Machine Learning to Language Model

Topic 02 - Word Embedding

Jaihua Yen
https://jaihuayen.github.io/homeweb/

Contents

Review Bi-Gram Model

Bi-Gram Model

A Very Simple Language Model

- **Intuition:** We only predict the next word based on the previous word.
- Model the predicted probability of a certain word based on a given word.

$$P(c_i \mid c_{i-1})$$

c_i is the word in the i position.

- Here we give an example of a sentence:

$\boxed{\text{<s>}}$ AI could finally be introduced into practice in general tasks $\boxed{\text{<e>}}$

Start Token

End Token

Bi-Gram Model

A Very Simple Language Model

<s> AI could finally be introduced into practice in general tasks <e>

$$c_{i-1} \qquad\qquad c_i$$

<s> AI

Bi-Gram Model

A Very Simple Language Model

<s> $\boxed{\text{AI could}}$ finally be introduced into practice in general tasks <e>

c_{i-1} c_i

AI could

Bi-Gram Model

A Very Simple Language Model

<s> AI could finally be introduced into practice in general $\boxed{\text{tasks <e>}}$

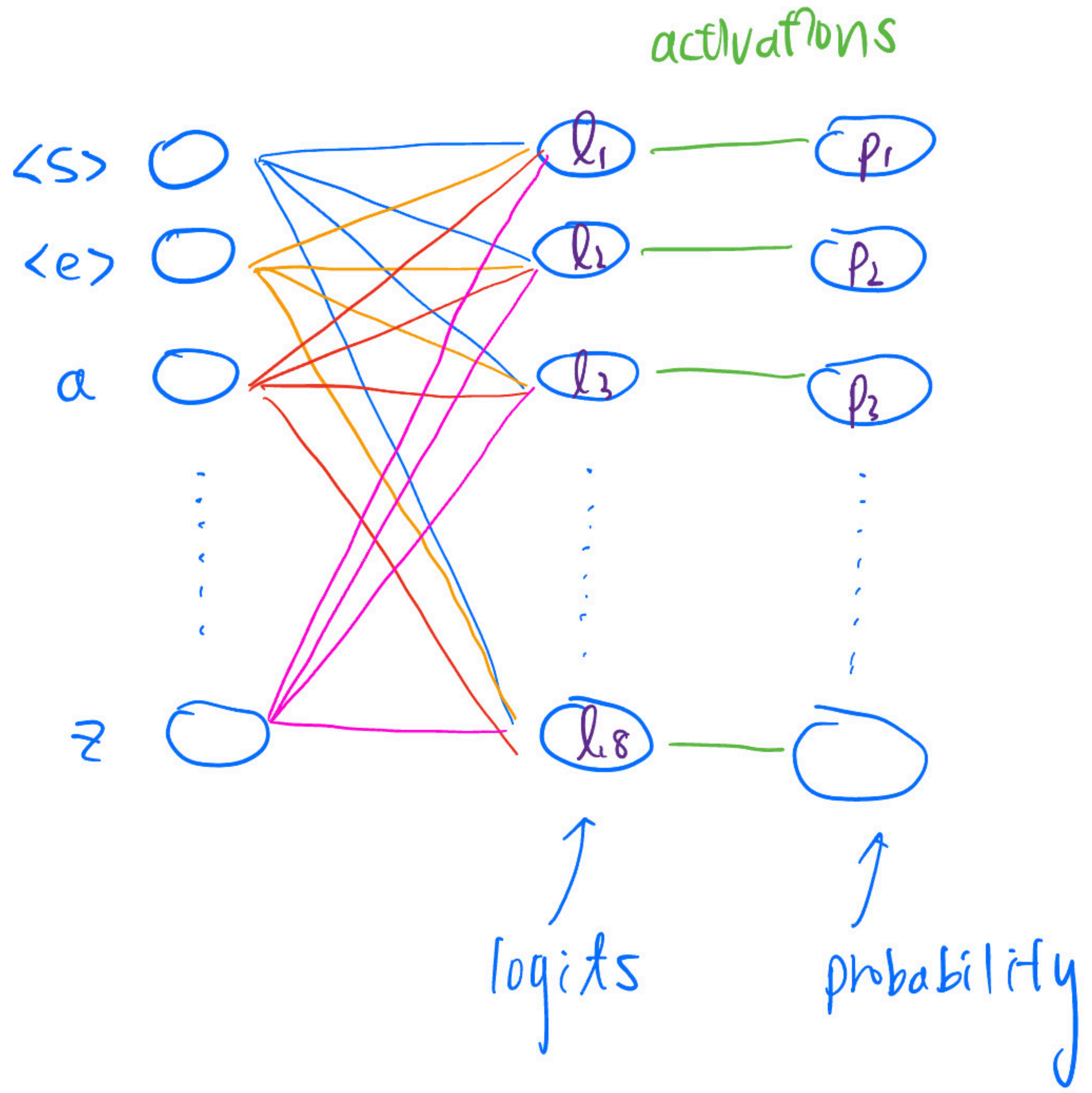
c_{i-1} c_i

tasks <e>

Bi-Gram Model

A Very Simple Language Model

- What if the features cannot be extracted only by one layer of neural network?

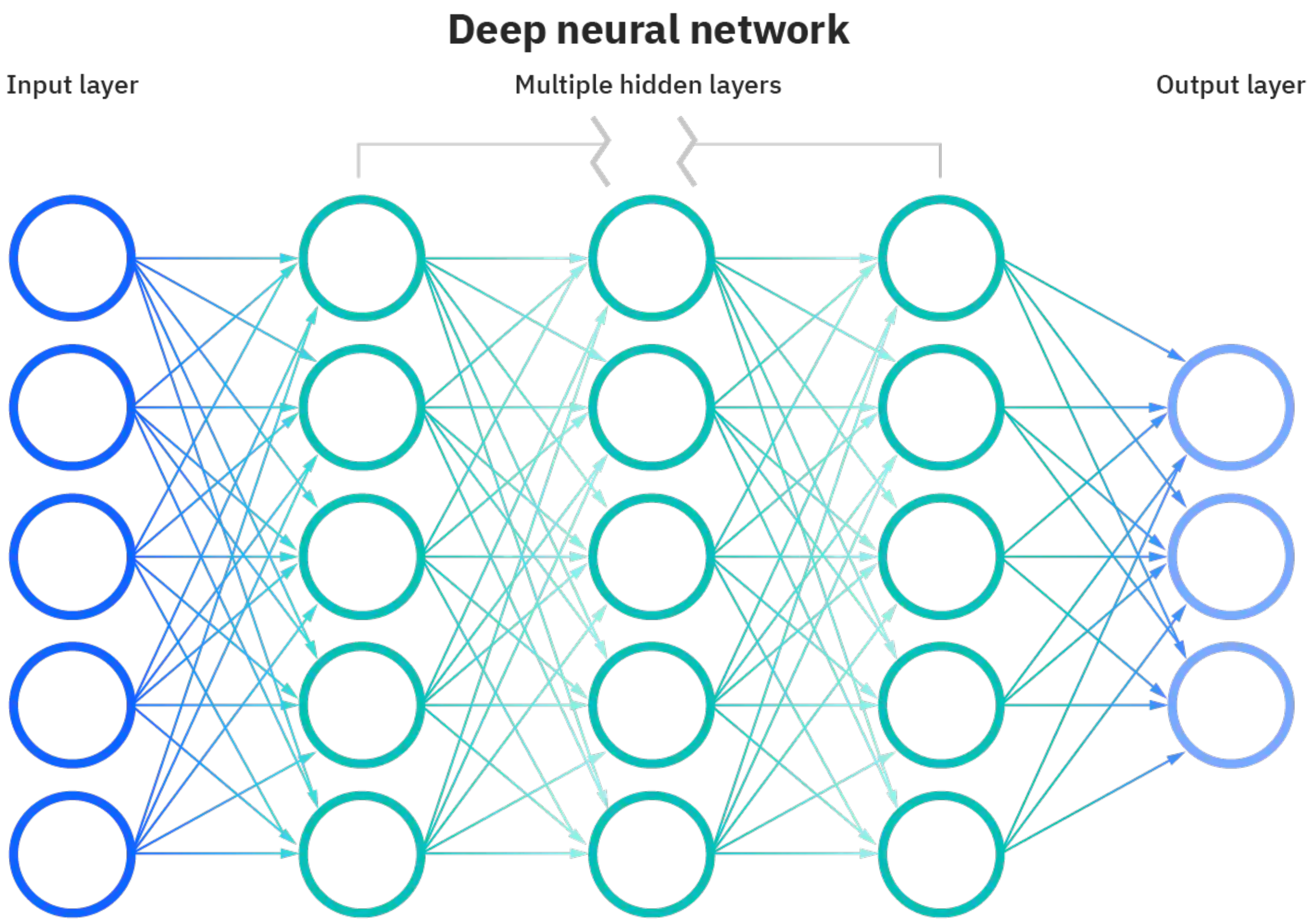


$$p_i = \frac{e^{l_i}}{\sum_{i=1}^{28} e^{l_i}}$$

Deep Neural Network

What Deeper?

- Deep Neural Networks (DNN) extract text semantics meanings on a deeper level.



https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

Word Embedding

Word Embedding

Word Representation

- From all the experiments above, we all use one-hot encodings.

$$v_{dog} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad v_{cat} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Word Embedding

Word Representation

- However, we cannot extract the meaning between those two words.

$$v_{dog} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad v_{cat} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

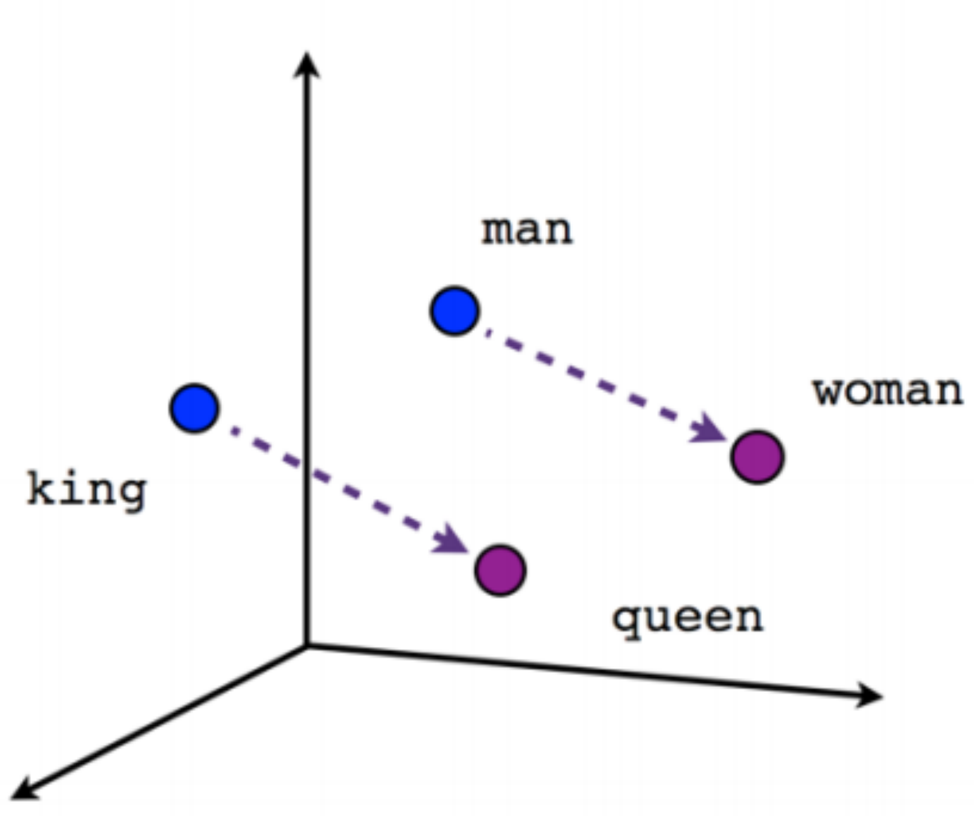
$$\cos(\theta) = \frac{v_{dog}^T v_{cat}}{\|v_{dog}\| \|v_{cat}\|} = 0$$

$$\cos(\theta) = \frac{v_{dog}^T v_{table}}{\|v_{dog}\| \|v_{table}\|} = 0$$

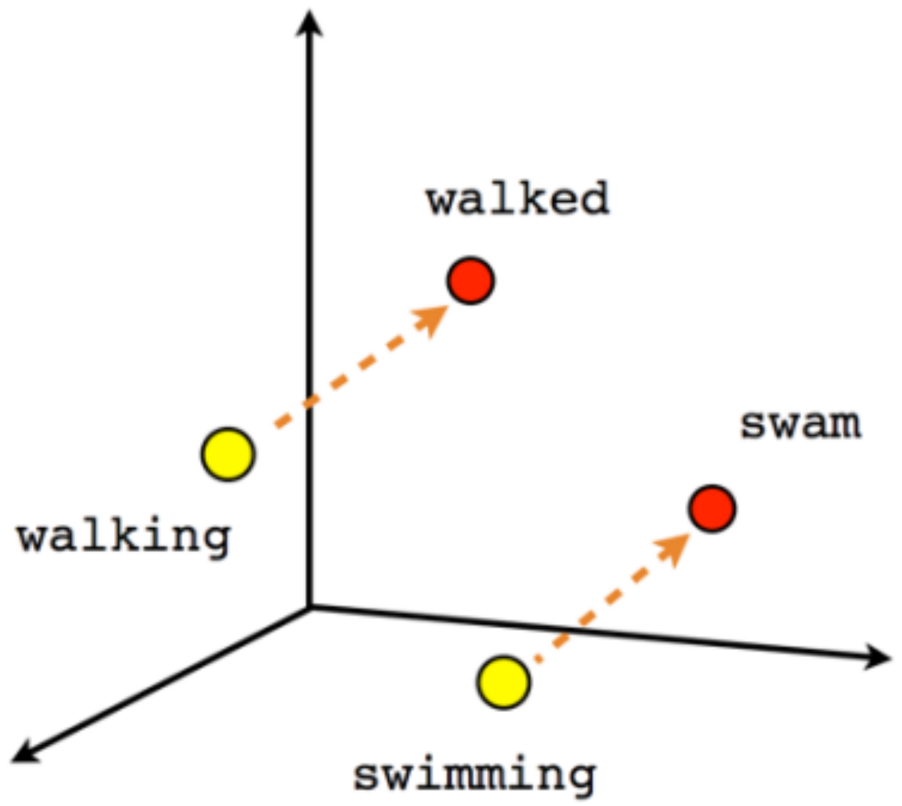
Word Embedding

Word Representation

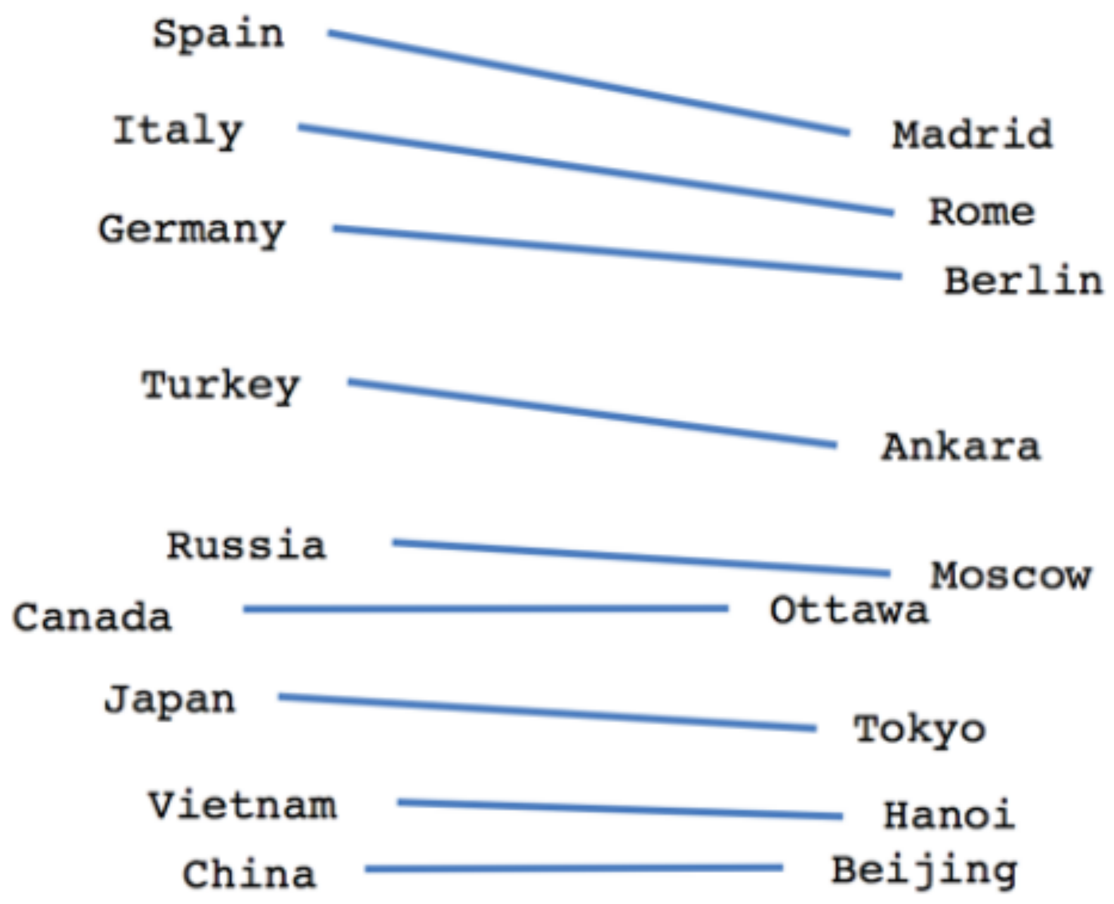
- Word embedding uses a vector representation which could indicate the semantic relationship between words.



Male-Female



Verb tense



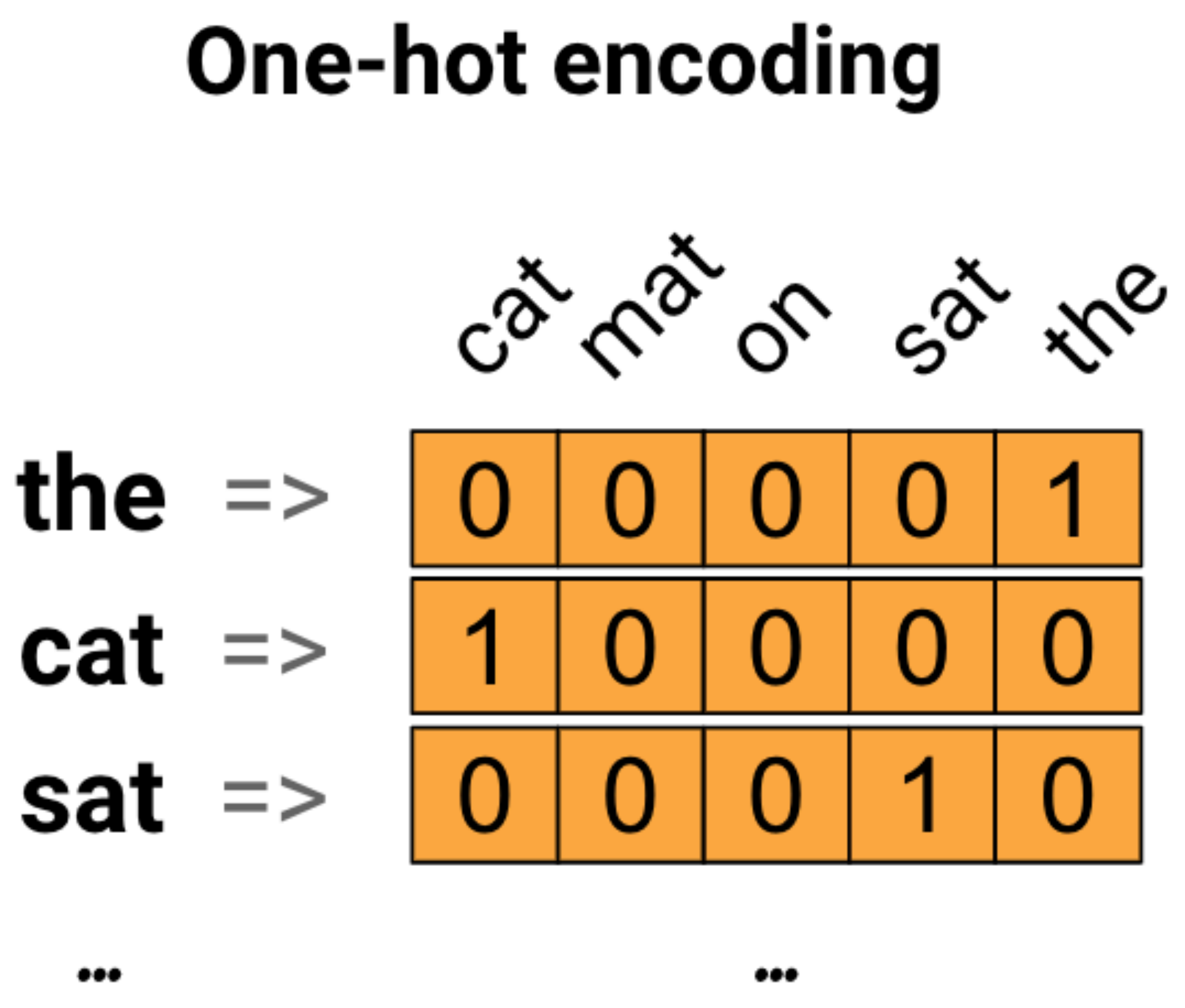
Country-Capital

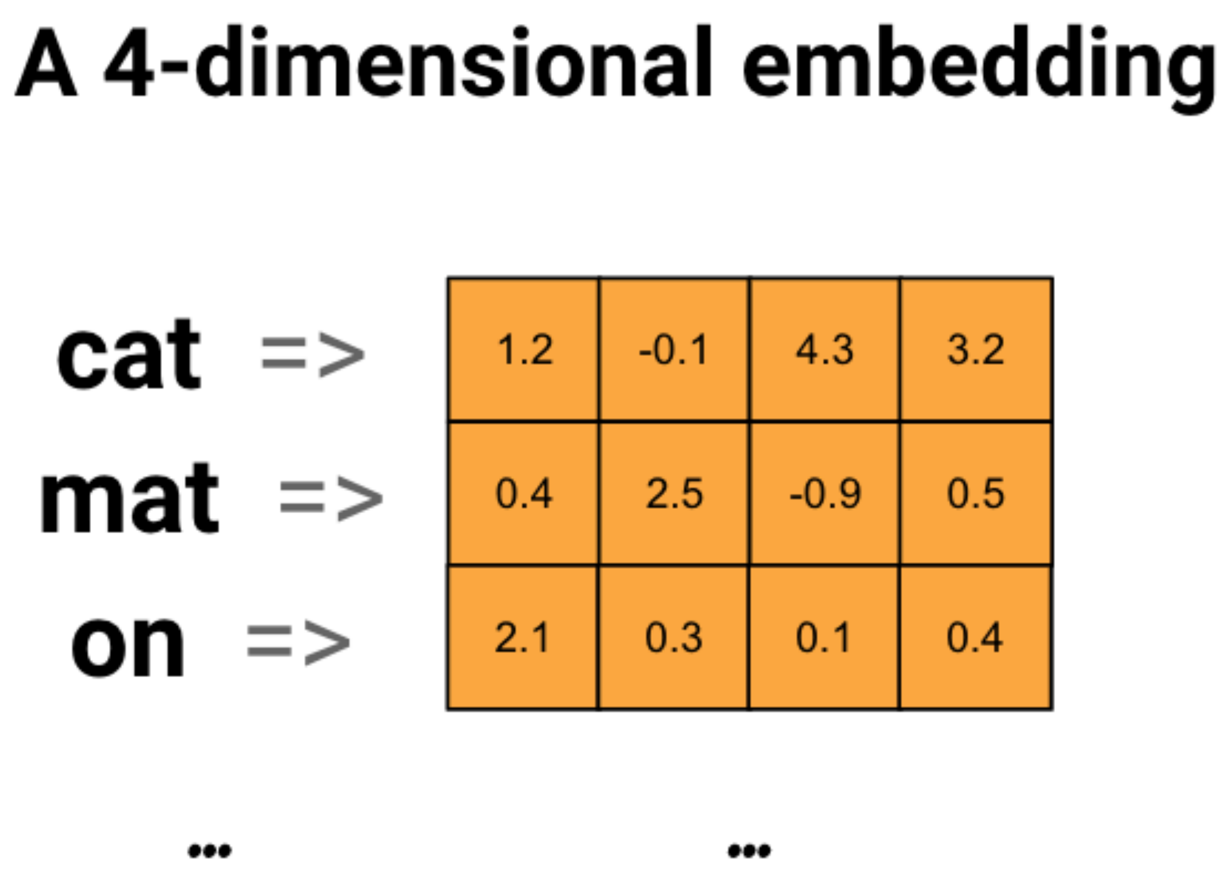
https://leemeng.tw/find-word-semantic-by-using-word2vec-in-tensorflow.html

Word Embedding

Advantages using word embeddings

- Find the semantic relationship between words.
- Map a high-dimensional one-hot encoding vector to a lower-dimensional word embedding vector



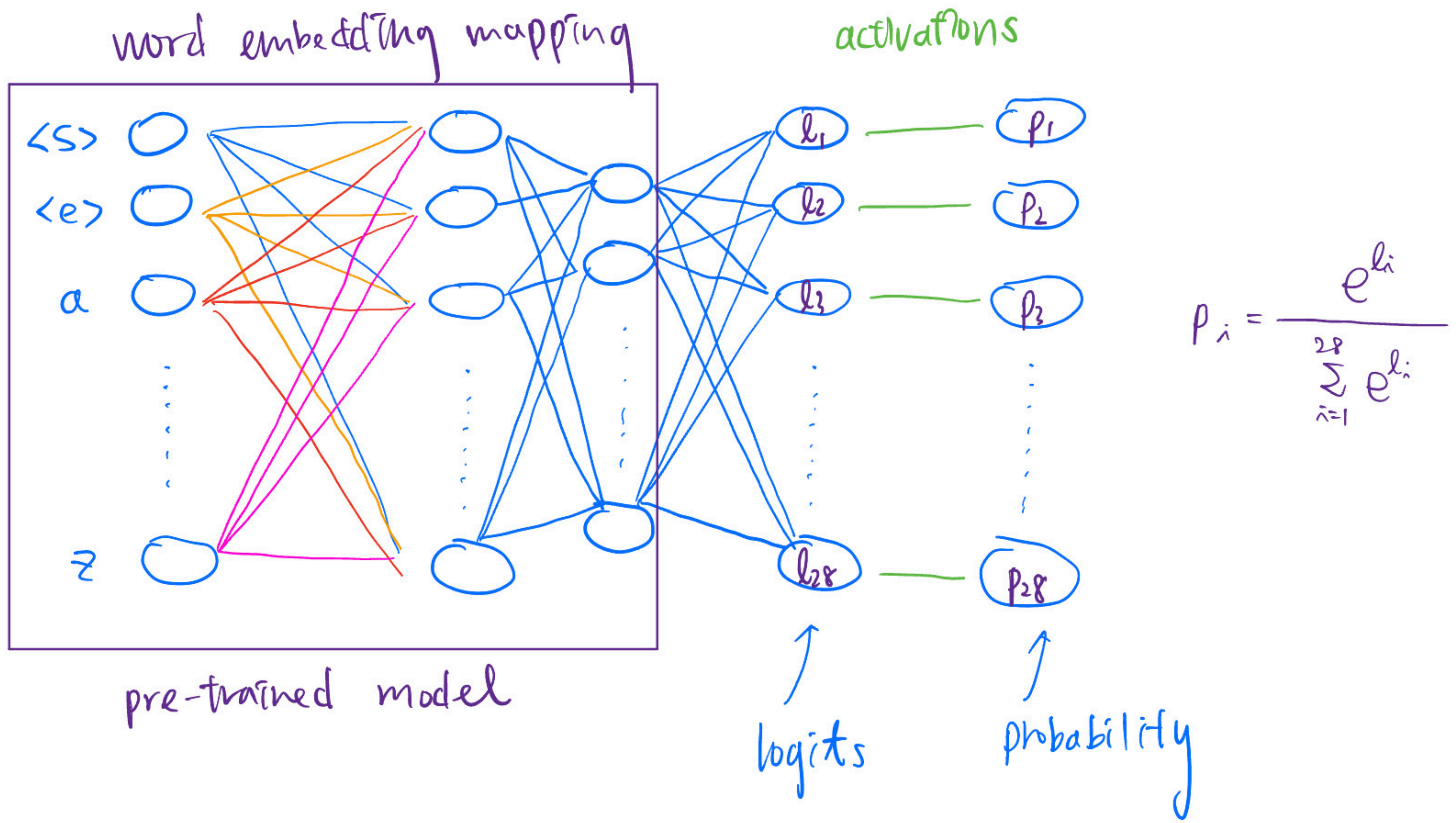


https://www.tensorflow.org/text/guide/word_embeddings

How to Train Word Embedding?

Deep Neural Network

Word embedding layer is in the hidden layer of DNN



Let’s do this in Colab!

Questions

- What if we have a massive dataset that cannot fit in memory?
- How can we compute gradient with more than one hidden layer?

Further Reading

A Neural Probabilistic Language Model

Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03

A Neural Probabilistic Language Model

Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Réjean Ducharme DUCHARME@IRO.UMONTREAL.CA
Pascal Vincent VINCENTP@IRO.UMONTREAL.CA
Christian Jauvin JAUVINC@IRO.UMONTREAL.CA
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal, Montréal, Québec, Canada

Editors: Jaz Kandola, Thomas Hofmann, Tomaso Poggio and John Shawe-Taylor

Abstract

A goal of statistical language modeling is to learn the joint probability function of sequences of words in a language. This is intrinsically difficult because of the **curse of dimensionality**: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training. Traditional but very successful approaches based on n-grams obtain generalization

Further Reading

Efficient Estimation of Word Representations in Vector Space (Word2Vec)

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

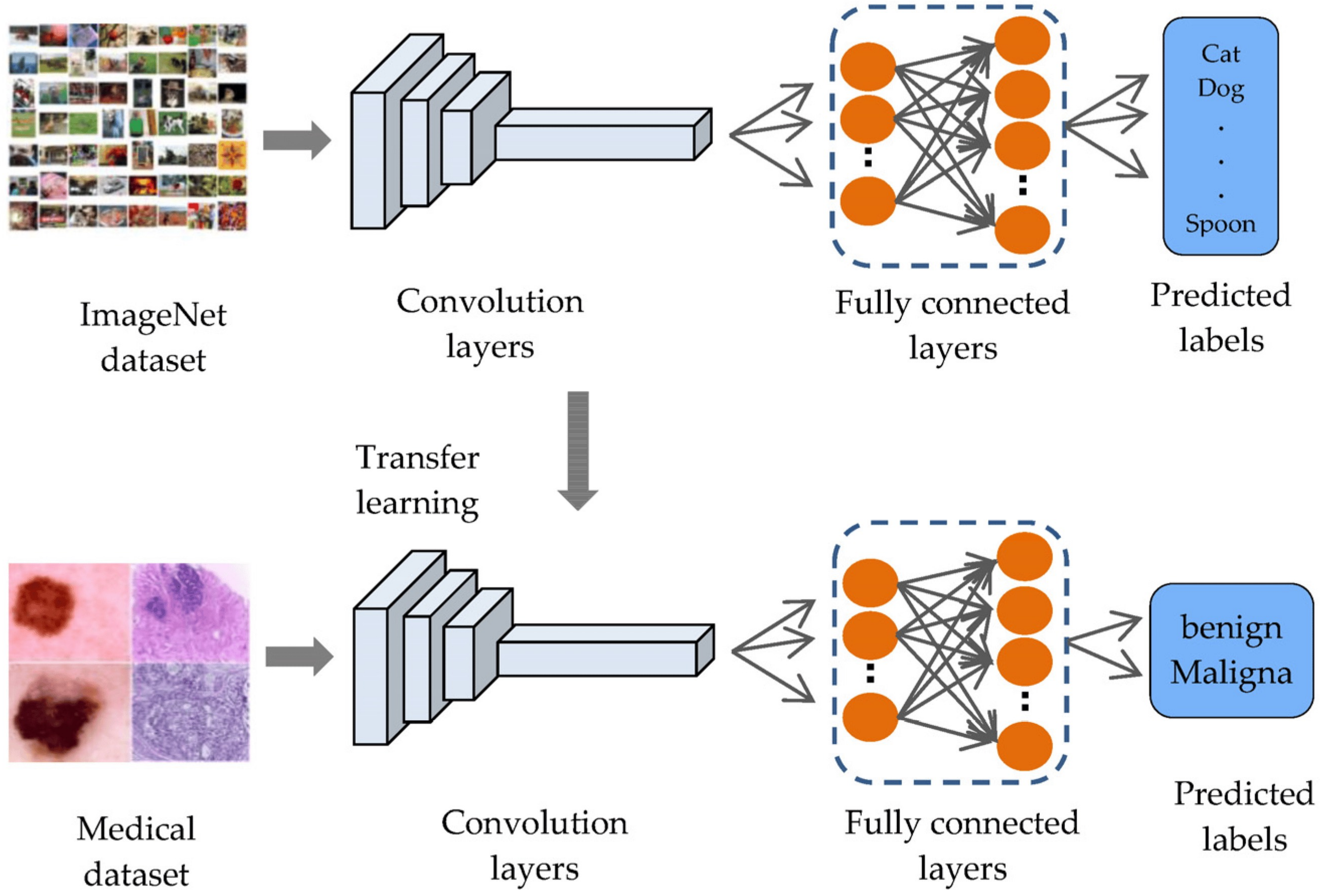
Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations

Transfer Learning

Transfer Learning

Use Pre-trained Model in other tasks

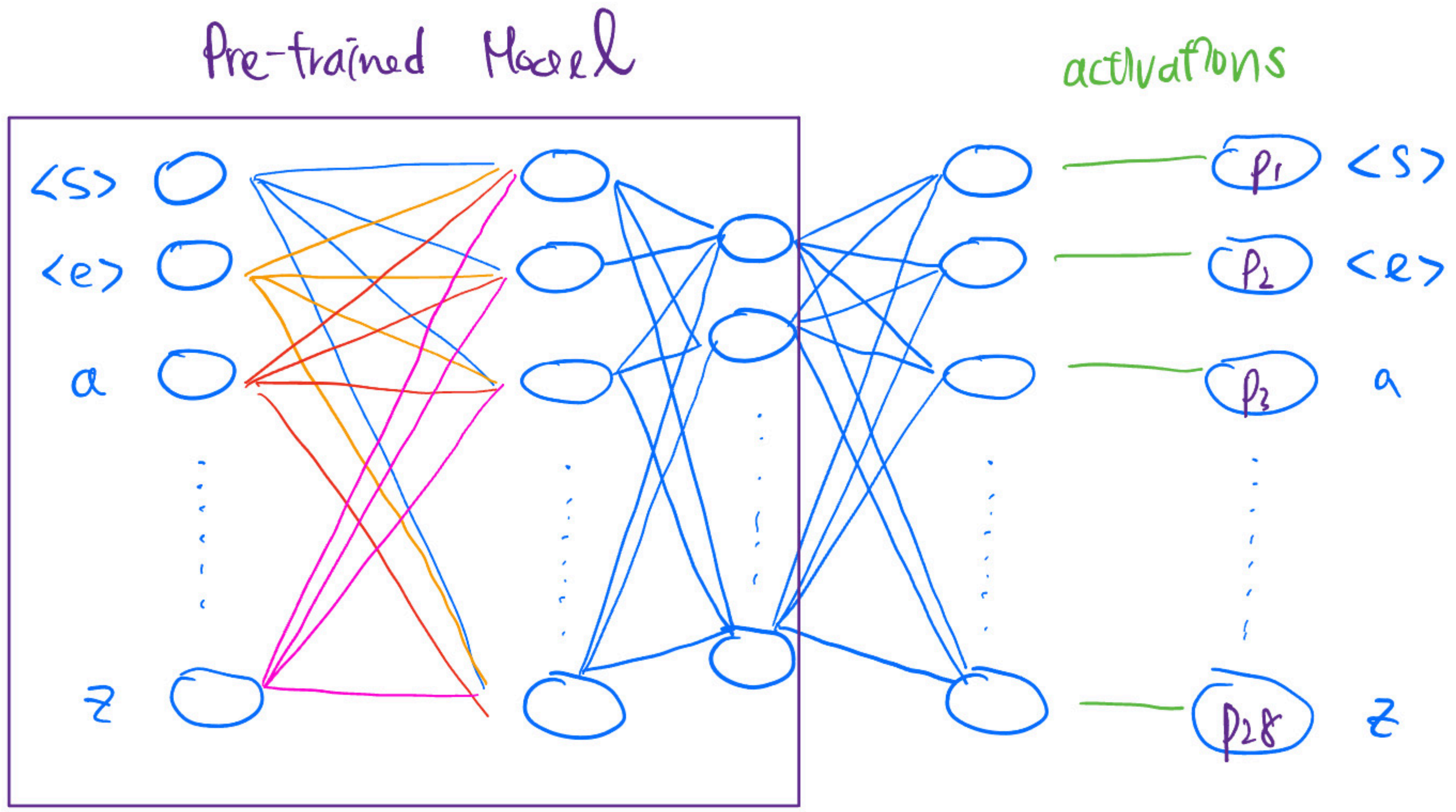


https://www.mdpi.com/1424-8220/23/2/570

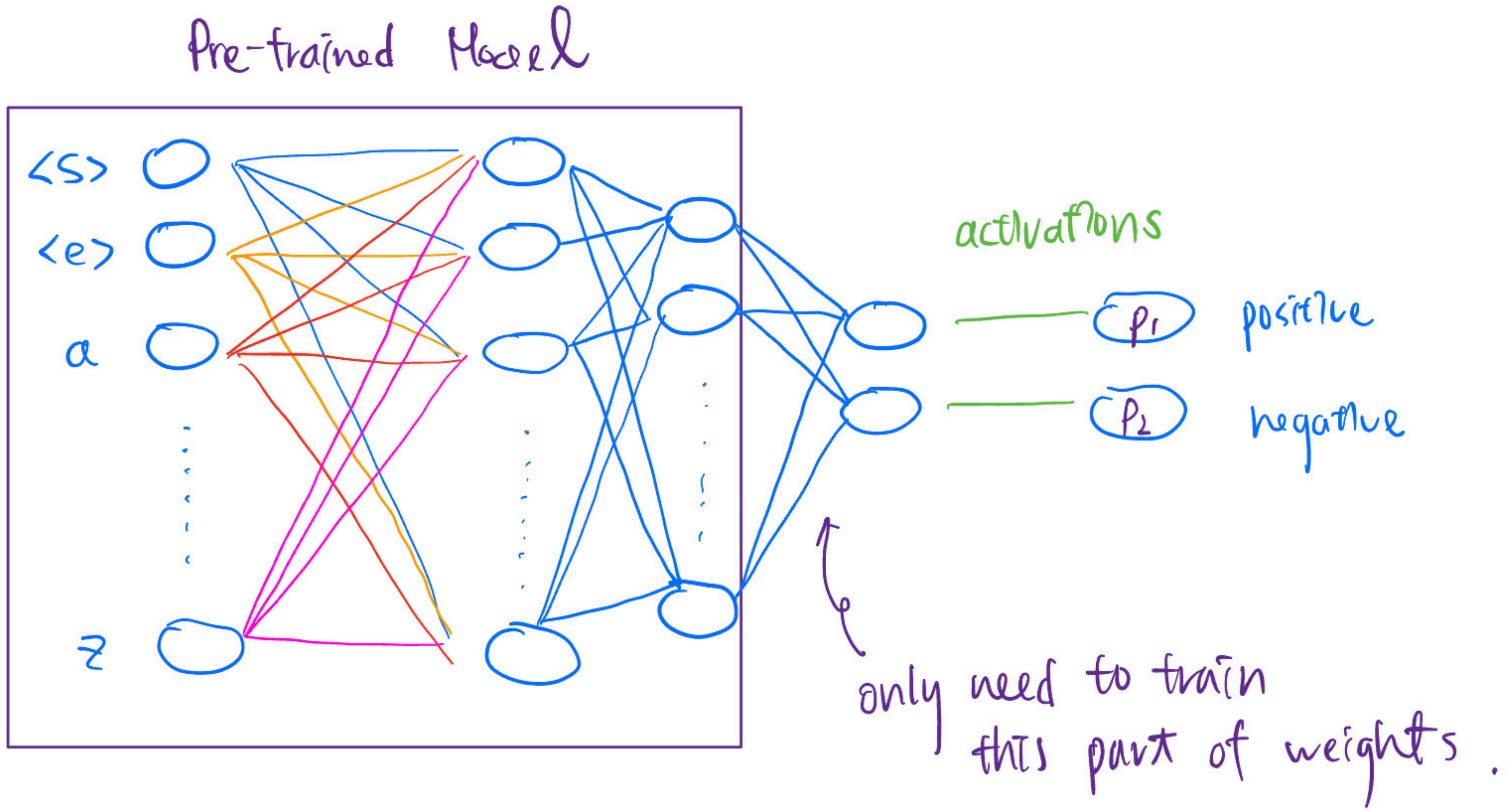
Transfer Learning

Use Pre-trained Model in other tasks

Original Task



New Classification Task



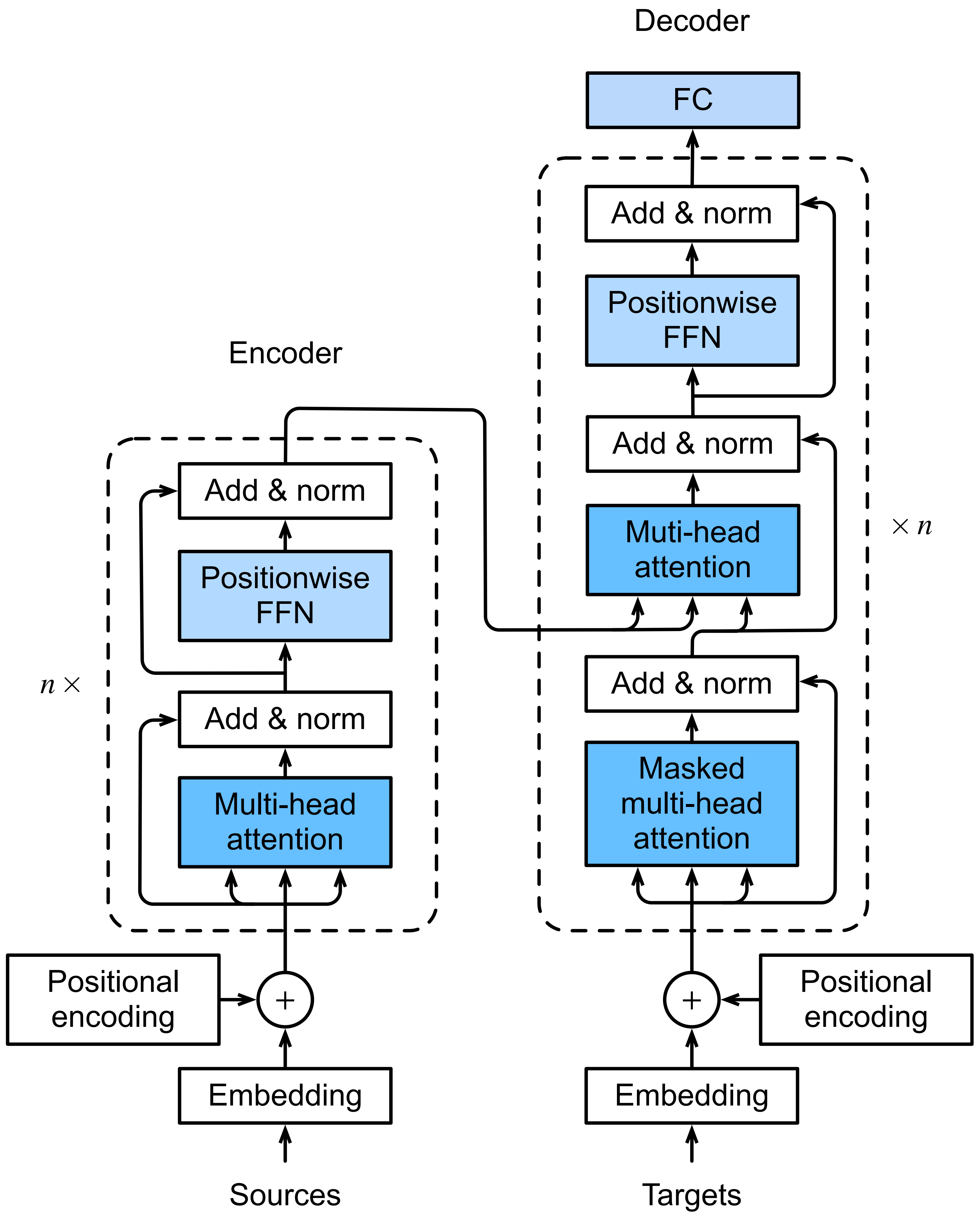
Wrap Up

What We Have Gone Through

- Deep Neural Network
- Word Embedding
- Transfer Learning

What's Next

- Transformer - Self Attention



Q & A