word vector embedding similar to word2vec

Available freely, and often used off-the-shelf:
- English word2vec weights trained on Google News data
- GloVe vectors trained on the Common Crawl dataset and a Twitter dataset.

If you have a lot of training data or a very different / niche domain (e.g., medical text), you might want to train your own word vectors on your dataset!

Summary So Far

- The journey to compact vector representations of words by their context:
  - term frequency vectors
  - term-term vectors
  - word embeddings (learned by NN)
- We can use measures of vector similarity (e.g., cosine similarity, L1 or L2 distance, others) to find related terms

Words in Context

- While word2vec is trained based on context, after training, it is applied independently to each word
  - E.g., train linear regression of sum of word vectors, or n-grams
- **Why is this problematic?**
  - “He ate a tasty apple”
  - “He wrote his essay on his Apple computer”
- Both use the same embedding!

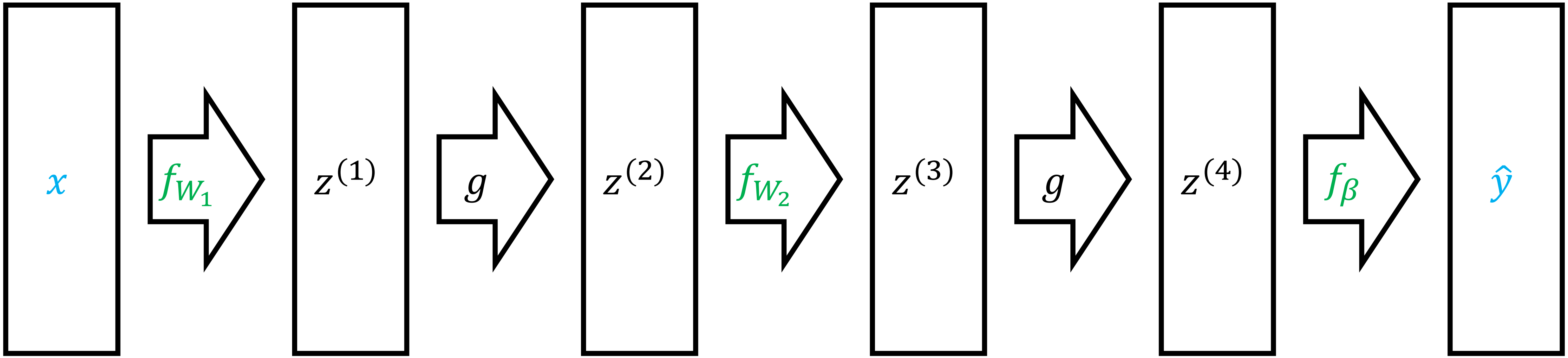
Recurrent Neural Networks

- Handle inputs/outputs that are **sequences**

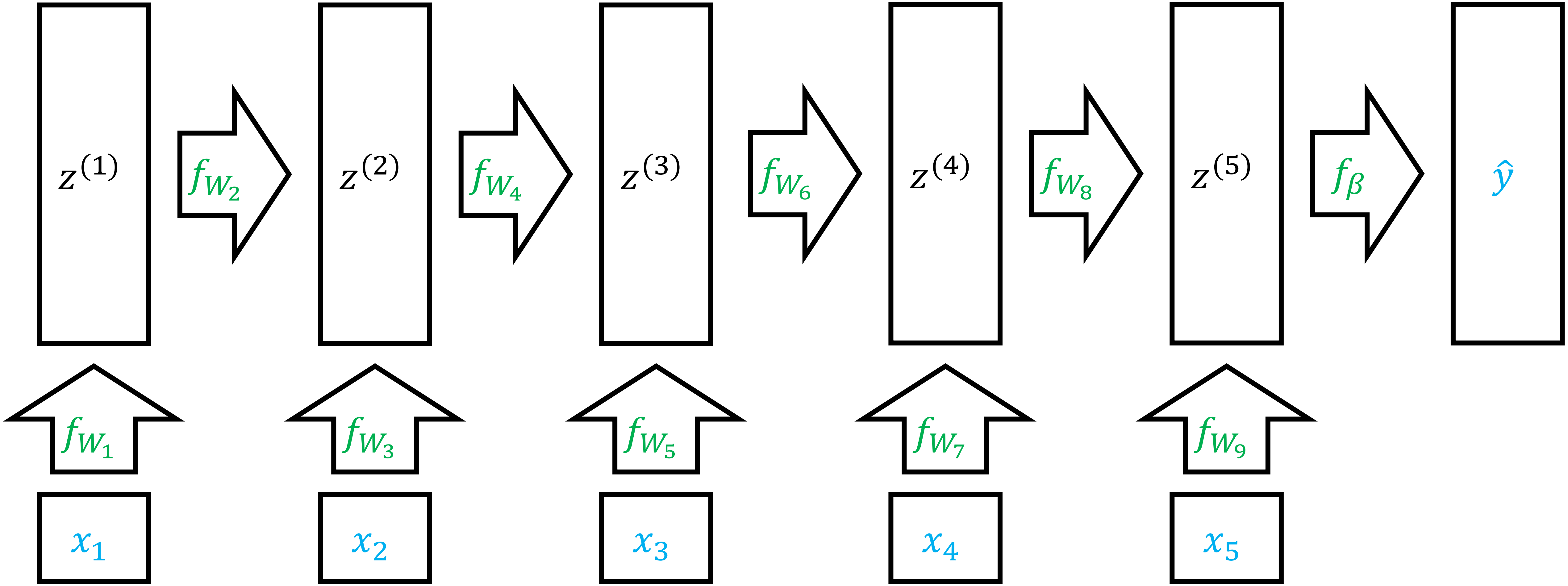
- **Naïve strategy**
  - Pad inputs to fixed length and use feedforward network
  - **Ignores temporal structure**

- **Recurrent neural networks (RNNs):** Process input sequentially

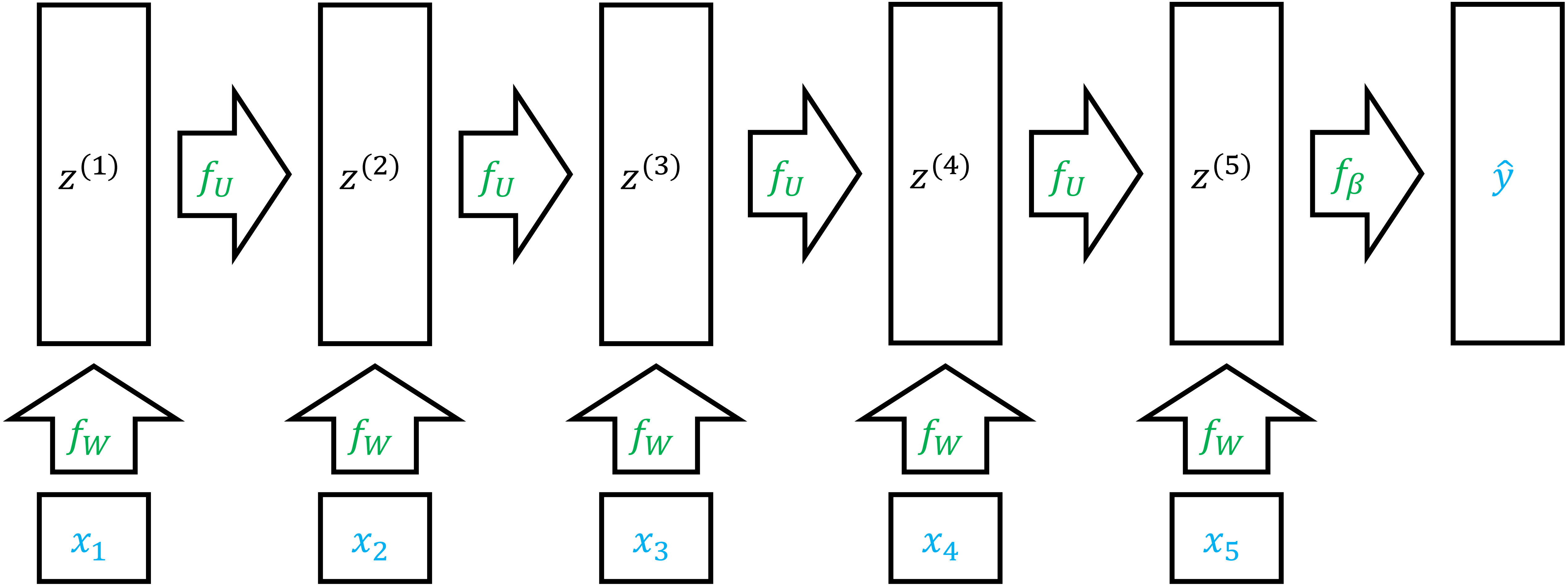
Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks

- Initialize $z^{(0)} = \vec{0}$
- Iteratively compute (for $t \in \{1, \dots, \mathrm{T}\}$):

$$z^{(t)} = g\left(Wx_t + Uz^{(t-1)}\right)$$

- Compute output:

$$y = \beta^\top z^{(T)}$$

Recurrent Neural Networks

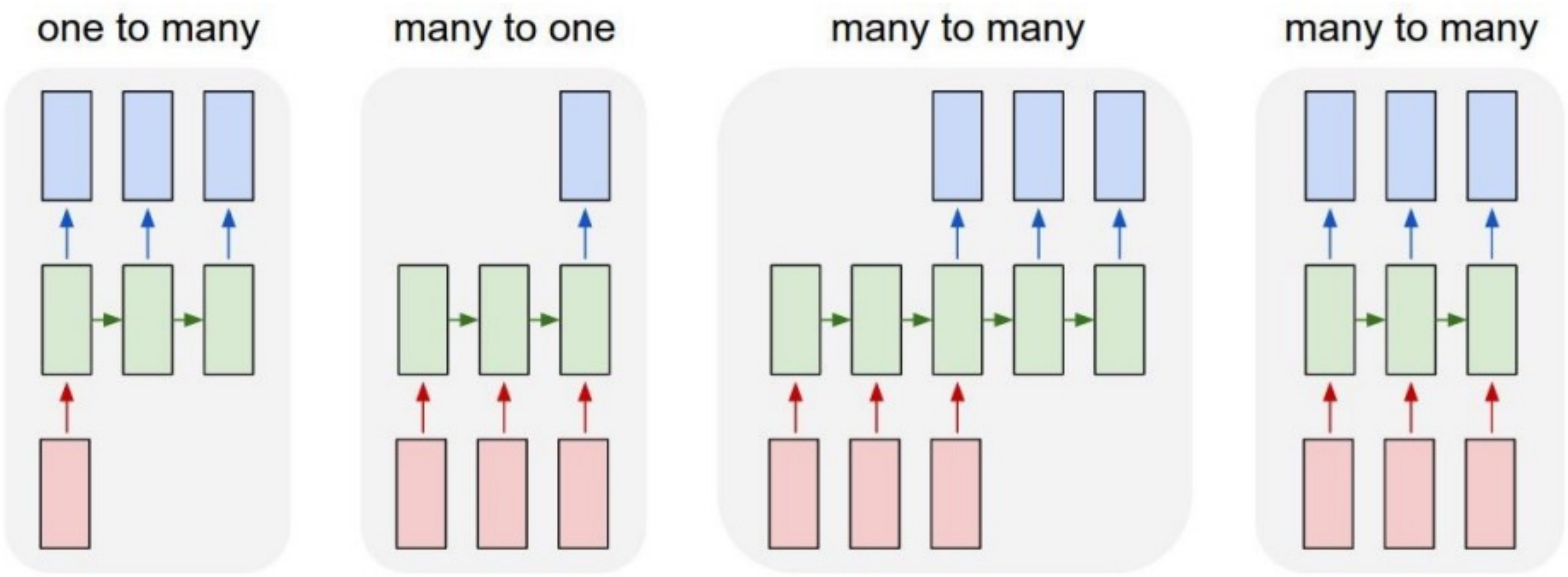


Image captioning | Sentiment prediction | Machine translation | Video captioning

Recurrent Neural Networks

- Backpropagation works as before
  - For shared parameters, overall gradient is sum of gradient at each usage

- Exploding/vanishing gradients can be particularly problematic

- LSTM (“long short-term memory”) and GRU (“gated recurrent unit”) do clever things to better maintain hidden state

RNNs for Natural Language

- Apply RNN to sequence of words
  - **Encoding 1:** One-hot encoding of each word
  - **Encoding 2:** Sequence of word vectors
- **Unsupervised pretraining**
  - Train on dataset of text to predict next word (classification problem)
  - $x = w_1 w_2 \ldots w_t$ and $y = w_{t+1}$ (usually y is one-hot even if x is not)
- Finetune pretrained RNN on downstream task

“Transfer Learning” Strategy

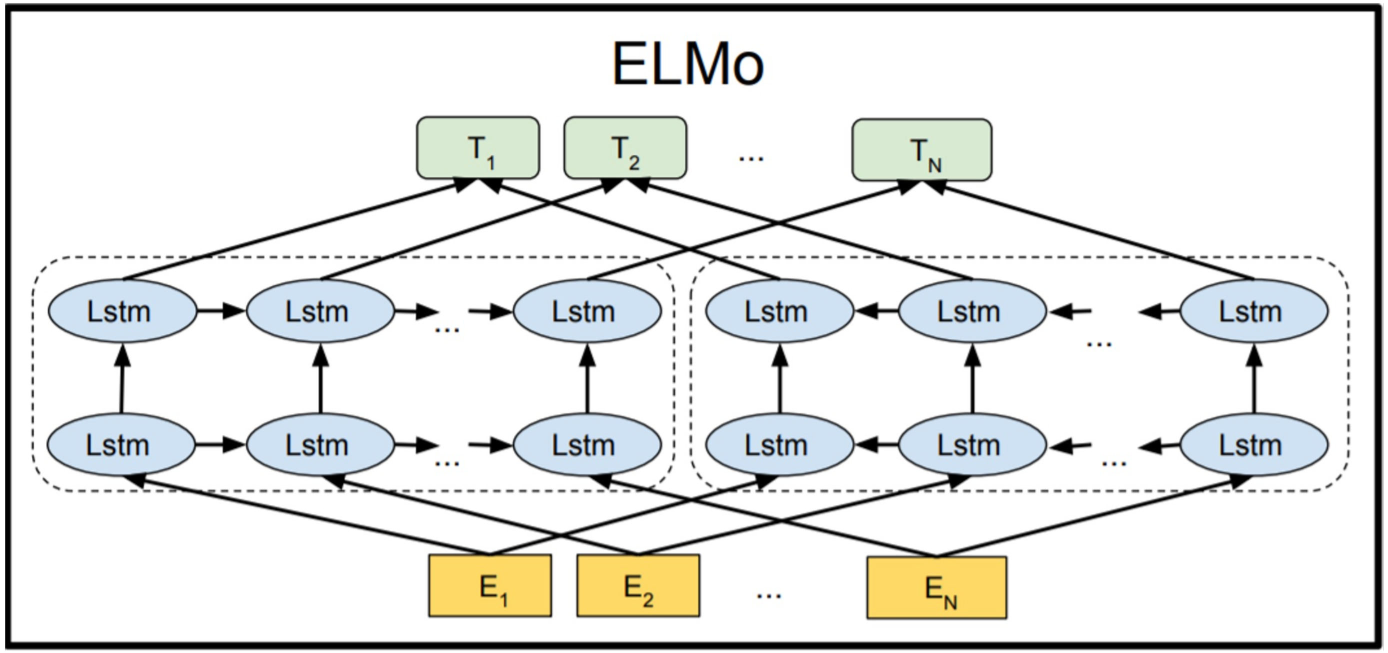
- **Step 0:** Pretrained on a large **unlabeled** text dataset
  - Also called “self-supervised”
  - Trained using supervised learning, but labels are predicting data itself
- **Step 1:** Replace next-word prediction layer with new layer for task
- **Step 2:** Train new layer or finetune end-to-end

RNNs for Natural Language

- **Shortcomings**
  - Unidirectional information flow (must remember everything relevant)
  - Computation time is proportional to sequence length
- **Improvements/alternatives**
  - Bidirectional LSTMs
  - CNNs
  - Transformers

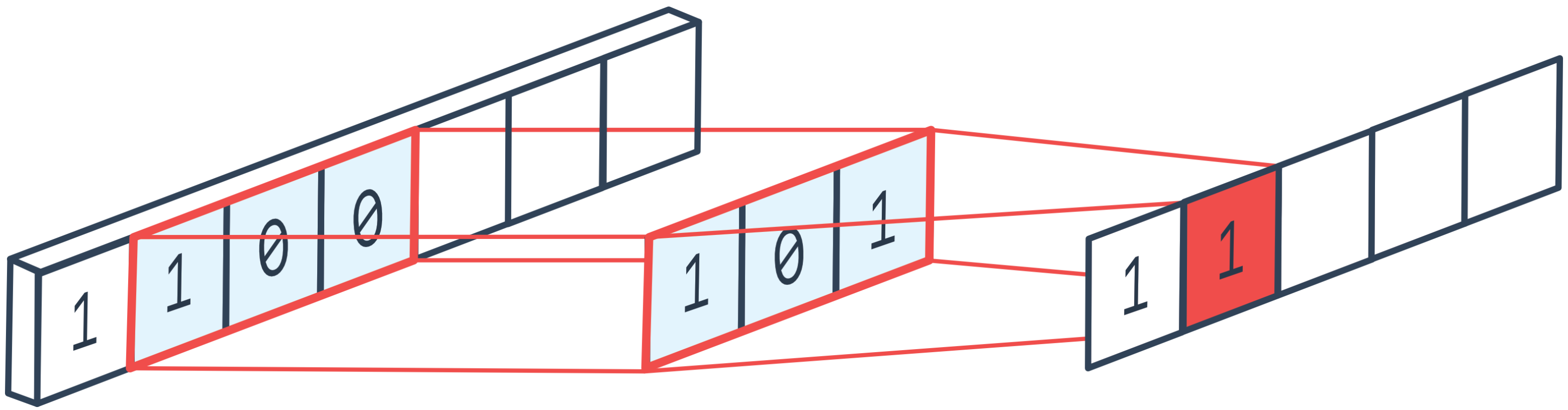
ELMo Model

- **Bidirectional LSTM:** Combine one LSTM to predict next word given previous words, another to predict previous word given later words



CNNs

- **Model**
  - 1D convolutional layers
  - Input is word embedding sequence
  - # channels is word embedding dimension



CNNs

- **Shortcomings**
  - Hard to reason about interactions between words that are far apart

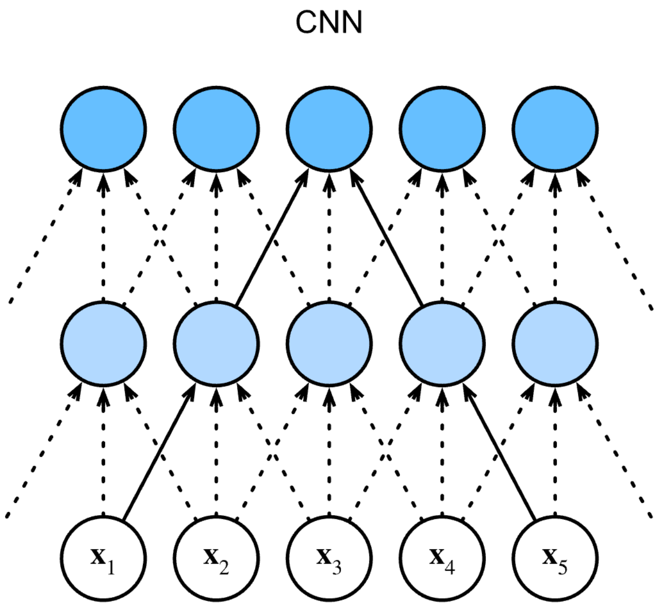


Figure credit to d2l.ai

Transformers

- Composition of **self-attention layers**

- **Intuition**
  - Want sparse connection structure of CNNs, but with different structure
  - Can we **learn** the connection structure?

Self-Attention Layer

- **Self-attention layer:**

$$y[t] = \sum_{s=1}^{T} \text{attention}(x[s], x[t]) \cdot f(x[s])$$

  - Input first processed by local layer f
  - All inputs can affect $y[t]$
  - But weighted by $\text{attention}(x[s], x[t])$

- Resembles convolution but connection is learned instead of hardcoded

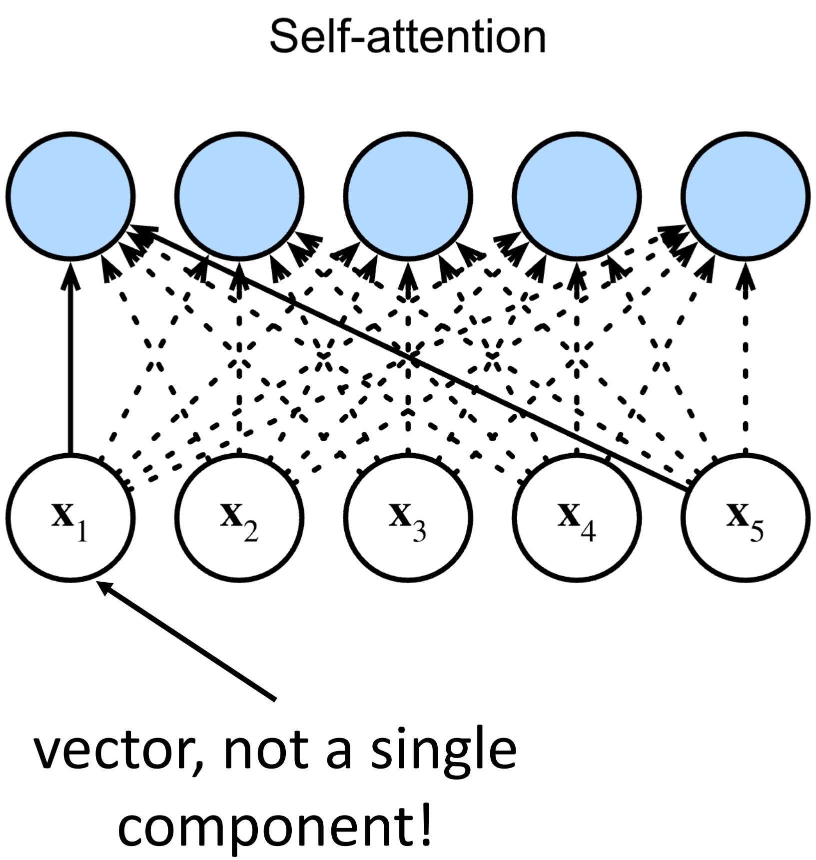


Figure credit to d2l.ai

Self-Attention Layer

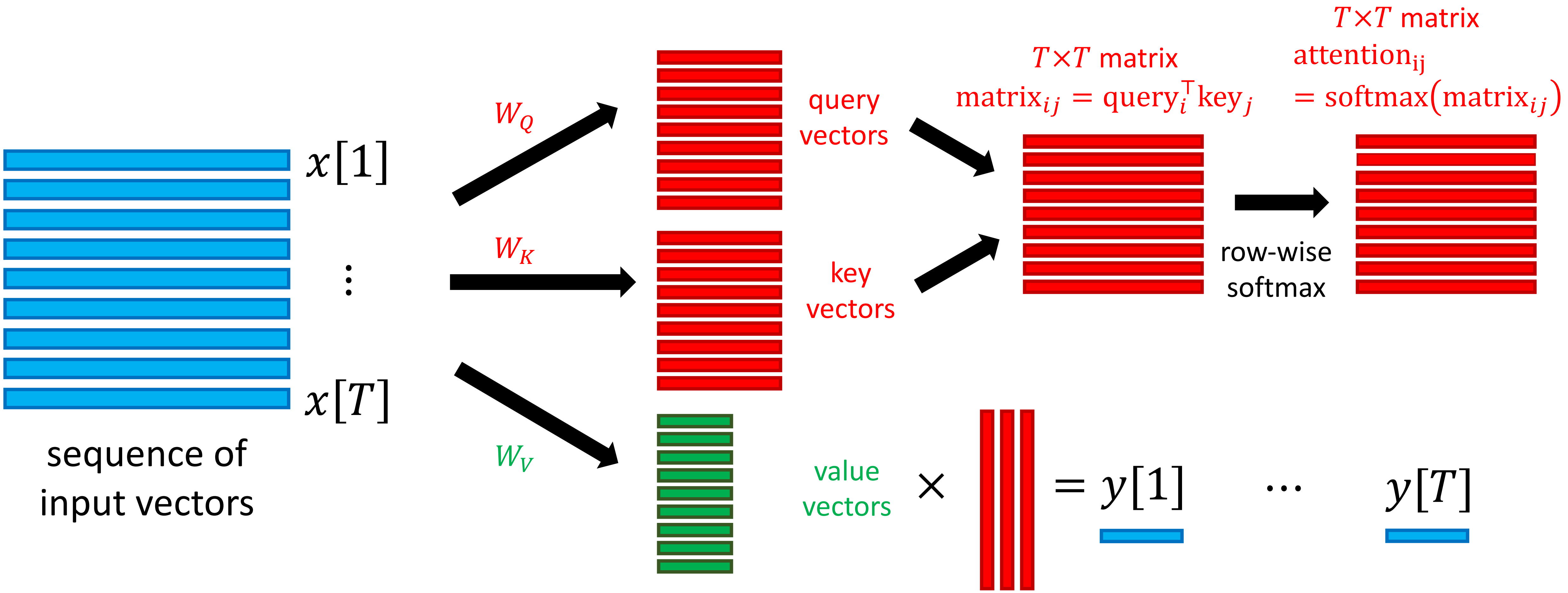
- **Self-attention layer:**

$$y[t] = \sum_{s=1}^{T} \mathrm{softmax}([\mathrm{query}(x[t])^\top \mathrm{key}(x[s])]) \cdot \mathrm{value}(x[s])$$

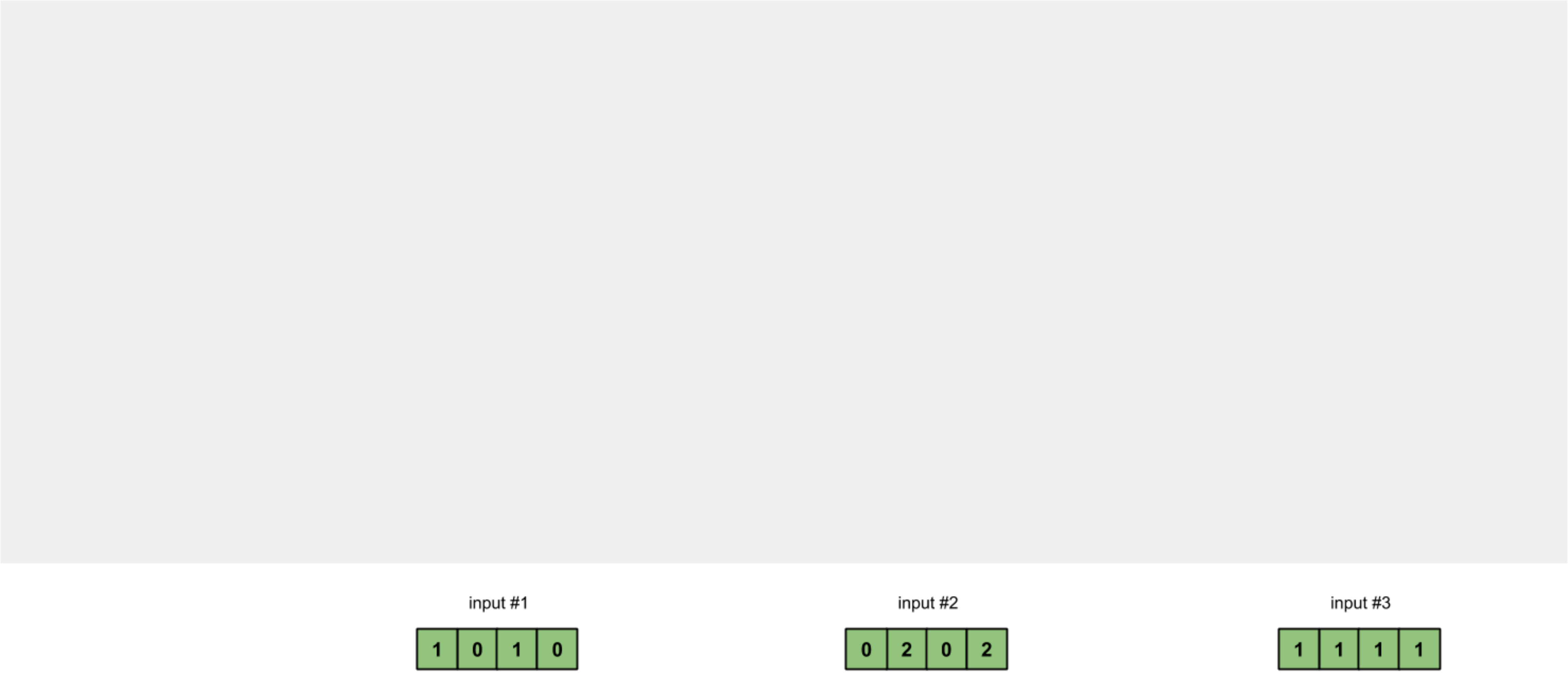
- Here, we have (learnable parameters are W_Q, W_K, and W_V):

$$\mathrm{query}(x[s]) = W_Q x[s]$$
$$\mathrm{key}(x[s]) = W_K x[s]$$
$$\mathrm{value}(x[s]) = W_V x[s]$$

Self-Attention Layer



Self-attention



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

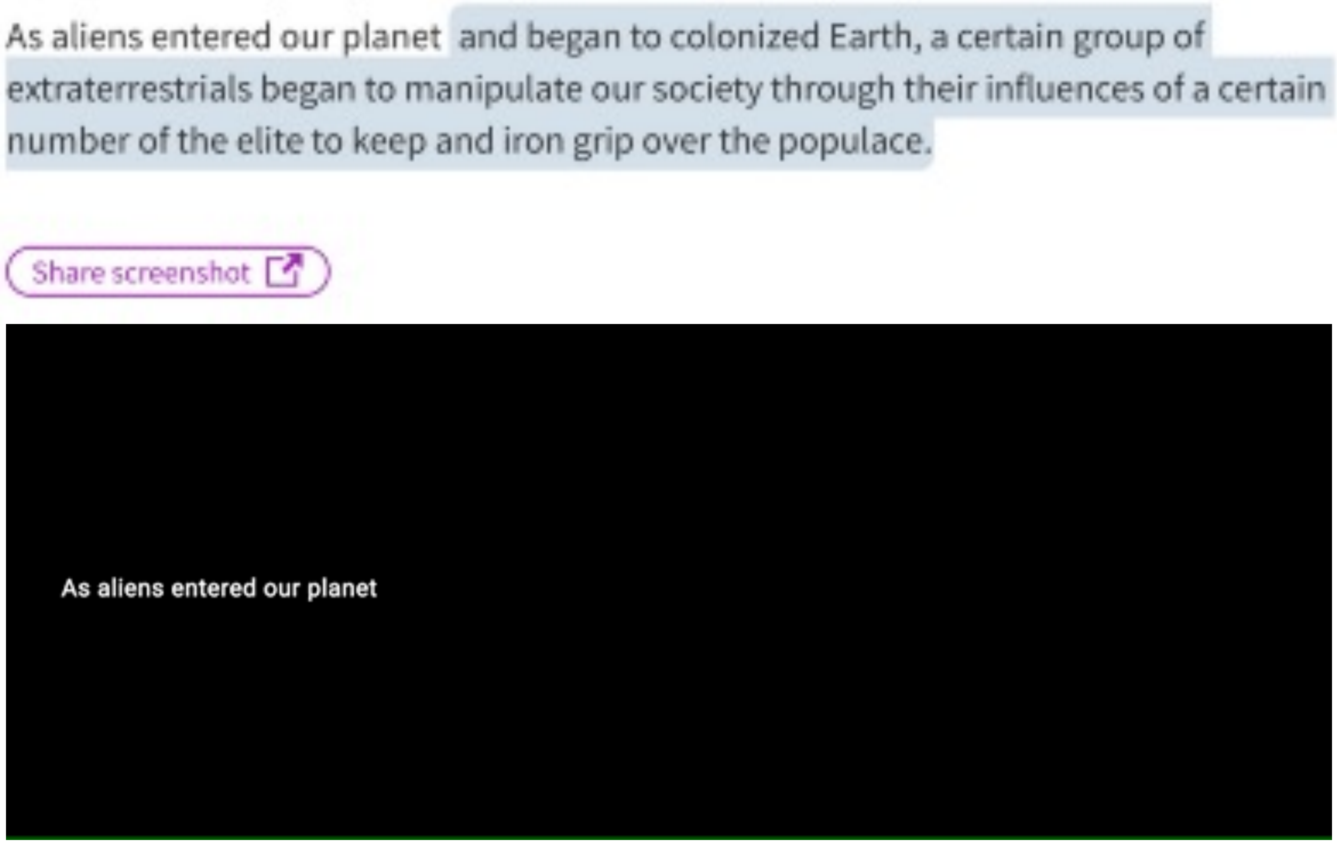
Transformers

- Stack self-attention layers to form a neural network architecture

- **Examples:**
  - **BERT:** Bidirectional transformer similar to ELMo, useful for prediction
  - **GPT:** Unidirecitonal model suited to text generation

- **Aside:** Self-attention layers subsume convolutional layers
  - Use "positional encodings" as auxiliary input so each input knows its position
  - https://d2l.ai/chapter_attention-mechanisms/self-attention-and-positional-encoding.html#
  - Then, the attention mechanism can learn convolutional connection structure

Visualizing Attention Outputs



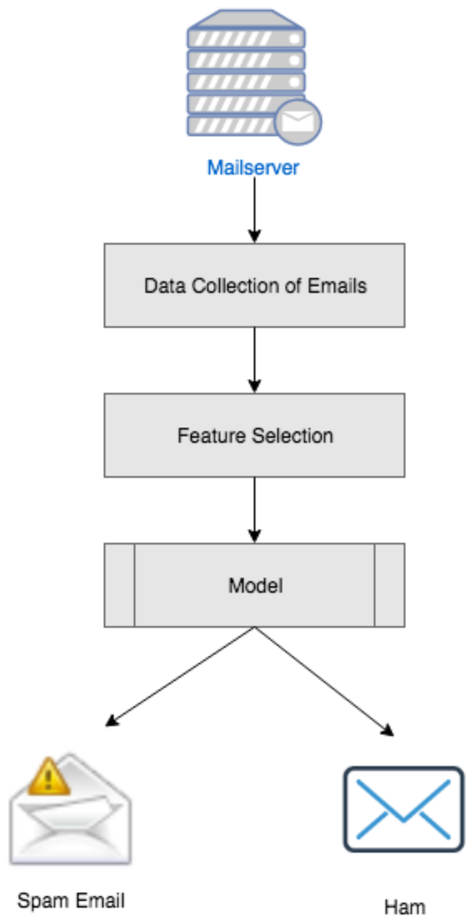
https://transformer.huggingface.co/

https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0

Applications: Spam Detection

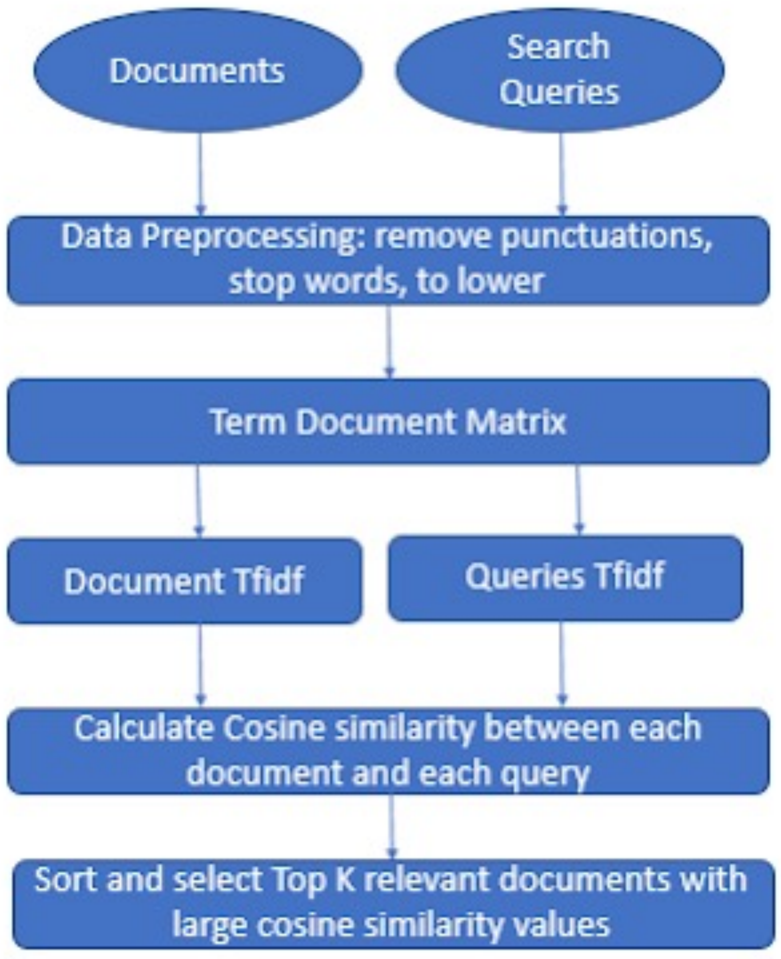
- “Bag of words” + SVMs for spam classification

- **Features:** Words like “western union”, “wire transfer”, “bank” are suggestive of spam



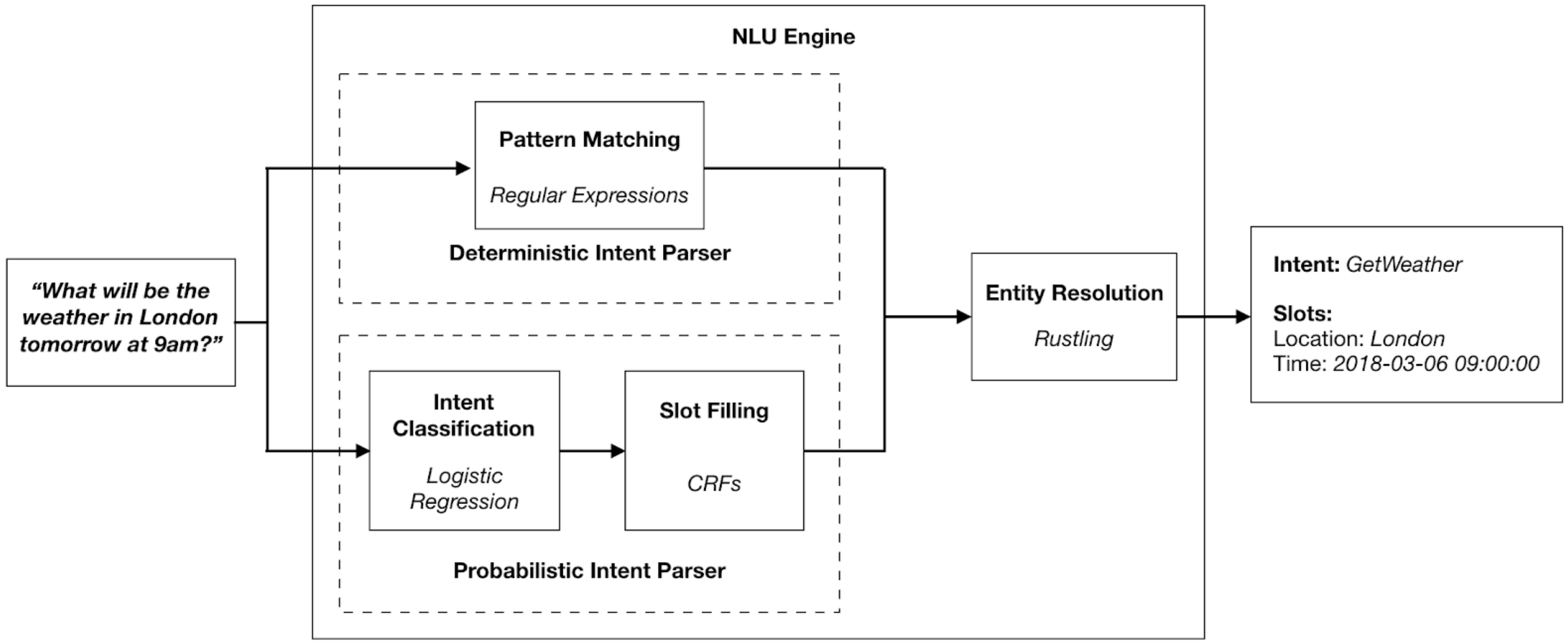
Applications: Search

- Use “bag of words” + TF-IDF to identify relevant documents for a search query



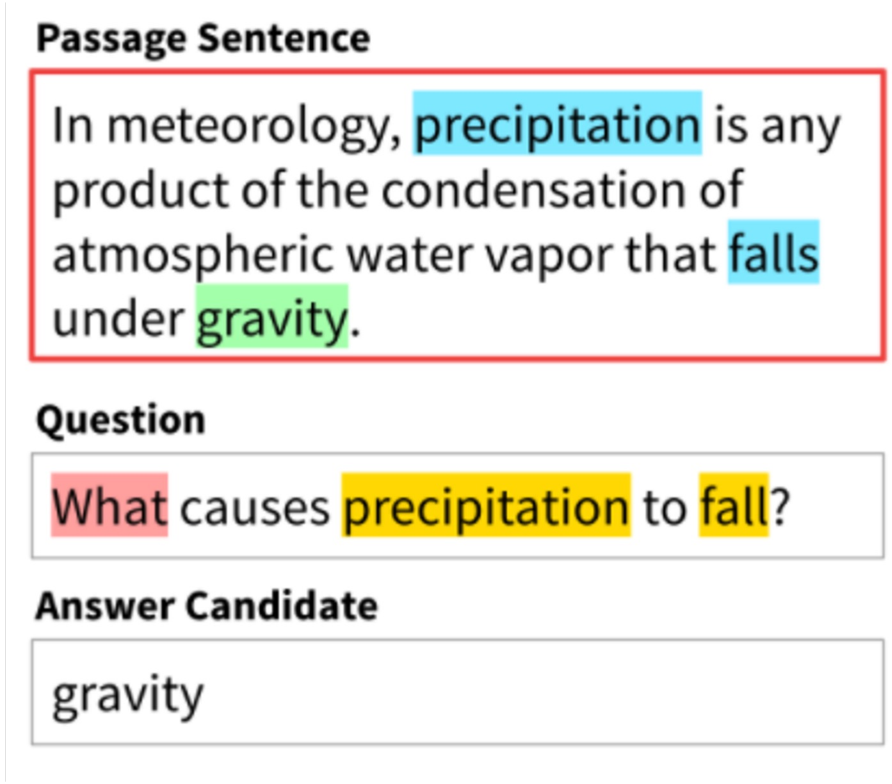
Applications: Virtual Assistants

- Use word vectors to predict intent of queries users ask



Applications: Question Answering

- Models like ELMo and BERT can be used to answer questions based on a given passage



Applications: Generation

- Language models such as GPT can automatically generate text for applications such as video games

AI DUNGEON

AI Dungeon, an infinitely generated text adventure powered by deep learning.

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Transformers for Computer Vision

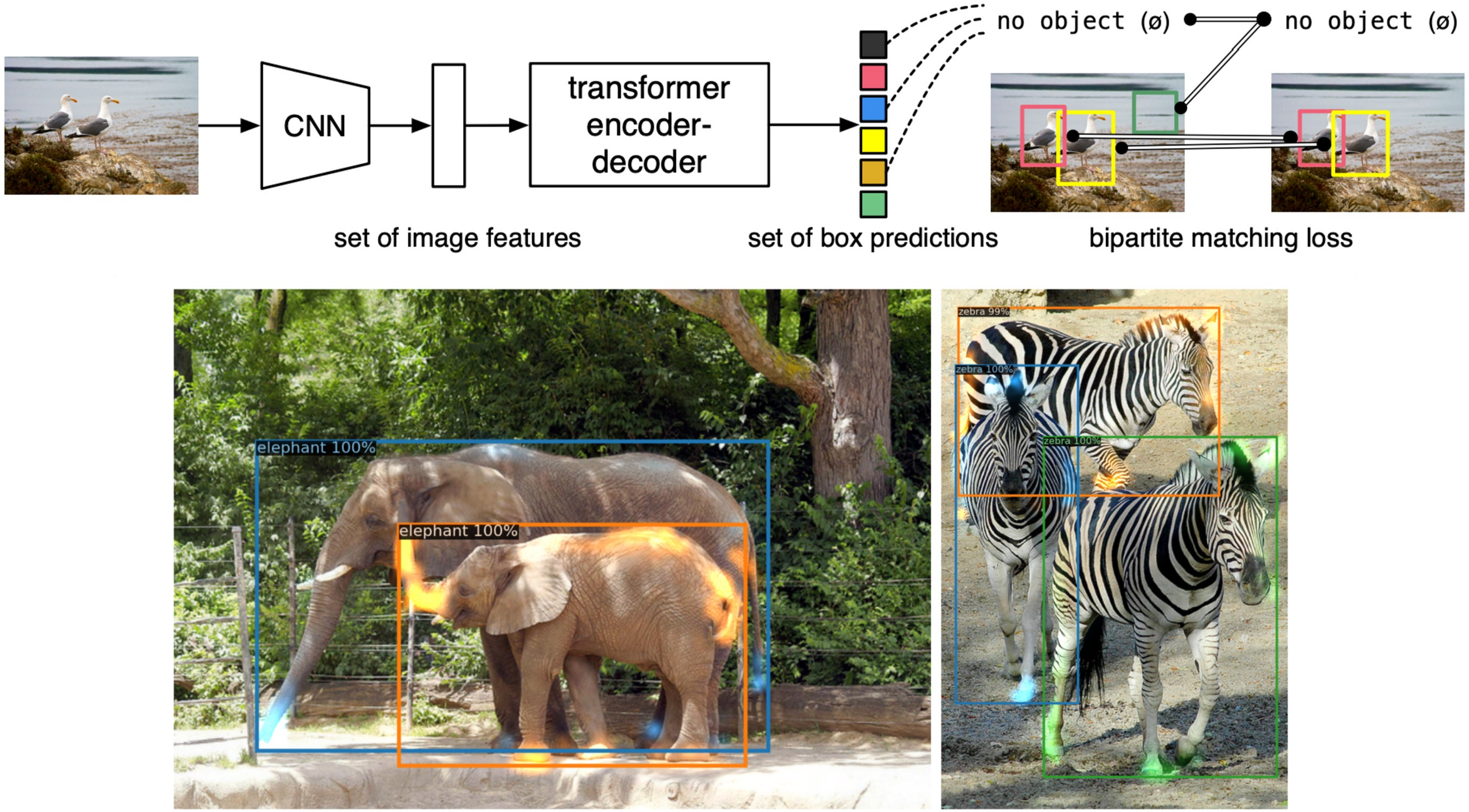


Figure credit to "End-to-End Object Detection with Transformers"