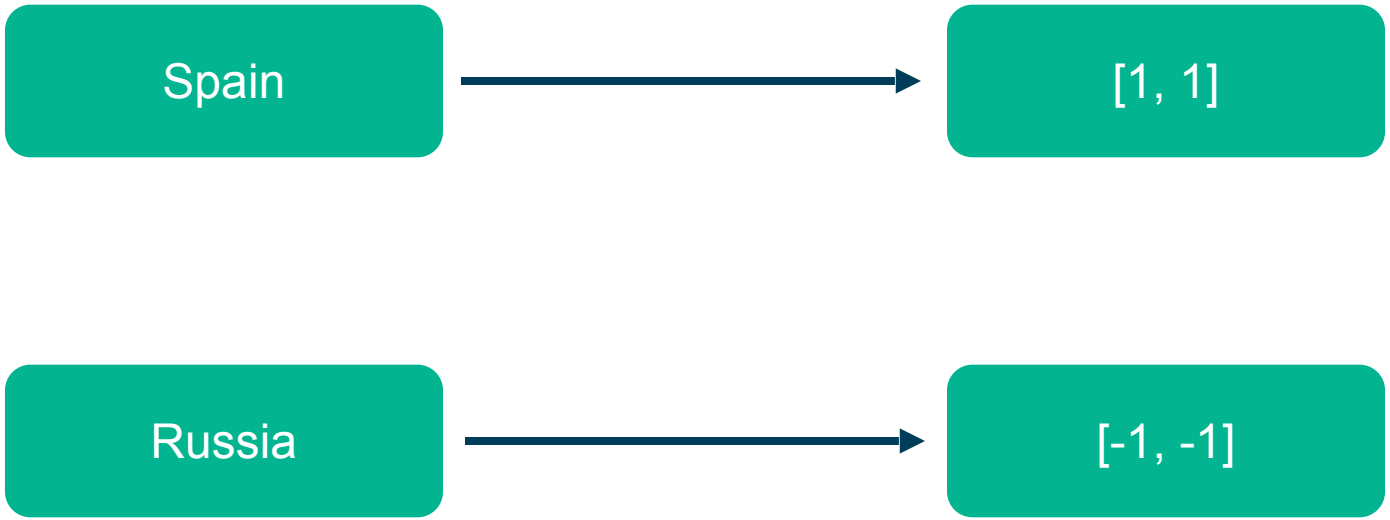
Word2Vec

Hands on Generative AI

Text Similarity

Data Trainers

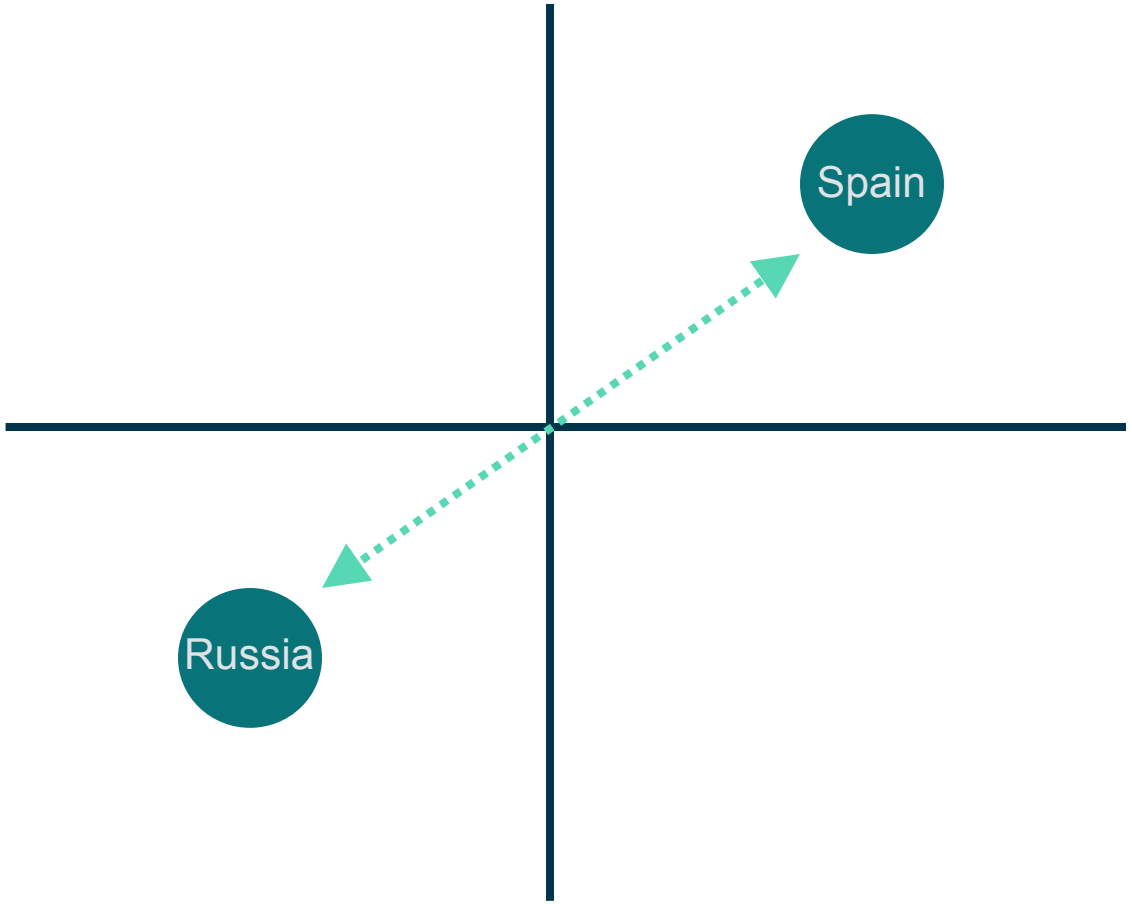
How do we represent words?



How do we represent words?

Spain → [1, 1]

Russia → [-1, -1]

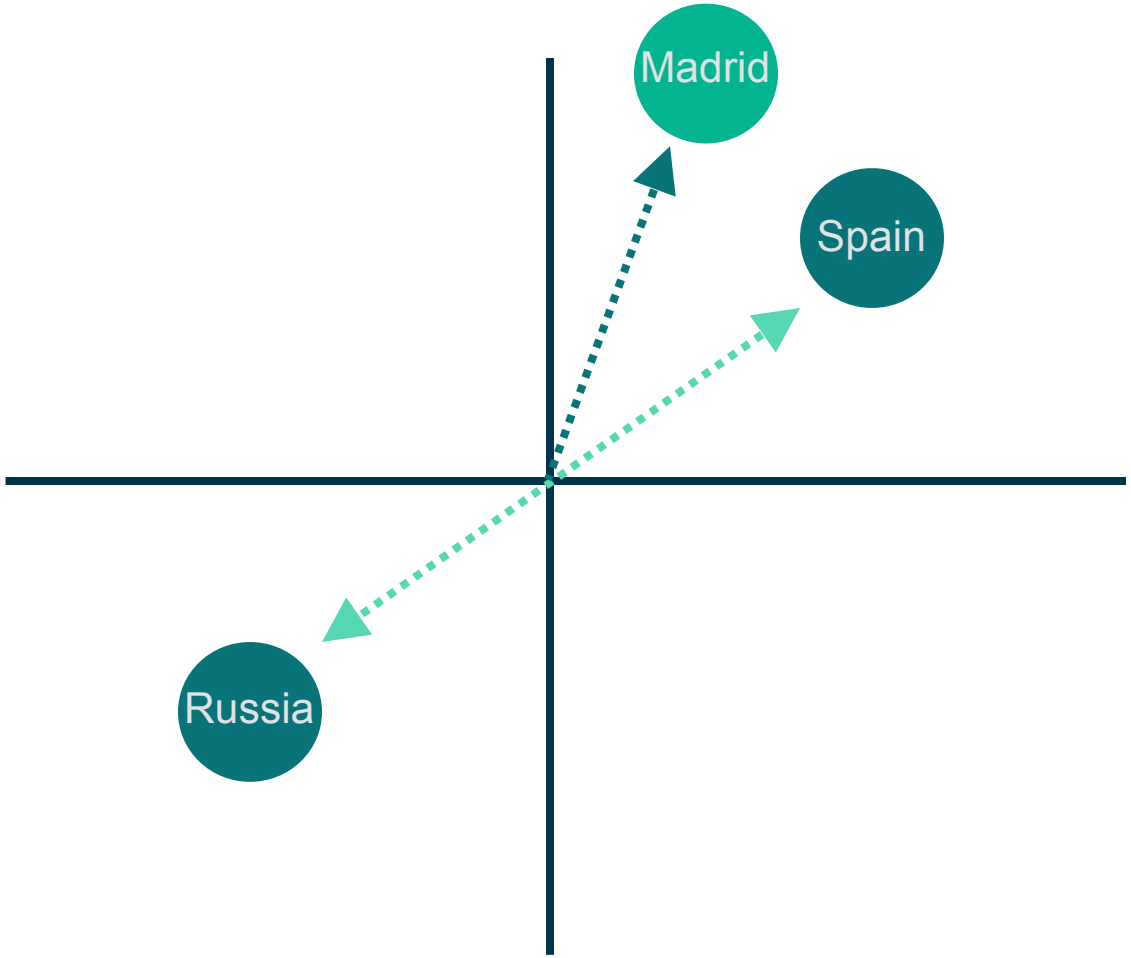


Madrid?

How do we represent words?

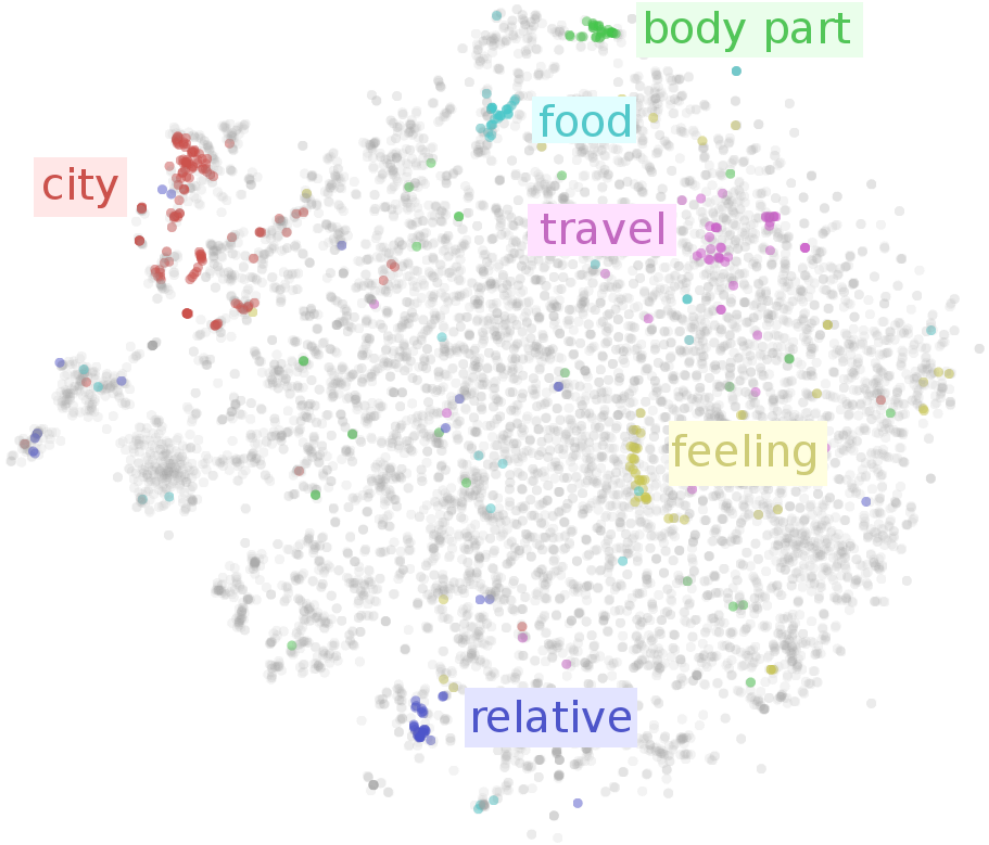
Spain → [1, 1]

Russia → [-1, -1]



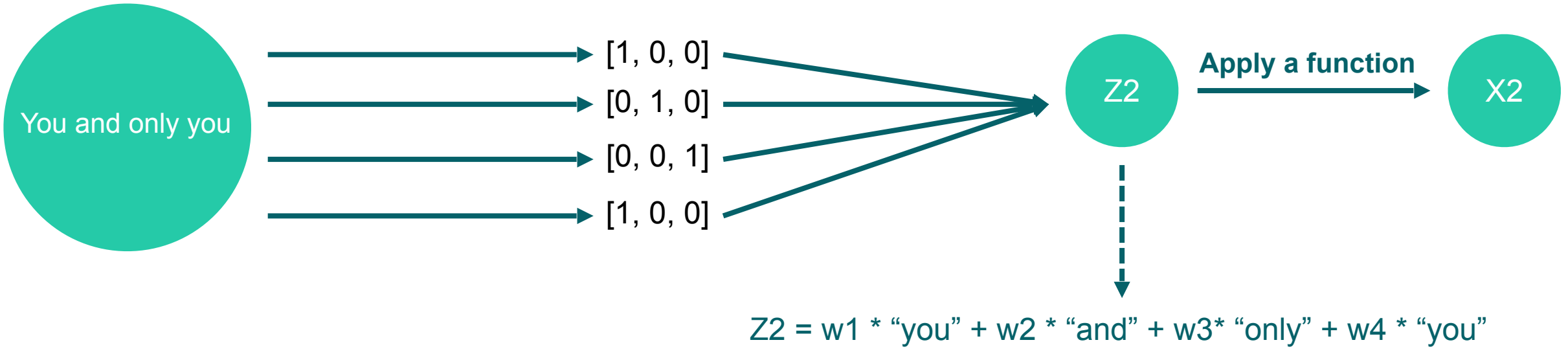
Madrid should be closer to Spain than Russia.

How do we represent words?



- This mapping, or function, from words to sequences of numbers is called a **word representation or embedding.**
- If we have a good word representation, then we can find similar words! And use those synonyms to expand the query.
- It has been shown that **neural networks** are great to find good embeddings.

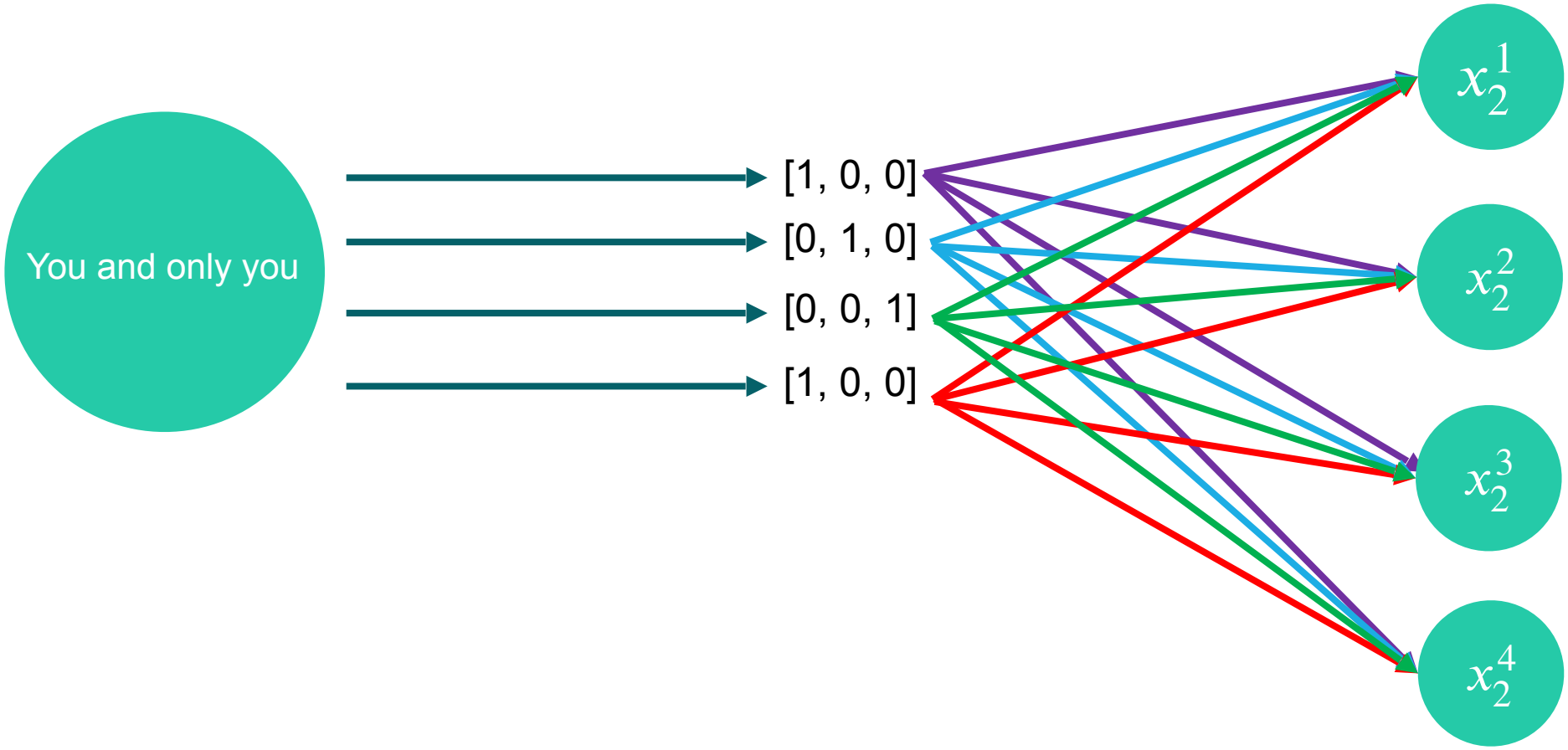
Neural Networks



This would create a 1 dimensional representation

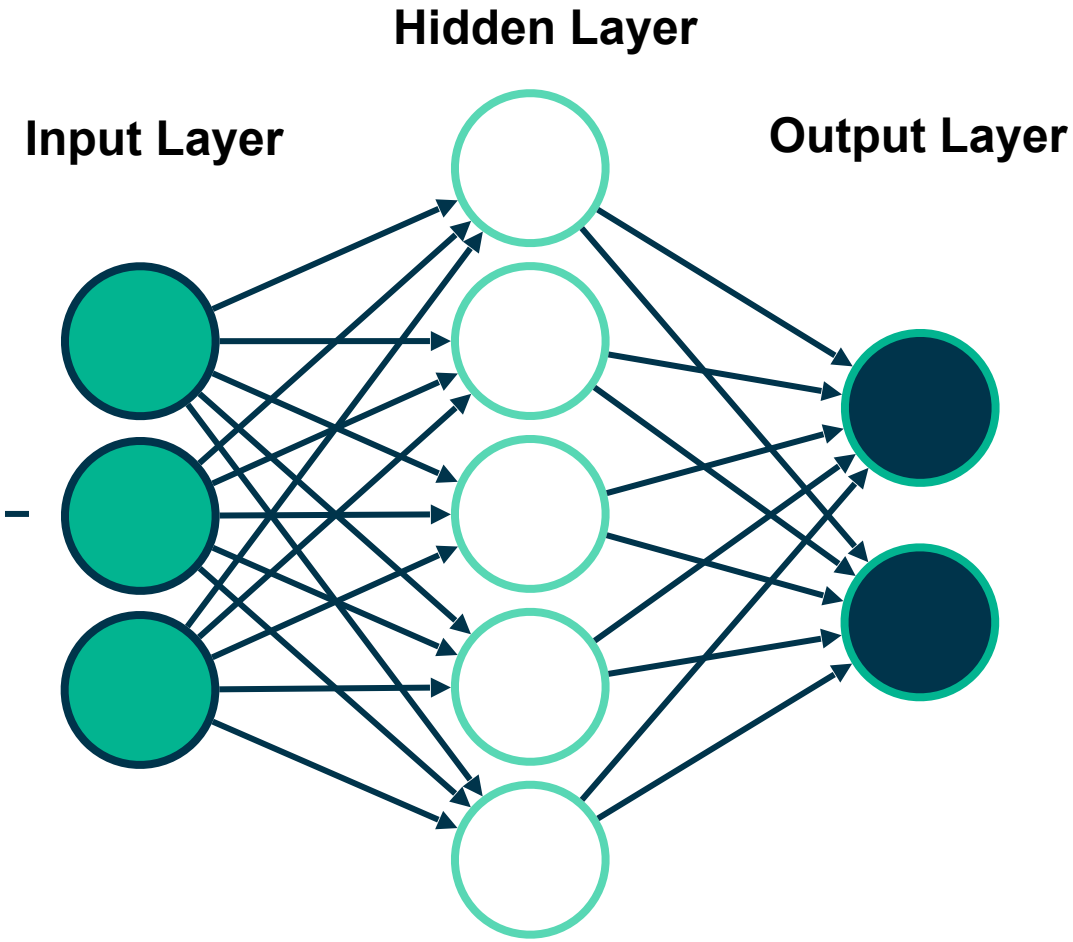
Neural Networks

We represent each of the initial numbers as circles too and we get…



This would create a 4 dimensional representation

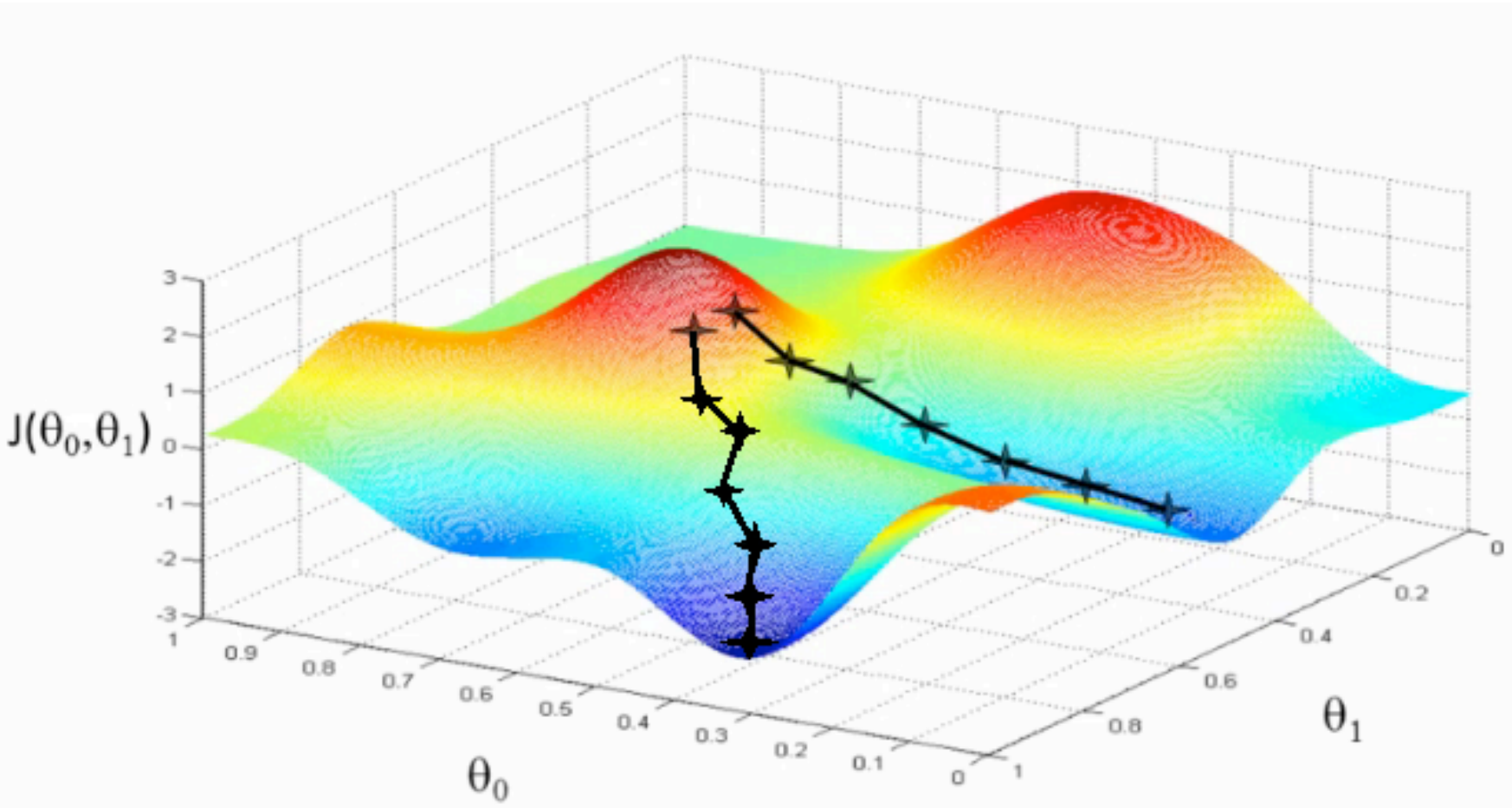
Neural Networks



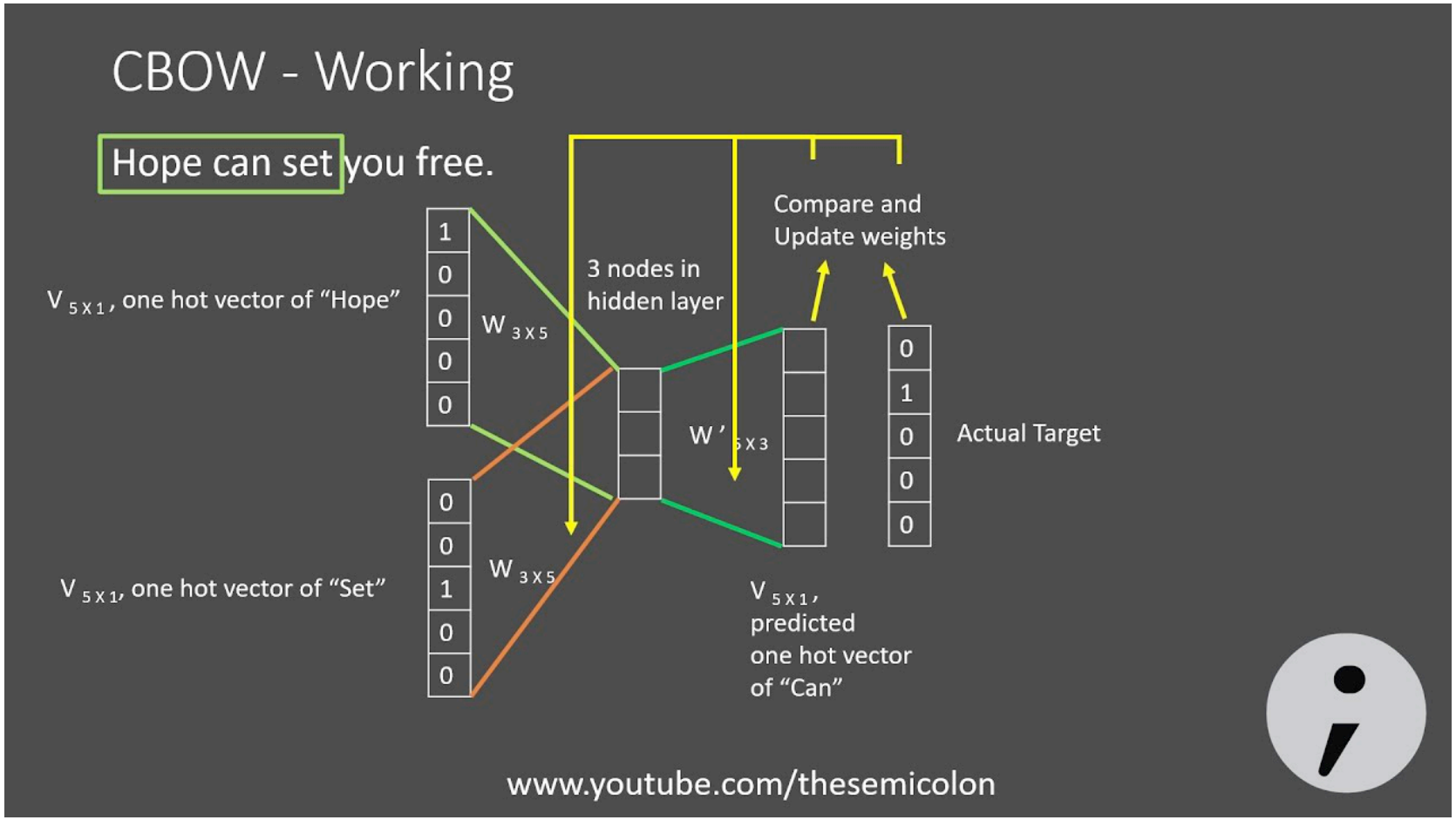
Each input word is a circle, for convenience.

Here we added another “layer” of doing the same, increasing complexity and prediction power.

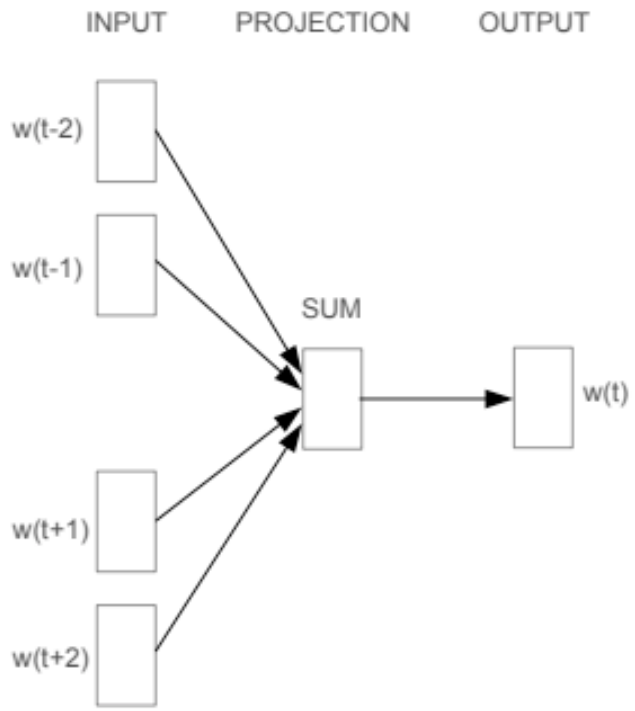
Training



CBOW


CBOW - Working
Hope can set you free.
V 5 X 1, one hot vector of "Hope"
1
0
0
0
0
W 3 X 5
3 nodes in hidden layer
Compare and Update weights
W ' 5 X 3
0
1
0
0
0
Actual Target
V 5 X 1, one hot vector of "Set"
0
0
1
0
0
W 3 X 5
V 5 X 1, predicted one hot vector of "Can"
www.youtube.com/thesemicolon

Why we do this?

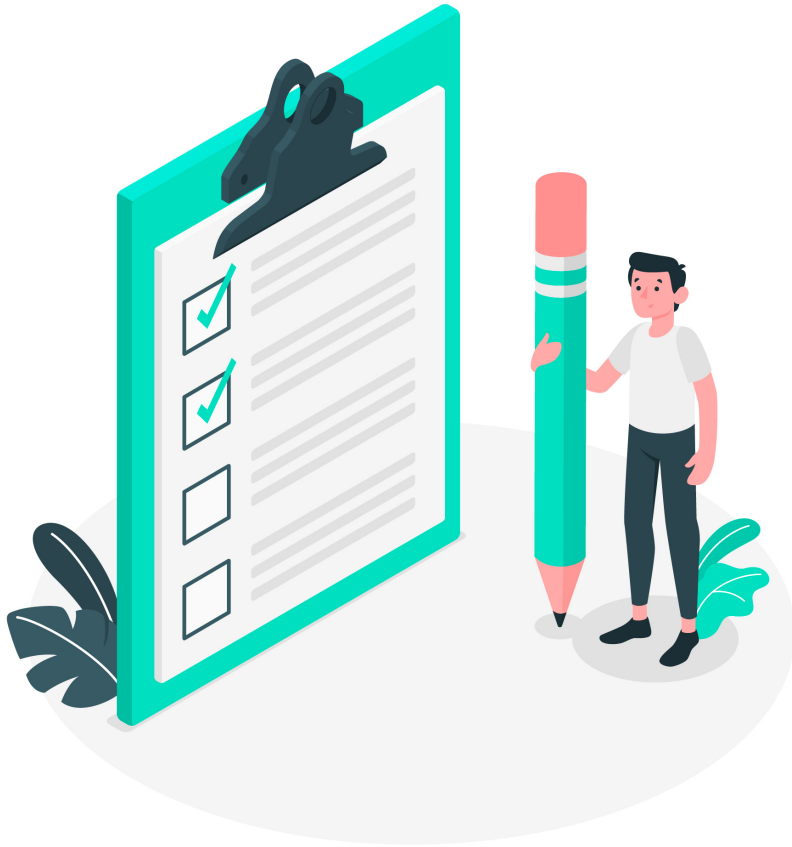


- This way, we can later get those *w* that mapped the words into this n-dimensional representation. **That is our embedding.**
- We used this trick with windows to be able to **train it** in an unlabeled fashion.
- It works incredibly well!

LAB

Classifying text

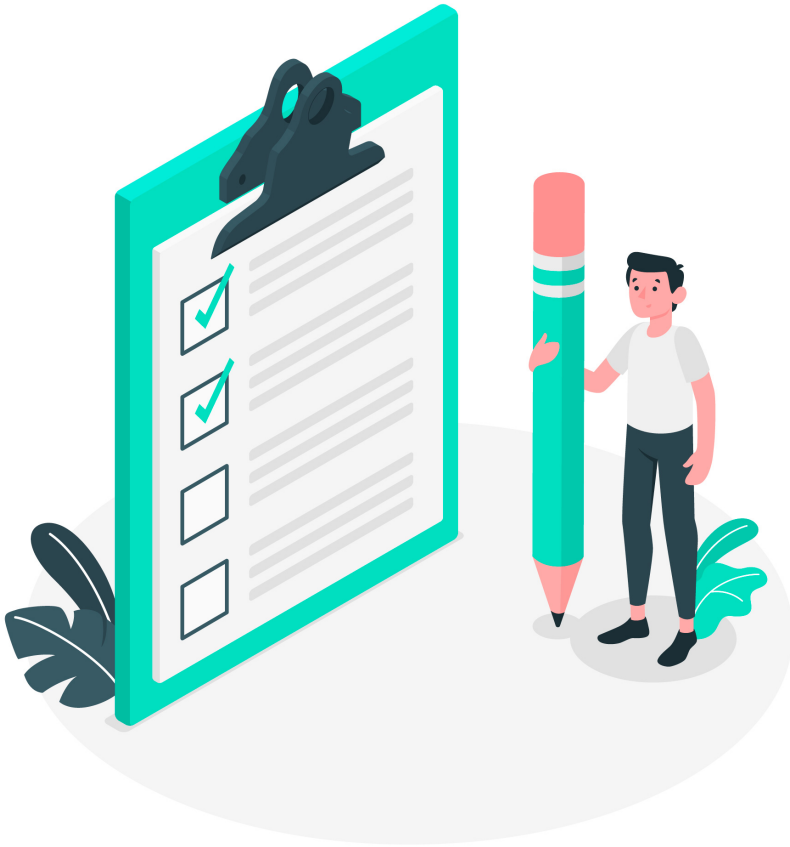
- Learn about text preprocessing in Keras
- Perform classification with logistic regression



LAB

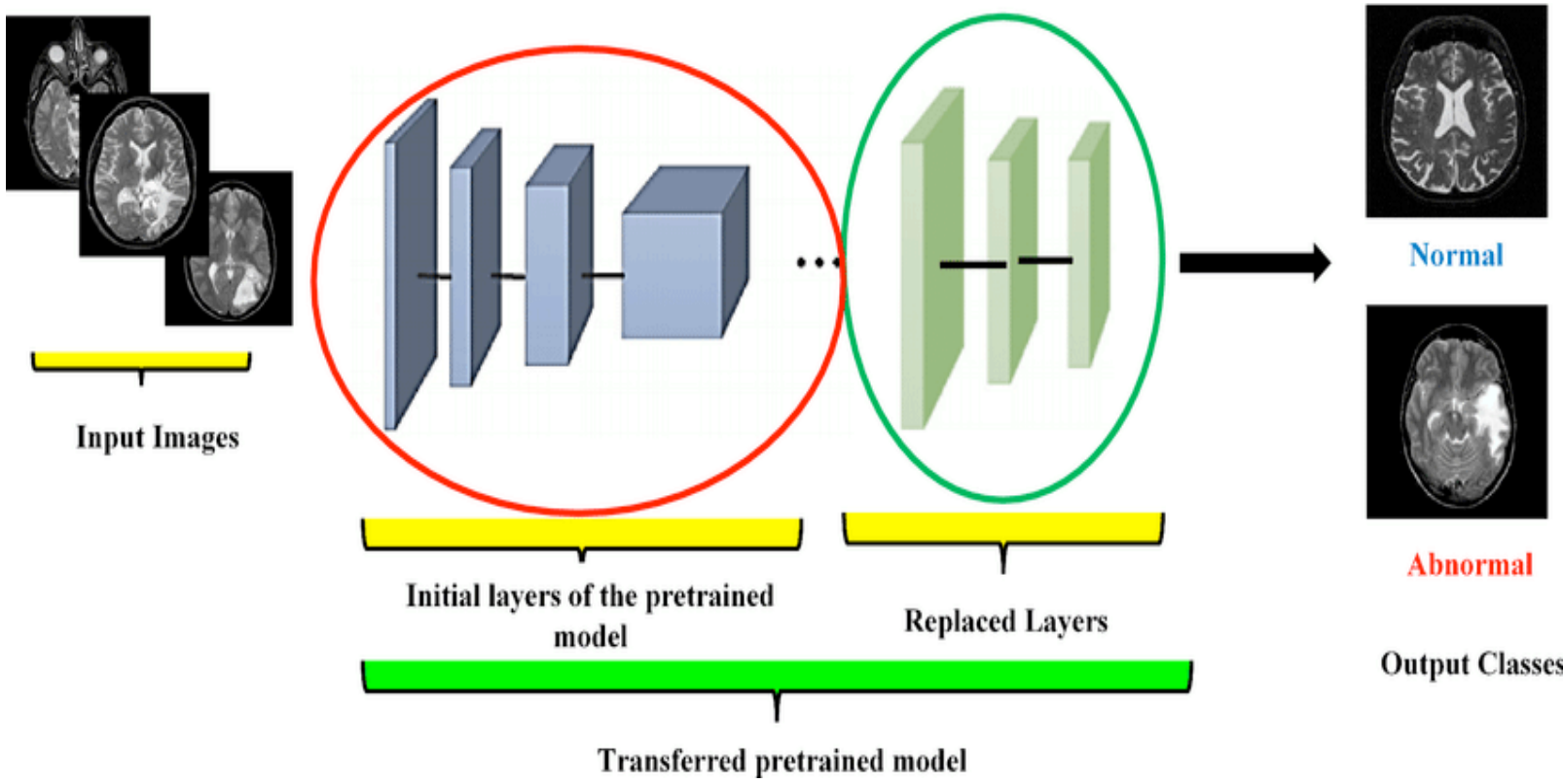
Training CBOW

- Train your first embedding using Keras.
- Transform text into a generator of training sequences.



Can we do better?

- Researchers found that one can do many things to improve.
- As layers “stack up” one can reuse a pretrained model on the first layers and just train the rest.
- Not only it speeds up, but makes it possible to do magic with little data.
- This is called **fine-tuning.**



LAB

Using GloVe pre trained Embedding

- Fine tune GloVe.
- Verify how easy it is!