Introduction to Word2Vec

t.me/cvision

Abolfazl Mahdizade

Outline

- *Word Vectors*
- *Word2Vec*

Word2Vec

http://rohanvarma.me/Word2Vec/

http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

http://ruder.io/word-embeddings-1/

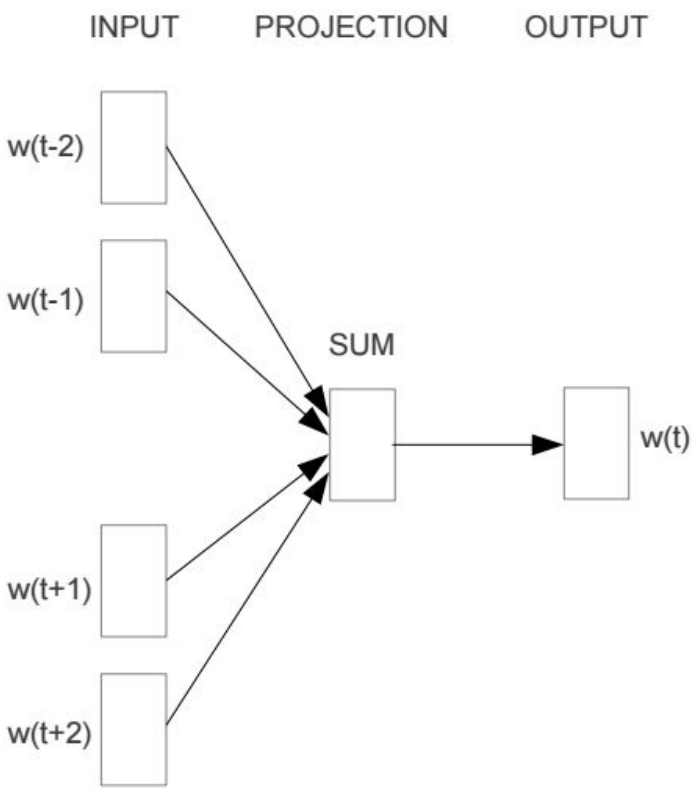
Word Vectors

- ***Word Vectors***, ***Word Embeddings*** *or* ***Distributed Representation of Words****, generally refer to a* ***Dense Vector*** *representation of a word, as compared to a sparse (ie* ***one-hot****) traditional representation*
- *Word2Vec is one of Distributed Representation of Words methods*

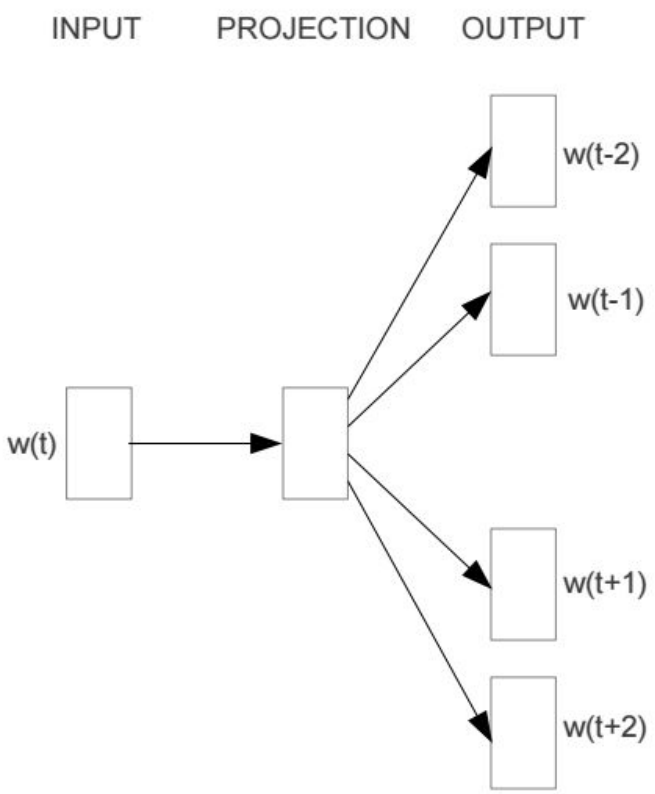
Word2Vec

- *Represent words as dense vectors*
- *Vectors feed into a discriminative model (typically an RNN)*
- *There are two different implementations of Word2Vec*
  - *Skip-Gram (SG)*
  - *Continuous Bag of Words (CBoW)*
- *Both of these models learn dense vector representation of words, based on the words that surround them (ie, their context)*
- *Skip-Gram model predicts context (surrounding) words given the current word*
- *Continuous Bag of Words model predicts the current word based on several surrounding words*

Word2Vec (Continue)

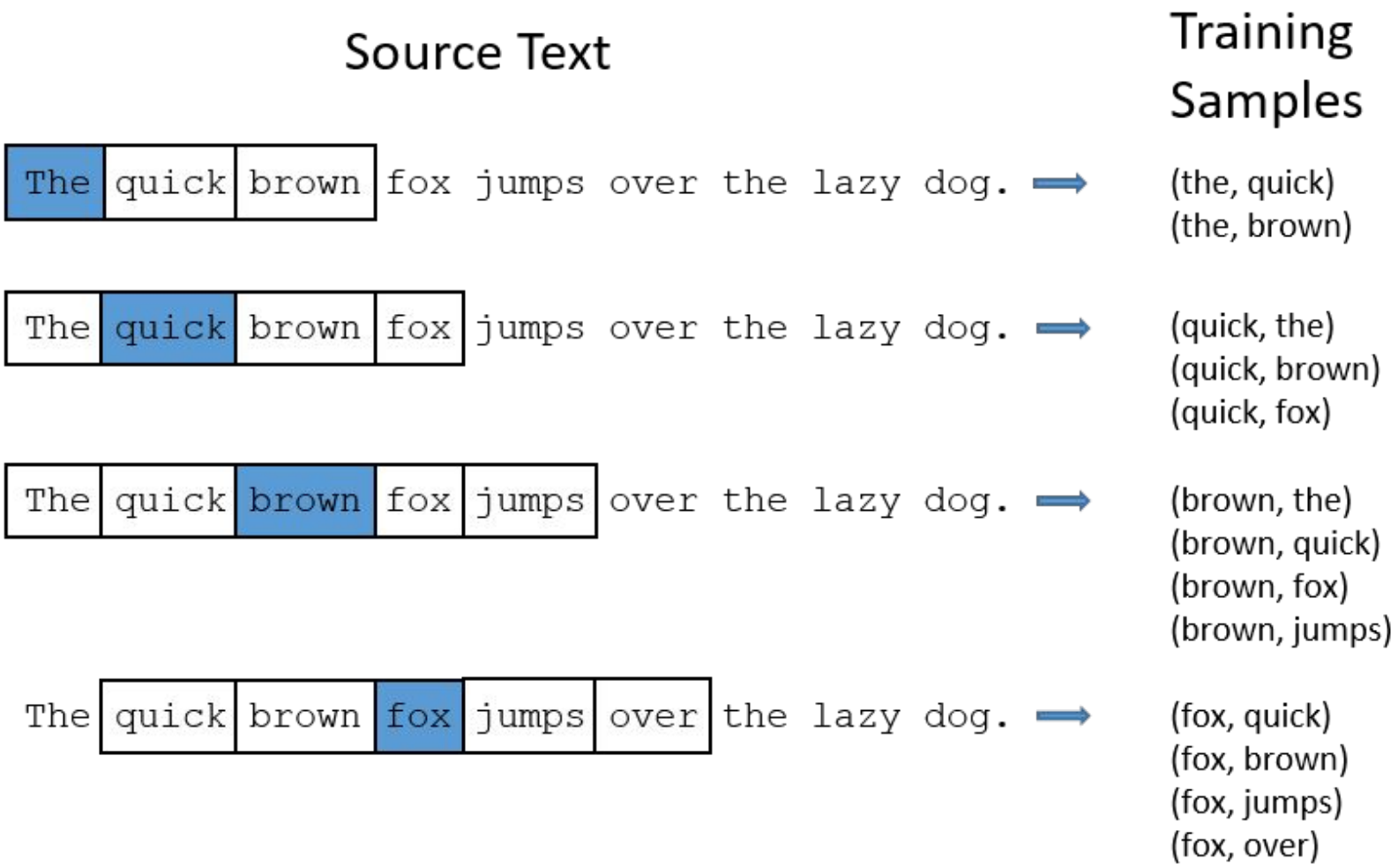


CBOW



Skip-gram

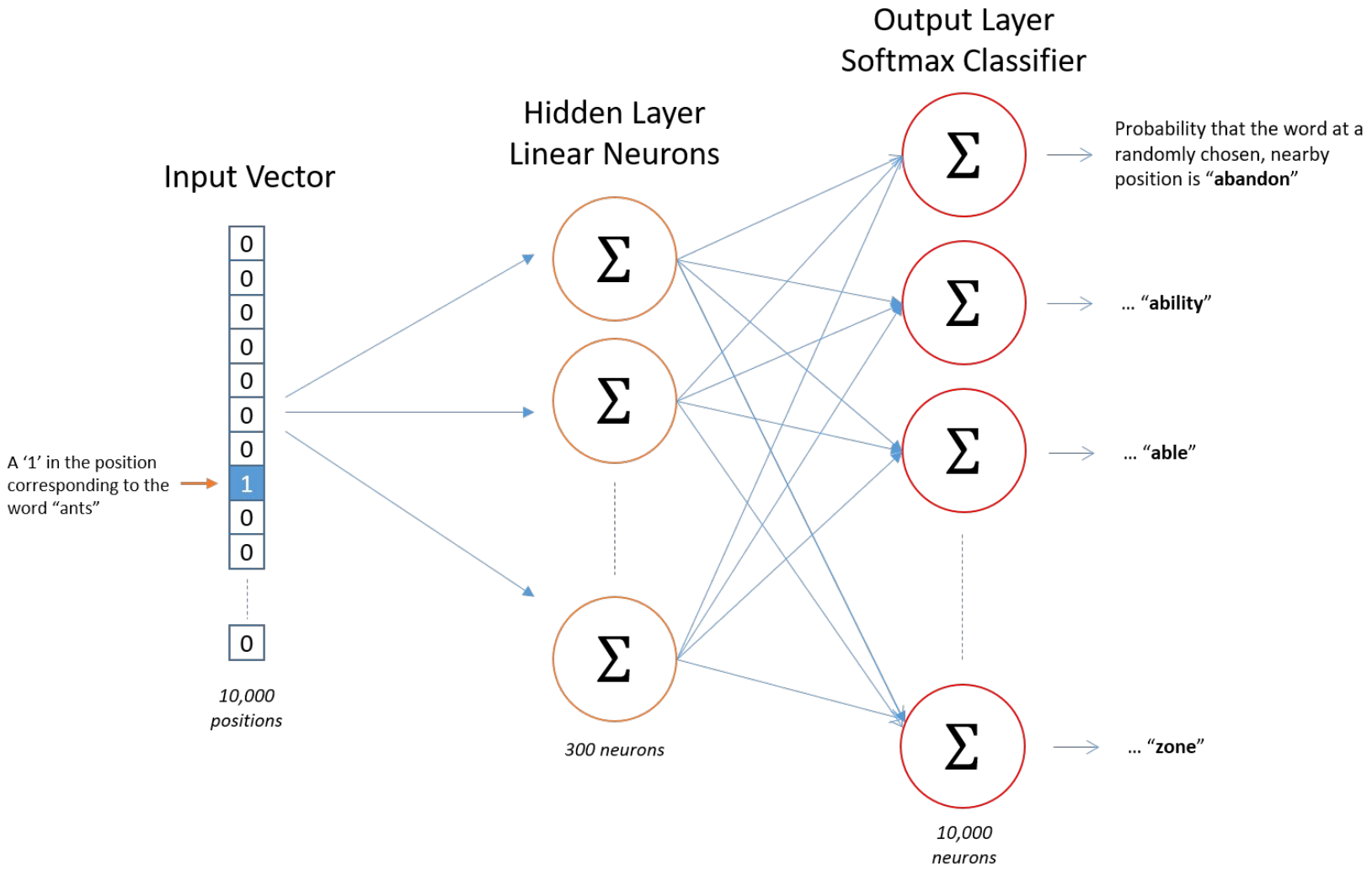
Word2Vec (Continue)



Word2Vec (Continue)

- *Word2Vec is a simple neural network with a single hidden layer*
- *This network train with raw text for specific task (language model), but the goal is actually just to learn the weights of the hidden layer*
- *The weights of hidden layer are actually the Word Vectors*
- *For learn Word Vectors with Word2Vec*
  - *Prepare training documents*
  - *Build a vocabulary of words from training documents (V unique words)*
  - *Represent each word as one hot vector (as input)*
  - *The output of the network have a same size as input (V)*

Word2Vec (Continue)

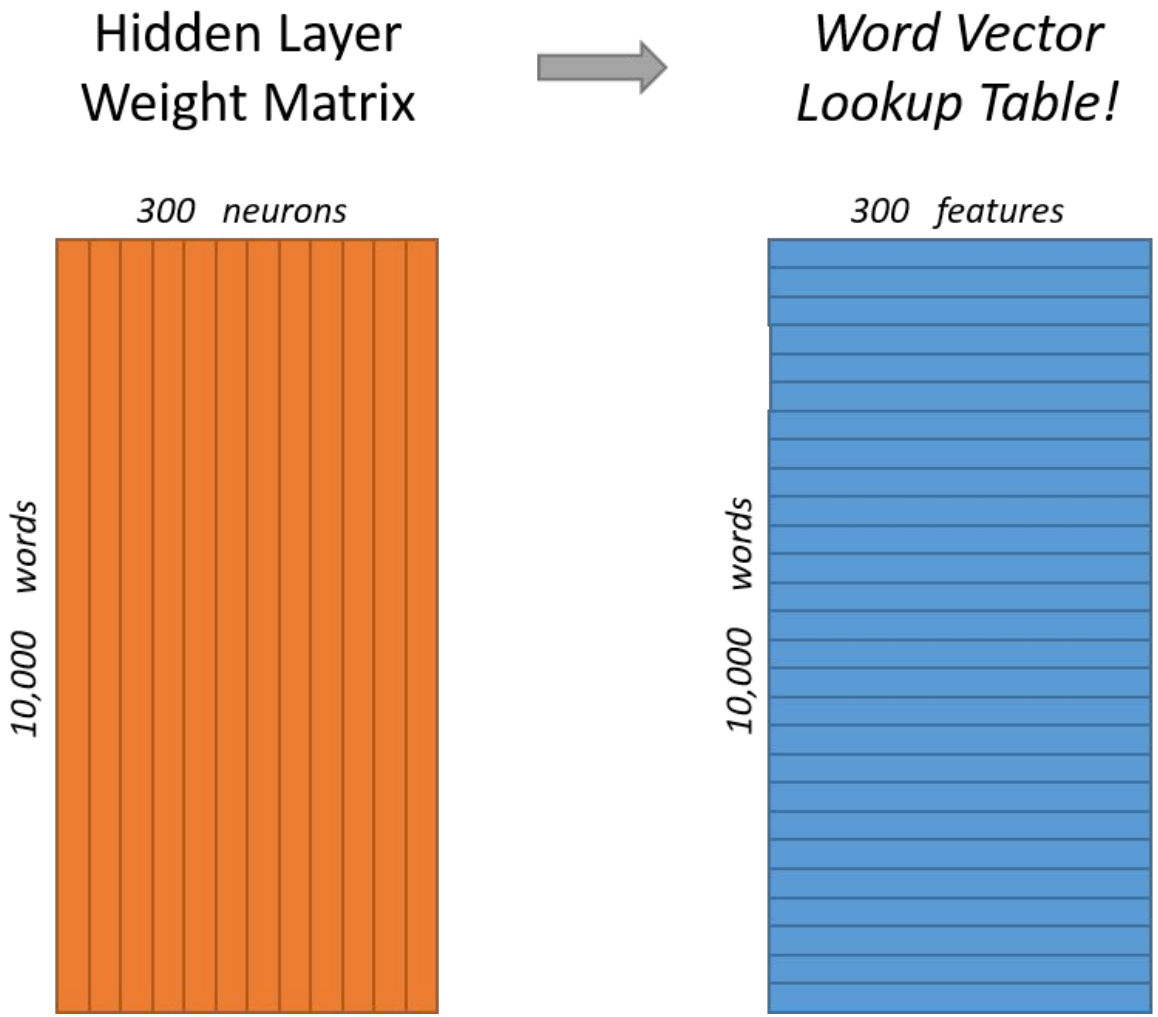


Word2Vec (Continue)

- *Hidden layer of this model is really just operating as a lookup table. The output of the hidden layer is just the "word vector" for the input word*

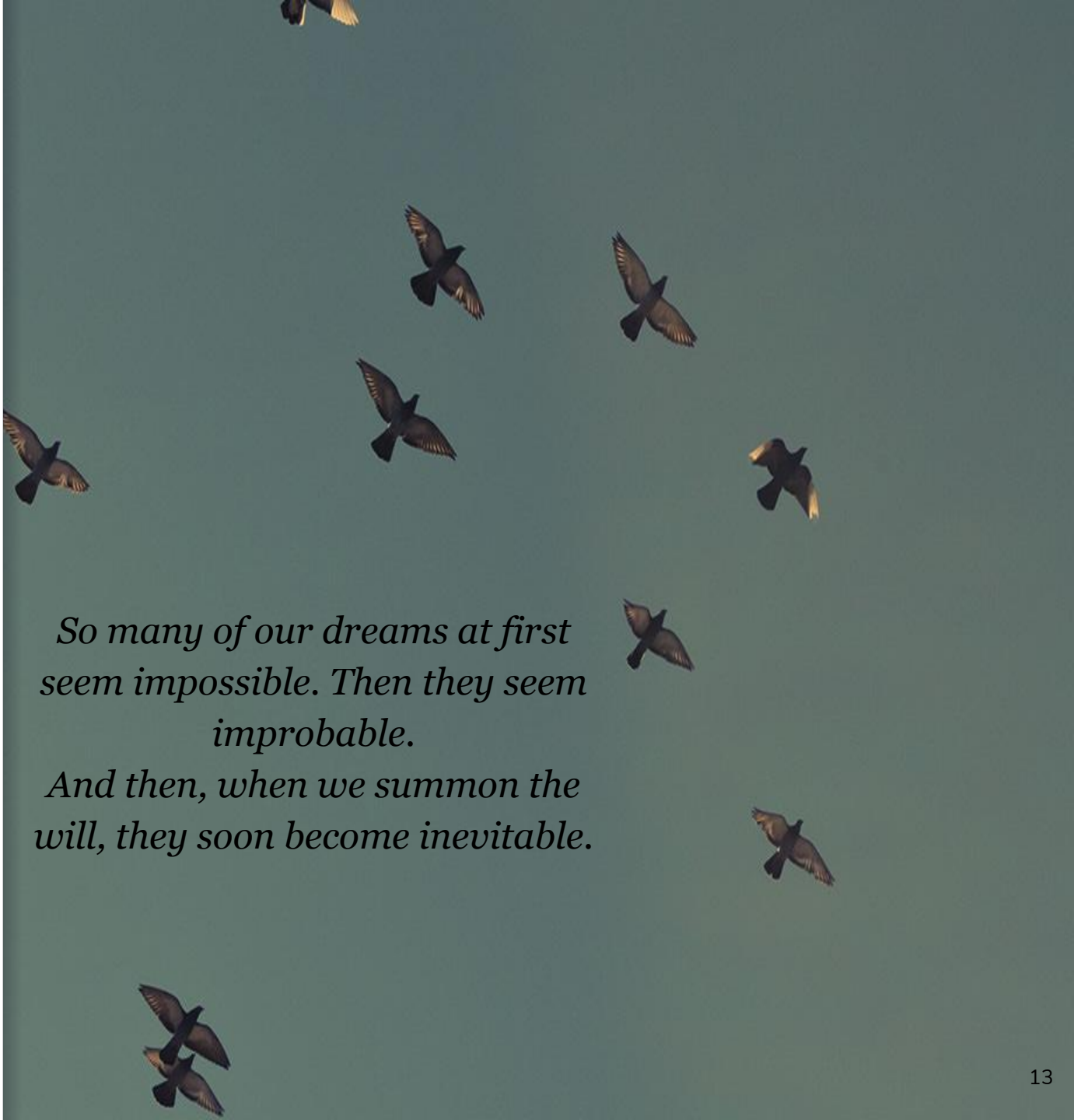
$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Word2Vec (Continue)



Word2Vec (Continue)

- *If two different words have very similar "contexts" (that is, what words are likely to appear around them - synonyms like "intelligent" and "smart" would have very similar contexts), the model needs to output very similar results for these two word.*
- *One way for the network to output similar context predictions for these two words is if the word vectors are similar.*
- *So, if two words have similar contexts, then the network is motivated to learn similar word vectors for these two words!*
- *Ta da!*


So many of our dreams at first seem impossible. Then they seem improbable.
And then, when we summon the will, they soon become inevitable.