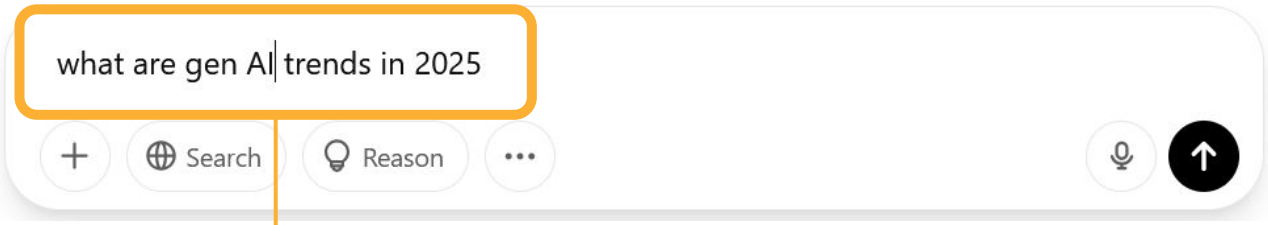
Vector DB & RAG

Sergey Tarasenko, PhD

TOC

Era of Generative AI

Origins of GenAI: Prompts



PROMPT

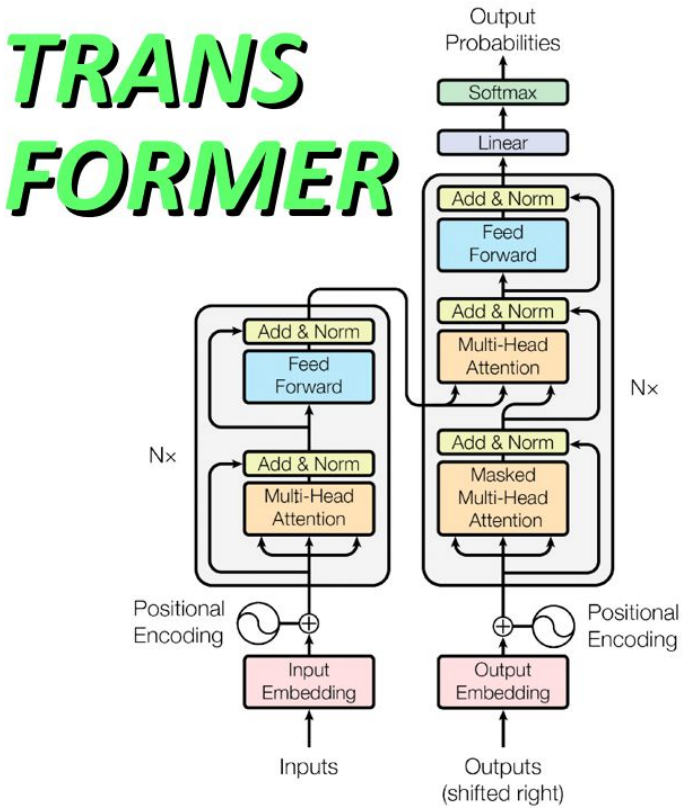
Response 1

Here are the **top generative AI trends shaping 2025**, based on industry momentum, research, and enterprise adoption:

1. Multimodal AI Becomes Mainstream

- **Trend:** Generative models now handle text, image, audio, and video seamlessly.
- **Key Players:** OpenAI's GPT-4.5 / GPT-5, Google Gemini, Meta's ImageBind, and Claude 3 Opus.

Origins of GenAI: Transformer Model



After introduction of Transformer model in 2017 and consequent release of ChatGPT model by OpenAI in 2022.

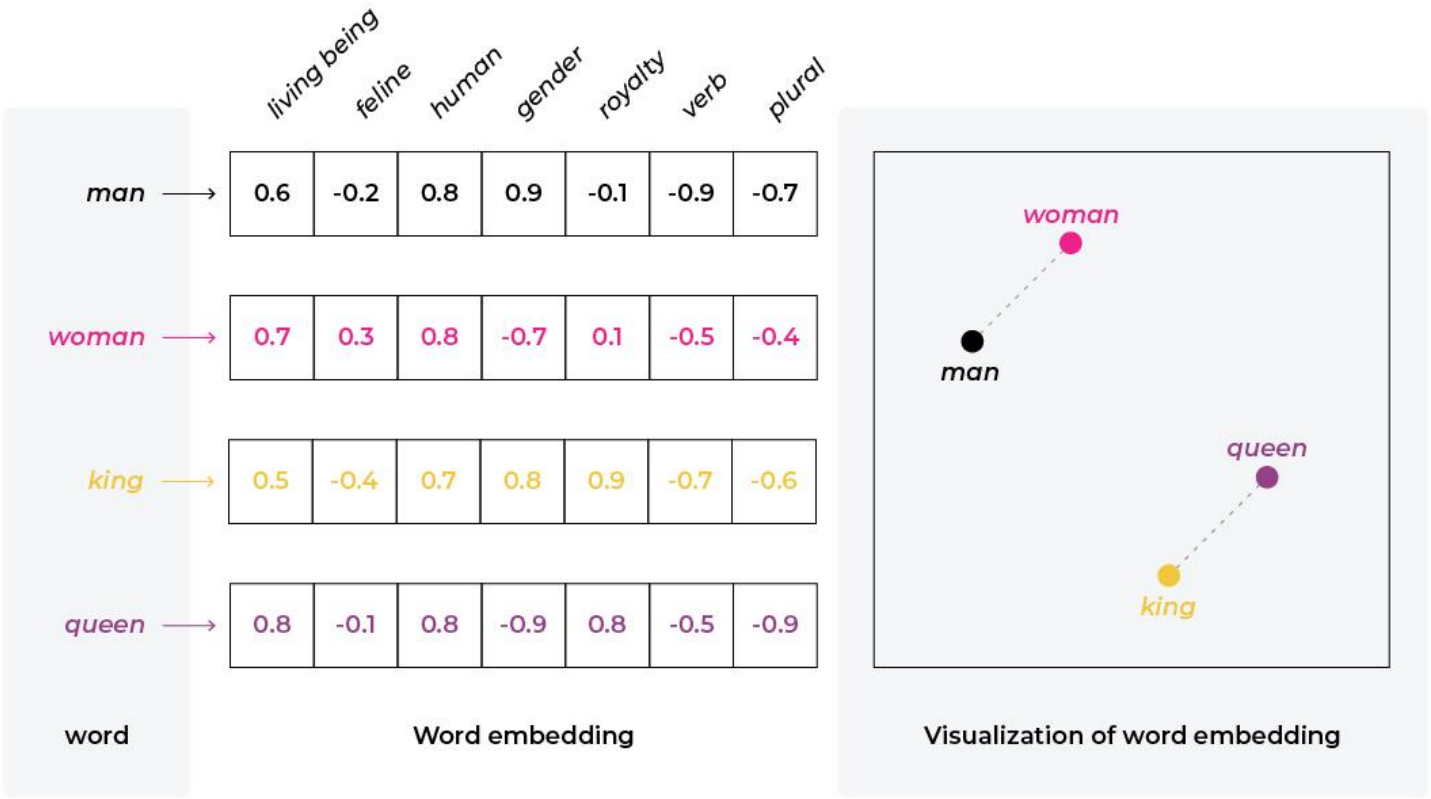
The world of AI has changed forever.

It became possible to interact with Large Language Models (LLMs) like with normal human vis-à-vis collocutor.

Word Embedding

Origins of GenAI: Embeddings

One of the key features enabled by LLMs is embedding -> a way to transform natural words and sentences into the multi-dim numerical vectors.



https://arize.com/blog-course/embeddings-meaning-examples-and-how-to-compute/

Origins of GenAI: Prompts

We use recently proposed techniques for measuring the quality of the resulting vector representations, with the expectation that not only will similar words tend to be close to each other, but that words can have **multiple degrees of similarity** [20]. This has been observed earlier in the context of inflectional languages - for example, nouns can have multiple word endings, and if we search for similar words in a subspace of the original vector space, it is possible to find words that have similar endings [13, 14].

Somewhat surprisingly, it was found that similarity of word representations goes beyond simple syntactic regularities. Using a word offset technique where simple algebraic operations are performed on the word vectors, it was shown for example that *vector("King") - vector("Man") + vector("Woman")* results in a vector that is closest to the vector representation of the word *Queen* [20].

Mikolov et al. (2013) Efficient Estimation of Word Representations in Vector Space
https://arxiv.org/pdf/1301.3781

Origins of GenAI: Prompts

Sentences:

1.The cat runs fast and the dog is slow.
2.The dog runs but the cat is very fast.
3. Slow is the dog, but the cat runs fast.

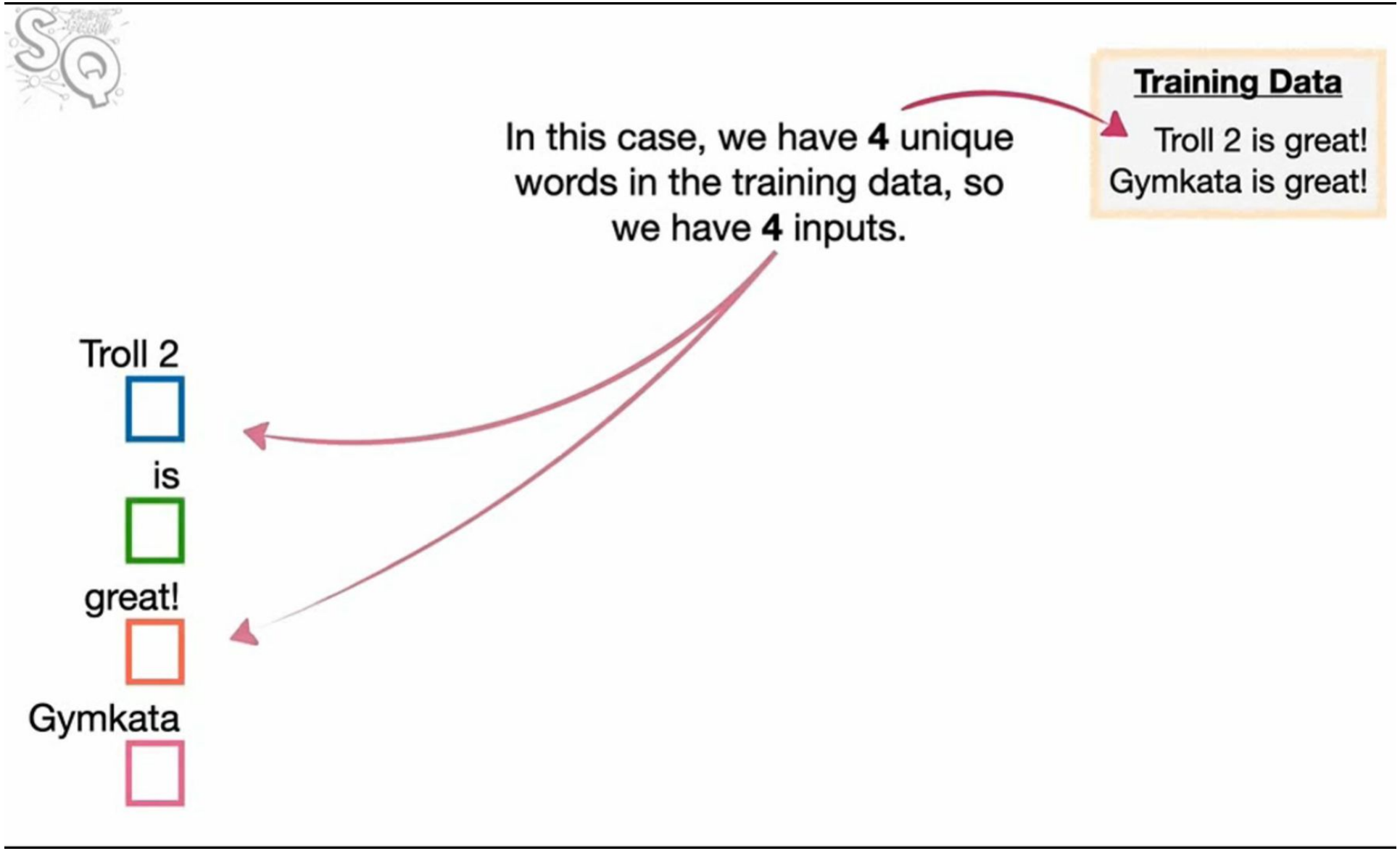
Unique Words: cat, runs, fast, dog, slow, but, the, and, is, very

cat	runs	fast	dog	slow	but	the	and	is	very
1	2	3	4	5	6	7	8	9	0

Origins of GenAI: One-Hot-Vector

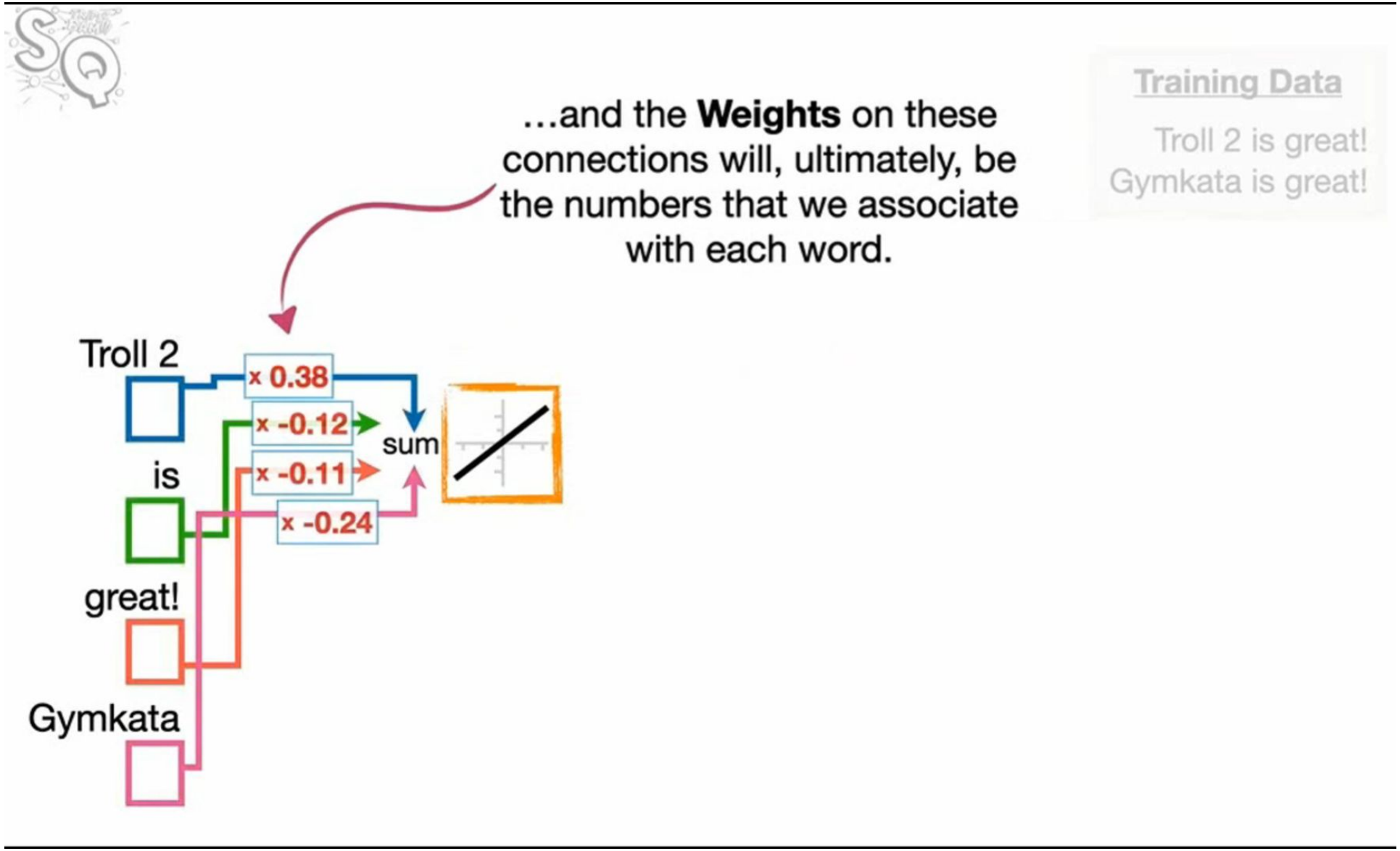
cat	Runs	Fast	dog	slow	but	the	and	is	very
1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

Word Embeddings: Word2Vec



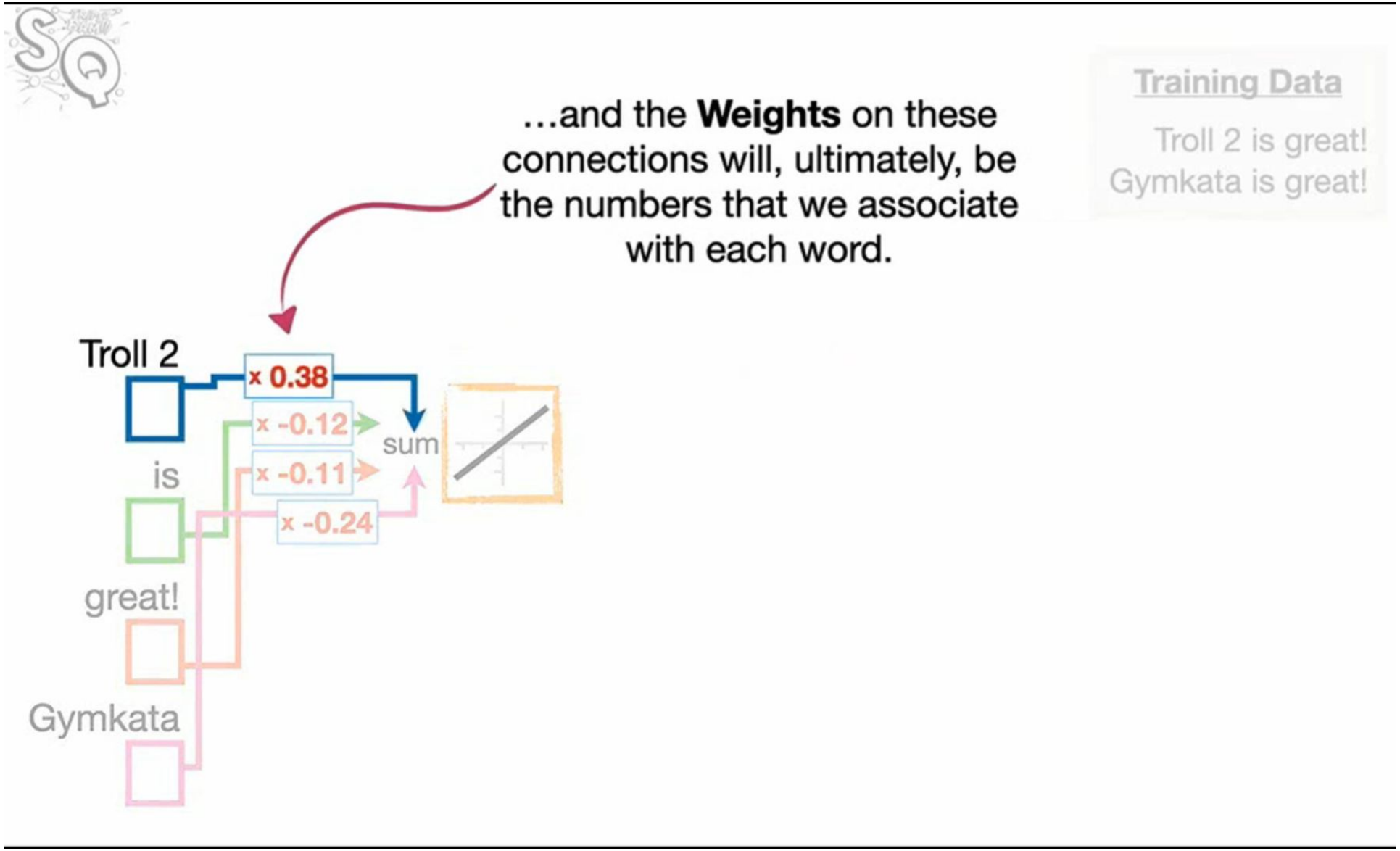
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



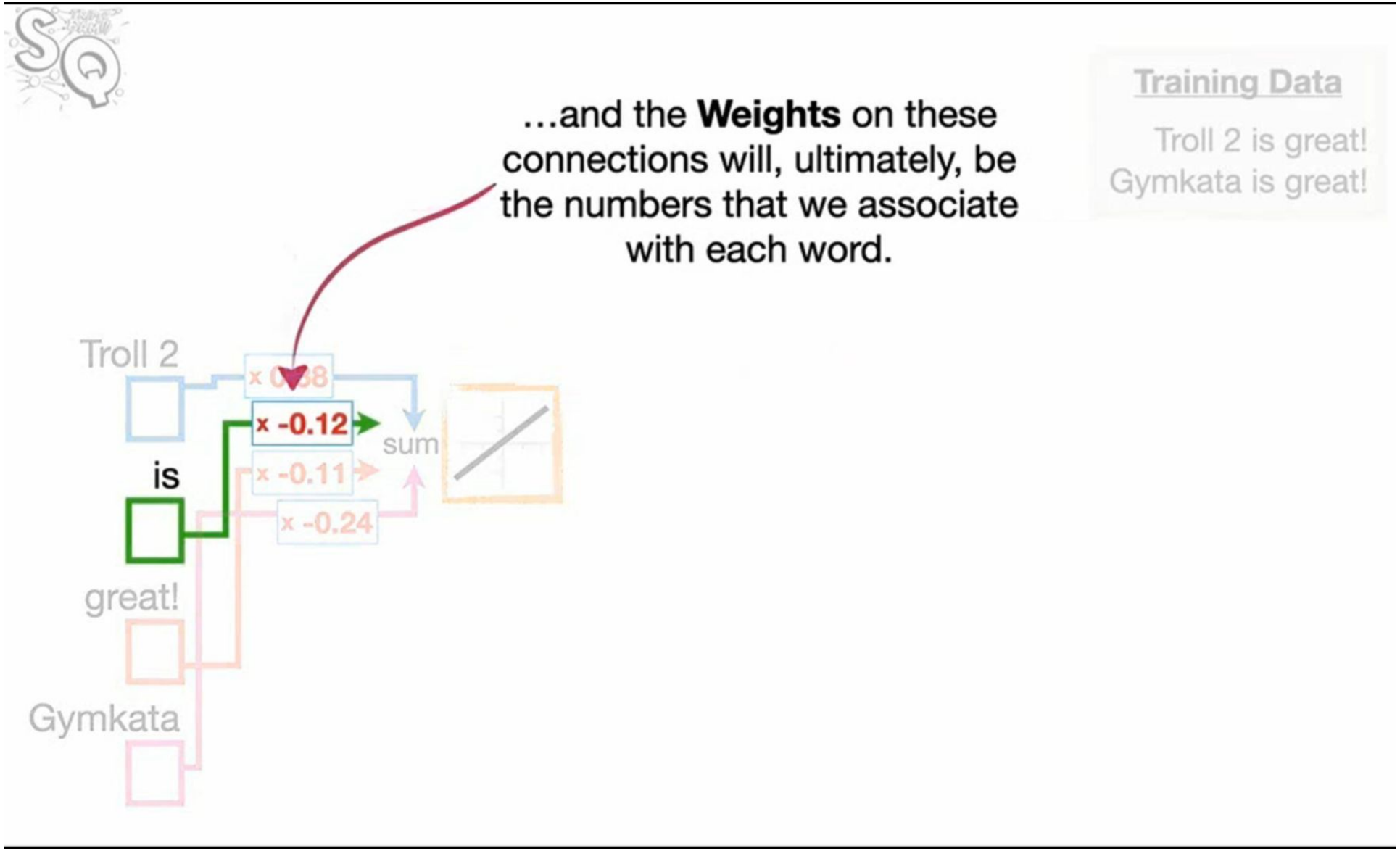
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



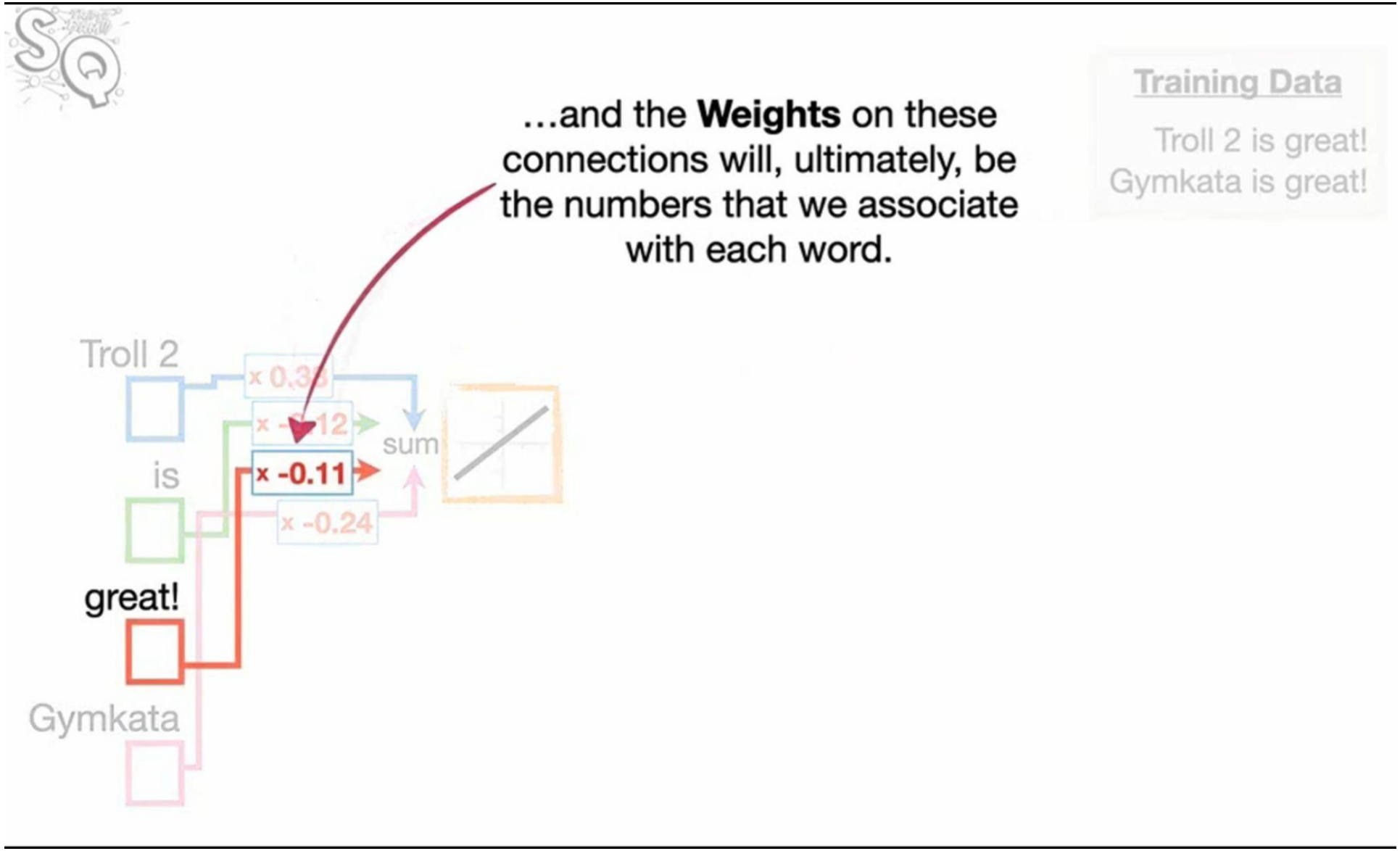
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



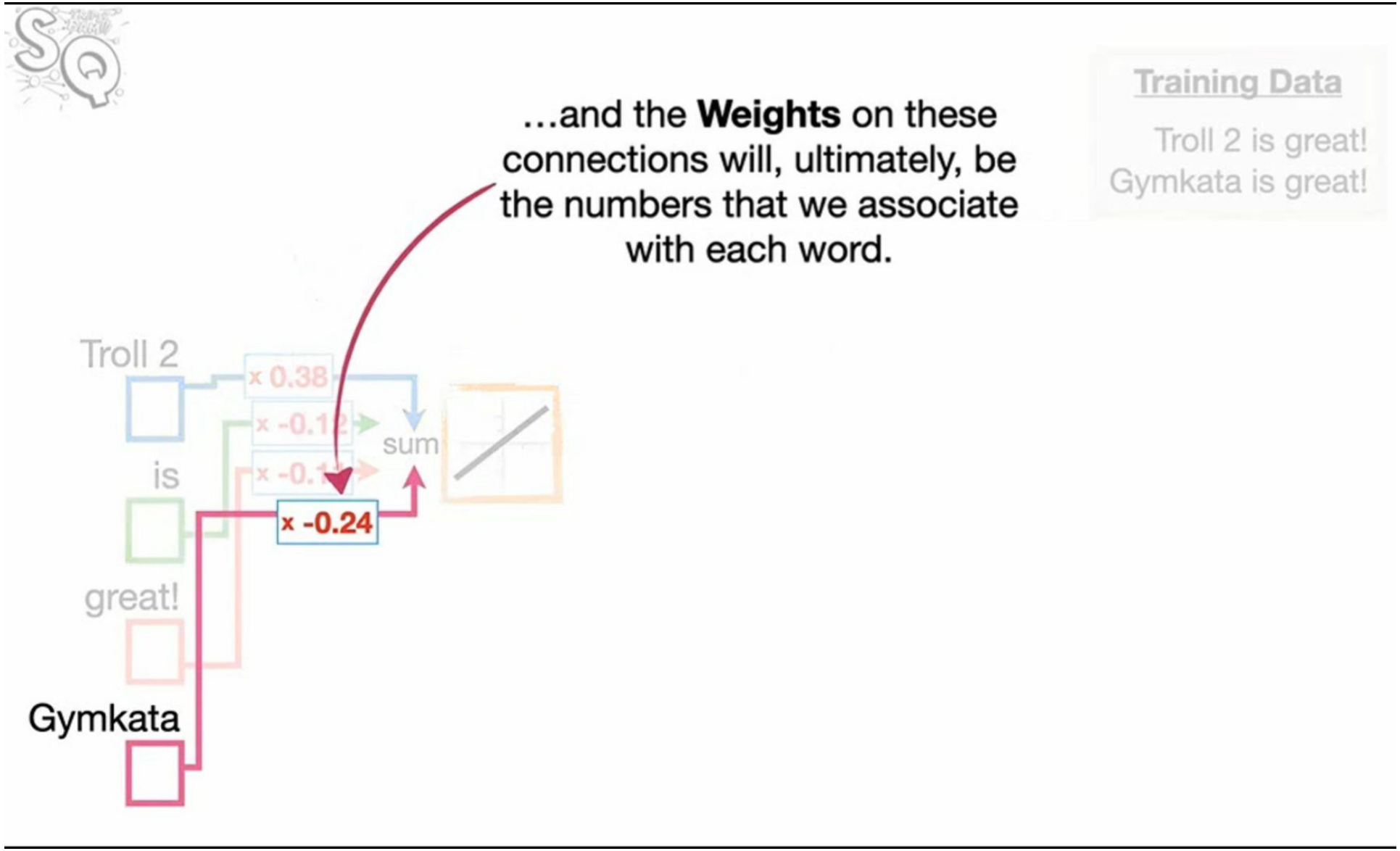
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



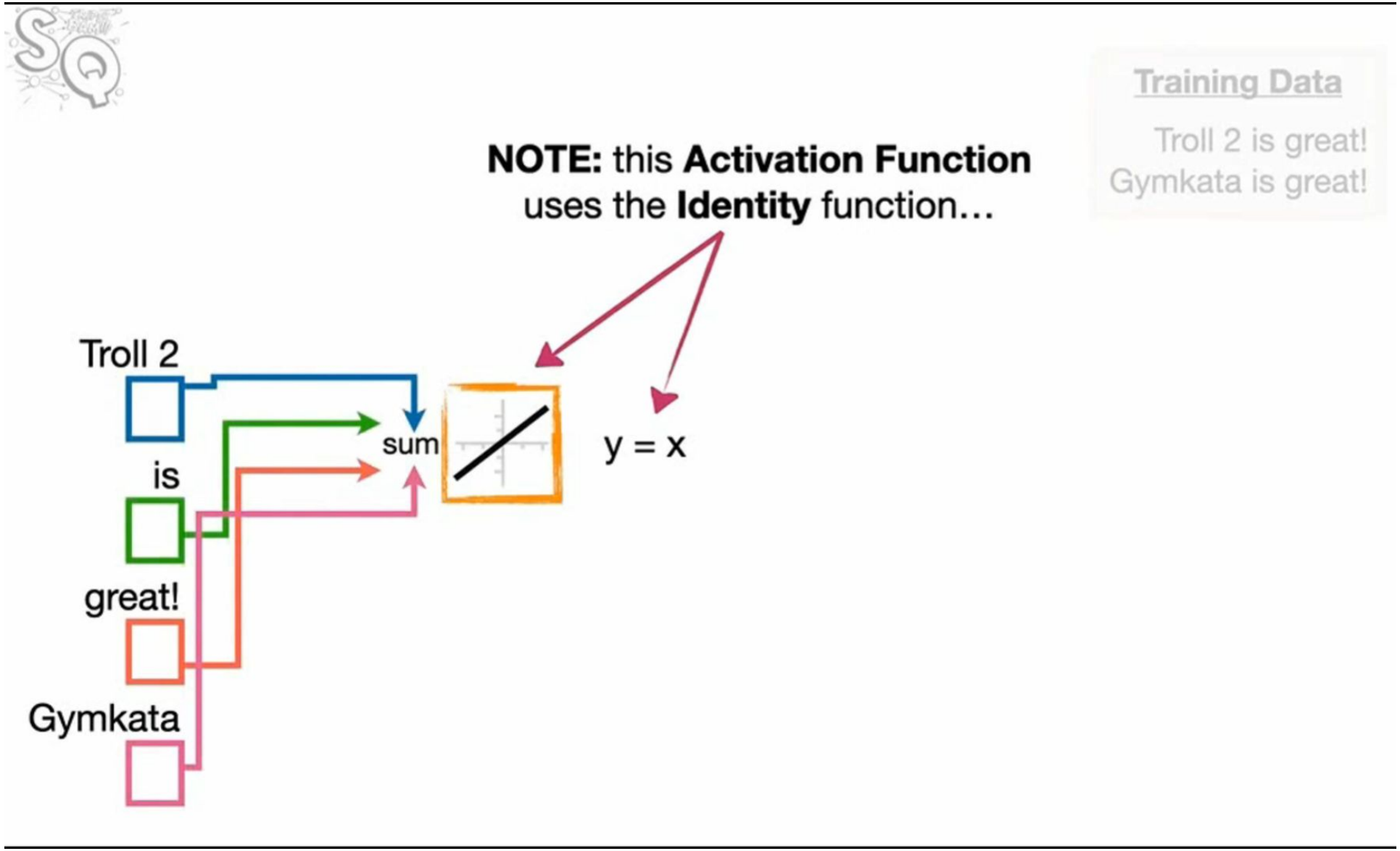
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



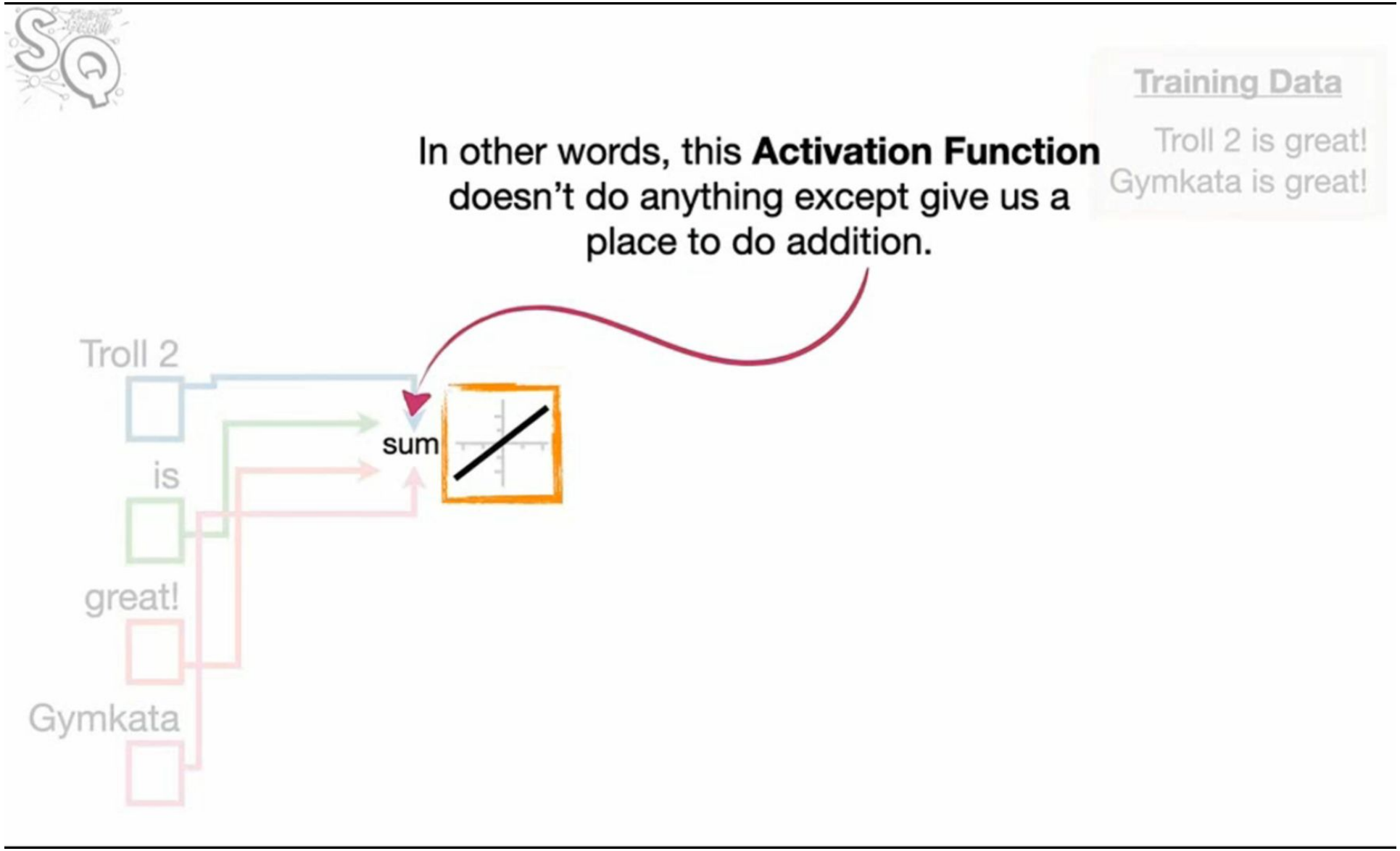
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



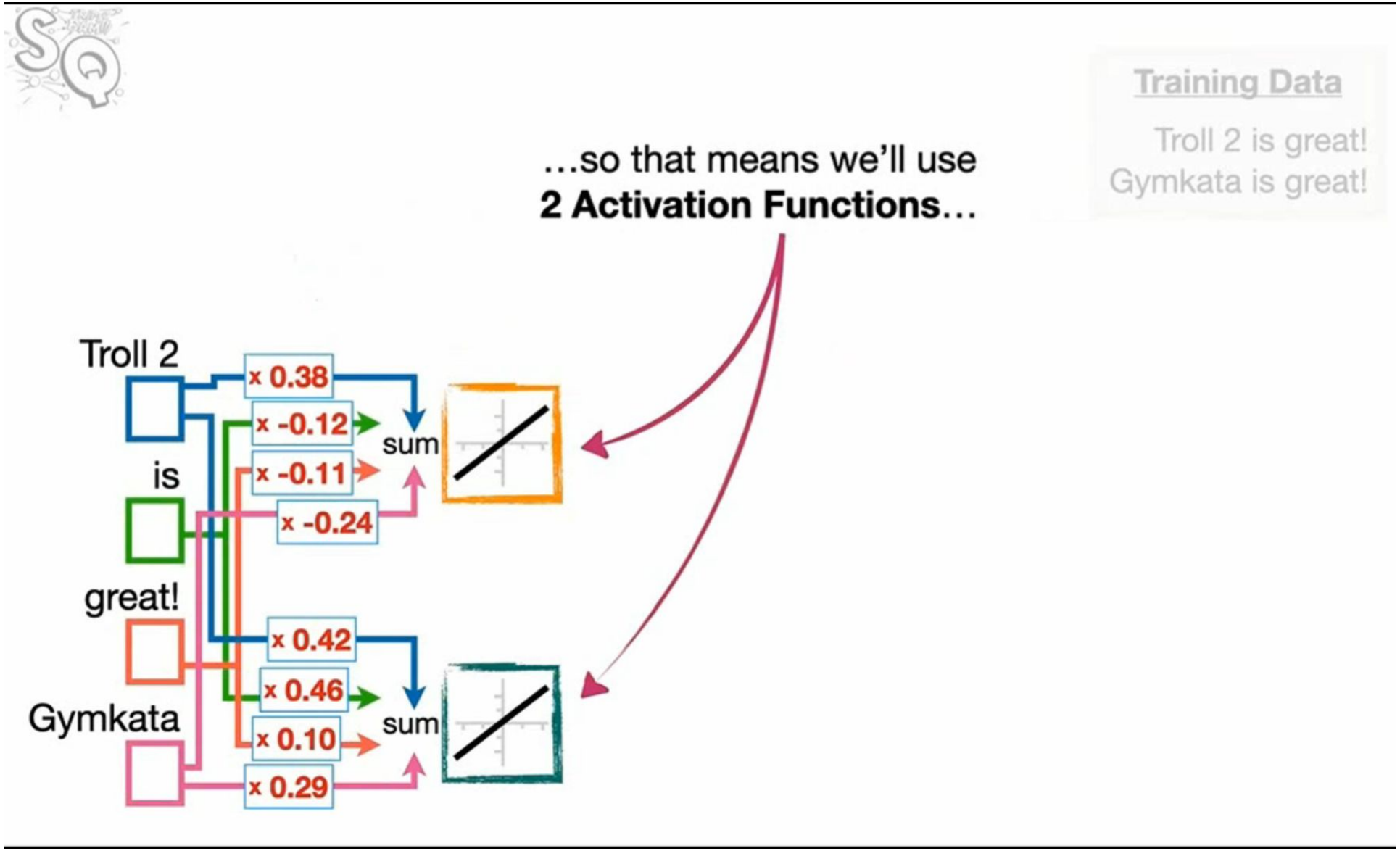
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



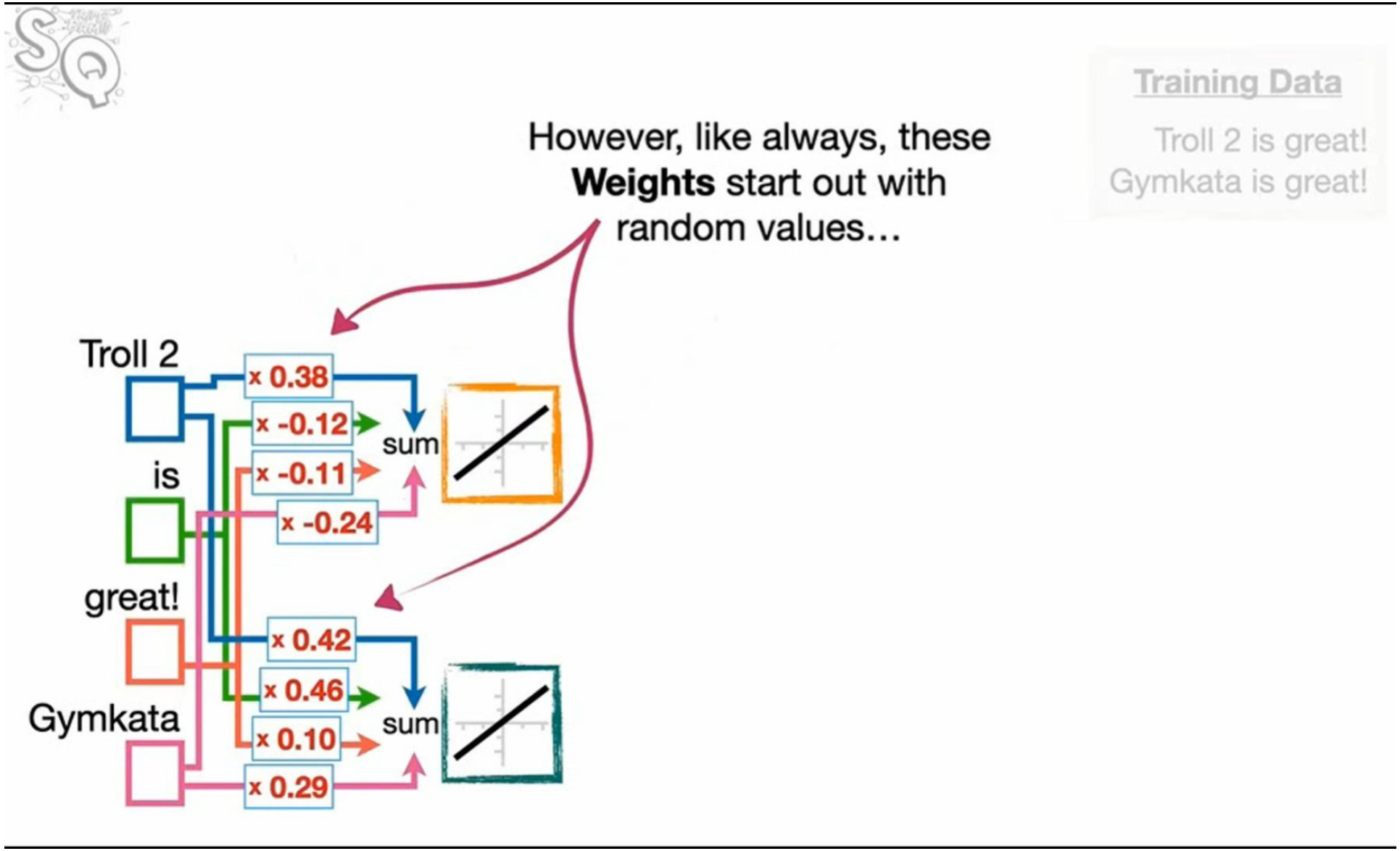
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



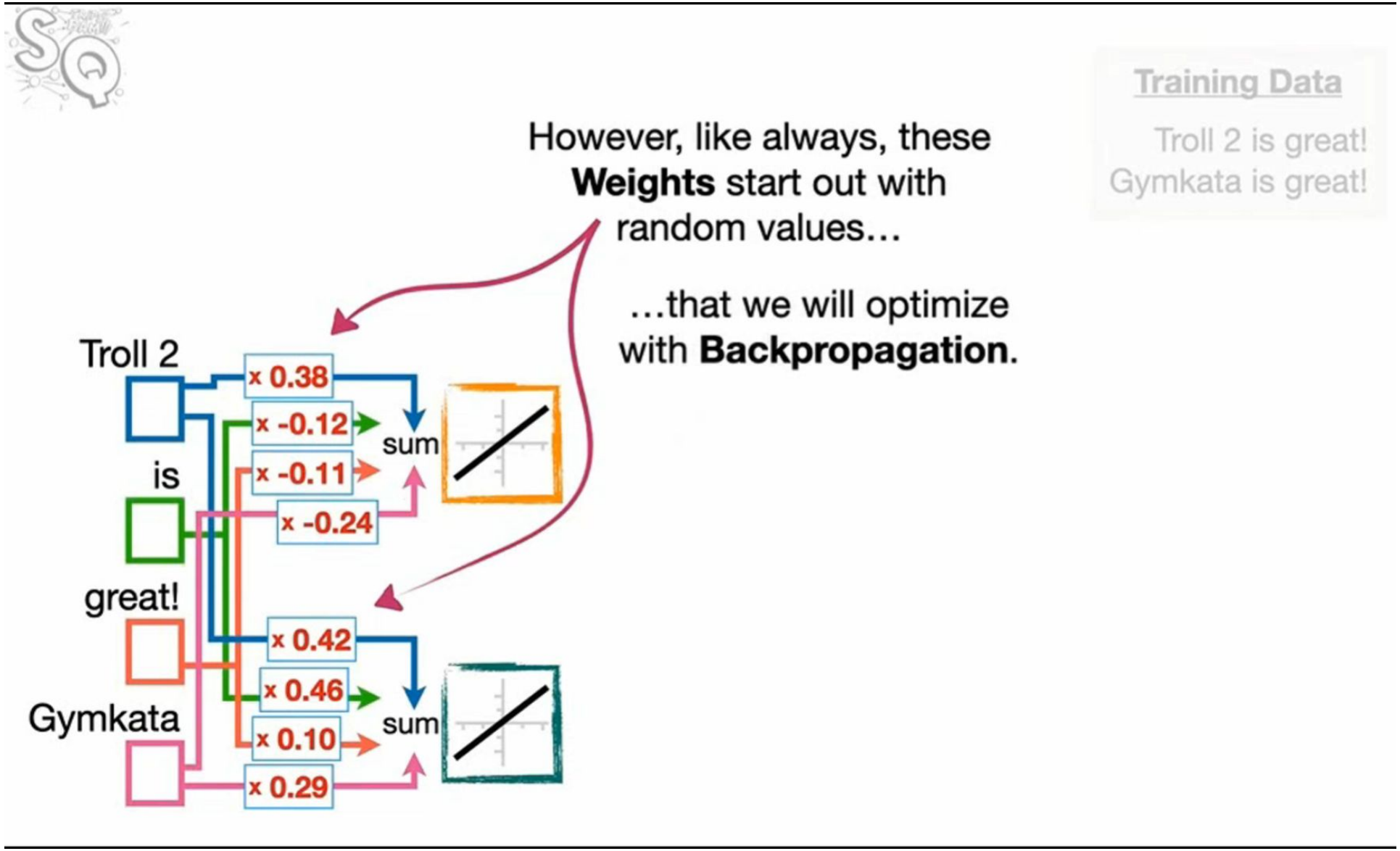
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



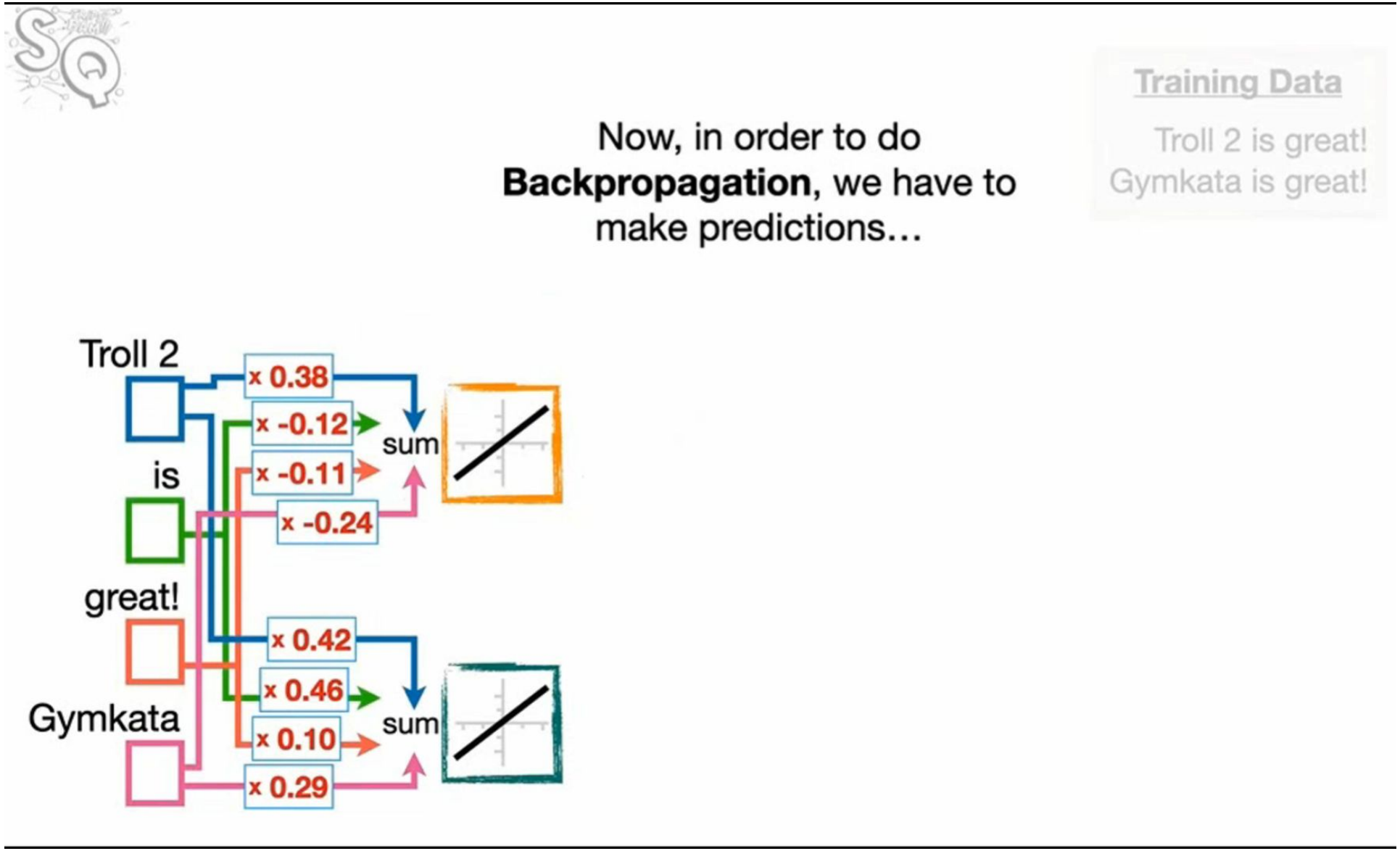
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

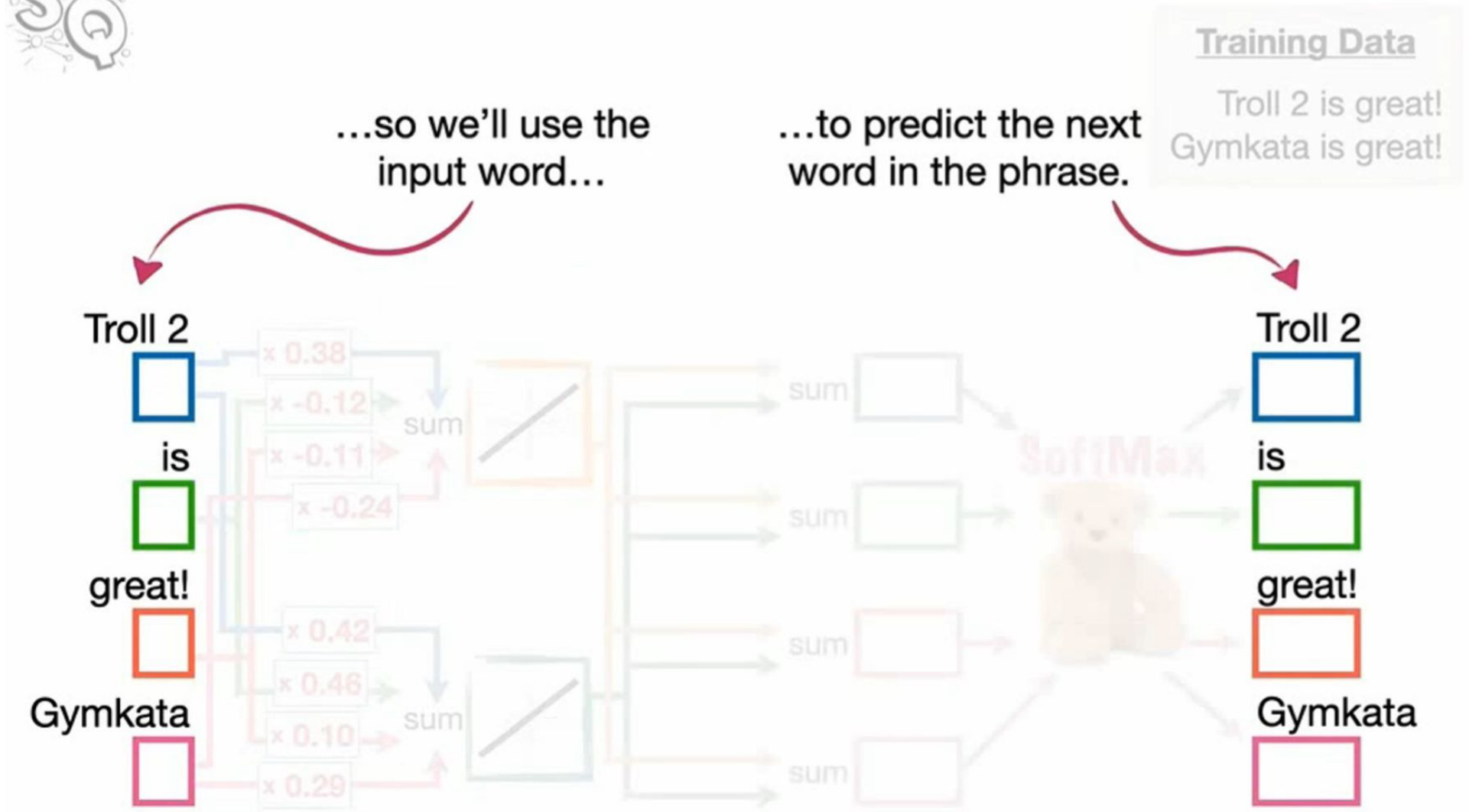
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

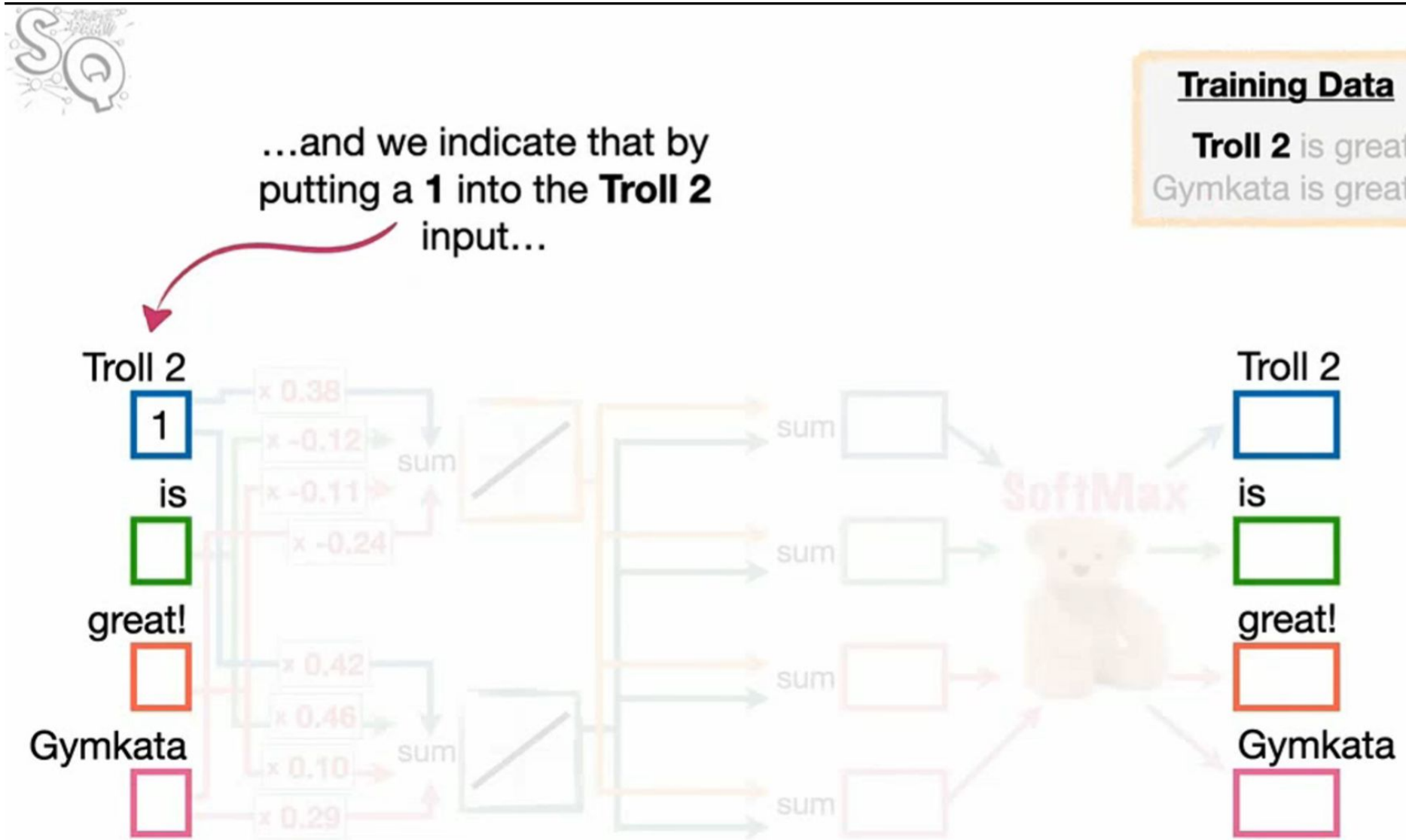
Word Embeddings: Word2Vec

Word Embeddings: Word2Vec



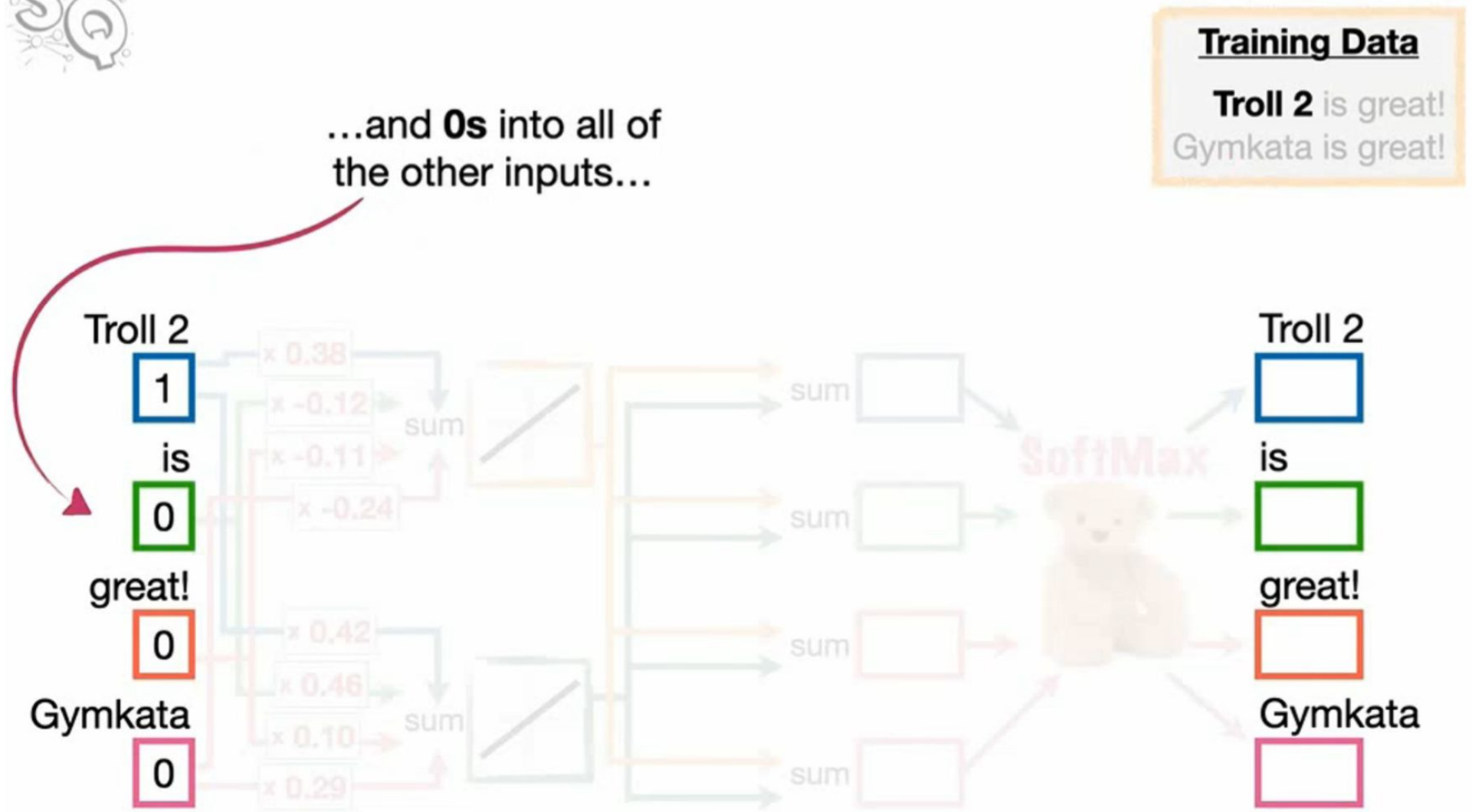
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0


Training Data
Troll 2 is great!
Gymkata is great!
…and we indicate that by putting a 1 into the Troll 2 input…
Troll 2
1
is
great!
Gymkata
x 0.38
x -0.12
x -0.11
x -0.24
x 0.42
x 0.46
x 0.10
x 0.29
sum
sum
sum
sum
sum
SoftMax
Troll 2
is
great!
Gymkata

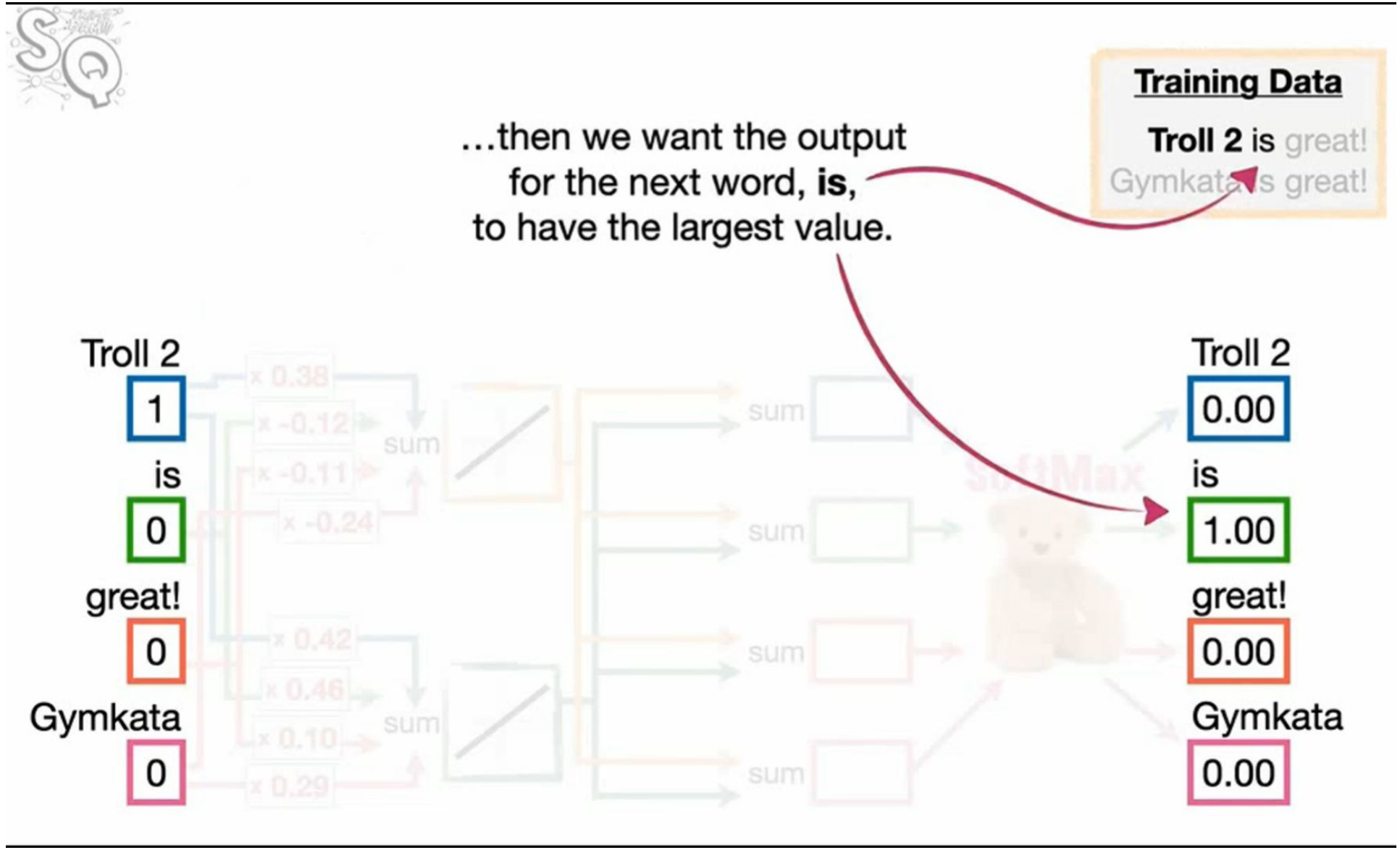
Word Embeddings: Word2Vec

Word Embeddings: Word2Vec

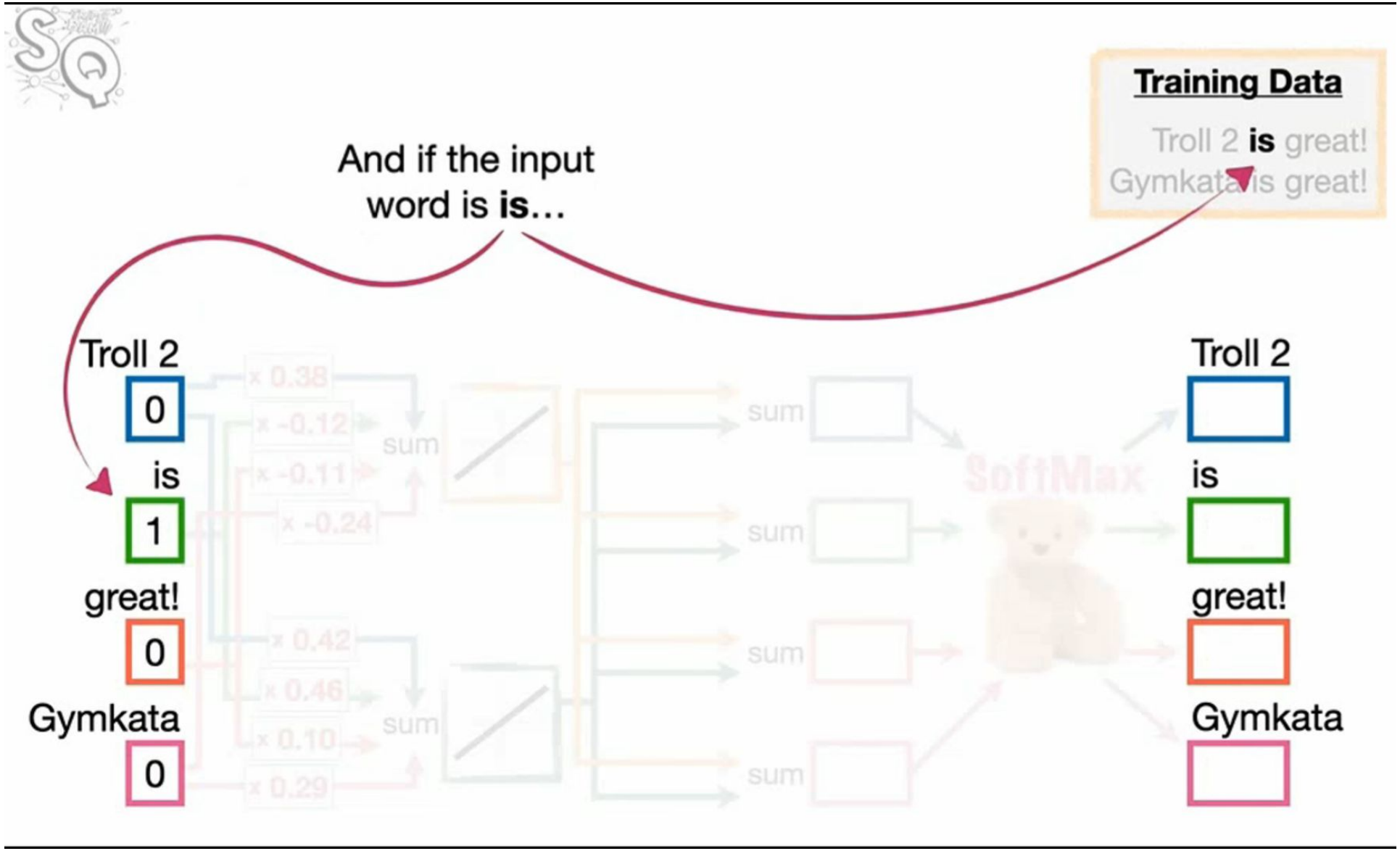


Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0


Training Data
Troll 2 is great!
Gymkata is great!
…then we want the output
for the next word, is,
to have the largest value.
Troll 2
1
is
0
great!
0
Gymkata
0
x 0.38
x -0.12
x -0.11
x -0.24
x 0.42
x 0.46
x 0.10
x 0.29
sum
sum
sum
sum
sum
SoftMax
Troll 2
0.00
is
1.00
great!
0.00
Gymkata
0.00

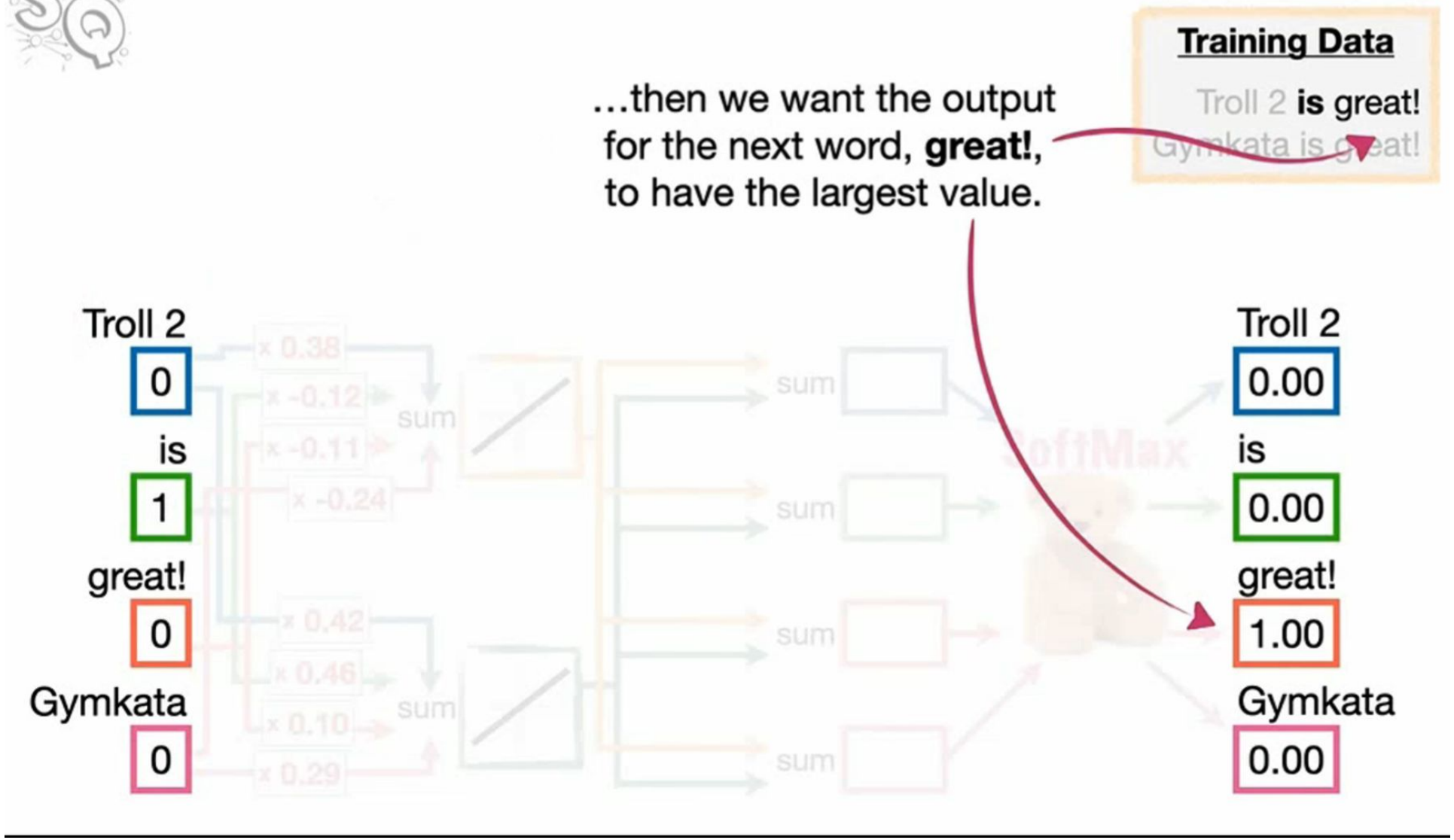
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec

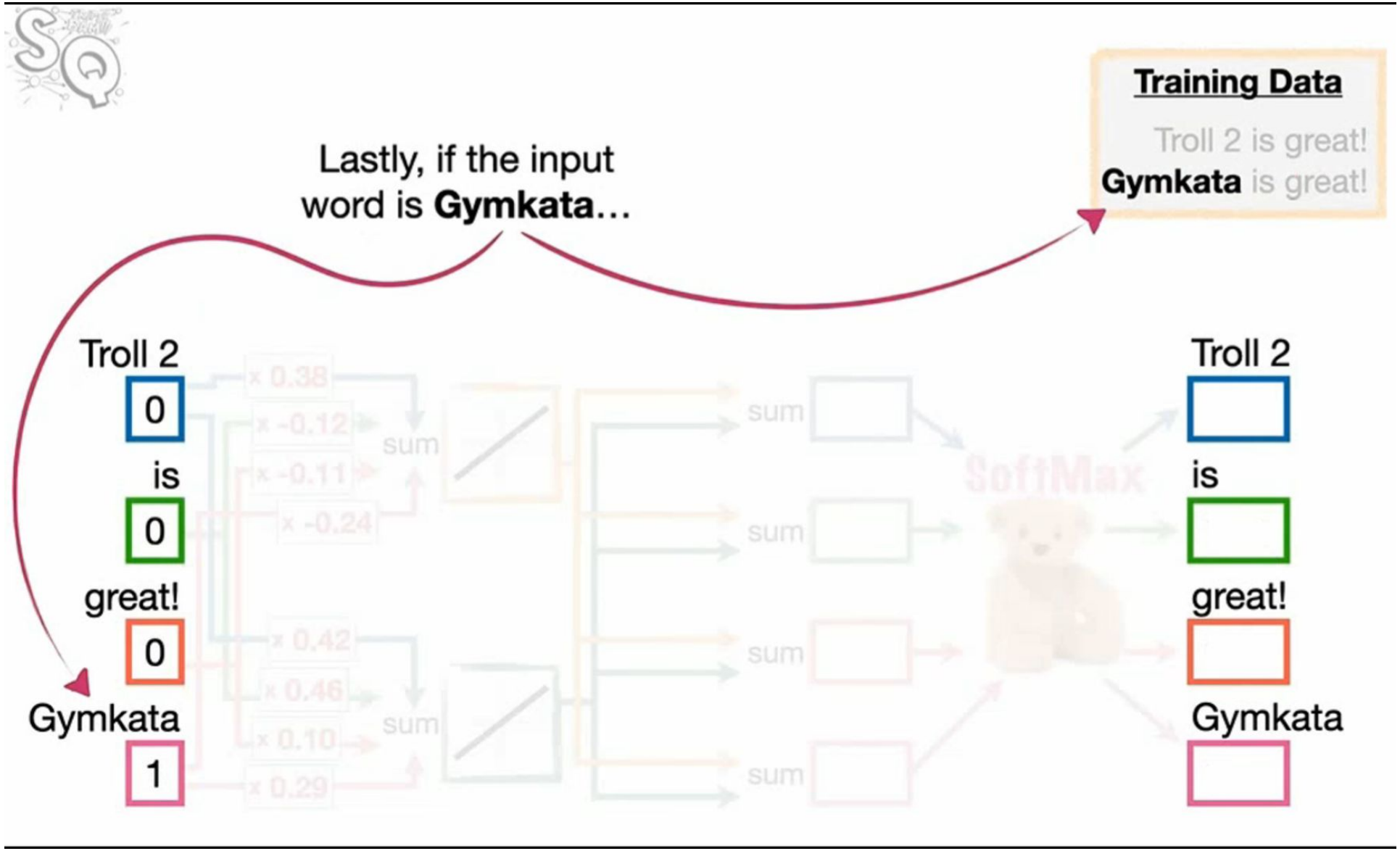
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

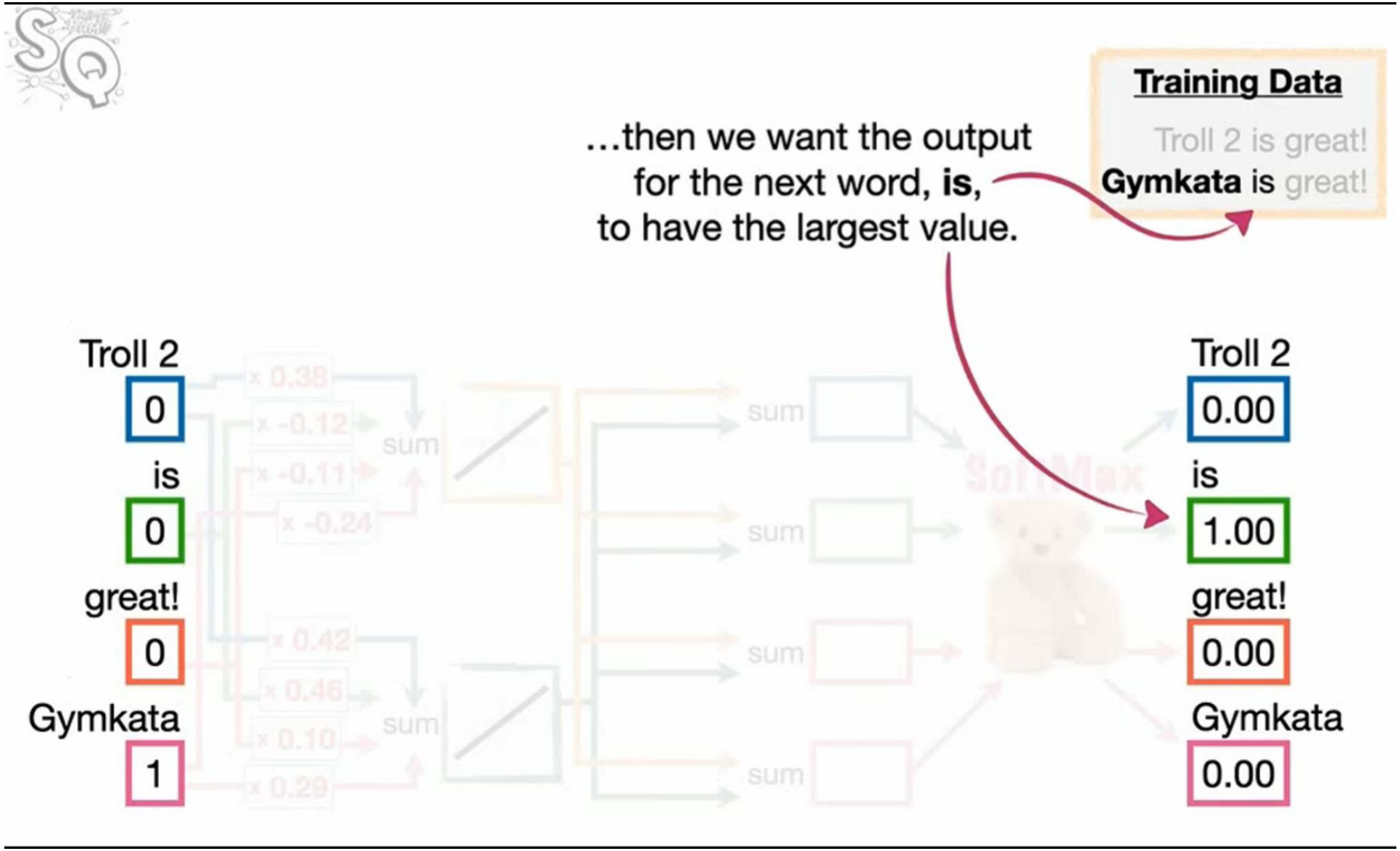
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec

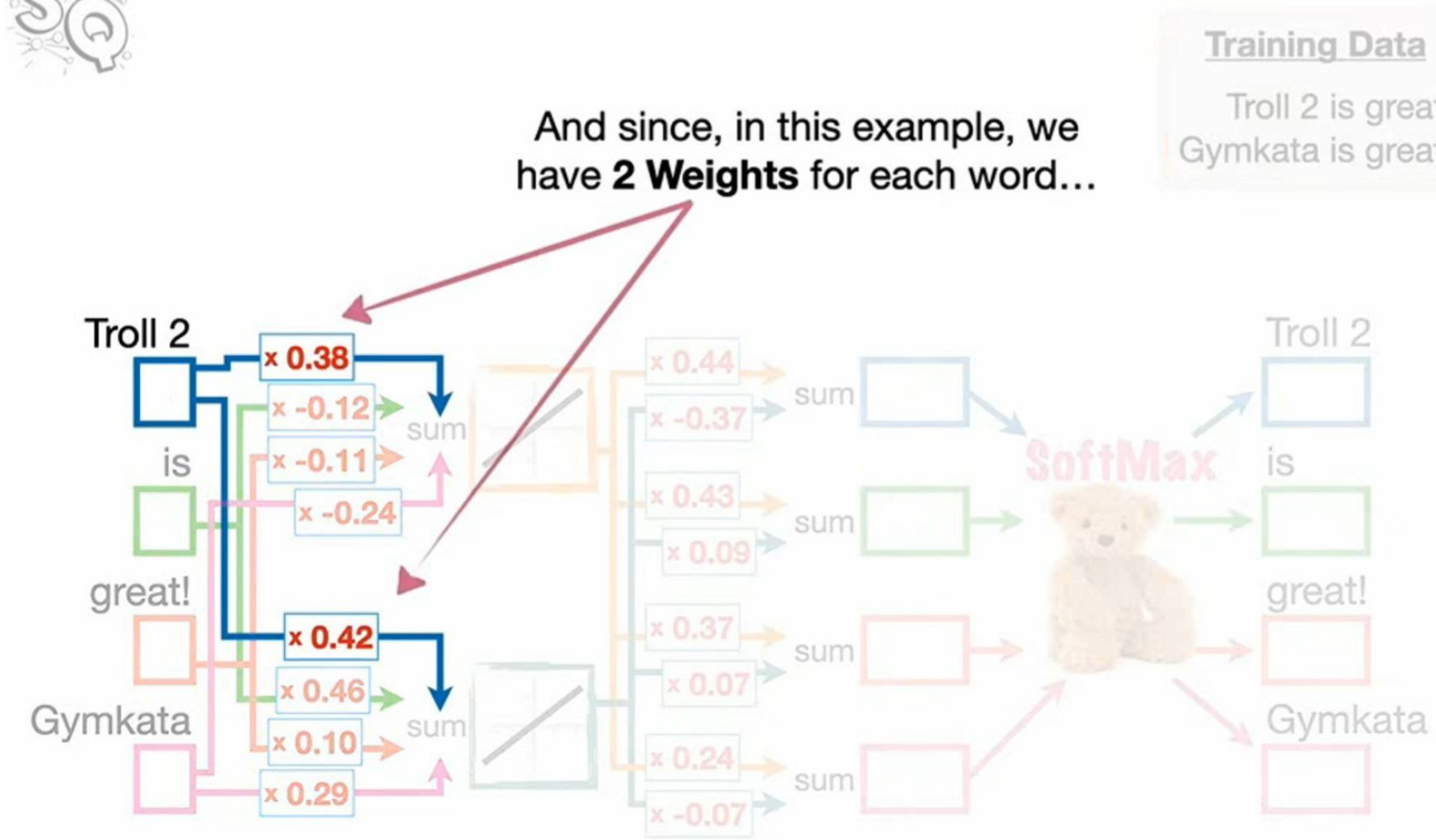


Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

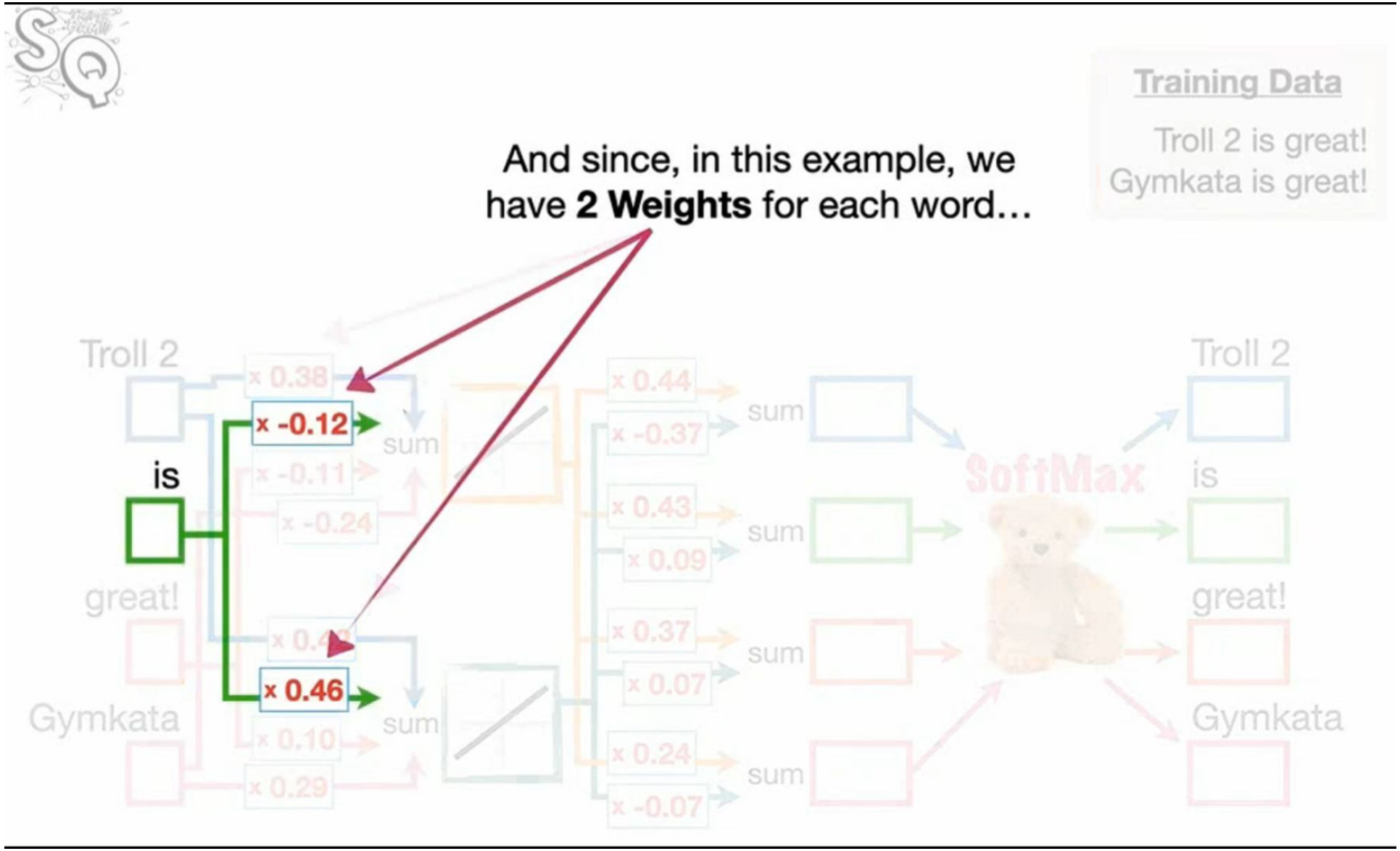
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

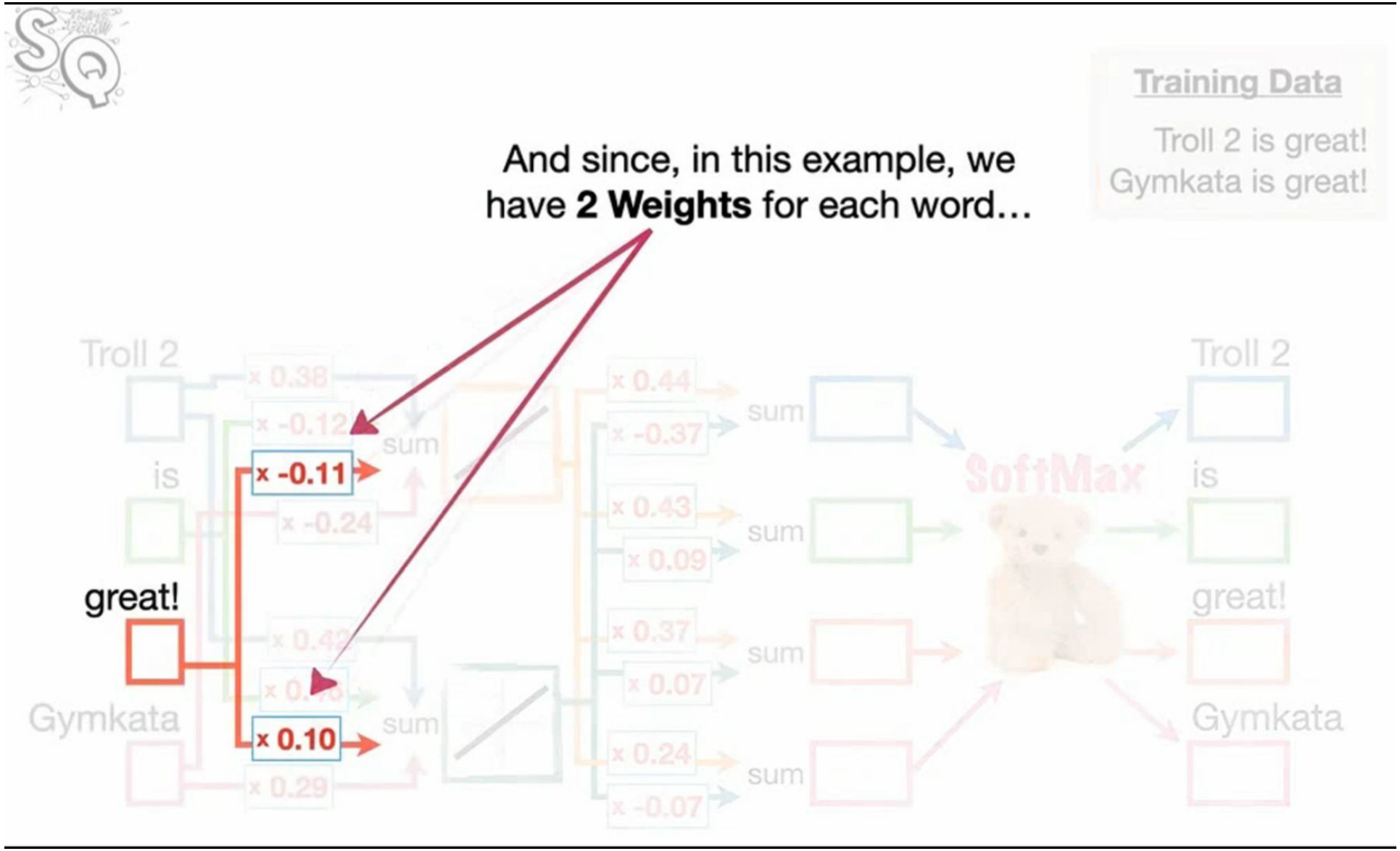

Training Data
Troll 2 is great!
Gymkata is great!
And since, in this example, we have 2 Weights for each word…
Troll 2
is
great!
Gymkata
x 0.38
x -0.12
x -0.11
x -0.24
x 0.42
x 0.46
x 0.10
x 0.29
sum
sum
x 0.44
x -0.37
x 0.43
x 0.09
x 0.37
x 0.07
x 0.24
x -0.07
sum
sum
sum
sum
SoftMax
Troll 2
is
great!
Gymkata

Word Embeddings: Word2Vec



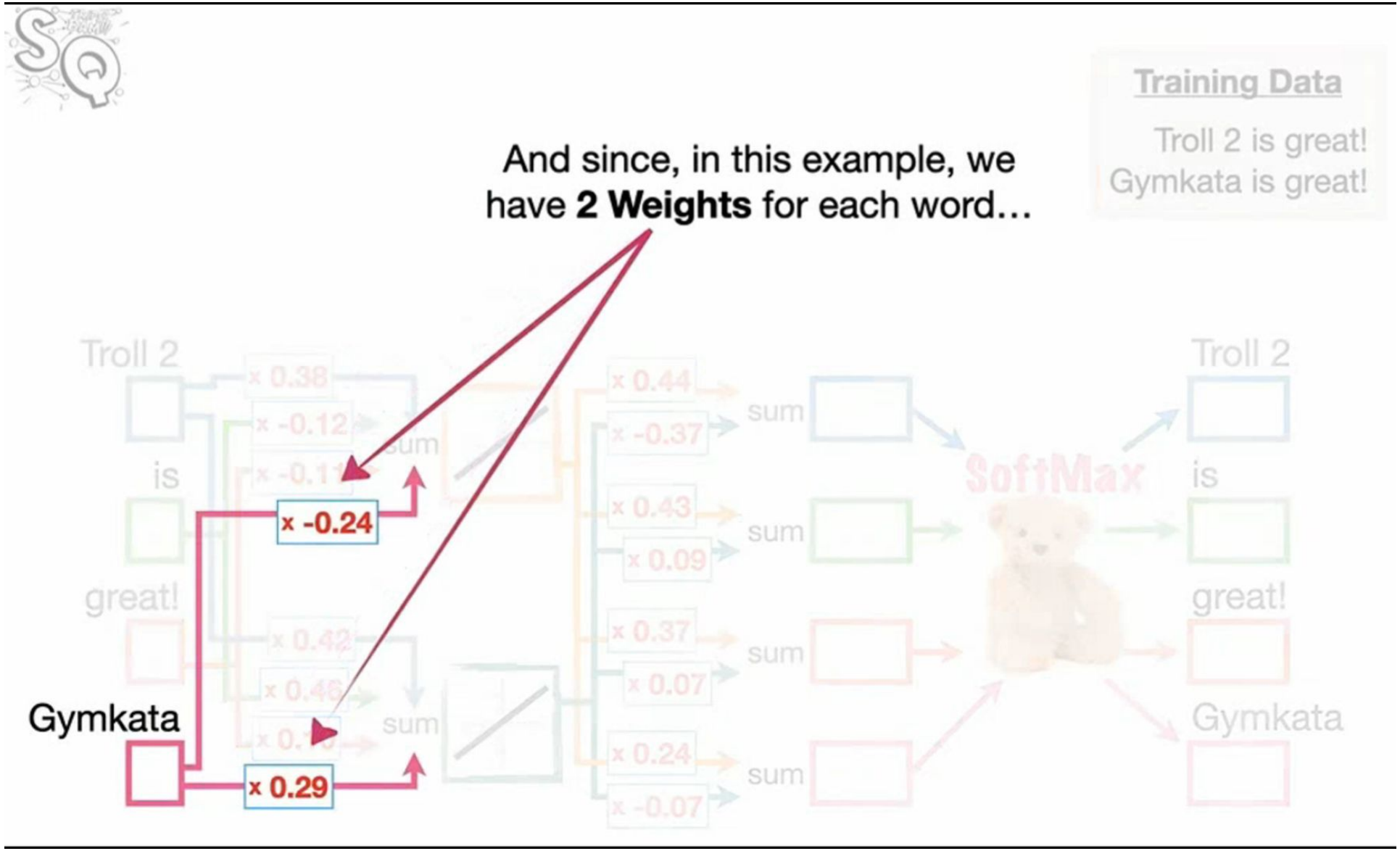
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



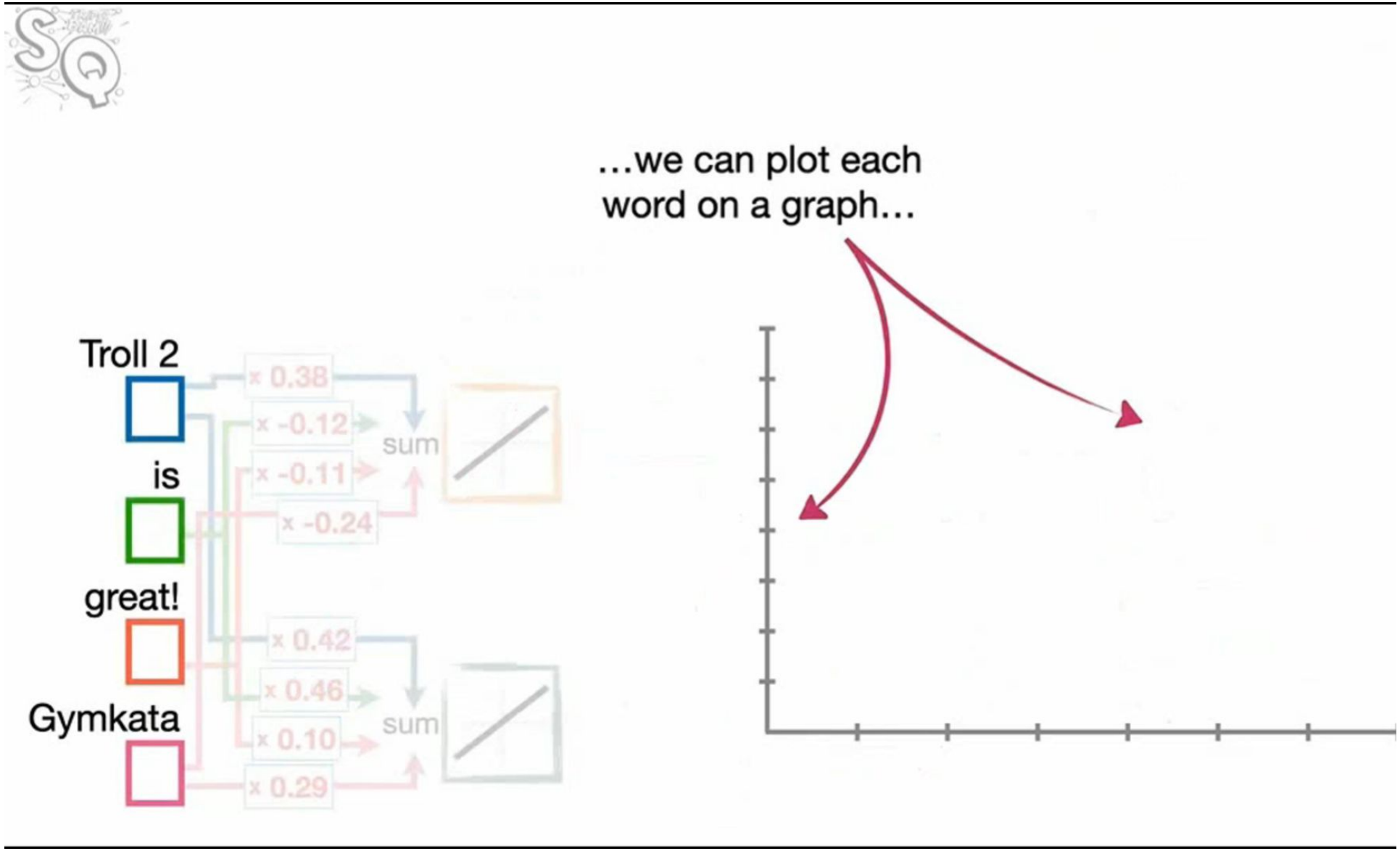
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



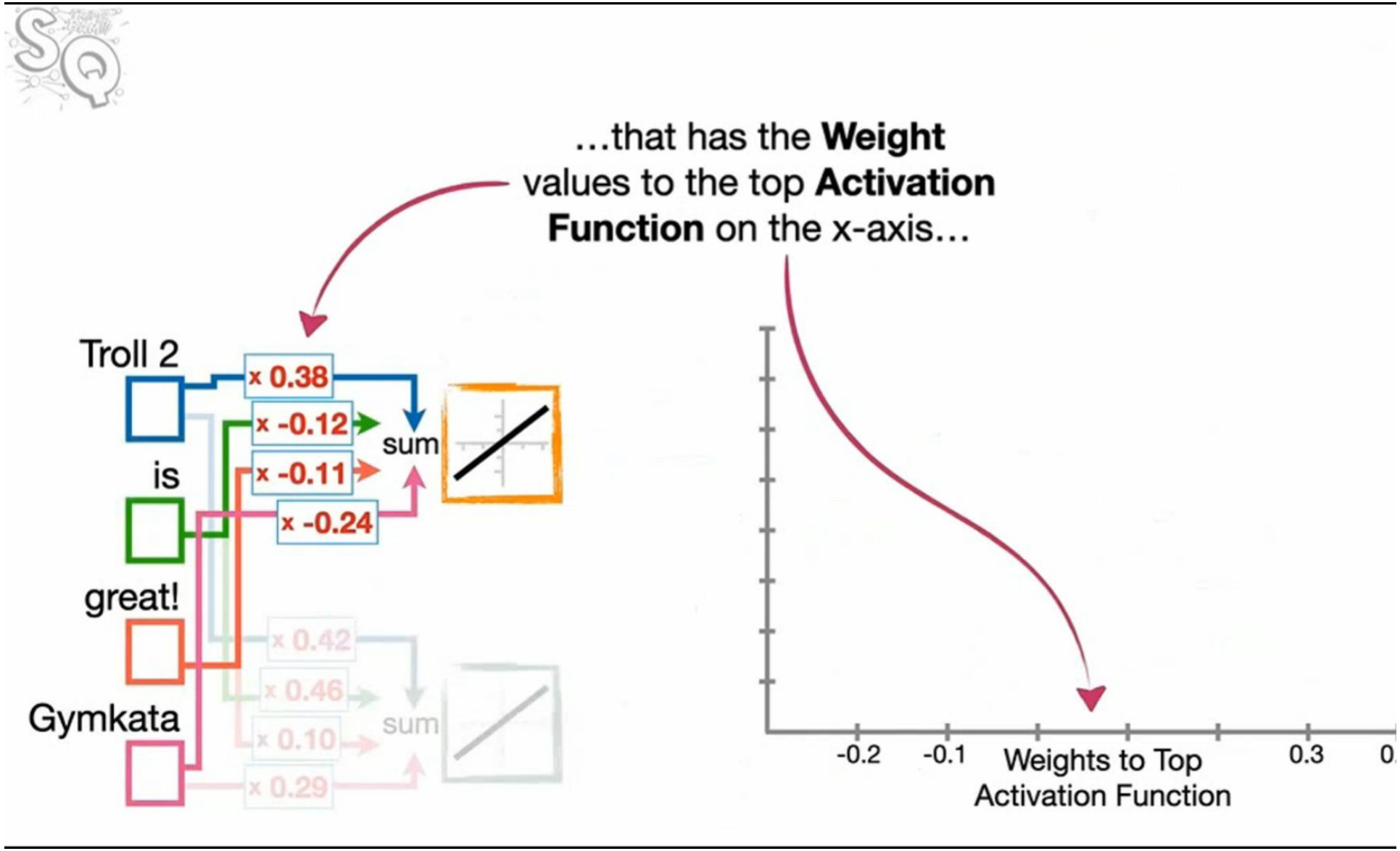
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



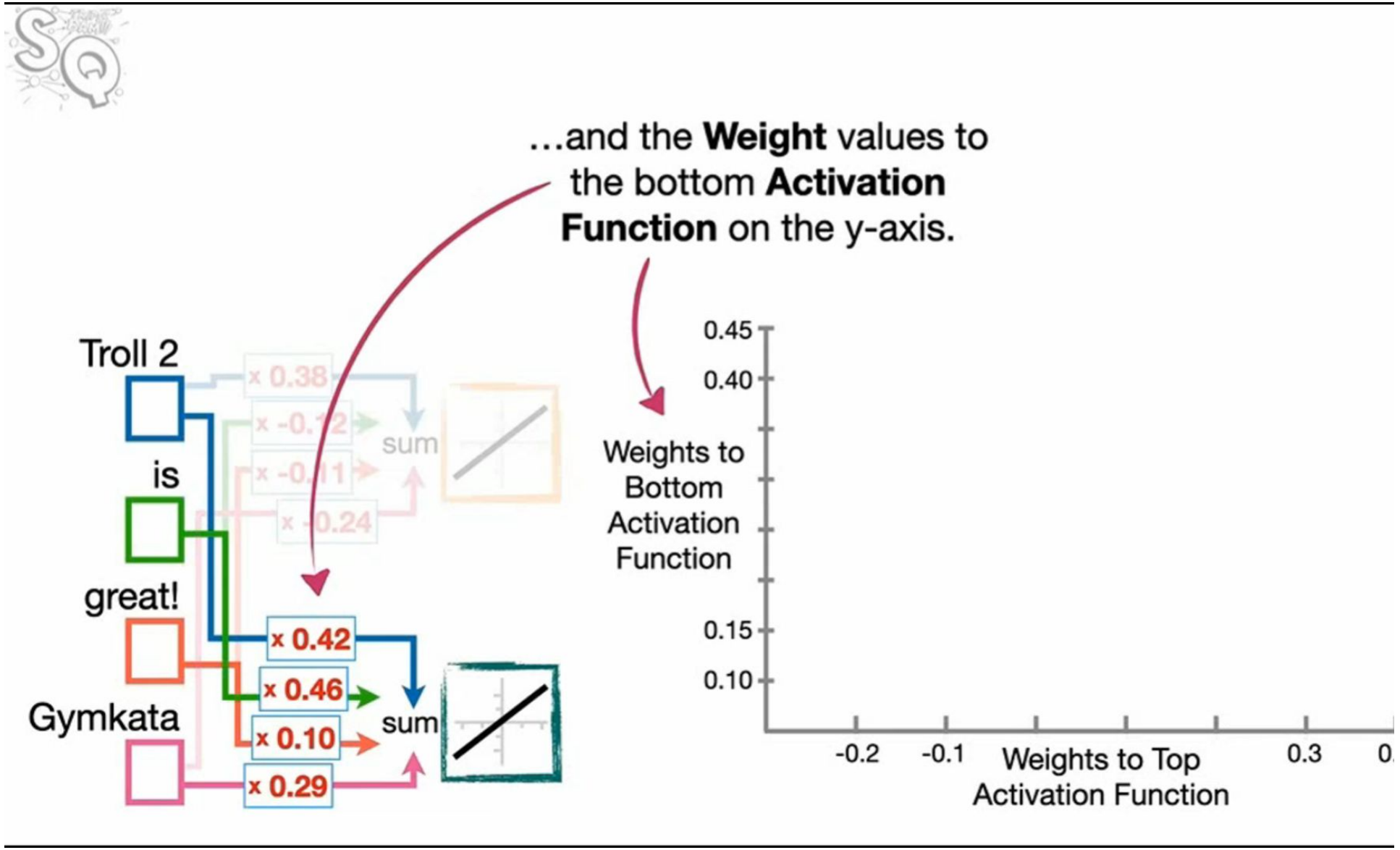
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

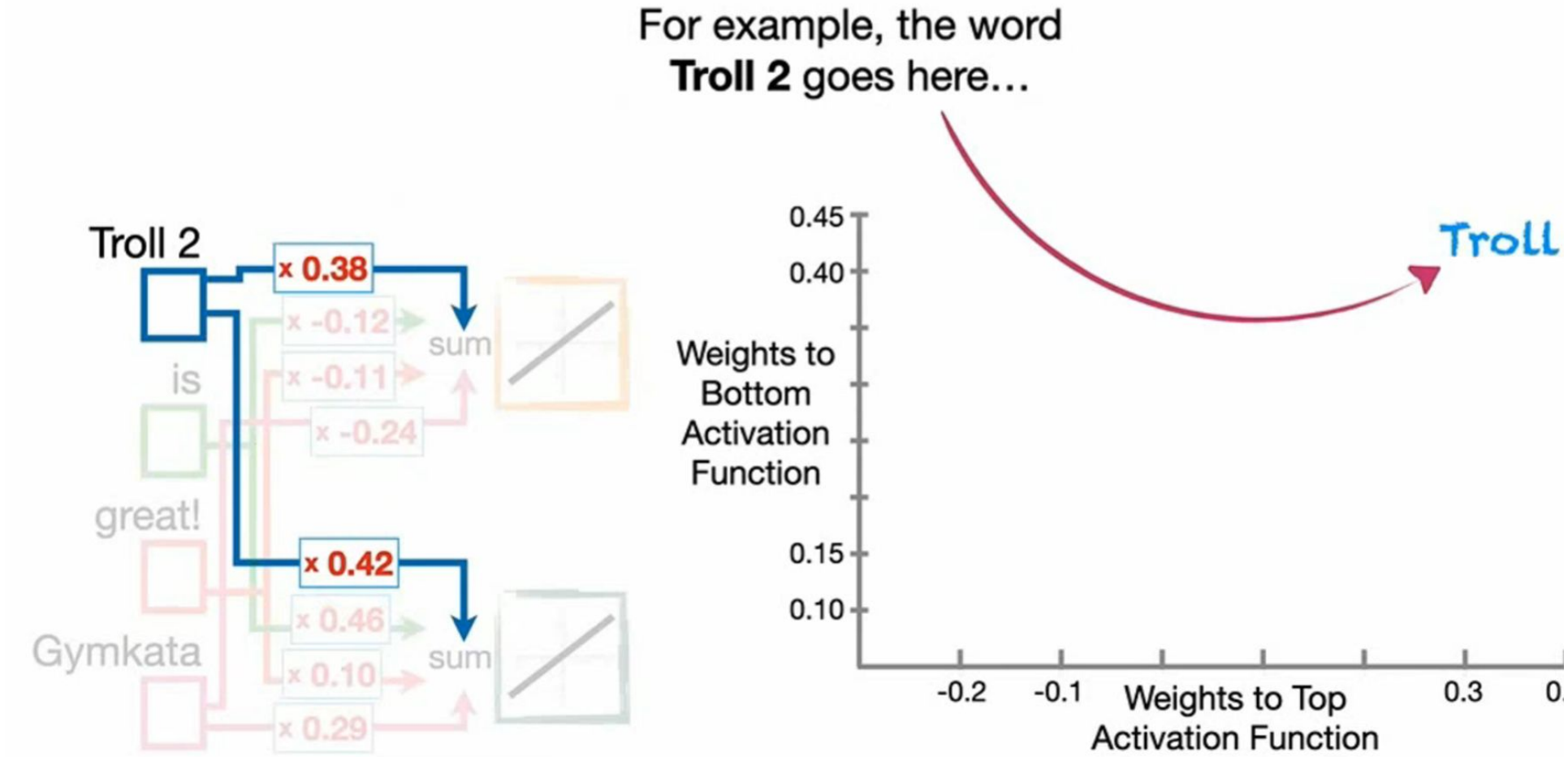
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

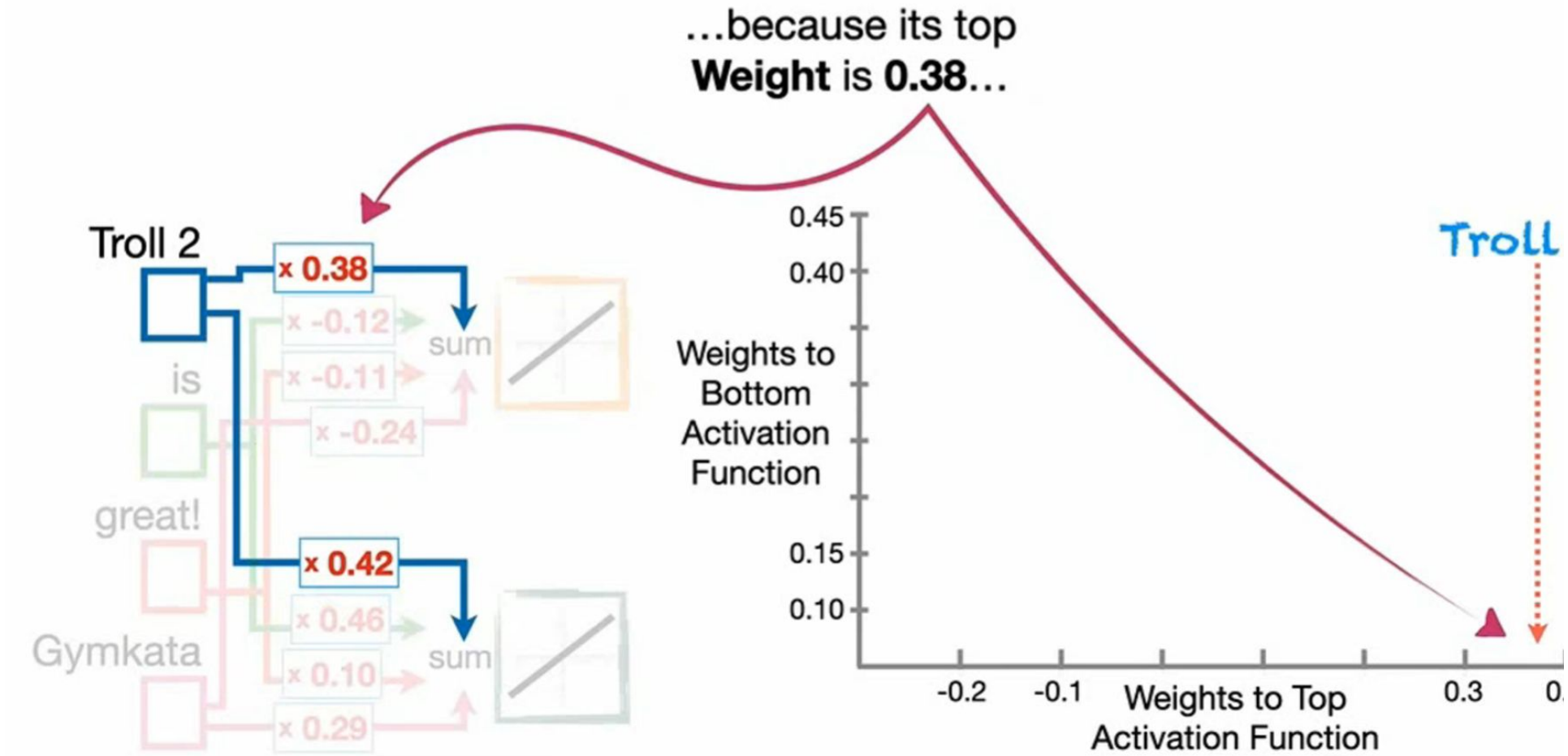
Word Embeddings: Word2Vec

Word Embeddings: Word2Vec



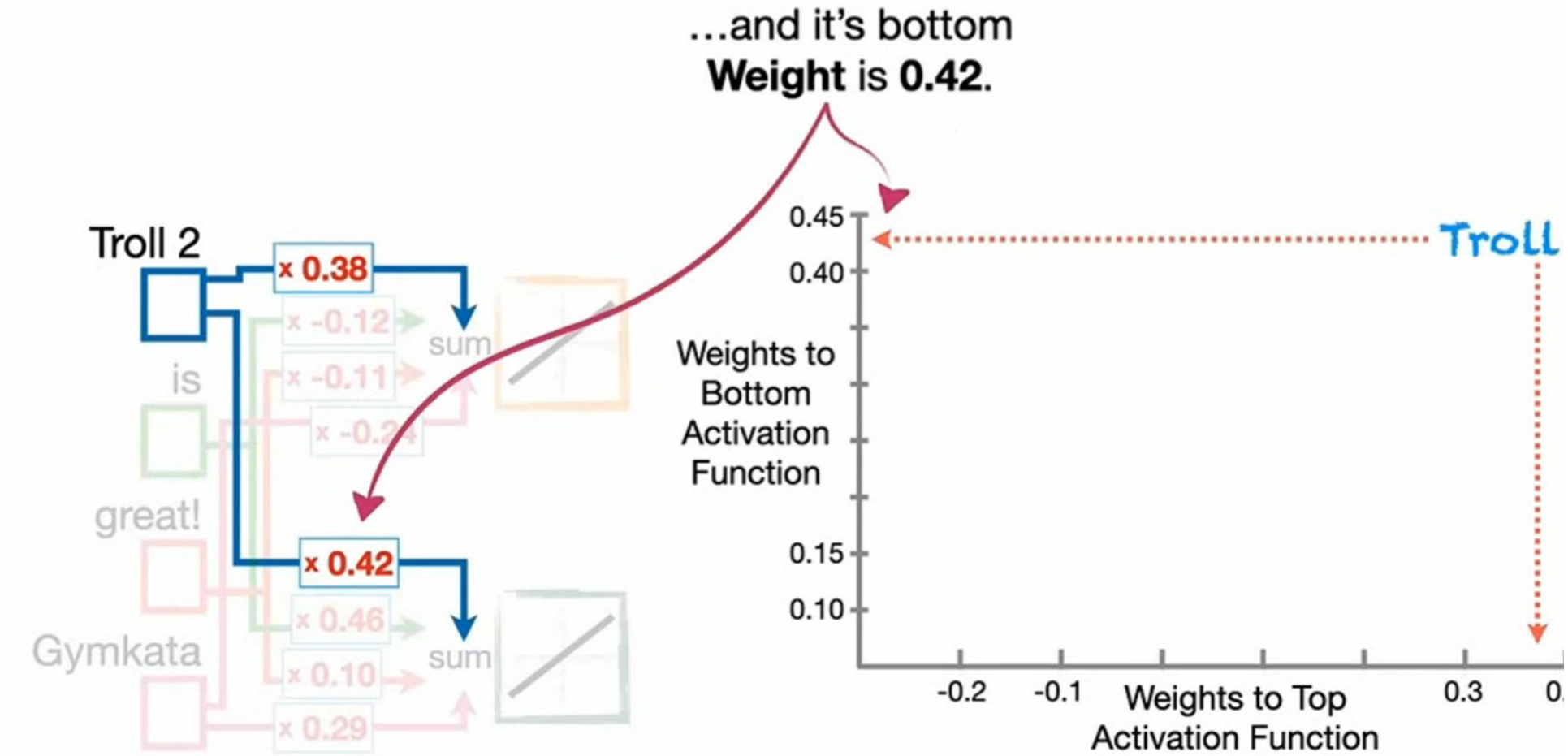
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0


…because its top
Weight is 0.38…
Troll 2
x 0.38
x -0.12
x -0.11
x -0.24
sum
is
great!
x 0.42
x 0.46
x 0.10
x 0.29
sum
Gymkata
0.45
0.40
Weights to Bottom Activation Function
0.15
0.10
Troll
-0.2
-0.1
Weights to Top Activation Function
0.3
0

Word Embeddings: Word2Vec

Word Embeddings: Word2Vec

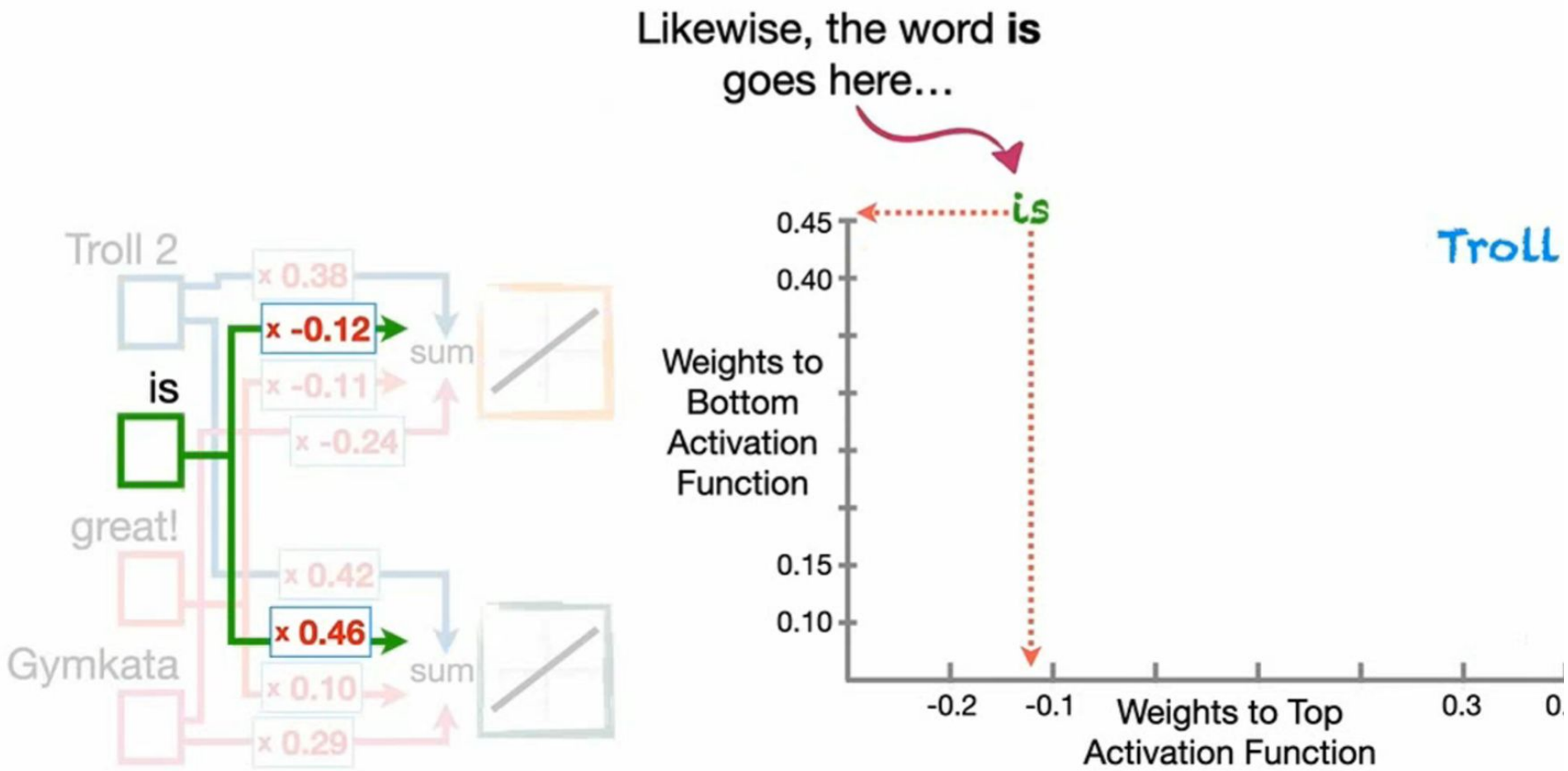


Word Embedding and Word2Vec, Clearly Explained!!!

Word Embedding and Word2Vec, Clearly Explained!!!

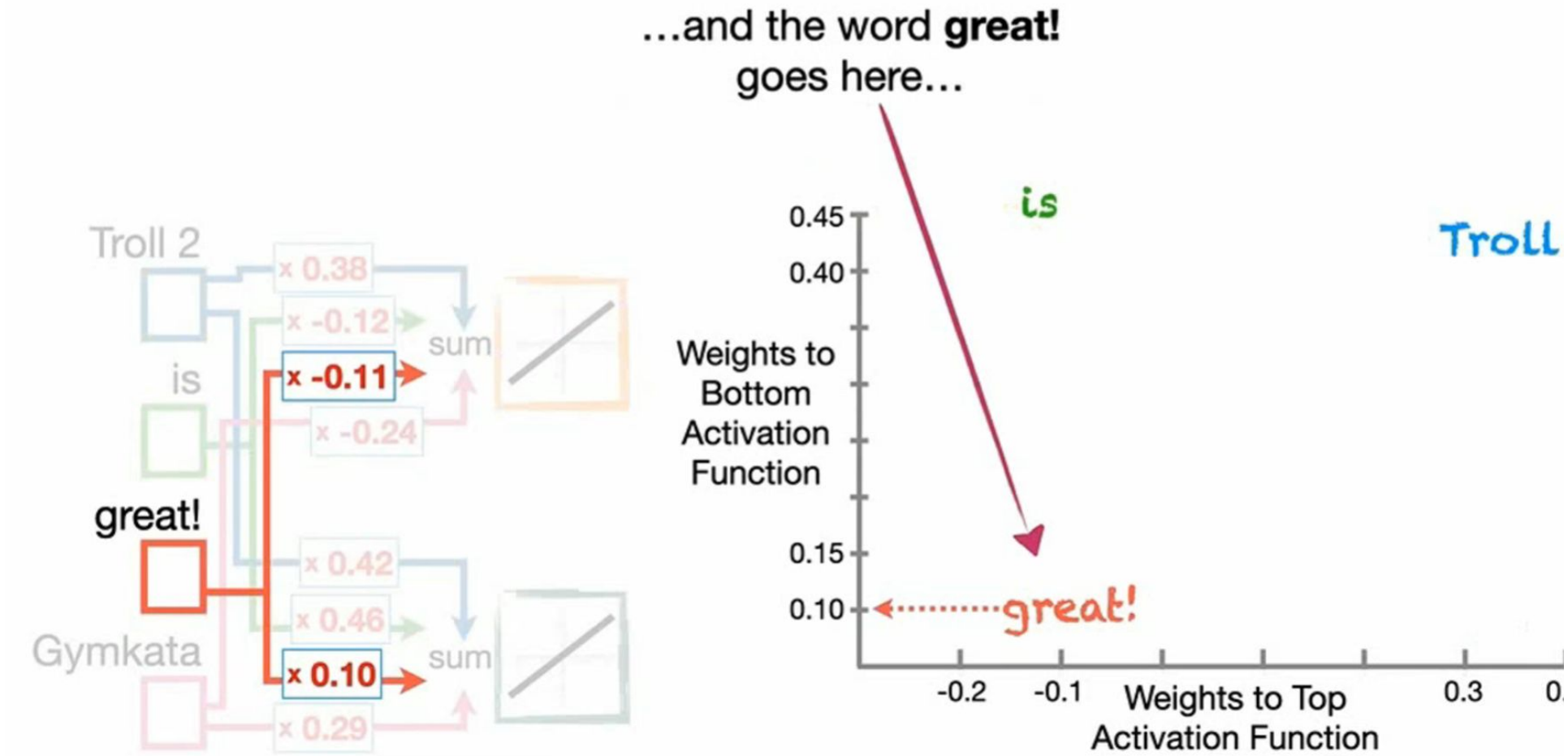
https://www.youtube.com/watch?v=viZrOnJclY0

https://www.youtube.com/watch?v=viZrOnJclY0


Likewise, the word is goes here…
Troll 2
is
great!
Gymkata
x 0.38
x -0.12
x -0.11
x -0.24
x 0.42
x 0.46
x 0.10
x 0.29
sum
sum
is
0.45
0.40
0.15
0.10
Weights to Bottom Activation Function
-0.2
-0.1
Weights to Top Activation Function
0.3
Troll

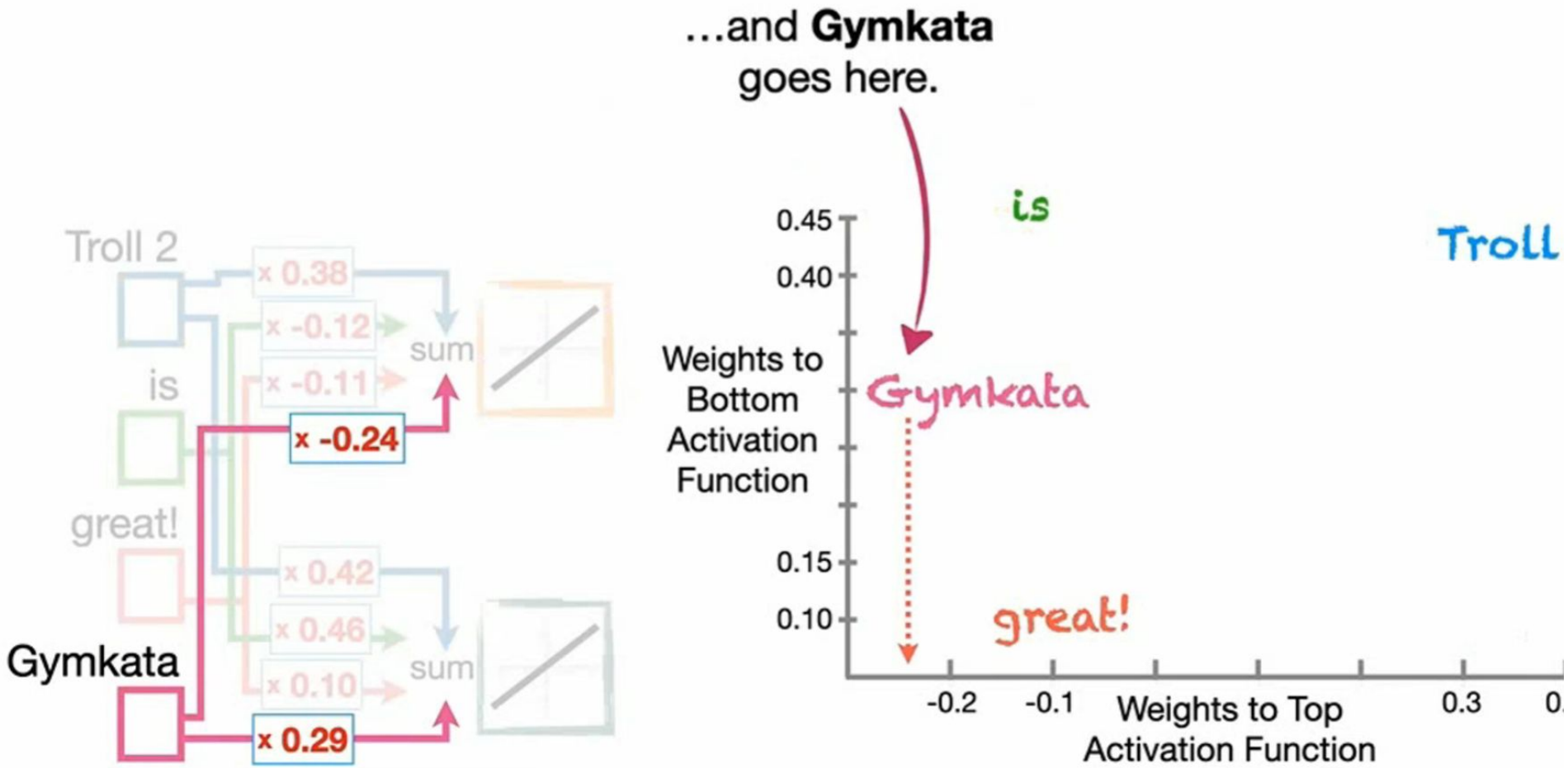
Word Embeddings: Word2Vec

Word Embeddings: Word2Vec

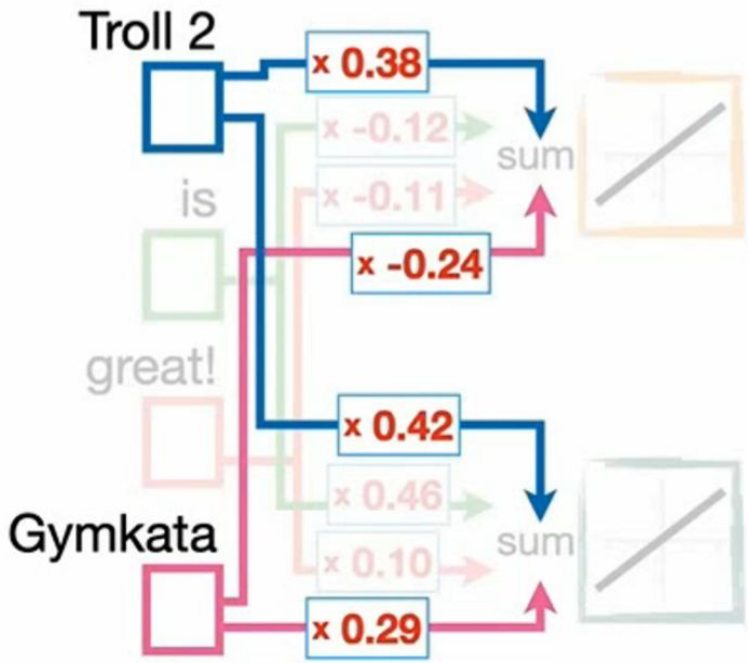


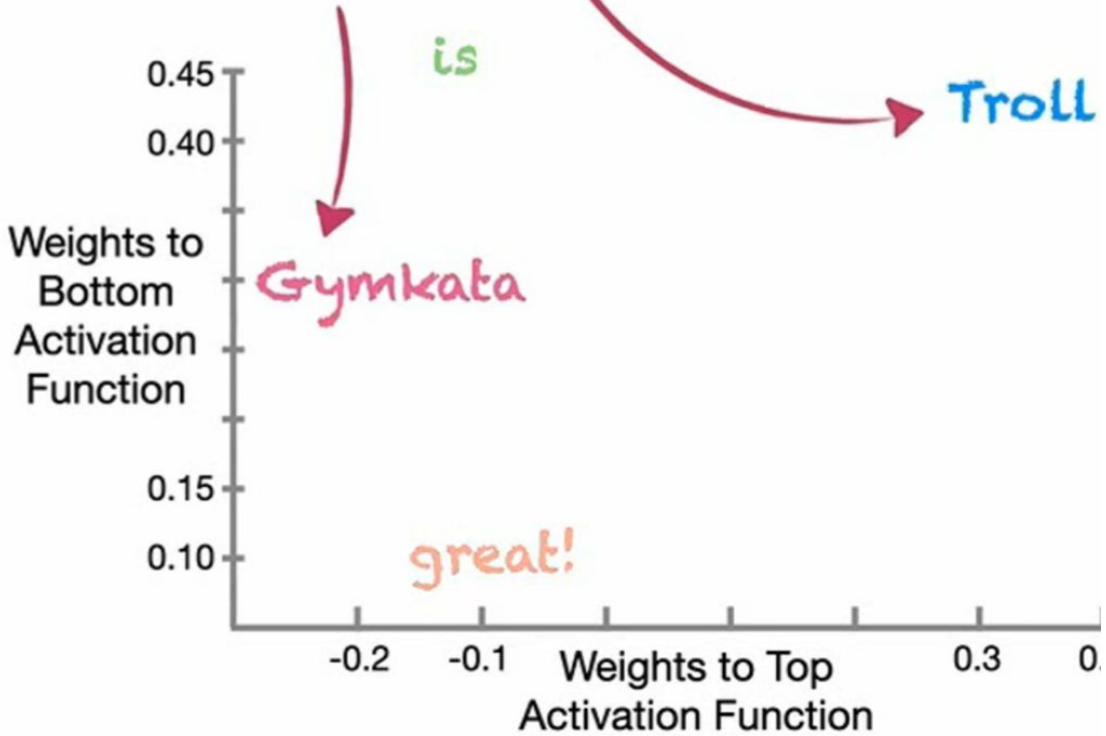
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

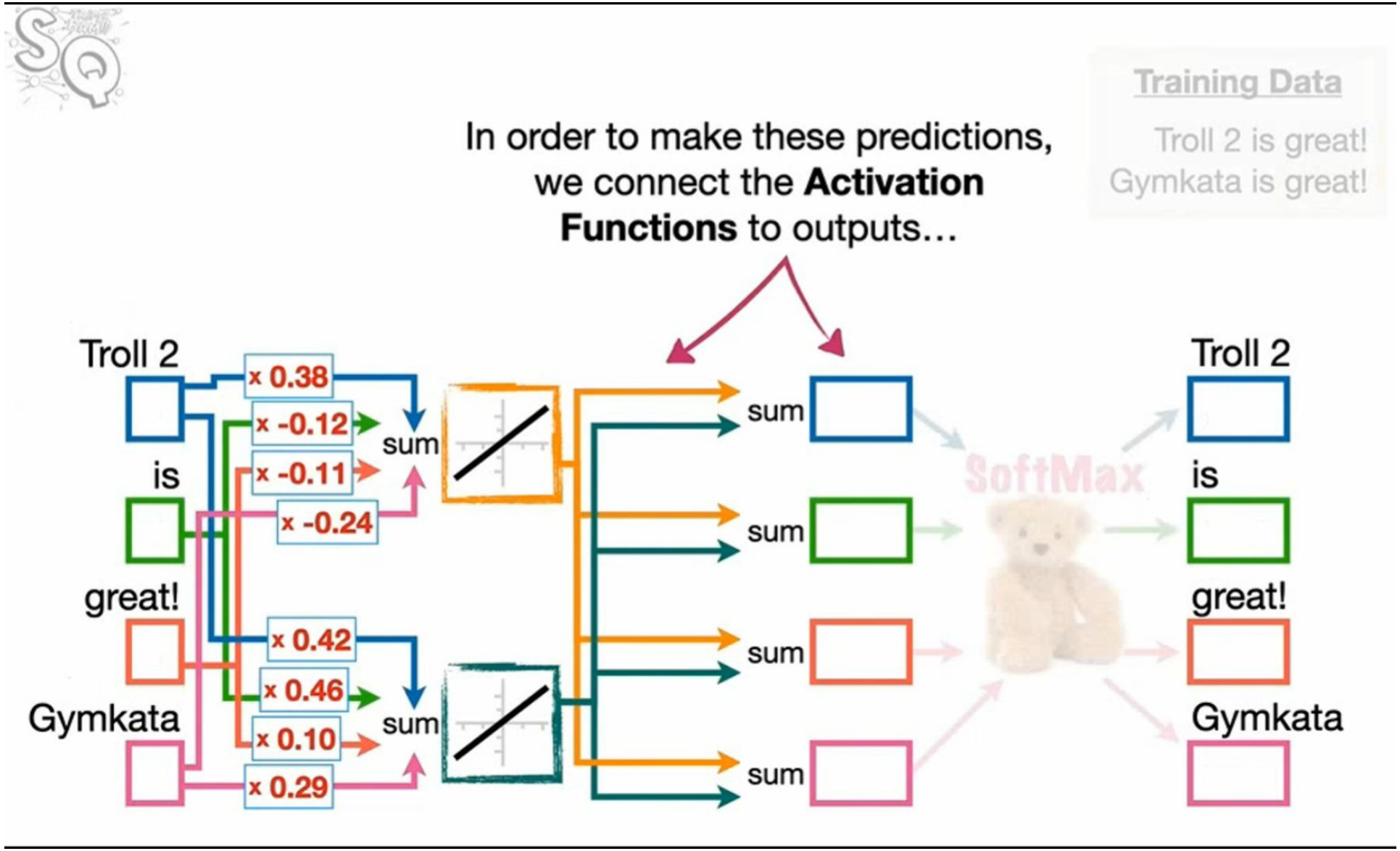

…and Gymkata
goes here.
Troll 2
is
great!
Gymkata
x 0.38
x -0.12
x -0.11
x -0.24
sum
x 0.42
x 0.46
x 0.10
x 0.29
sum
0.45
0.40
Weights to
Bottom
Activation
Function
0.15
0.10
is
Troll
Gymkata
great!
-0.2
-0.1
Weights to Top
Activation Function
0.3

Now, with this graph, we see that the words **Troll 2** and **Gymkata** are currently no more similar to each other as they are to any of the other words.



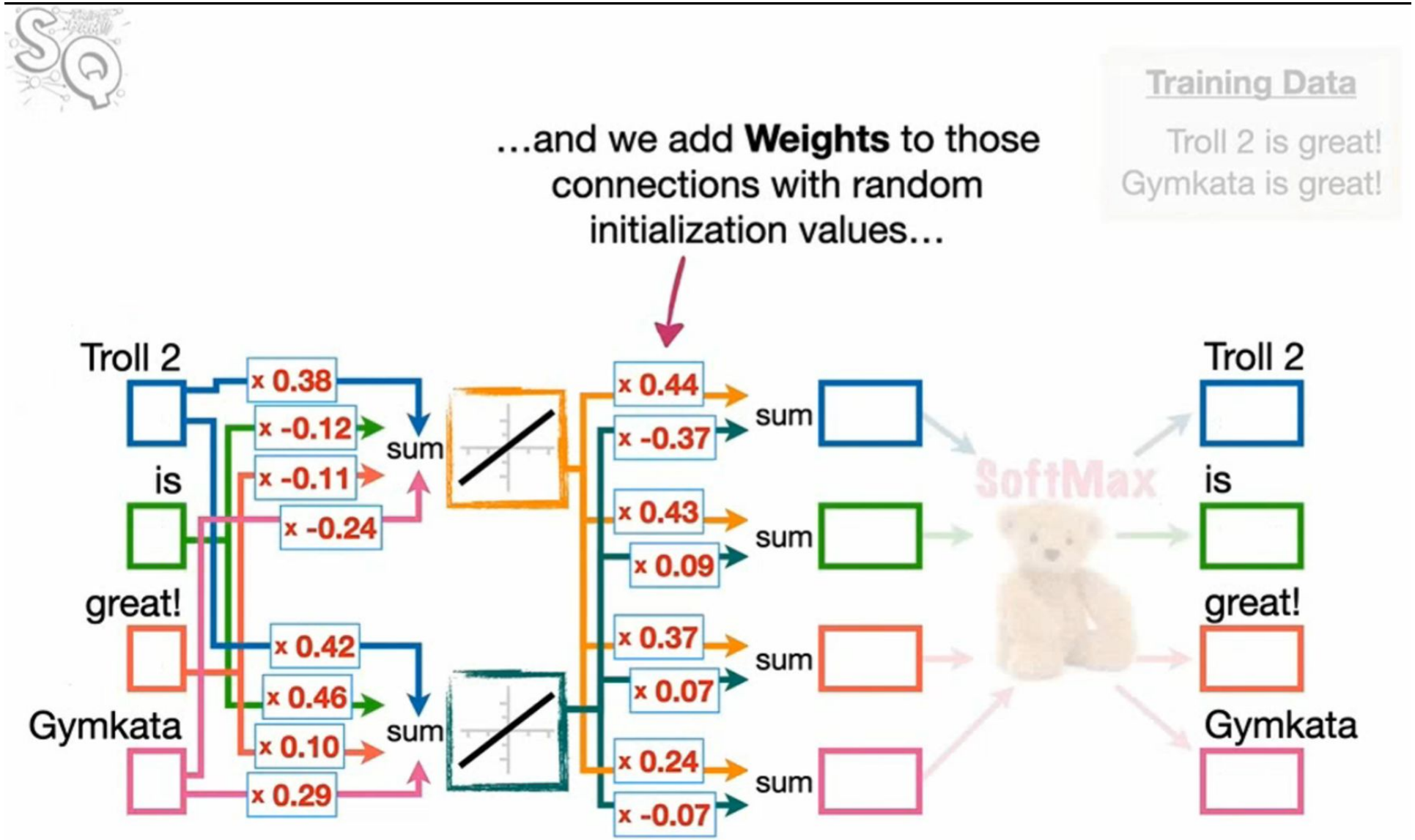


Word Embeddings: Word2Vec



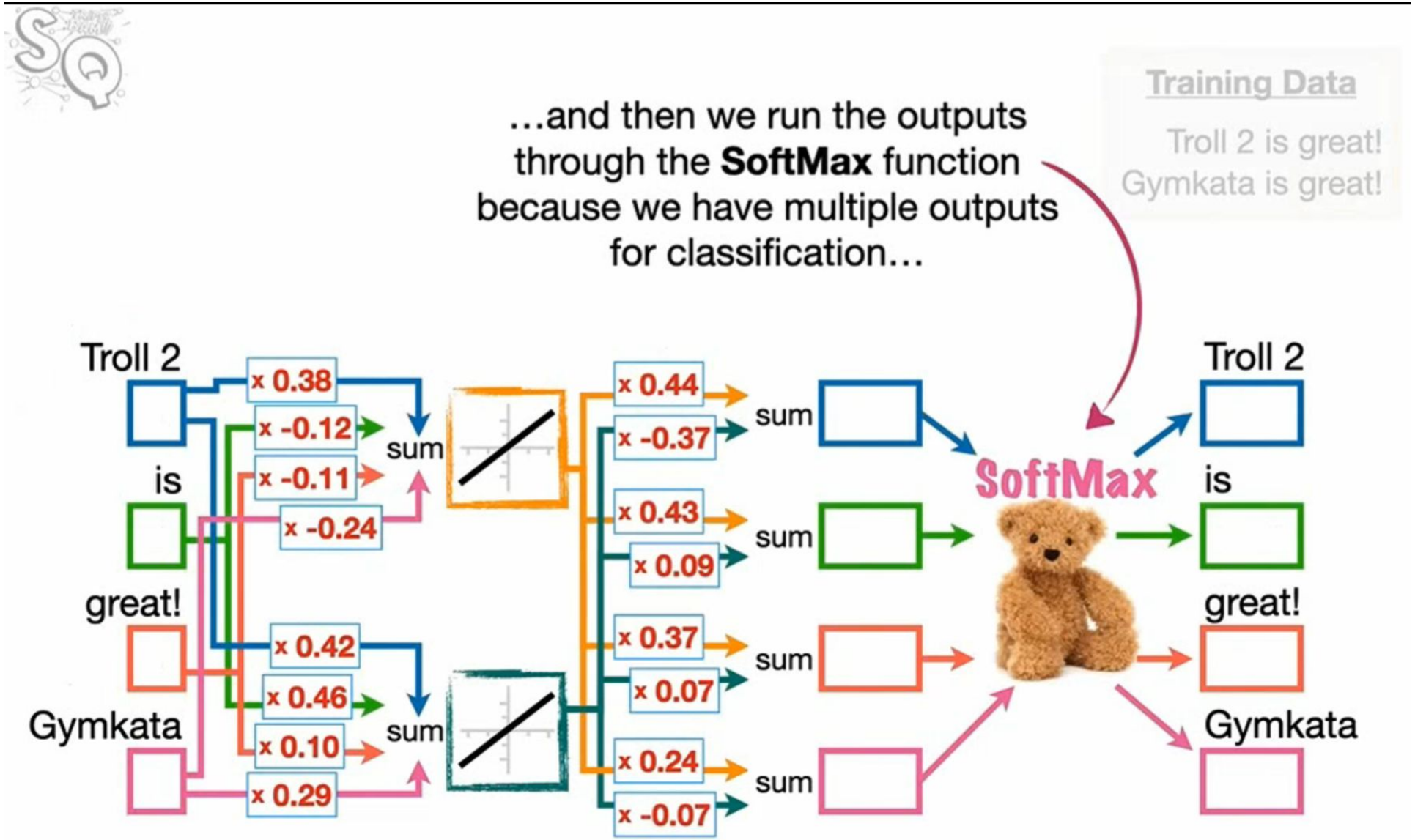
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

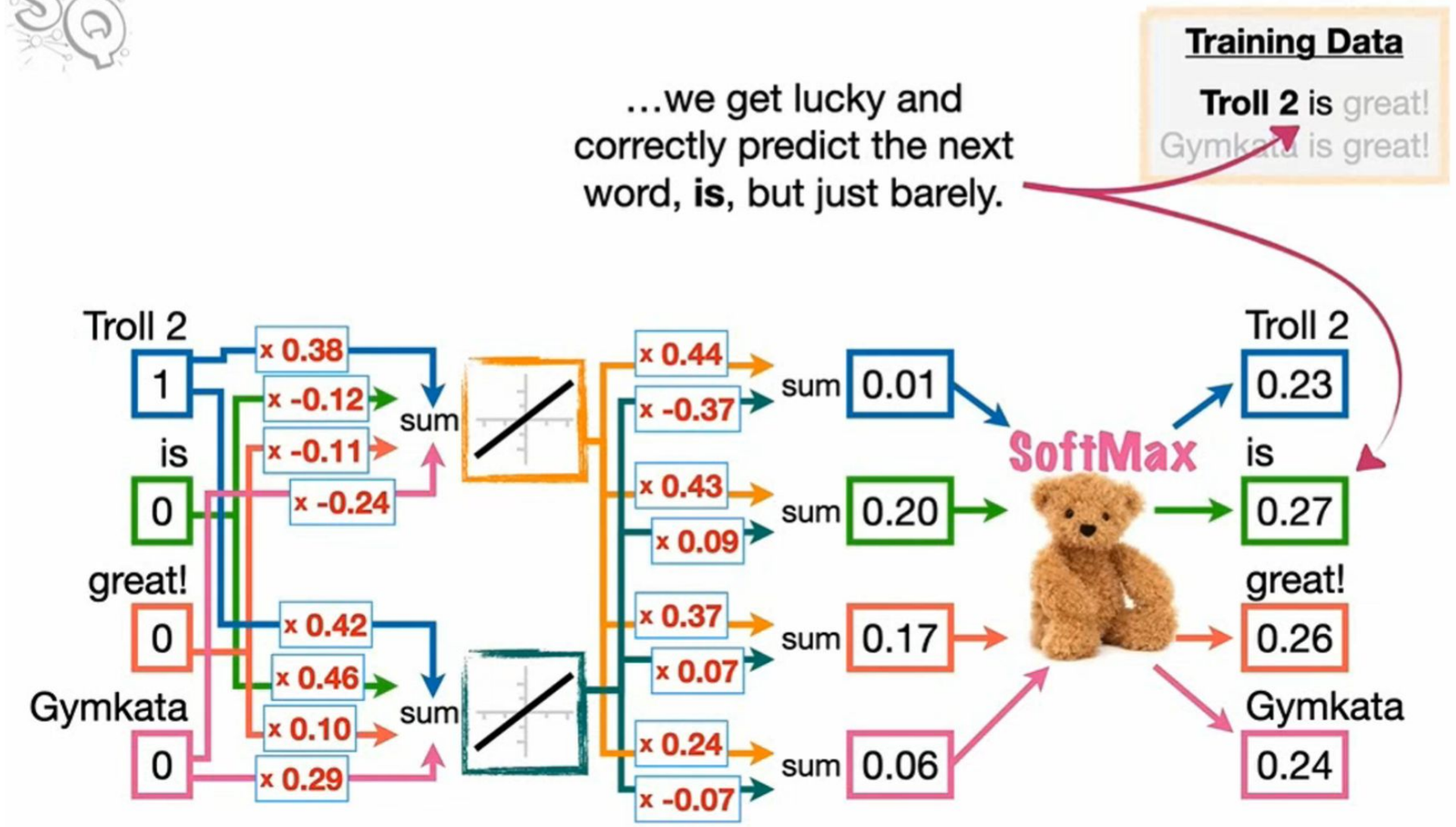
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

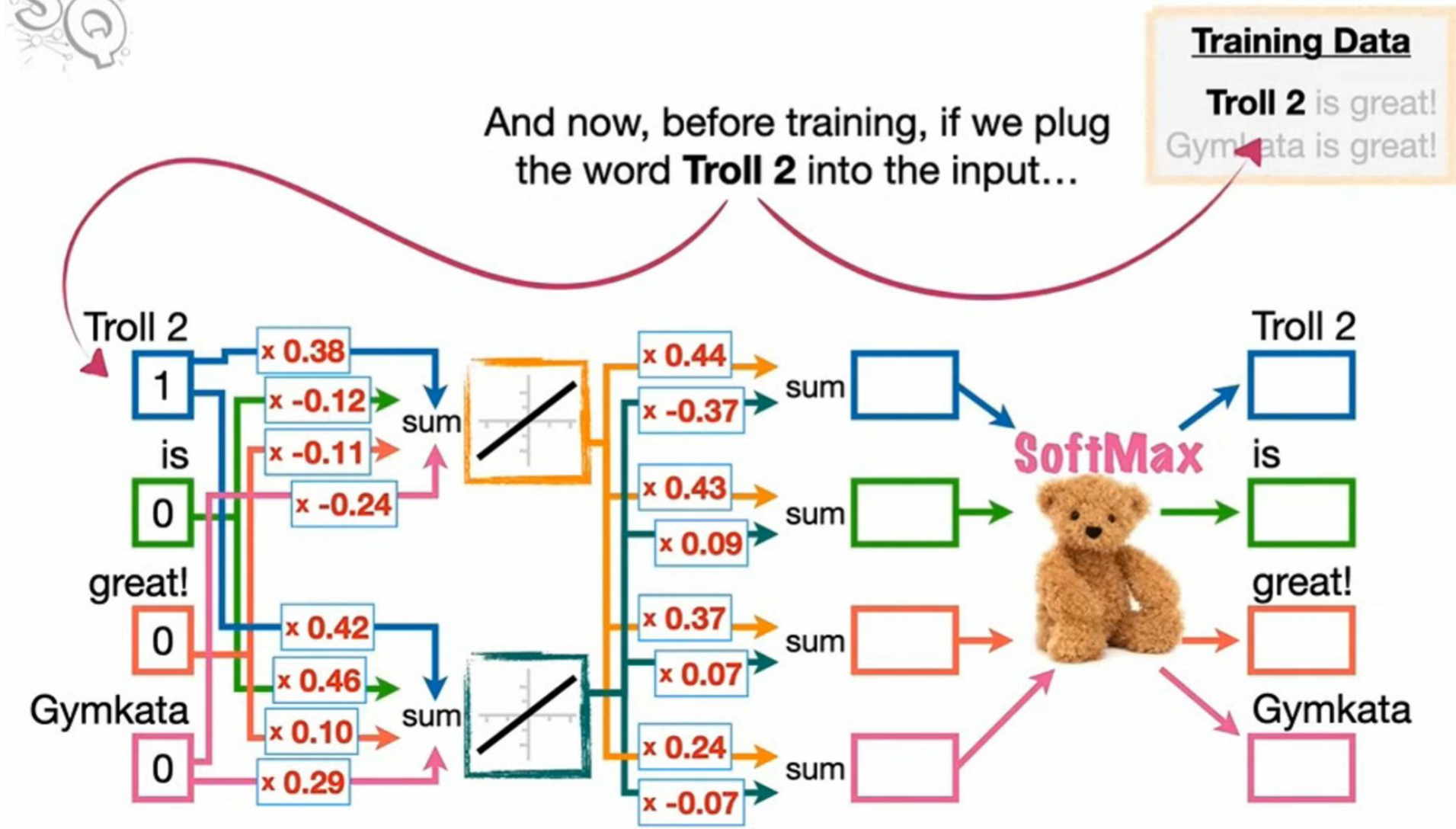
Word Embeddings: Word2Vec

Word Embeddings: Word2Vec

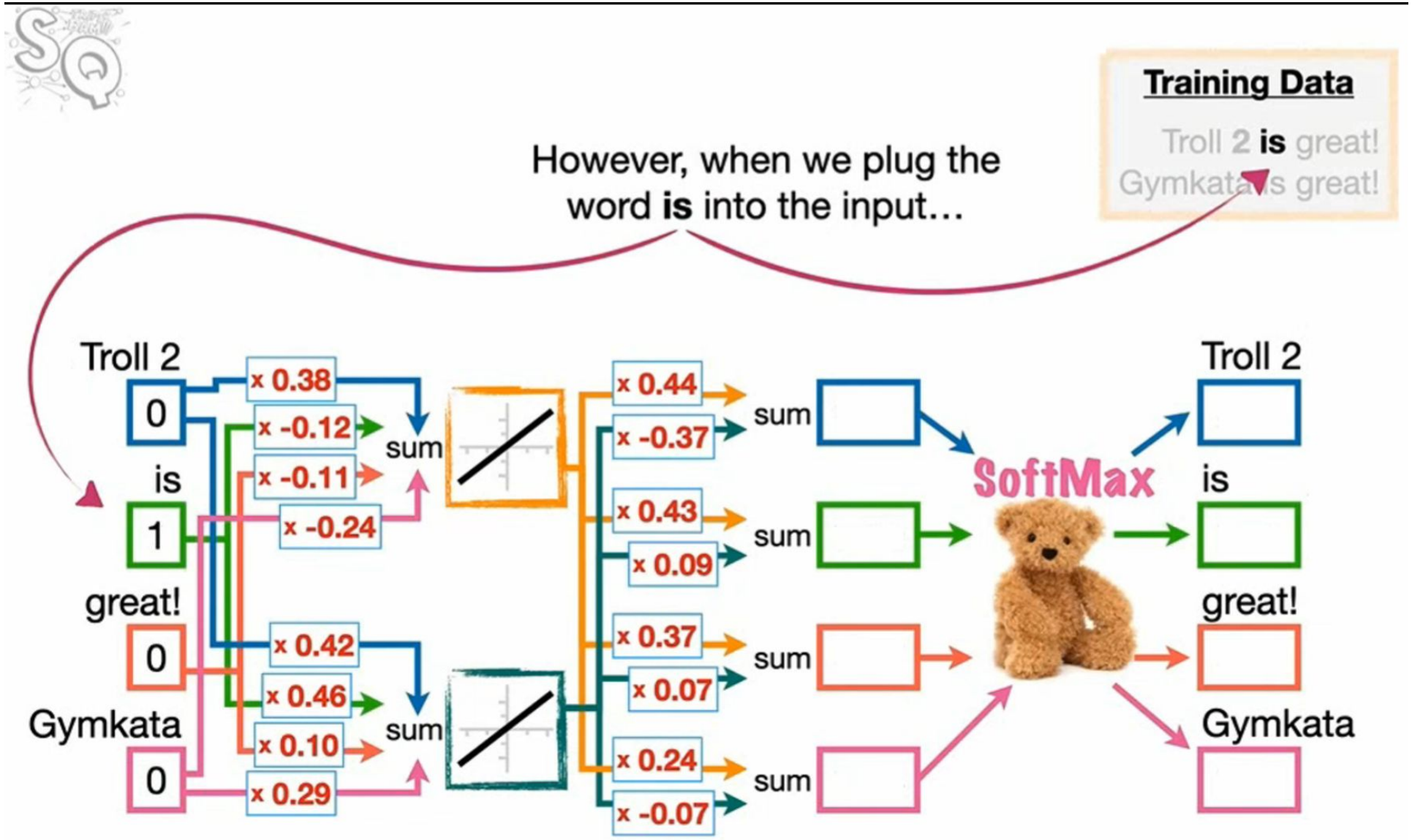


Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0


Training Data
Troll 2 is great!
Gymkata is great!
And now, before training, if we plug the word Troll 2 into the input…
Troll 2
1
is
0
great!
0
Gymkata
0
x 0.38
x -0.12
x -0.11
x -0.24
x 0.42
x 0.46
x 0.10
x 0.29
sum
sum
x 0.44
x -0.37
x 0.43
x 0.09
x 0.37
x 0.07
x 0.24
x -0.07
sum
sum
sum
sum
SoftMax
Troll 2
is
great!
Gymkata

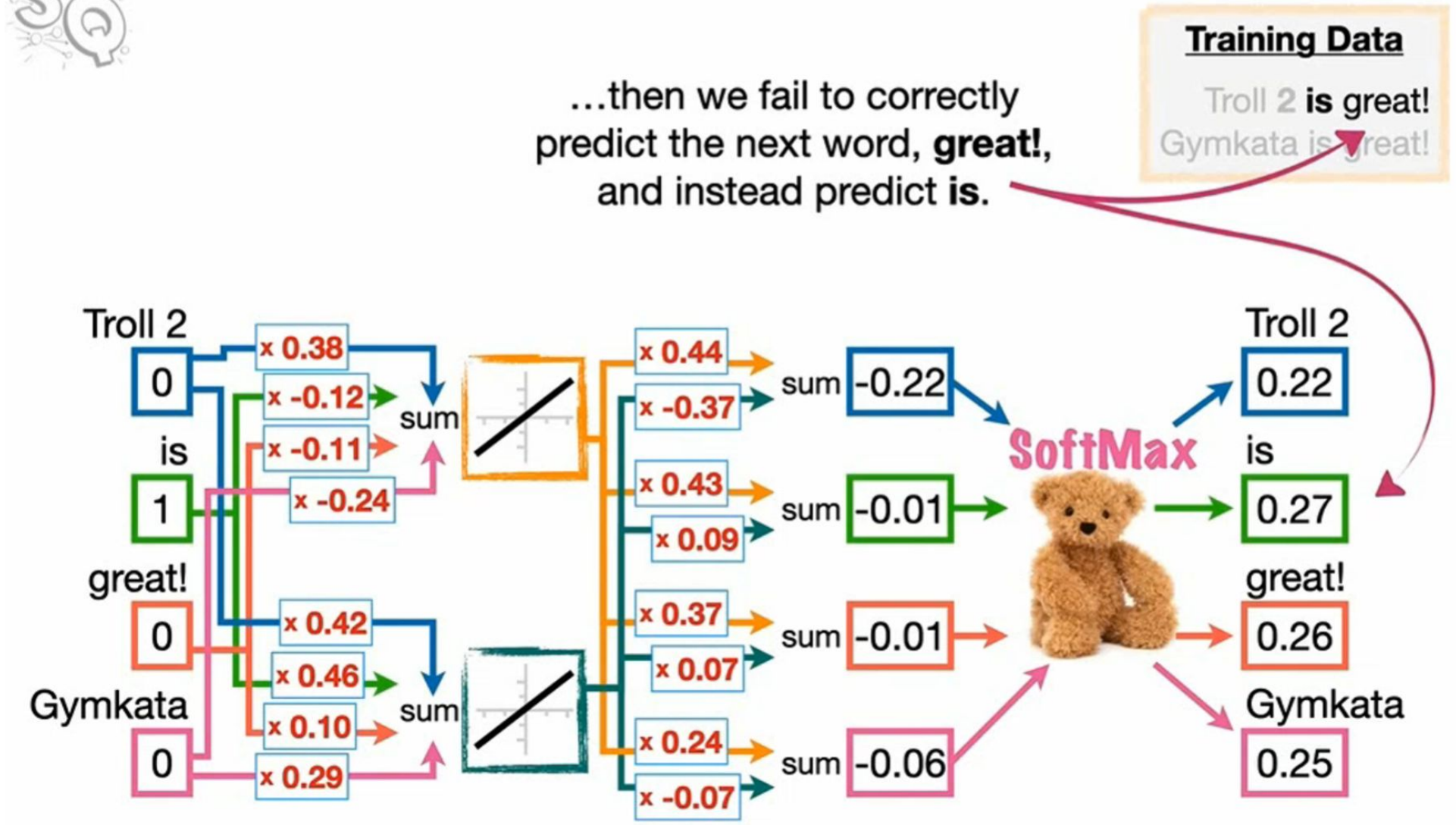
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec

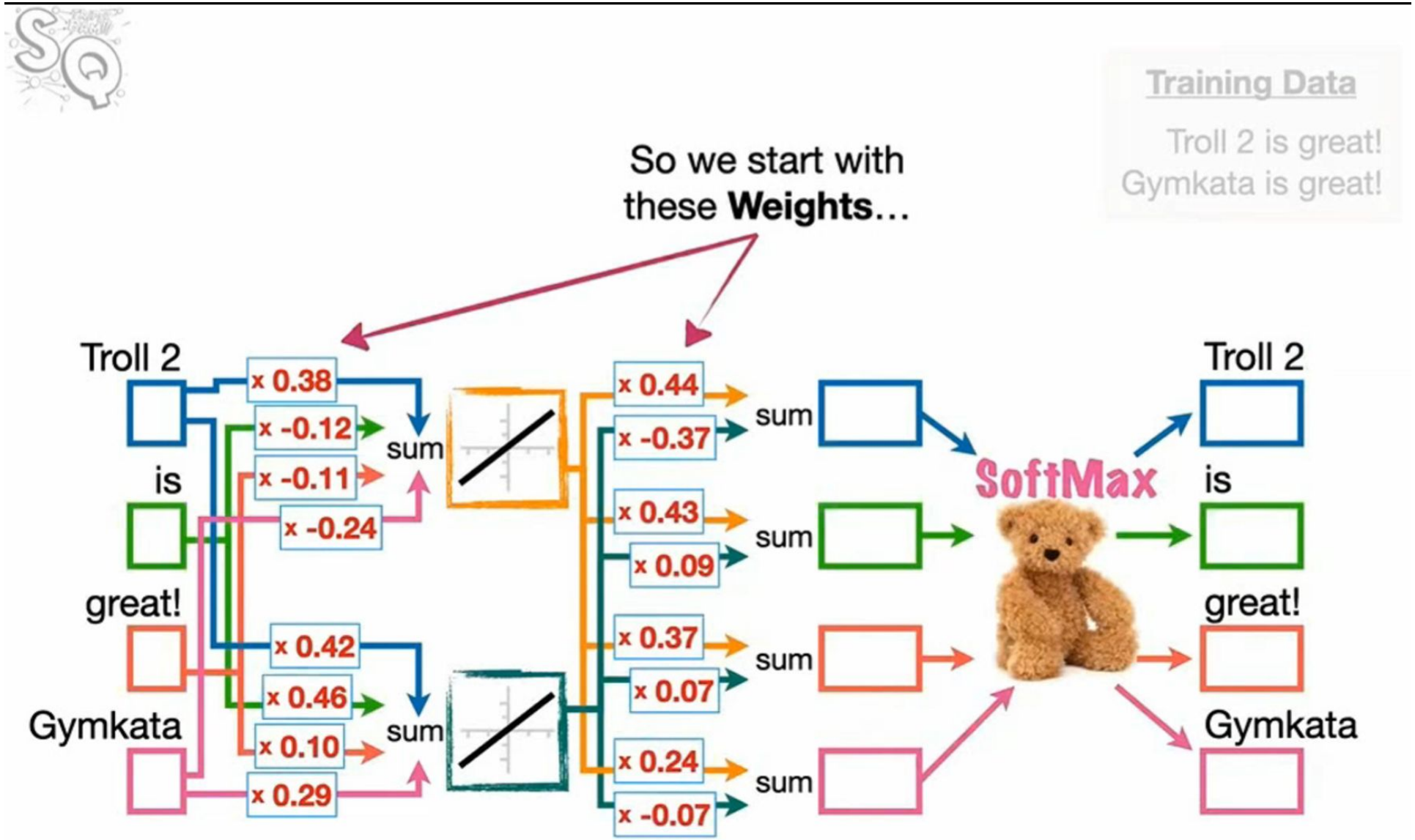
Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec

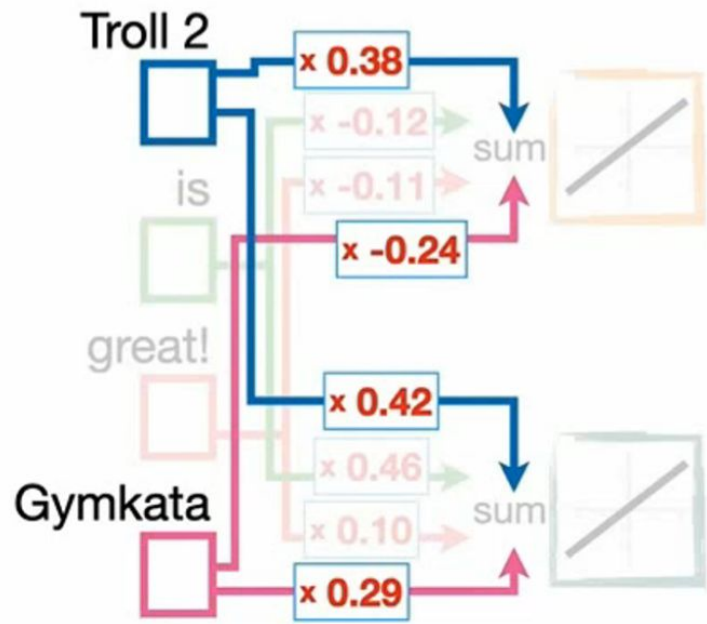


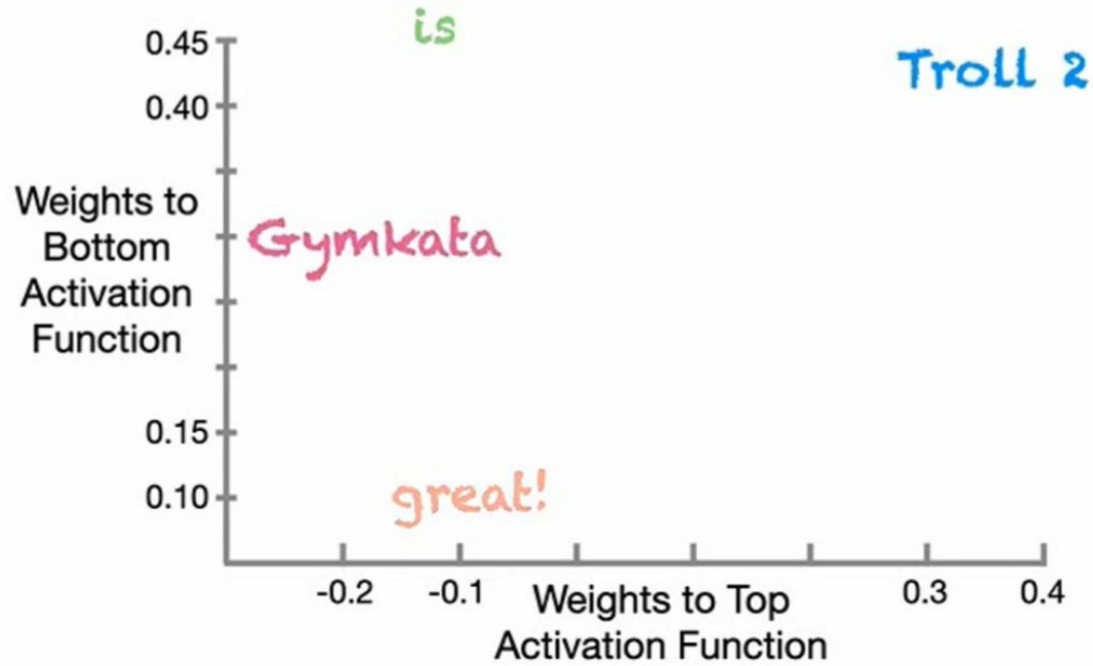
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

…we hope that **Backpropagation** will make their **Weights** more similar.

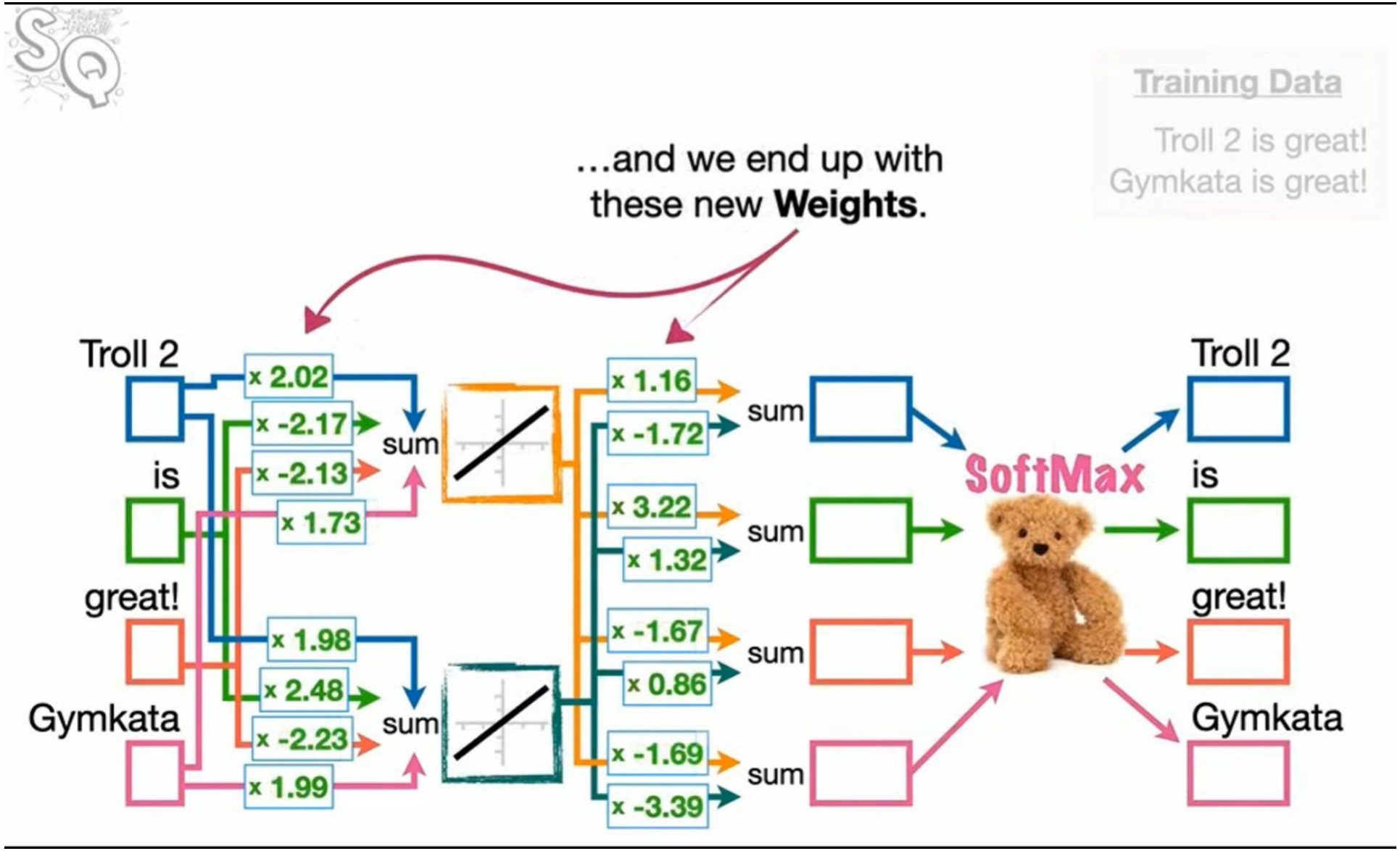
Training Data

Troll 2 is great!
Gymkata is great!



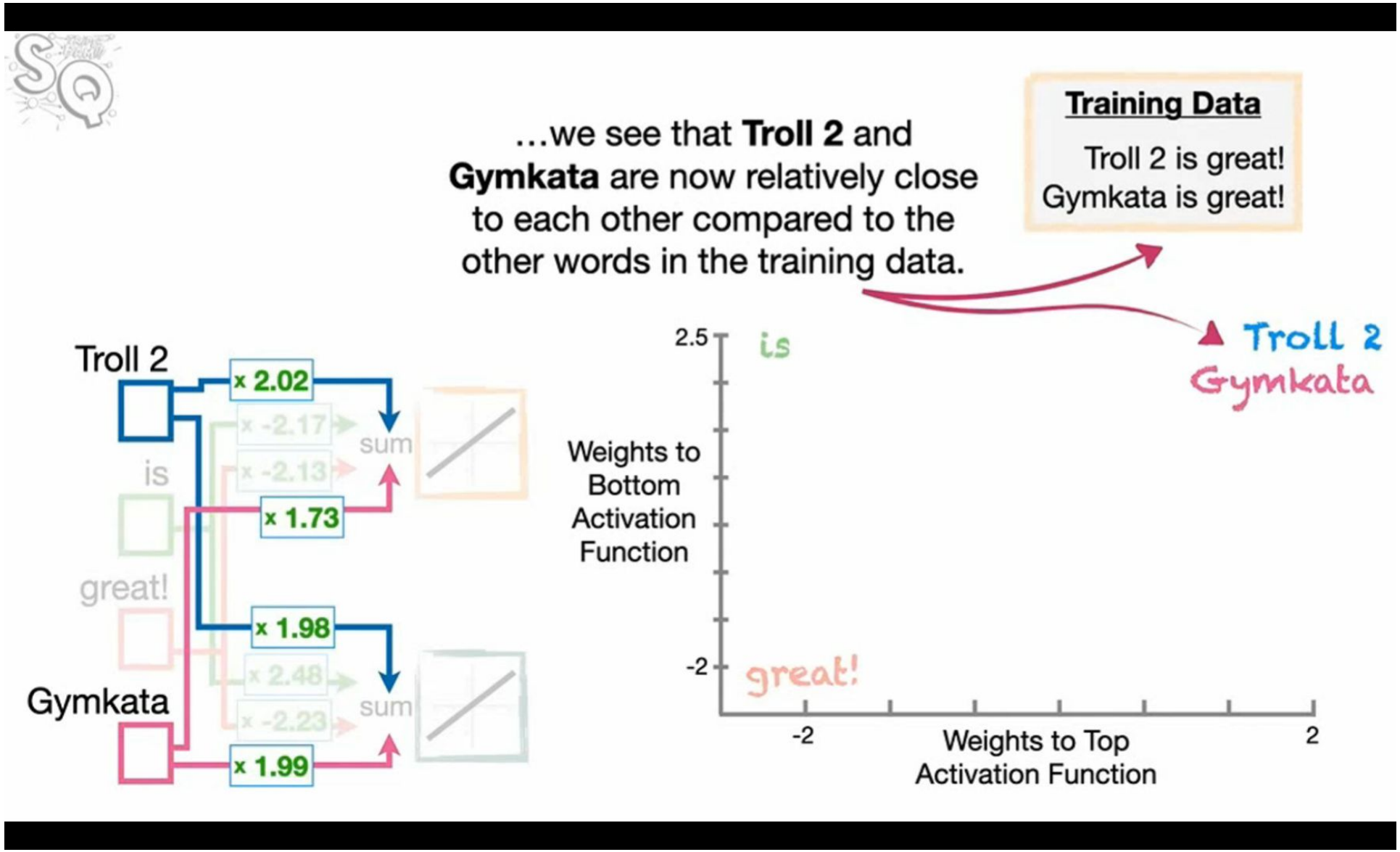


Word Embeddings: Word2Vec



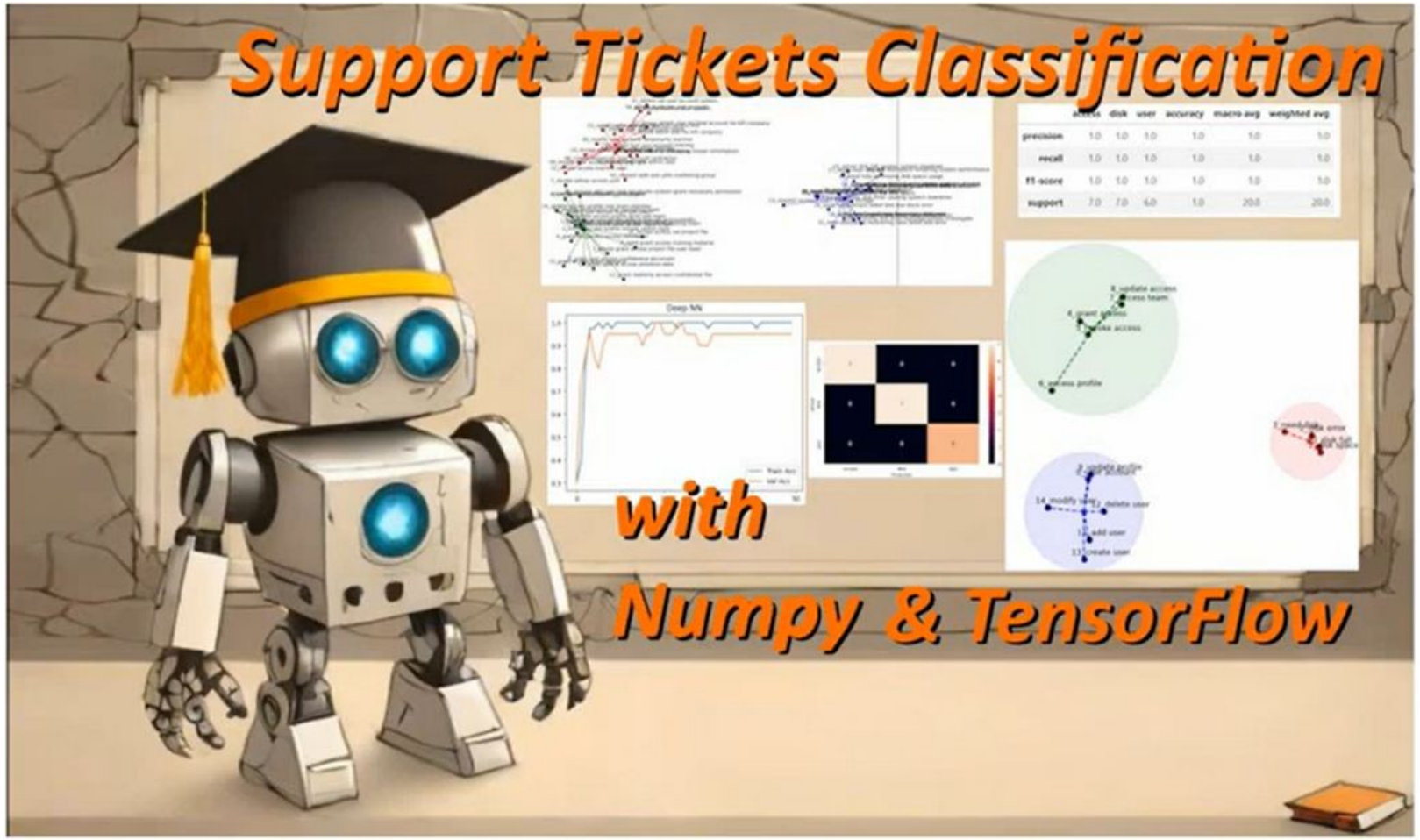
Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings: Word2Vec



Word Embedding and Word2Vec, Clearly Explained!!!
https://www.youtube.com/watch?v=viZrOnJclY0

Word Embeddings with Sentence Transformers



Support Ticket Classification
Github: https://github.com/enoten/support_ticket_analysis/blob/main/support_tickets_classification.ipynb
Youtube: https://www.youtube.com/watch?v=NlJ1yS0F03M

Word Embeddings with Sentence Transformers

3. Transform text: embedding, dim reduction

3.0 Import LLM library

In [33]:
```
import warnings
from sentence_transformers import SentenceTransformer
warnings.filterwarnings("ignore")
```

3.1 Create embeddings for preprocessed description

In [34]:
```
phrase_model = SentenceTransformer("all-mpnet-base-v2")
```

In [35]:
```
#conver df column to list
preproc_desc = tickets_df['preproc_description'].to_list()
```

In [36]:
```
embeds = phrase_model.encode(preproc_desc)
```

In [37]:
```
embeds.shape
```

Out[37]: (60, 768)

Support Ticket Classification
https://github.com/enoten/support_ticket_analysis/blob/main/support_tickets_classification.ipynb

Word Embeddings with Sentence Transformers

	subject	description	key_phrase	label	preproc_description	preproc_subject
0	Grant Access to Confidential Files	Grant user access to confidential documents.	grant access	0	grant user access confidential document	grant access confidential file
1	Revoke Access to Sensitive Data	Revoke user access to sensitive information.	revoke access	0	revoke user access sensitive information	revoke access sensitive data
2	Update Access Profile	Modify my access profile to include admin rights	access profile	0	modify access profile include admin right	update access profile
3	Grant Access Request for New Project	Please grant access to project files for users...	grant access	0	please grant access project file user team	grant access request new project
4	Access Removal for Departing Employee	Revoke access for departing employee.	revoke access	0	revoke access departing employee	access removal departing employee

Support Ticket Classification
Github: https://github.com/enoten/support_ticket_analysis/blob/main/support_tickets_classification.ipynb
Youtube: https://www.youtube.com/watch?v=NlJ1yS0F03M

Word Embeddings with Sentence Transformers

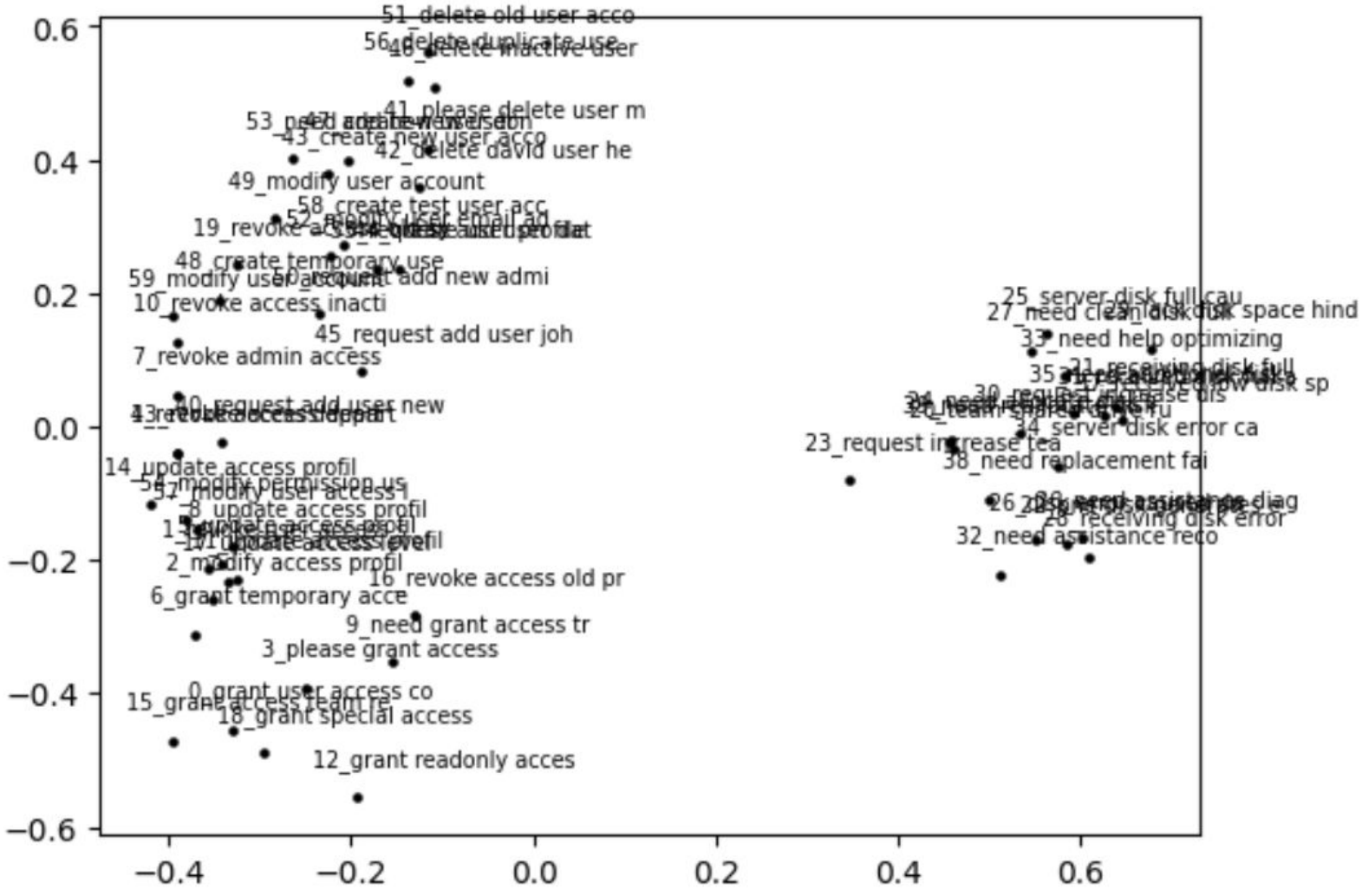
3.2 Do dim reduction for embeddings

```
import numpy as np
from sklearn.decomposition import PCA

pca = PCA(n_components=2, svd_solver='full')
x_embeds = embeds.copy()
X_transformed = pca.fit_transform(embeds)
```

```
X_transformed.shape
```

```
(60, 2)
```

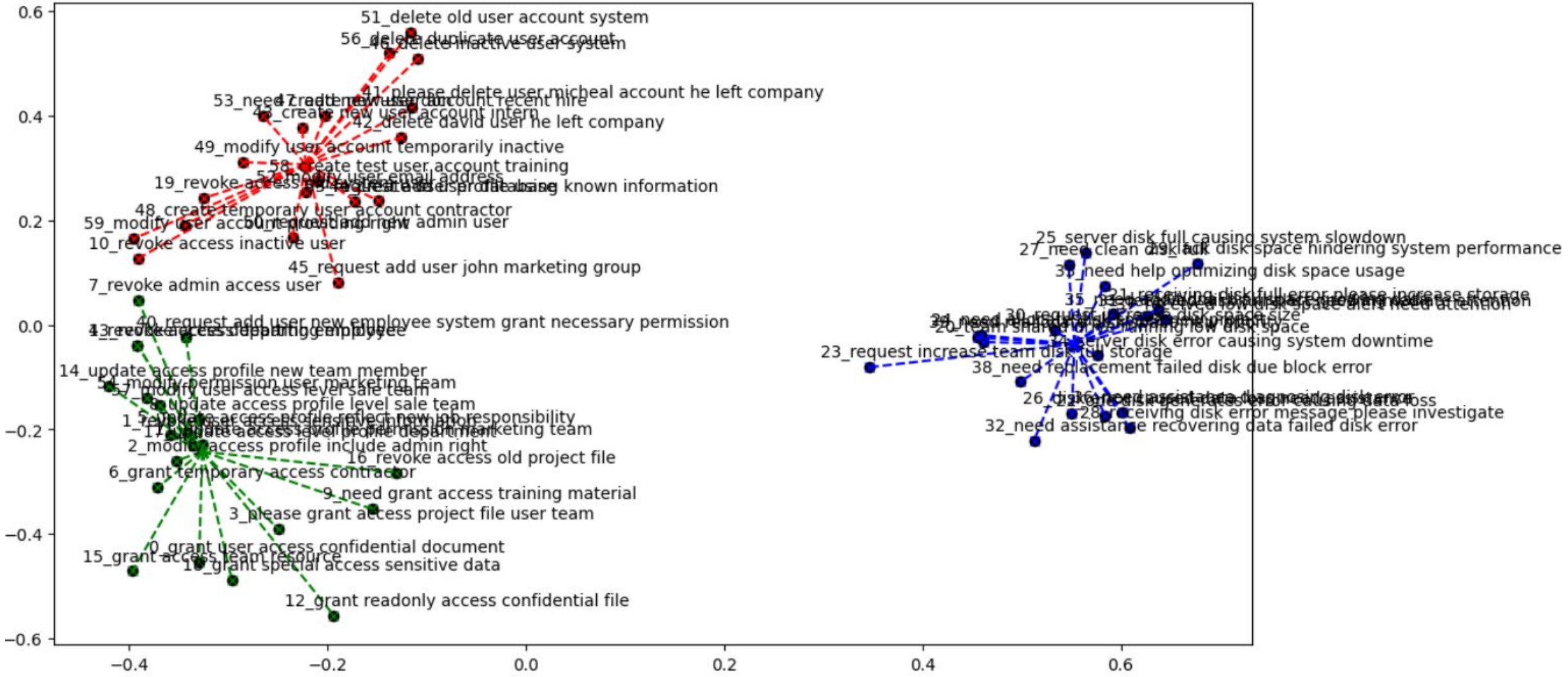


Support Ticket Classification

Github: https://github.com/enoten/support_ticket_analysis/blob/main/support_tickets_classification.ipynb

Youtube: https://www.youtube.com/watch?v=NlJ1yS0F03M

Clustering Embeddings



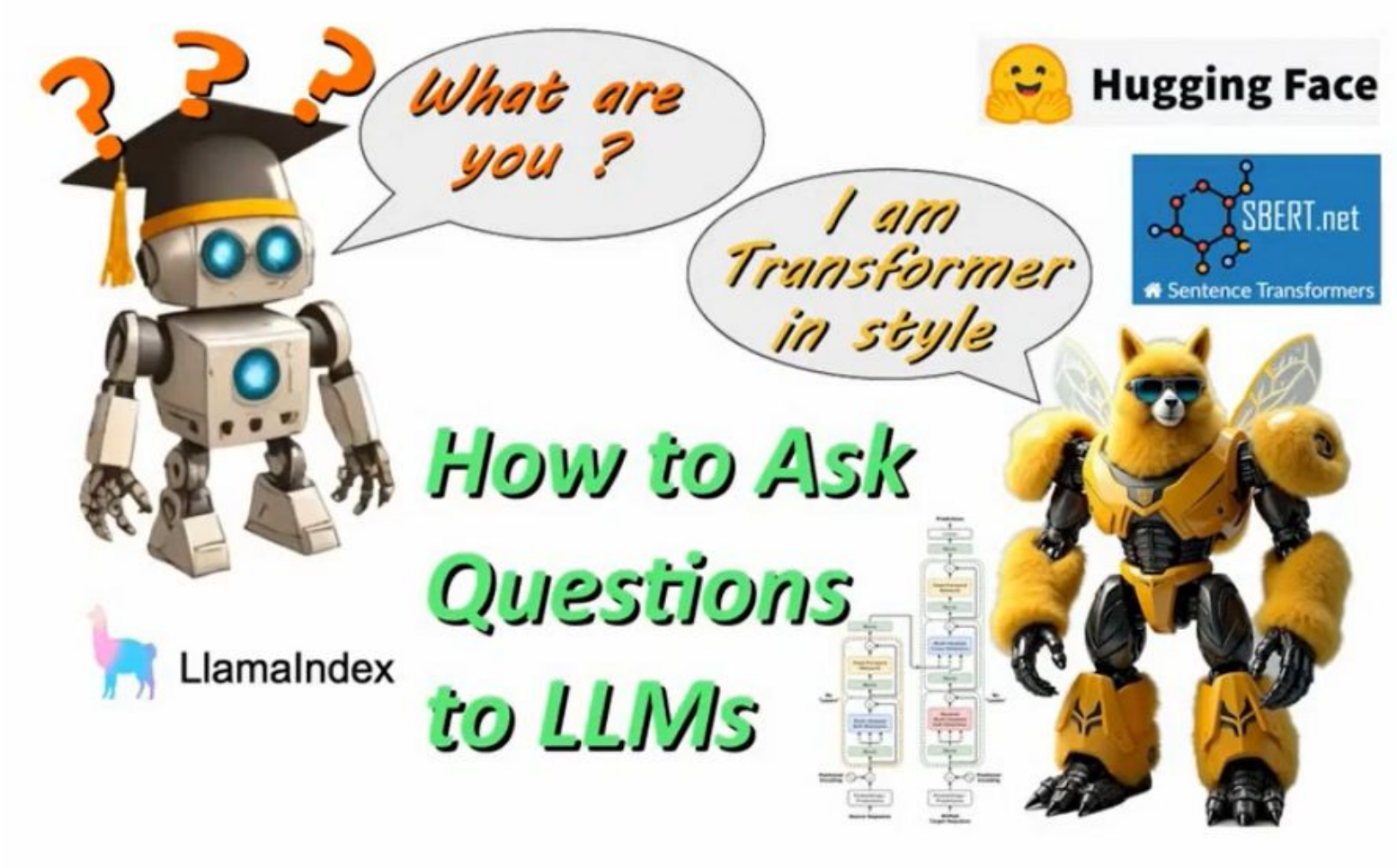
Support Ticket Classification

Github: https://github.com/enoten/support_ticket_analysis/blob/main/support_tickets_classification.ipynb

Youtube: https://www.youtube.com/watch?v=NlJ1yS0F03M

Semantic Search

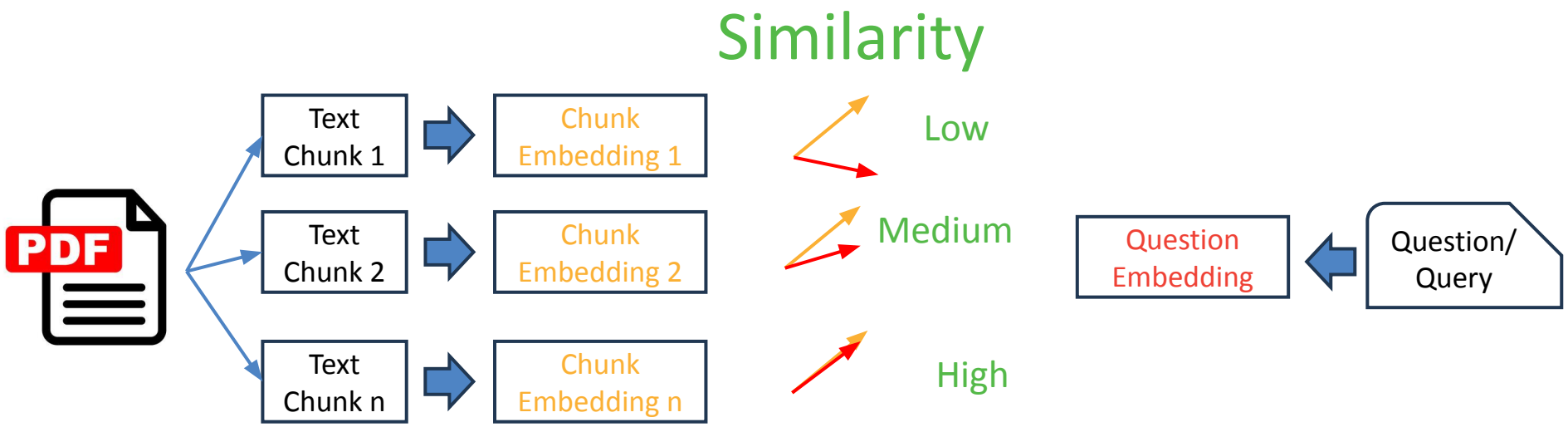
Semantic Search



How to ask Questions to LLMs
Github:
https://github.com/enoten/huggingface_llms_apps/blob/main/ask_questions_about_research_papers%20to%20LLMs.ipynb
Youtube: https://www.youtube.com/watch?v=E4k1qGFemSE

Origins of GenAI: Embeddings

Embeddings can be used to represented text/pdf documents in the numeric format.
On the other hand, we can convert Question/Query about text/pdf document into numeric vector as well.



Now We can Easily Find Most Relevant Piece of the Text

Semantic Search

3.2.3 Applications of Attention in our Model The Transformer uses multi-head attention in three different ways: • In "encoder-decoder attention" layers, the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder Score:0.50 With a single attention head, averaging inhibits this Score:0.46 In this work we employ h = 8 parallel attention layers, or heads Score:0.46 Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality Score:0.43 Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions Score:0.43 In a self-attention layer all of the keys, values and queries come from the same place, in this case, the output of the previous layer in the encoder Score:0.37 • The encoder contains self-attention layers Score:0.37 This mimics the typical encoder-decoder attention mechanisms in sequence-to-sequence models such as [38, 2, 9] Score:0.36 • Similarly, self-attention layers in the decoder allow each position in the decoder to attend to all positions in the decoder up to and including that position Score:0.33 3.3 Position-wise Feed-Forward Networks In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically Score:0.33 We implement this inside of scaled dot-product attention by masking out (setting to −∞) all values in the input of the softmax which correspond to illegal connections Score:0.26 MultiHead(Q, K, V) = Concat(head1, ...,headh)WO where headi = Attention(QWQ i , KWK i , V WV i) Where the projections are parameter matricesWQ i ∈ Rdmodel×dk , WK i ∈ Rdmodel×dk , WV i ∈ Rdmodel×dv and WO ∈ Rhdv×dmodel Score:0.24 This allows every position in the decoder to attend over all positions in the input sequence Score:0.20 Each position in the encoder can attend to all positions in the previous layer of the encoder Score:0.15 This consists of two linear transformations with a ReLU activation in between Score:0.15 3.4 Embeddings and Softmax Similarly to other sequence transduction models, we use learned embeddings to convert the input tokens and output tokens to vectors of dimension dmodel Score:0.14 See Figure 2 Score:0.14 We also use the usual learned linear transfor- mation and softmax function to convert the decoder output to predicted next-token probabilities Score:0.13 s Score:0.13 Another way of describing this is as two convolutions with kernel size 1 Score:0.12 The dimensionality of input and output is dmodel = 512, and the inner-layer has dimensionality dff = 2048 Score:0.10 output values Score:0.10 We need to prevent leftward information flow in the decoder to preserve the auto-regressive property Score:0.08 For each of these we use dk = dv = dmodel/h = 64 Score:0.04 These are concatenated and once again projected, resulting in the final values, as depicted in Figure 2 Score:0.03 In the embedding layers, we multiply those weights by √dmodel Score:0.02 In our model, we share the same weight matrix between the two embedding layers and the pre-softmax linear transformation similar to [30] Score:0.02 FFN(x) = max(0, xW1 + b1)W2 + b2 (2) While the linear transformations are the same across different positions, they use different parameters from layer to layer Score:-0.01

How to ask Questions to LLMs
Github:
https://github.com/enoten/huggingface_llms_apps/blob/main/ask_questions_about_research_papers%20to%20LLMs.ipynb
Youtube: https://www.youtube.com/watch?v=E4k1qGFemSE

RDB Search

Relational DB Search

id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

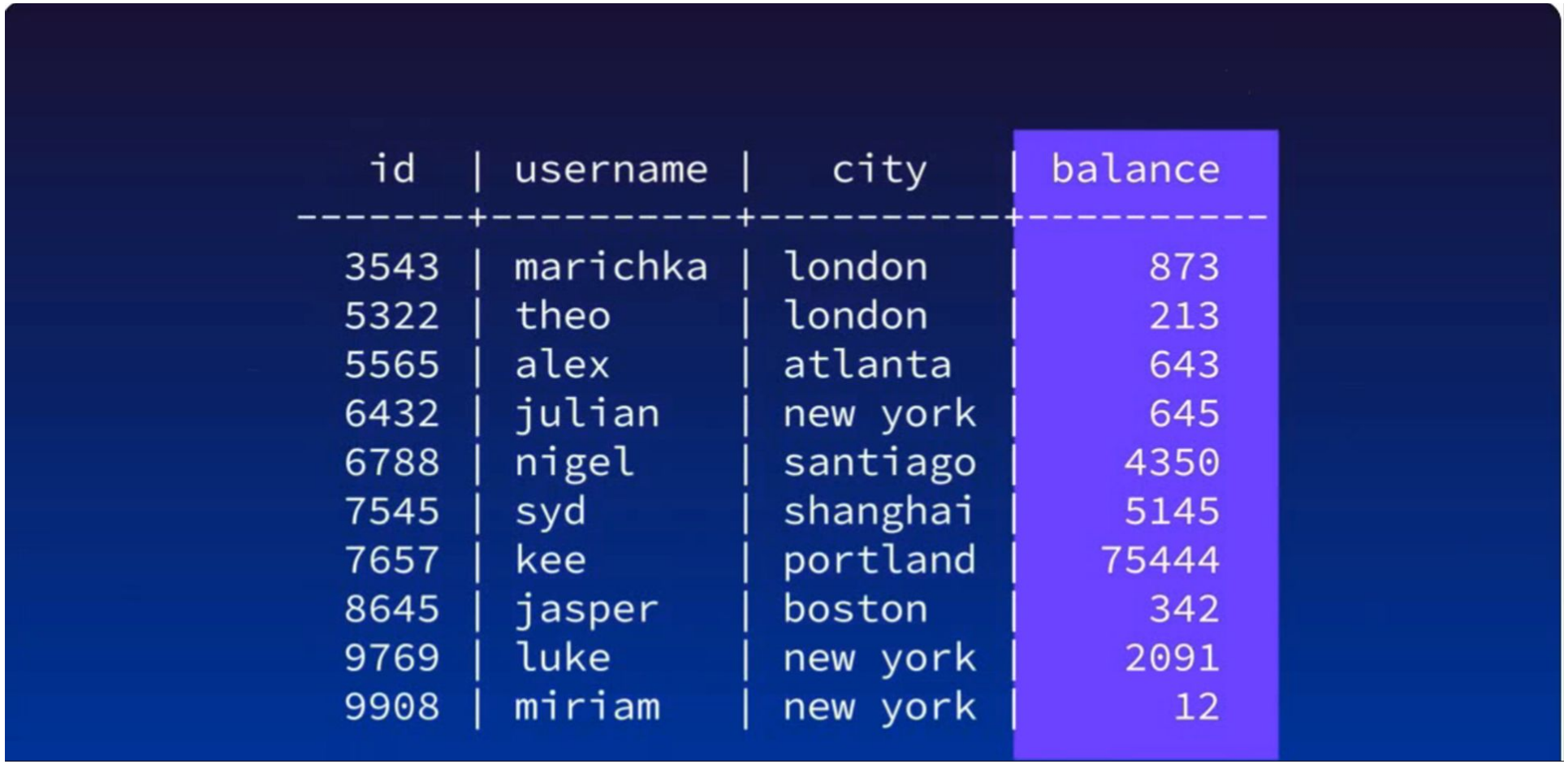
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

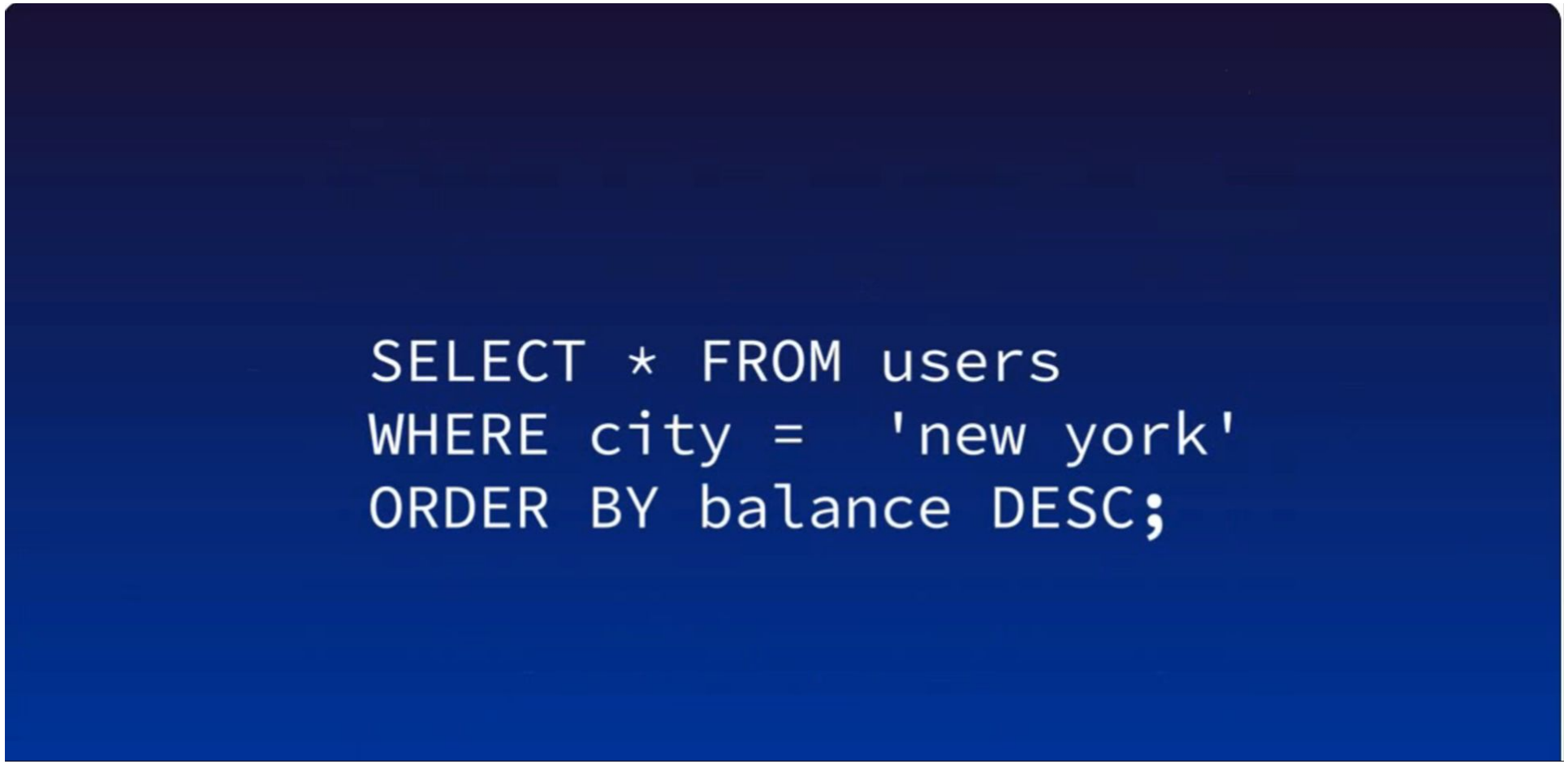
Relational DB Search



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search



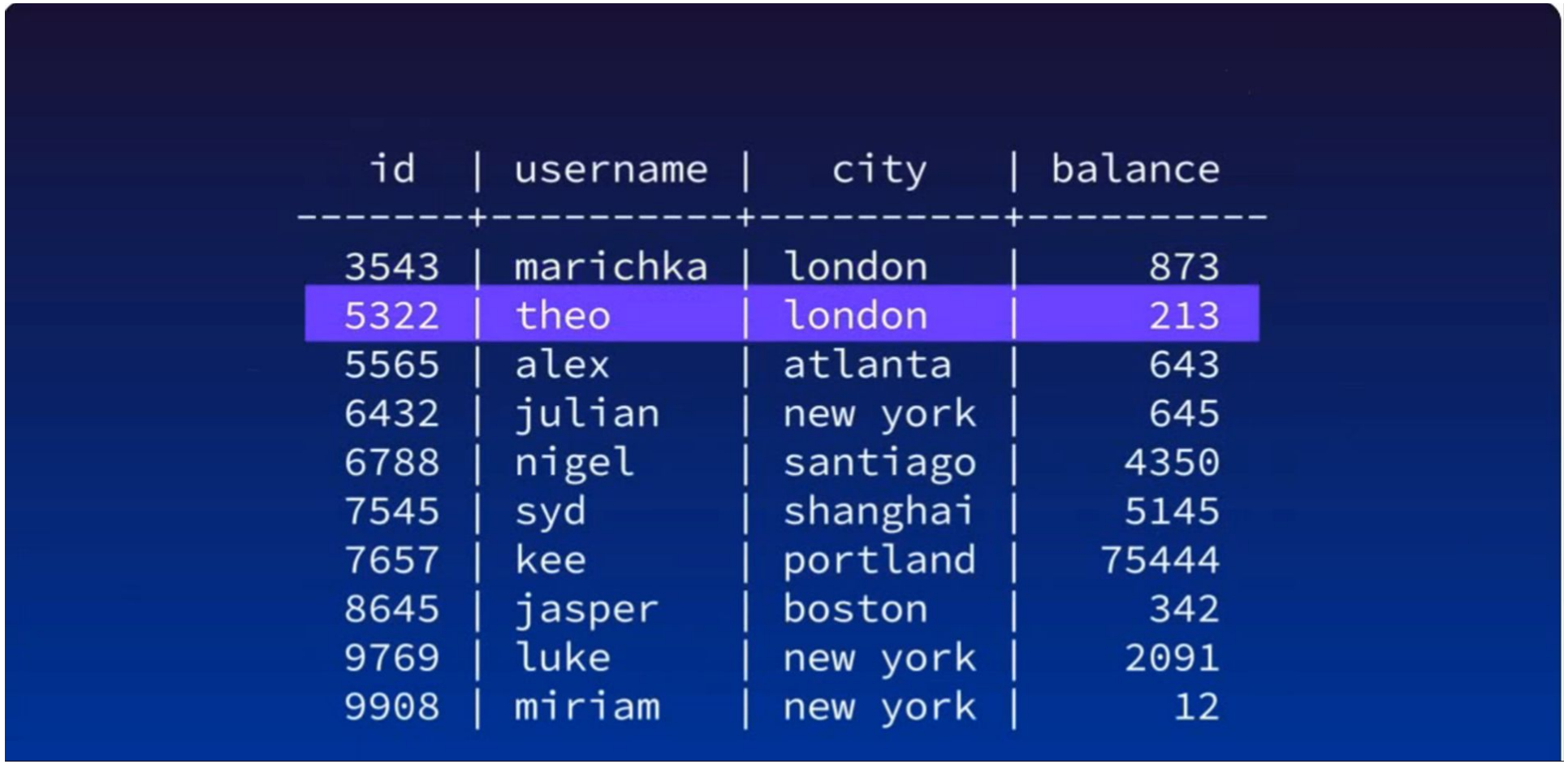
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

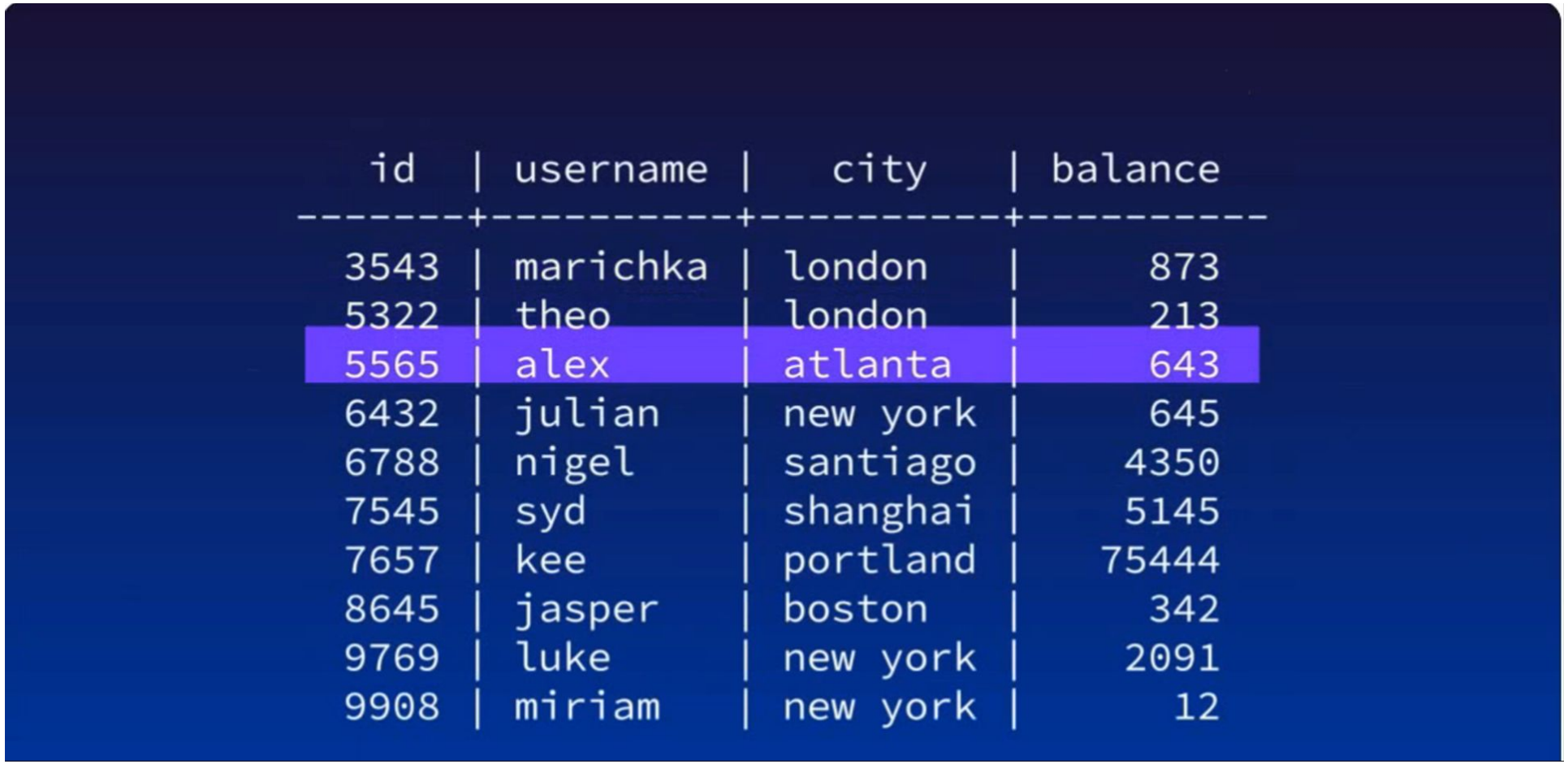
Relational DB Search



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

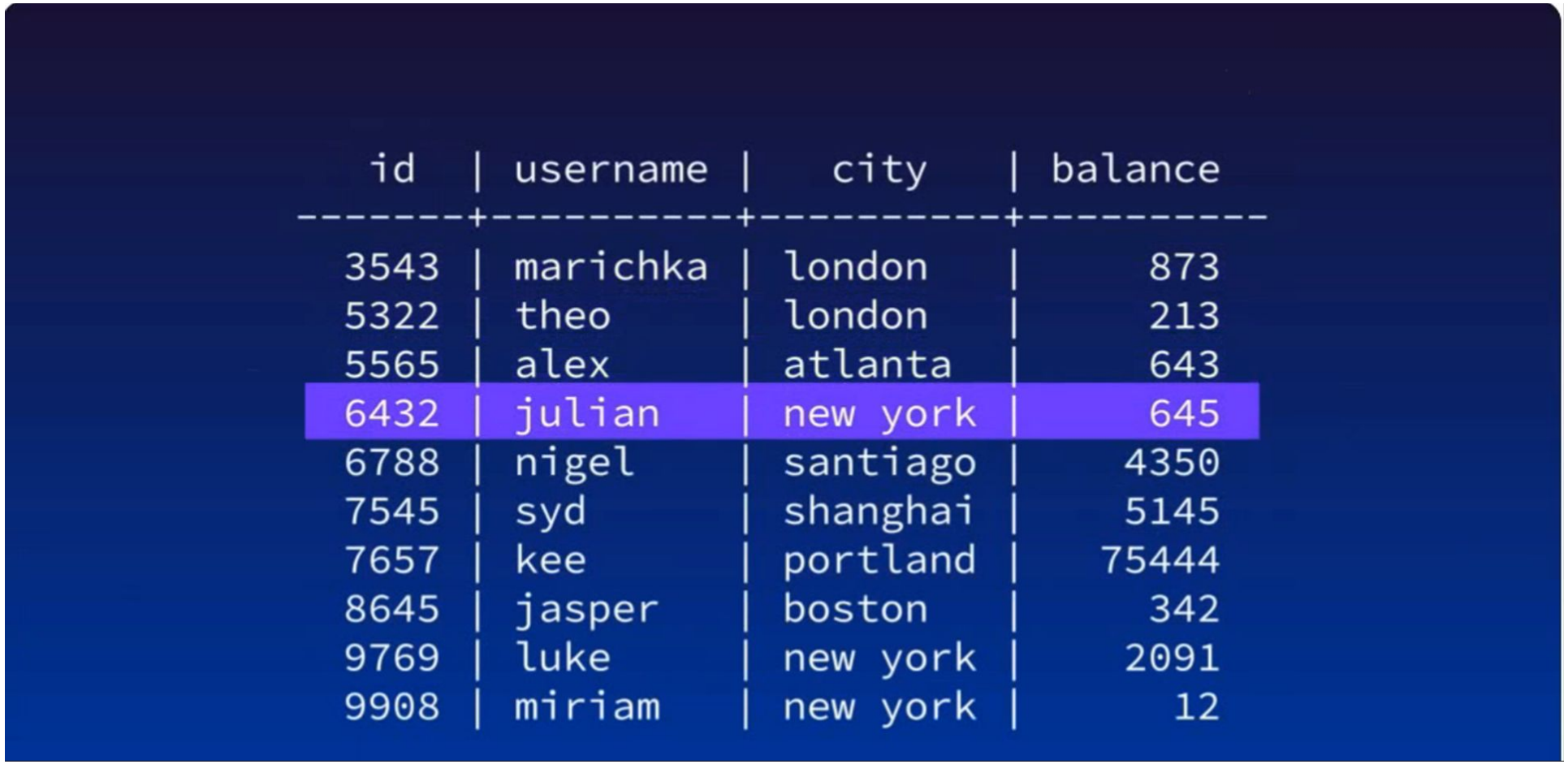
Relational DB Search



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

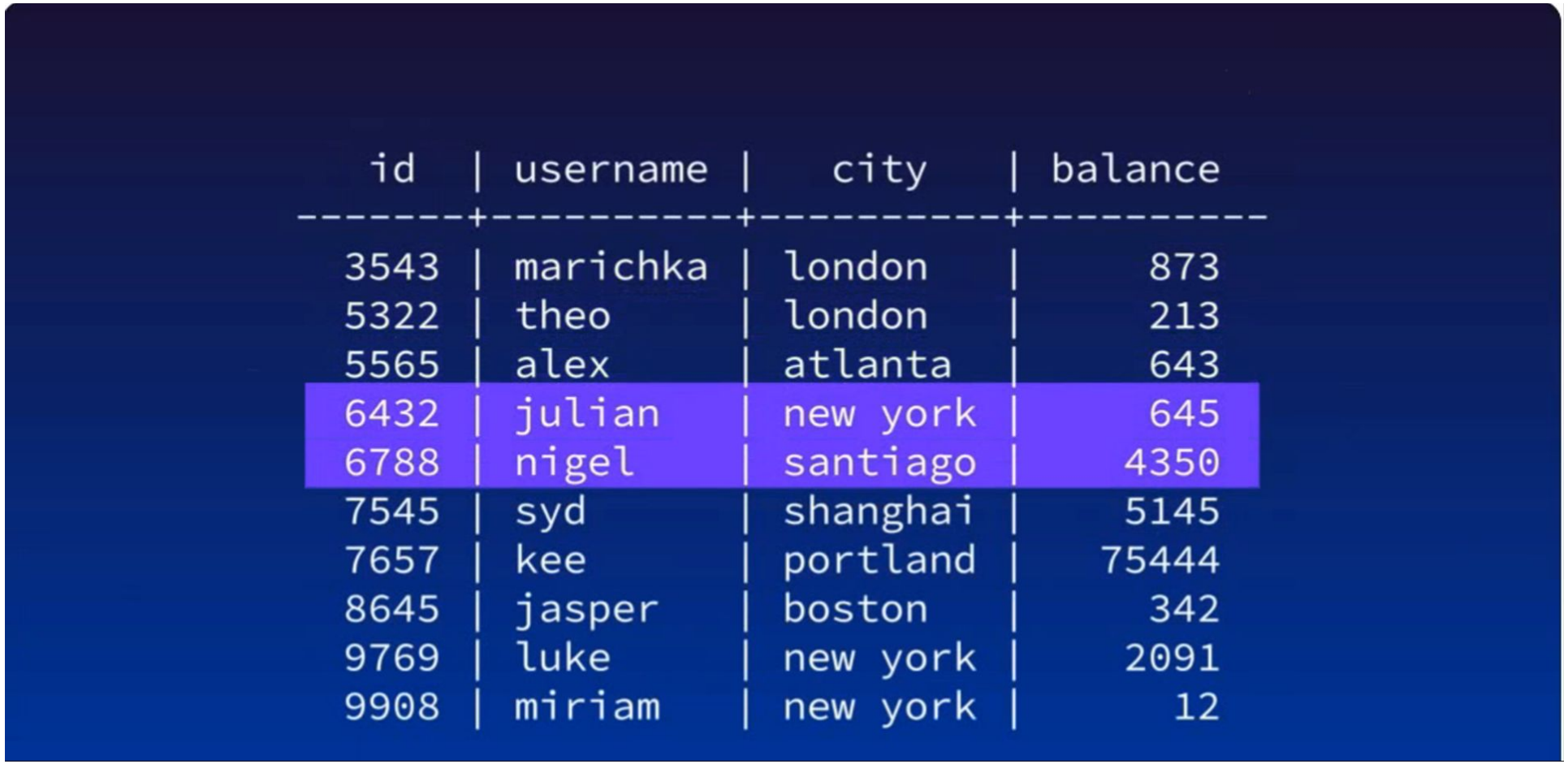
Relational DB Search



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

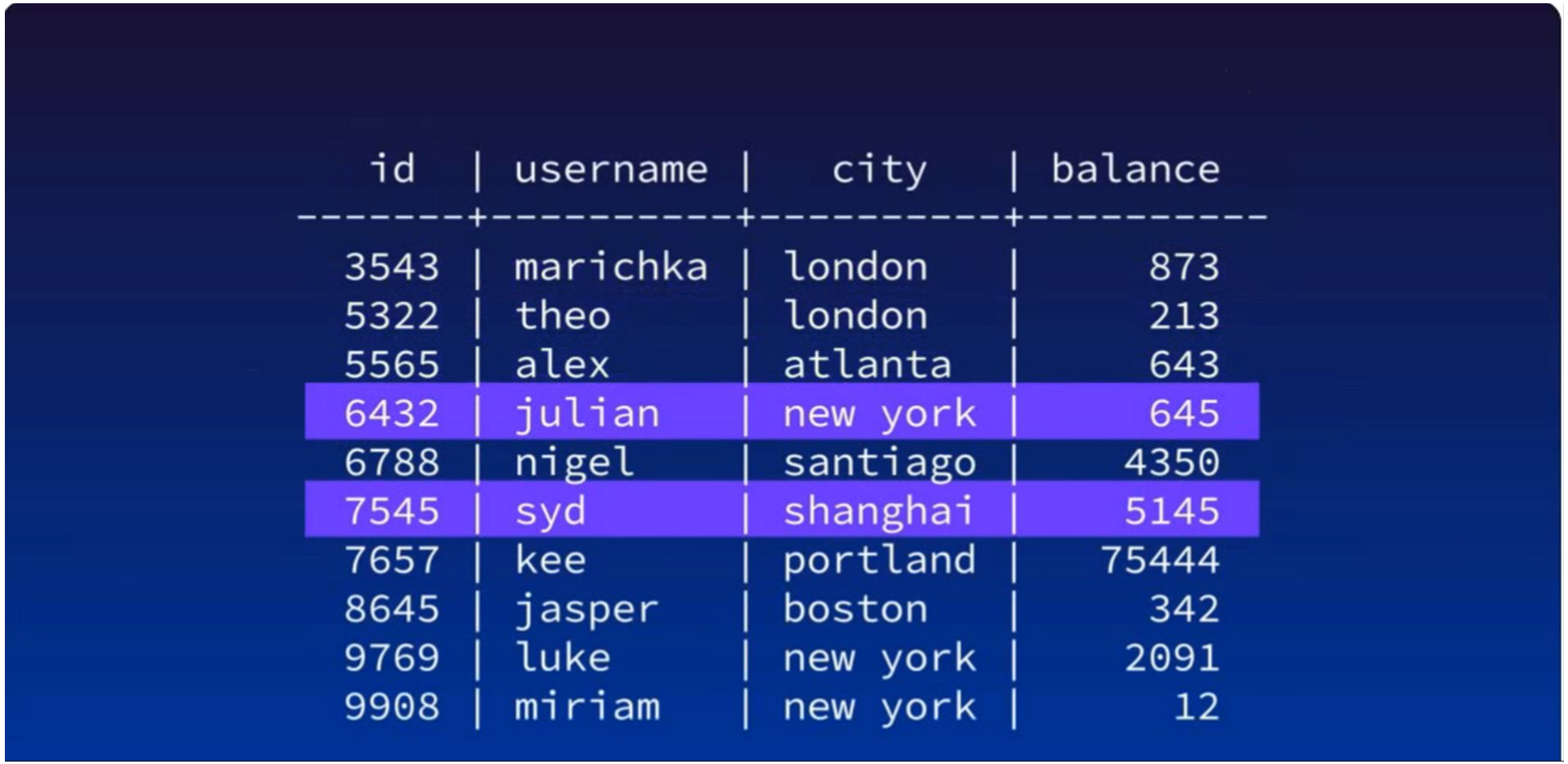
Relational DB Search



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

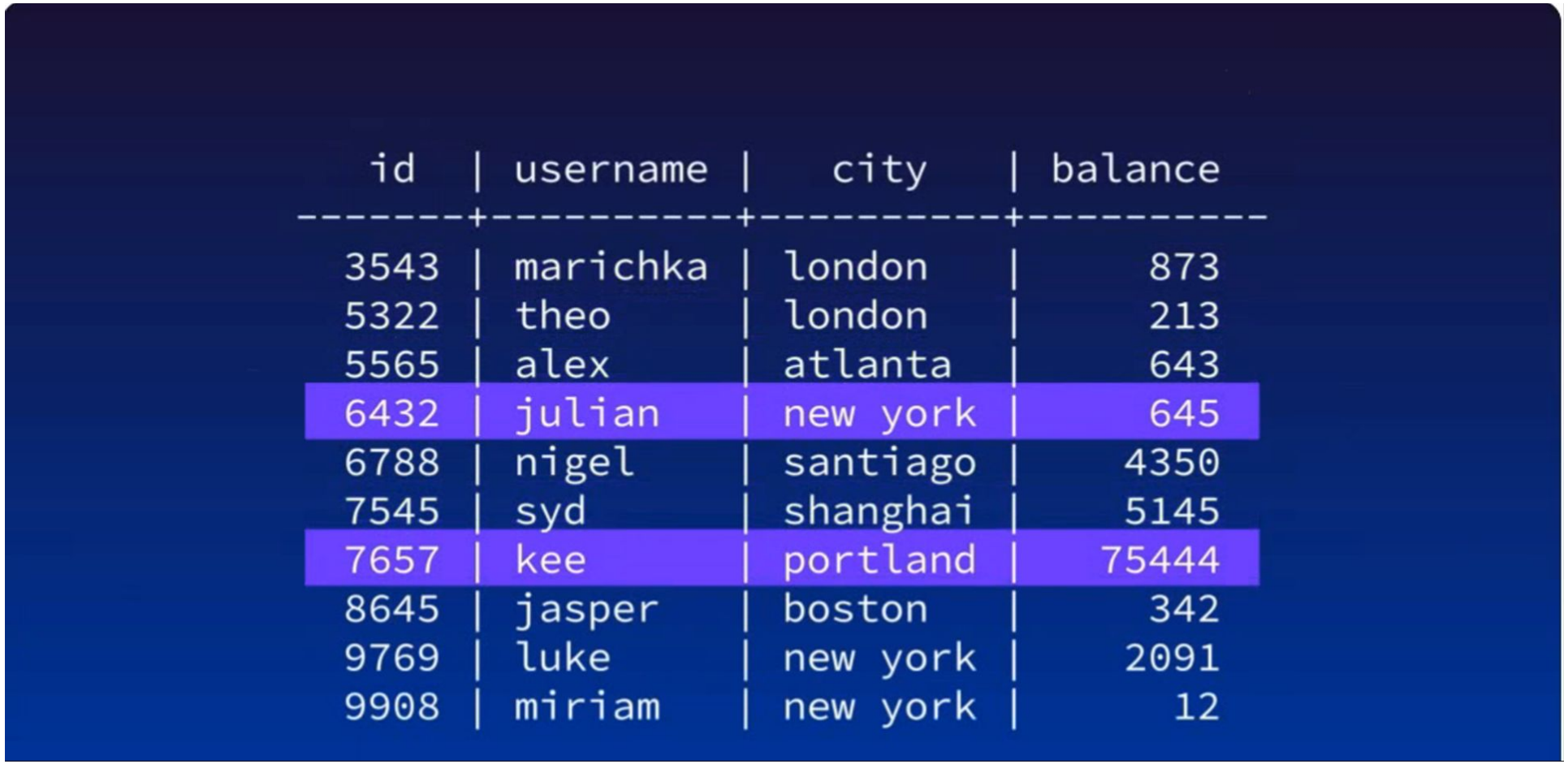
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search



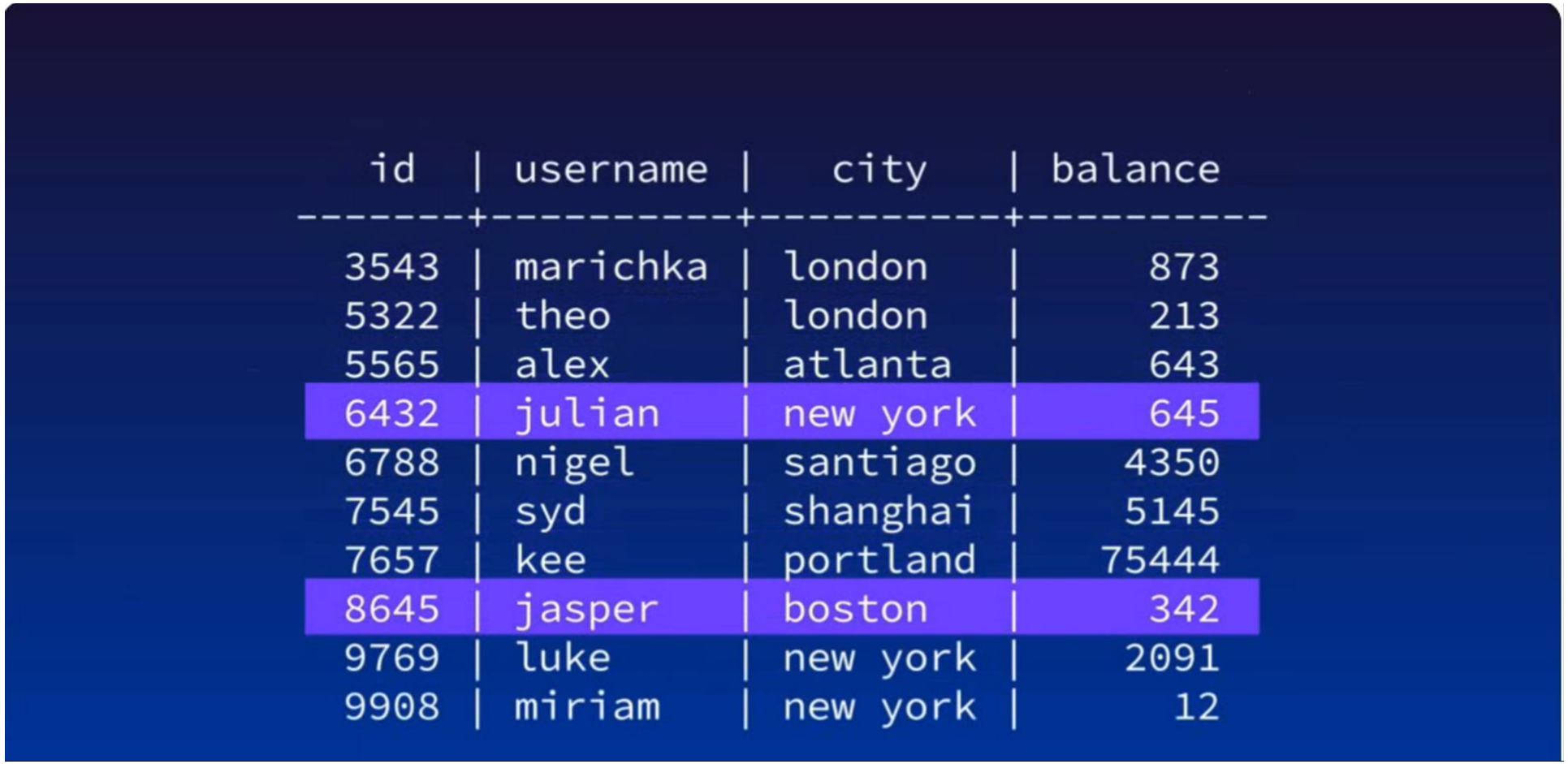
id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI



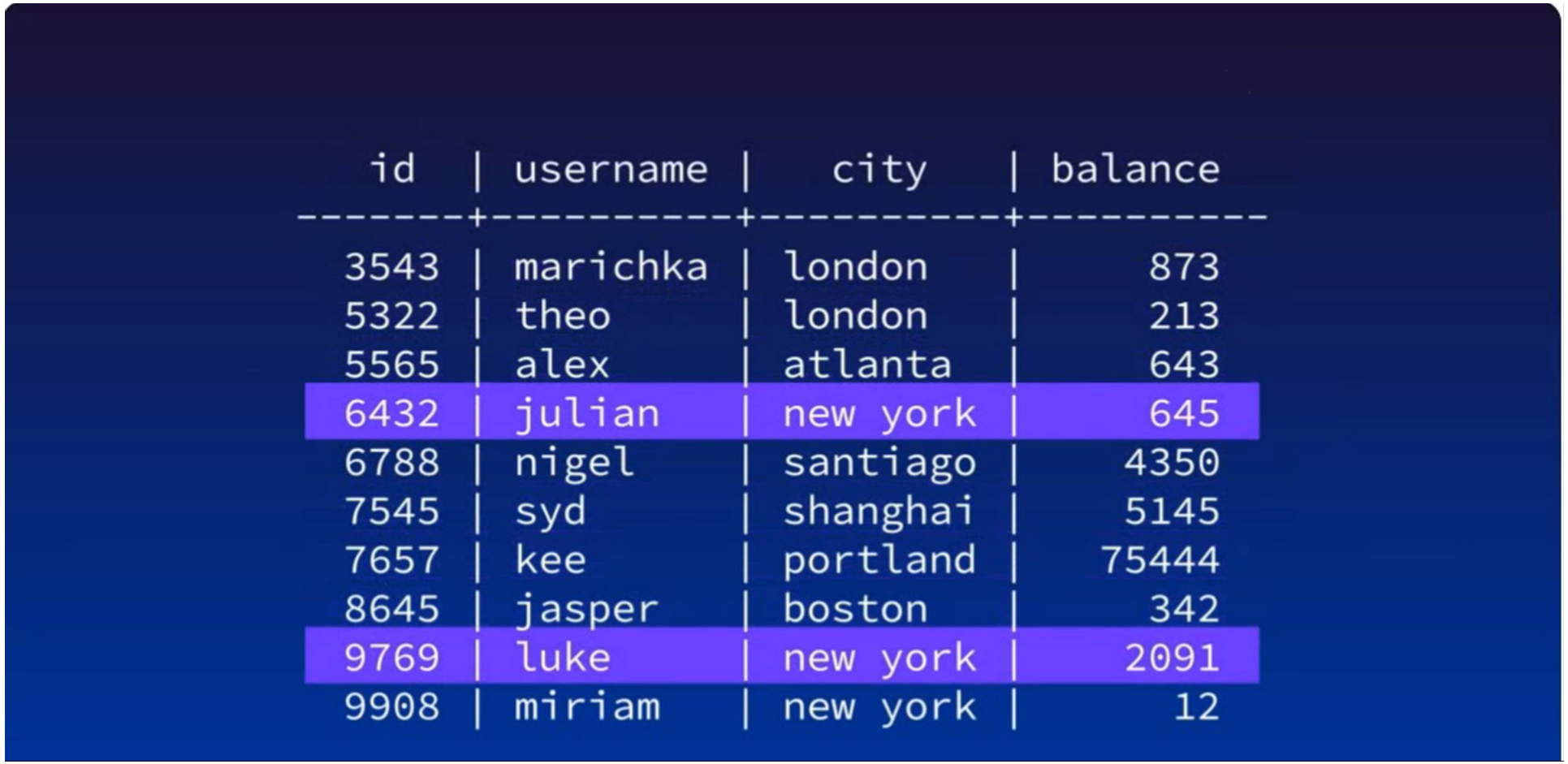
id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

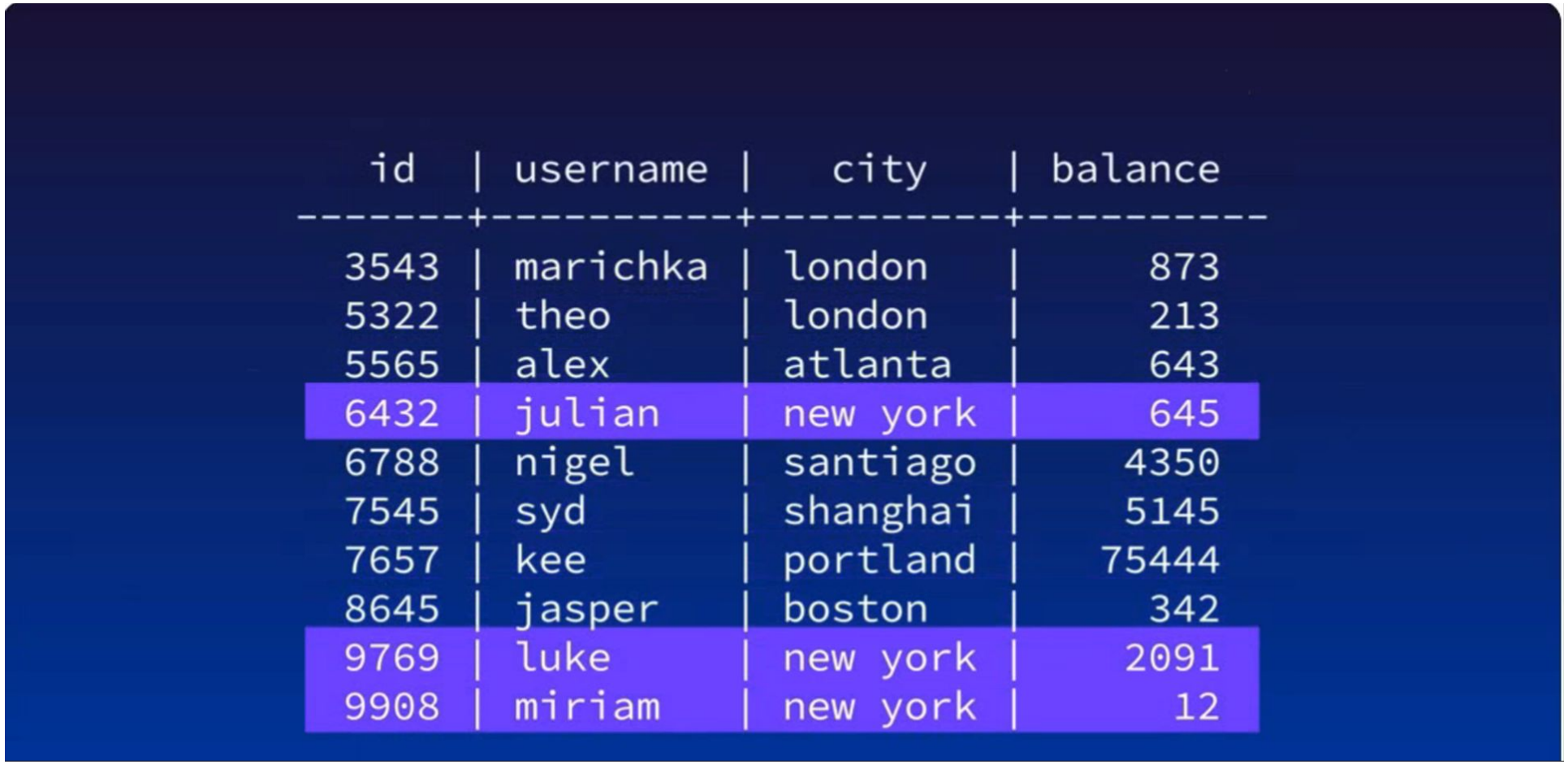
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search



id	username	city	balance
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

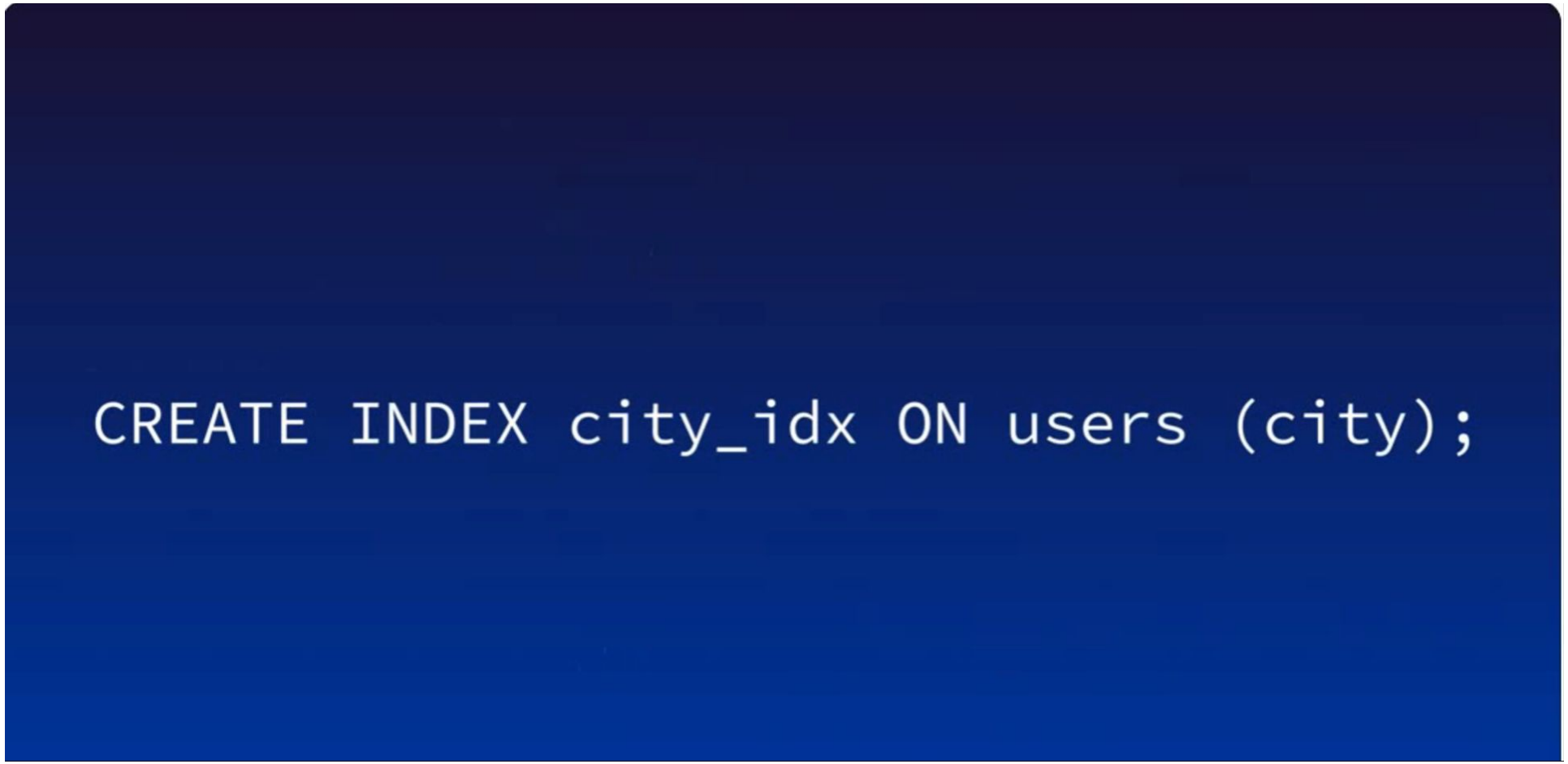
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12
3543	marichka	london	873
5322	theo	london	213
5565	alex	atlanta	643
6432	julian	new york	645
6788	nigel	santiago	4350
7545	syd	shanghai	5145
7657	kee	portland	75444
8645	jasper	boston	342
9769	luke	new york	2091
9908	miriam	new york	12
3543	marichka	london	873

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

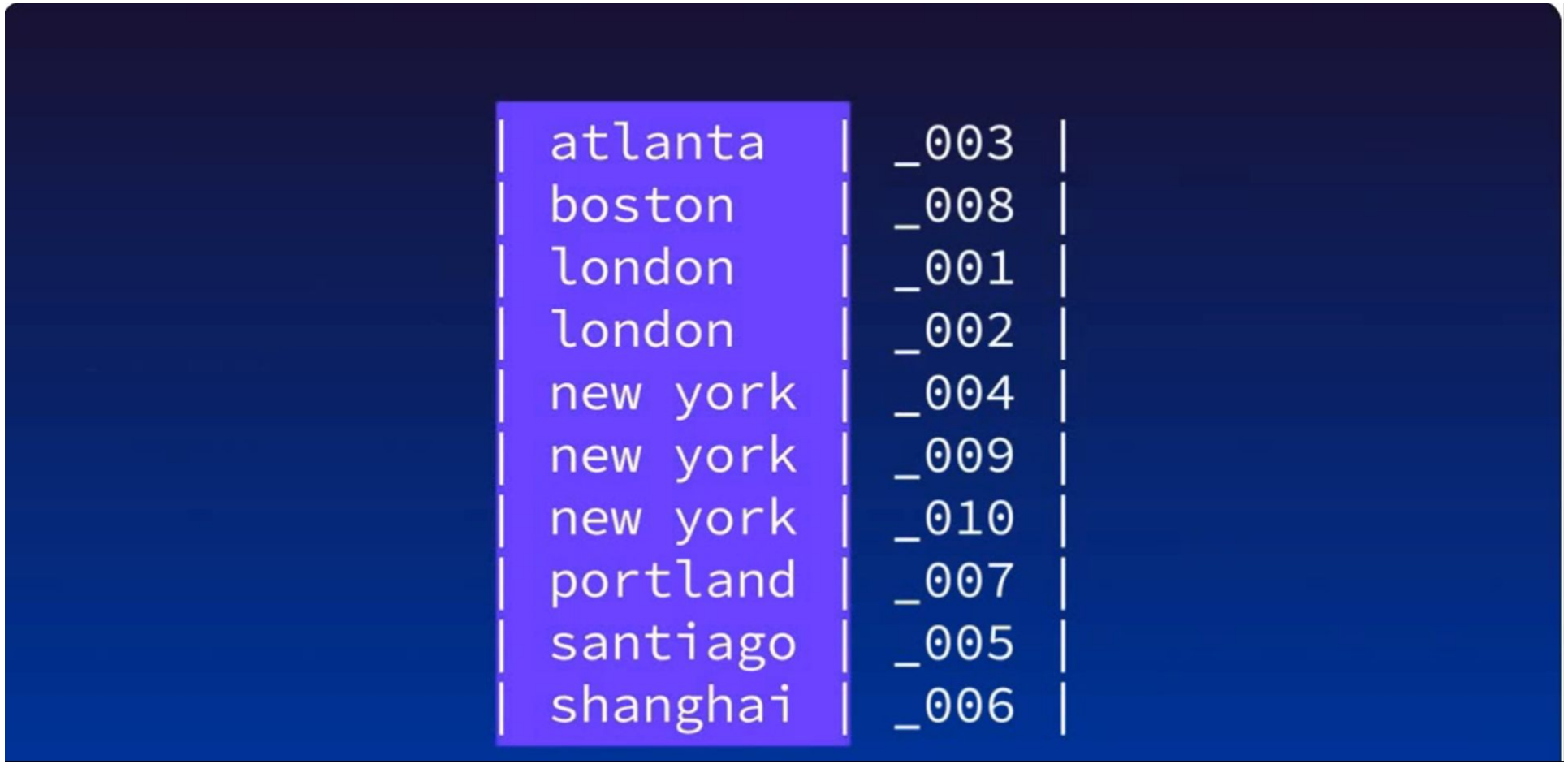
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search



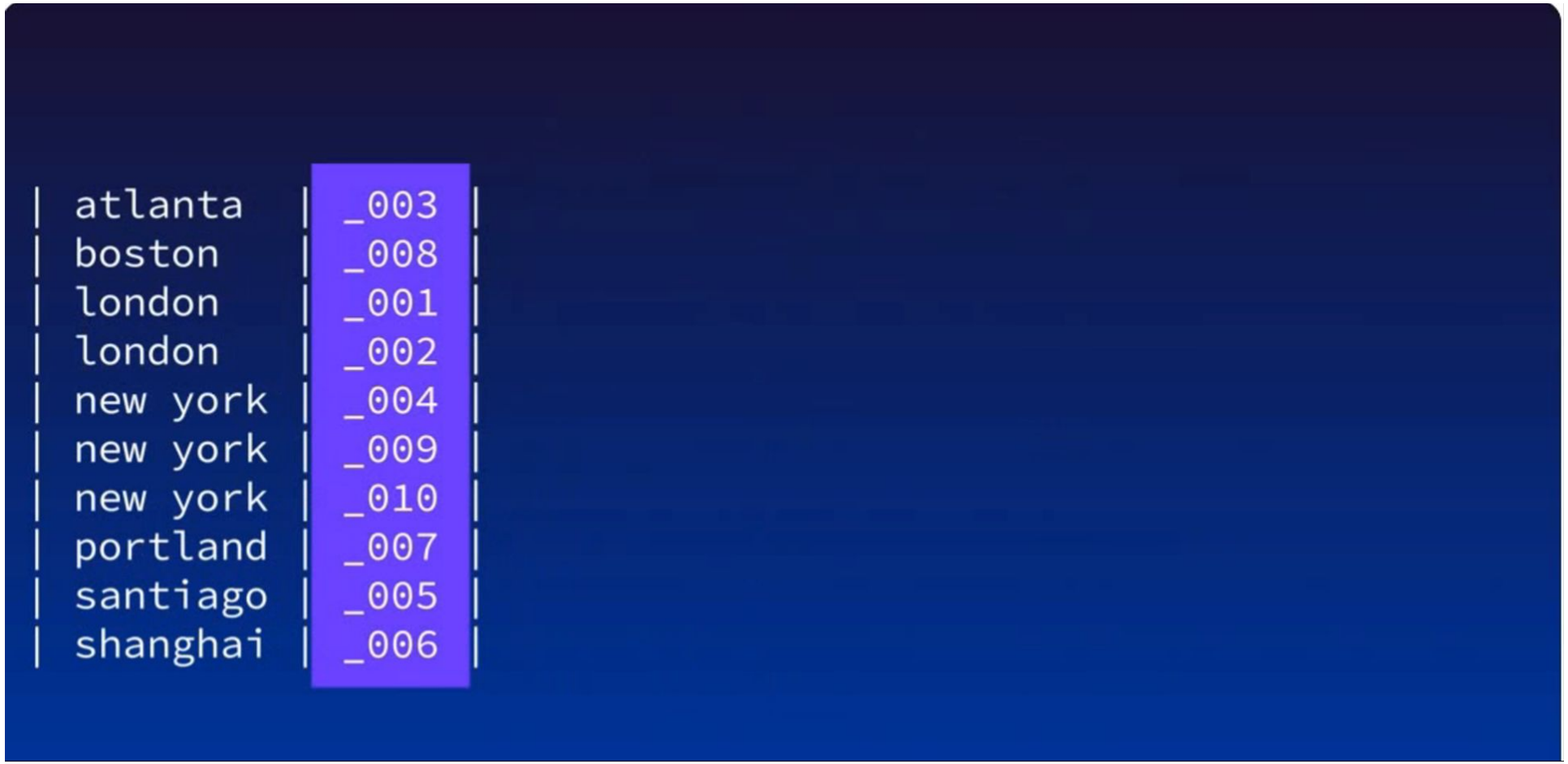
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search

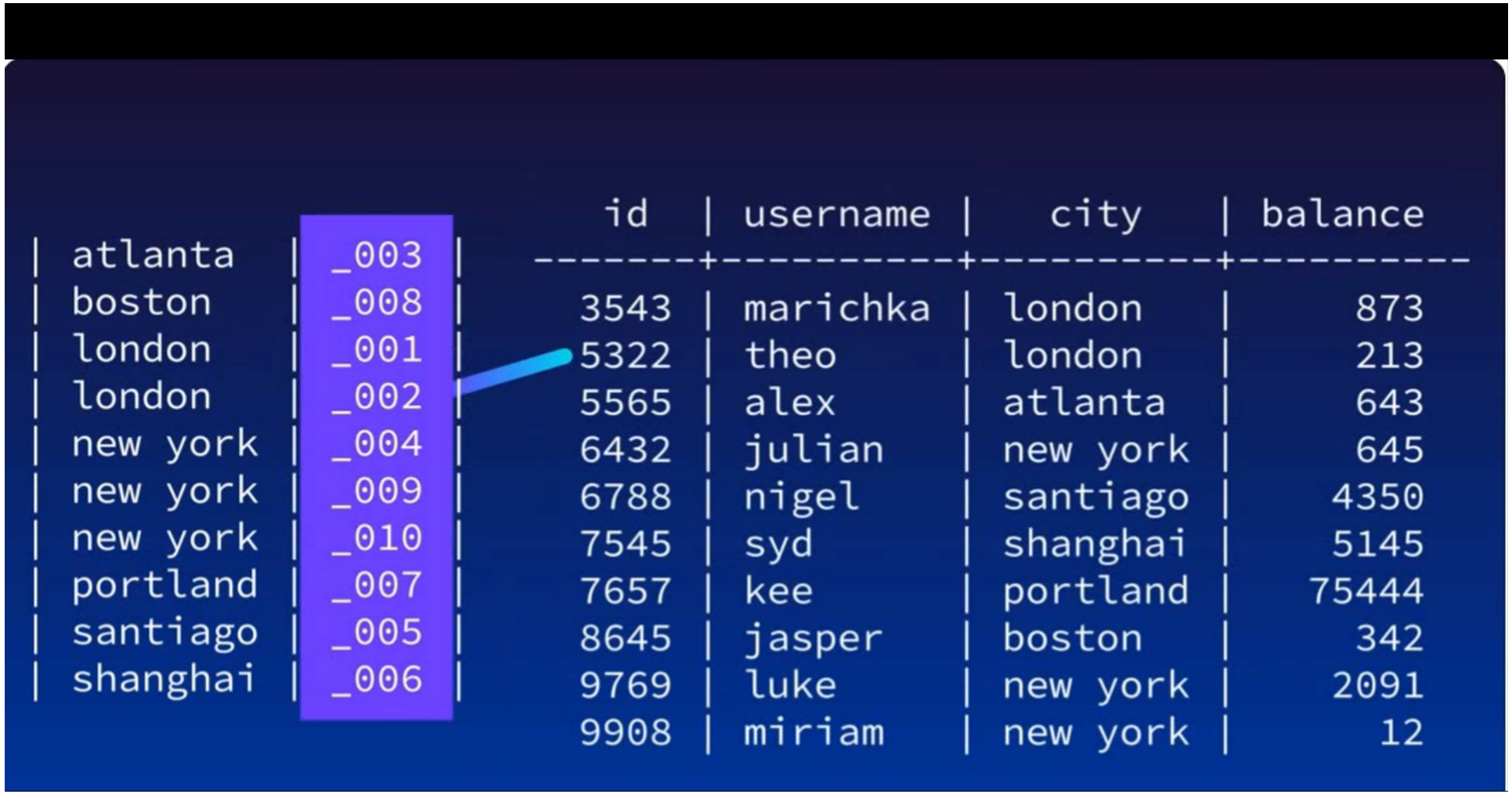


SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Relational DB Search



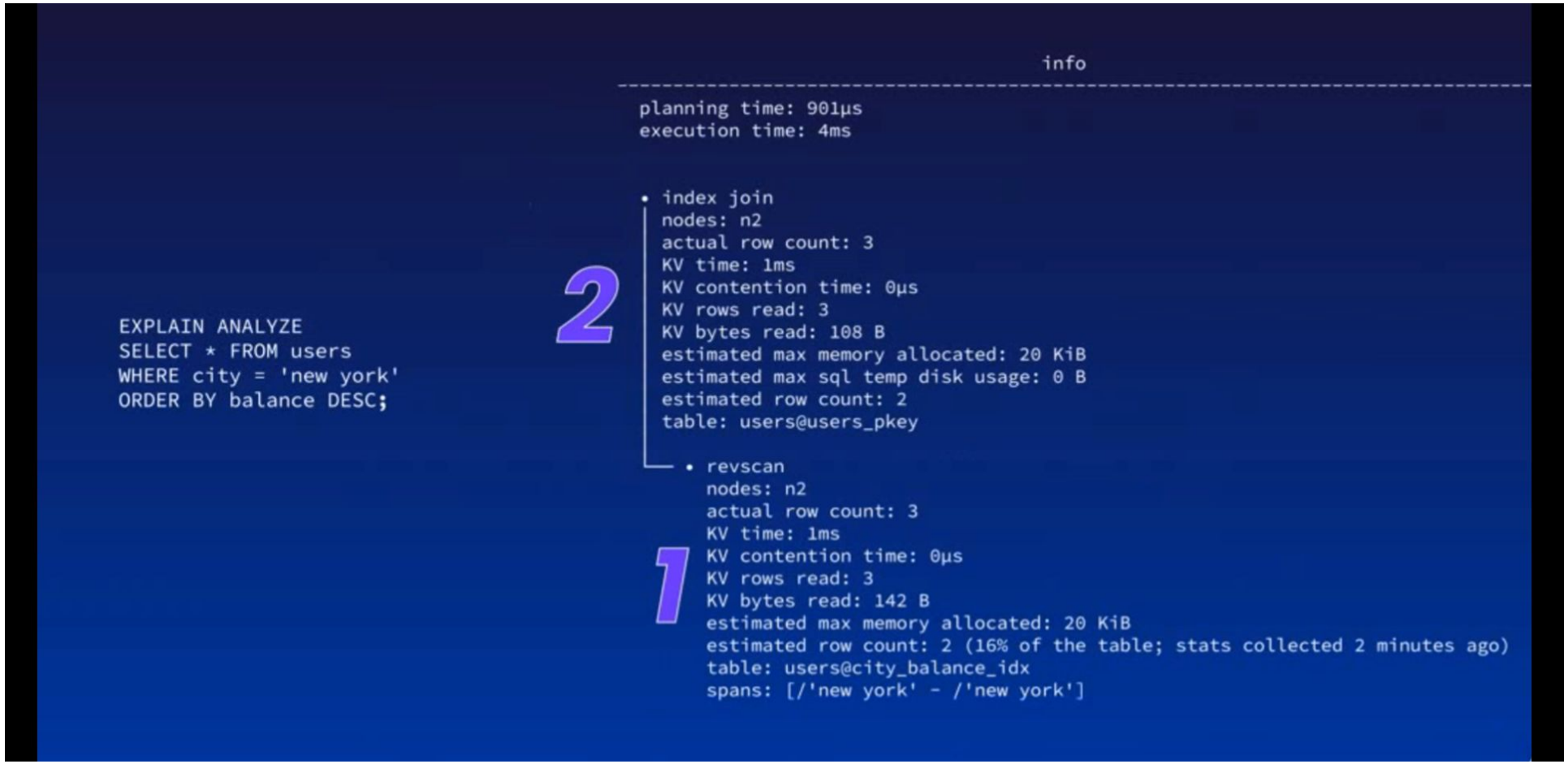
SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI



SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

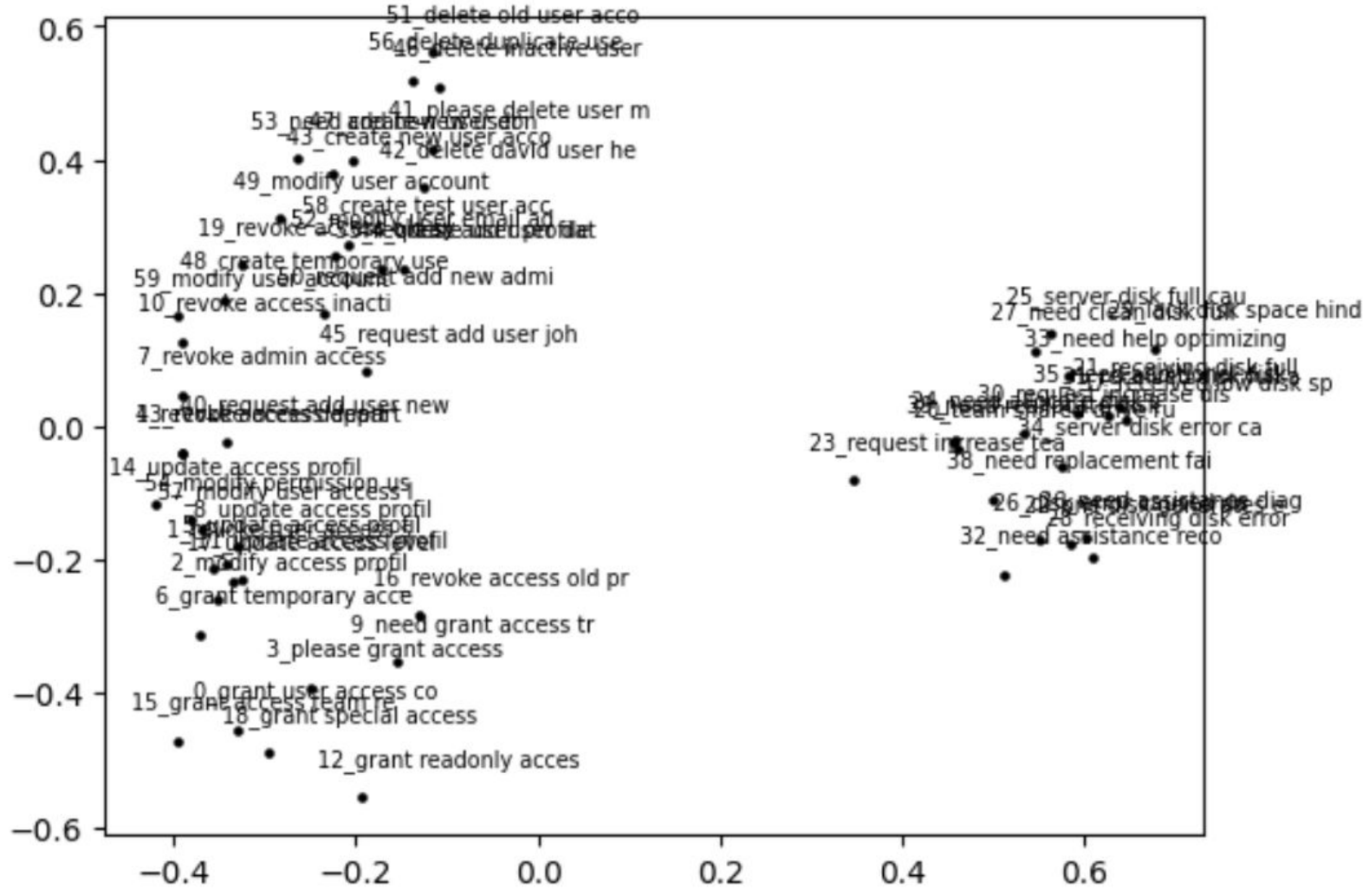
Relational DB Search



SQL indexing best practices | How to make your database FASTER!
https://www.youtube.com/watch?v=BIlFTFrEFOI

Vector Space Search

Vector Space Search

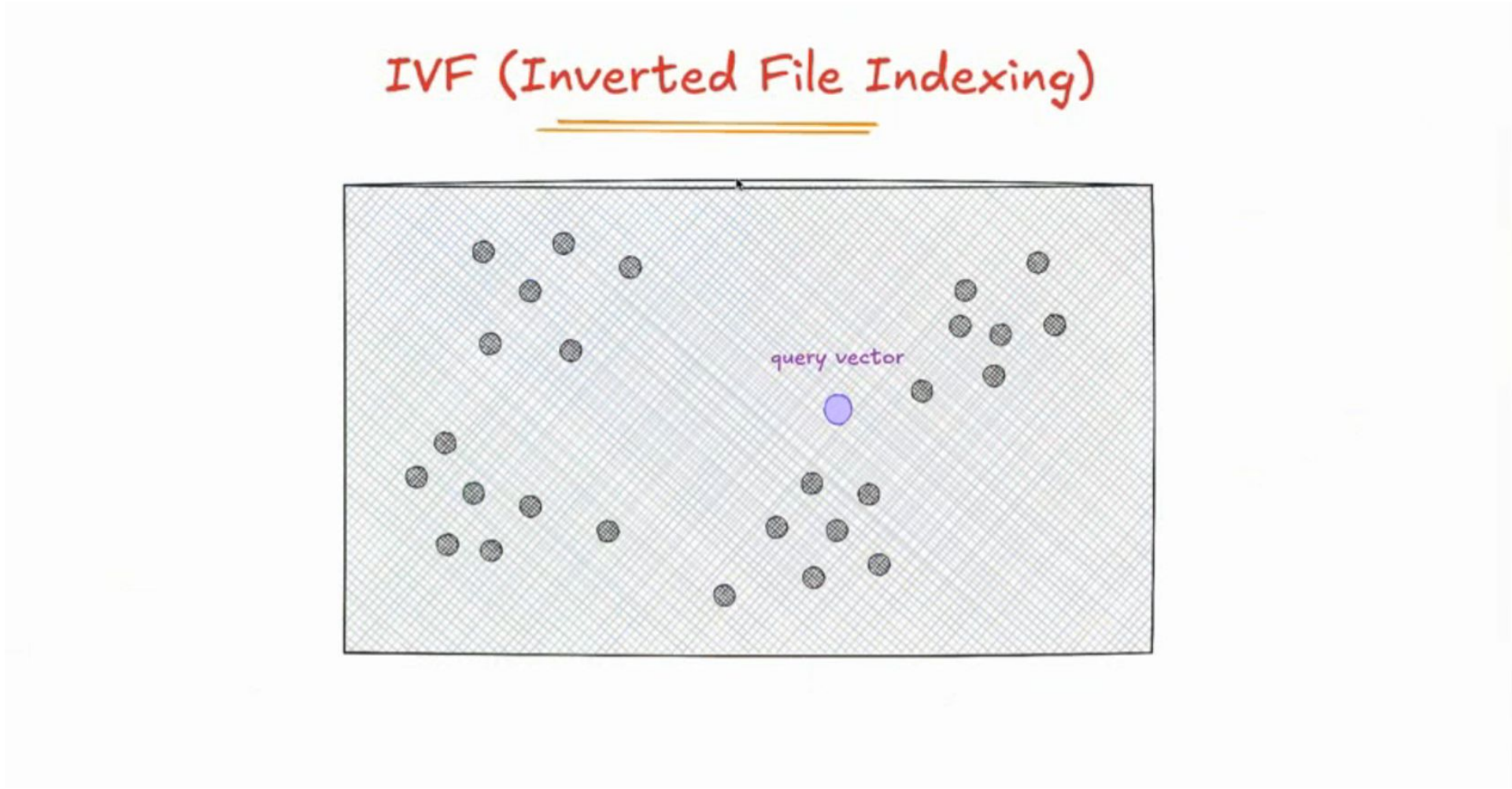


Support Ticket Classification

Github: https://github.com/enoten/support_ticket_analysis/blob/main/support_tickets_classification.ipynb

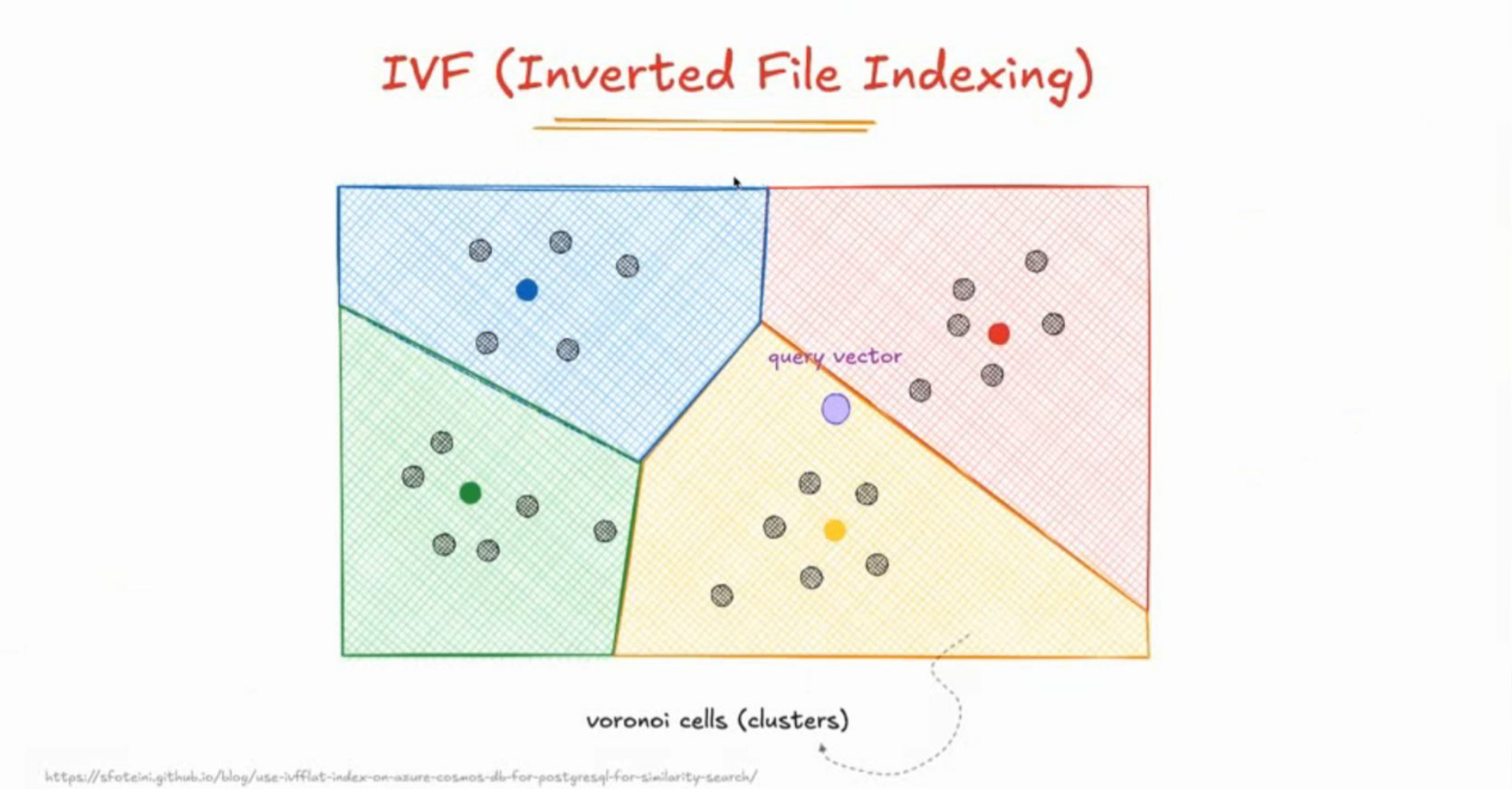
Youtube: https://www.youtube.com/watch?v=NlJ1yS0F03M

Vector Space Search: IVF indexing



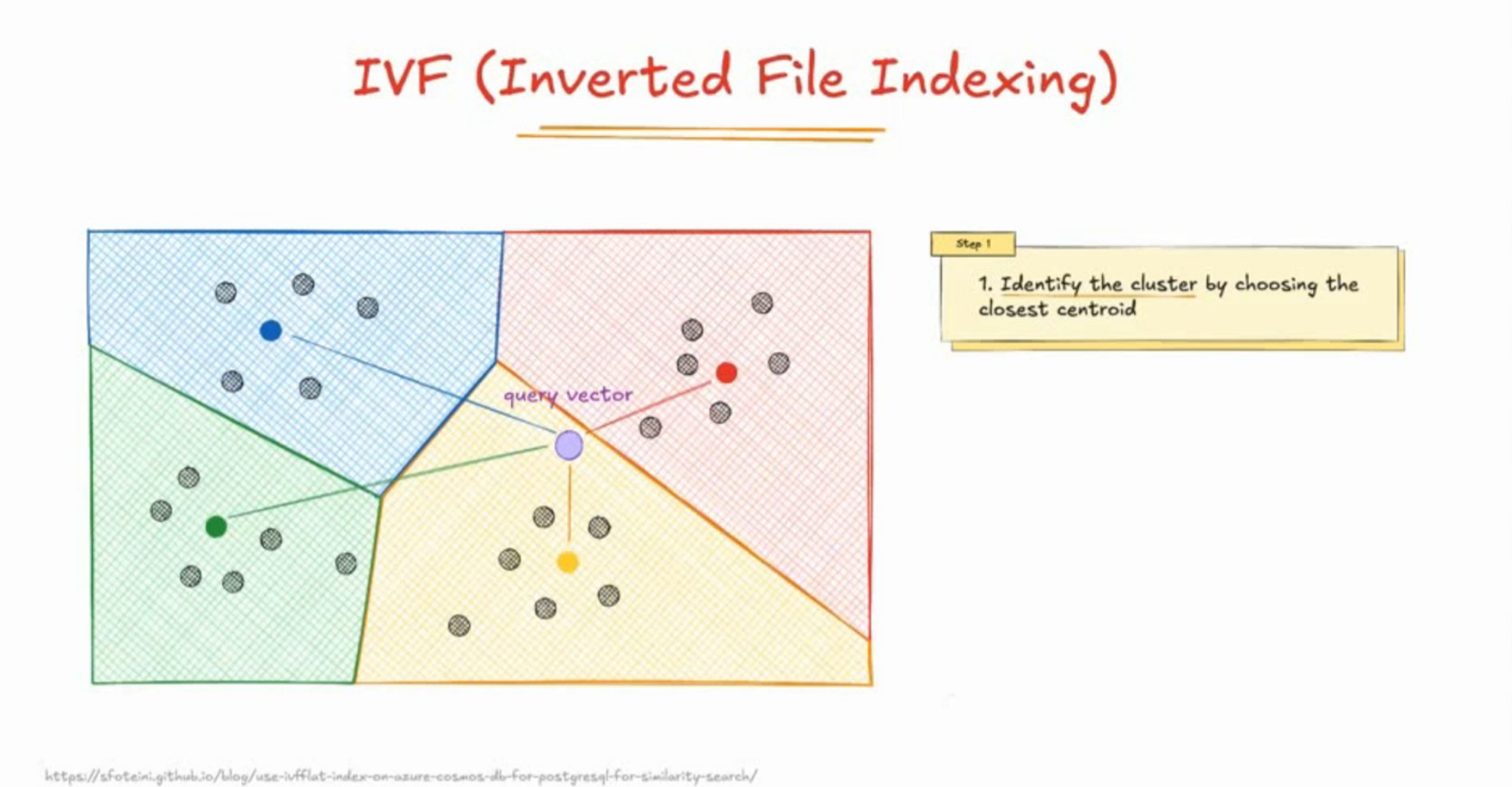
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: IVF indexing



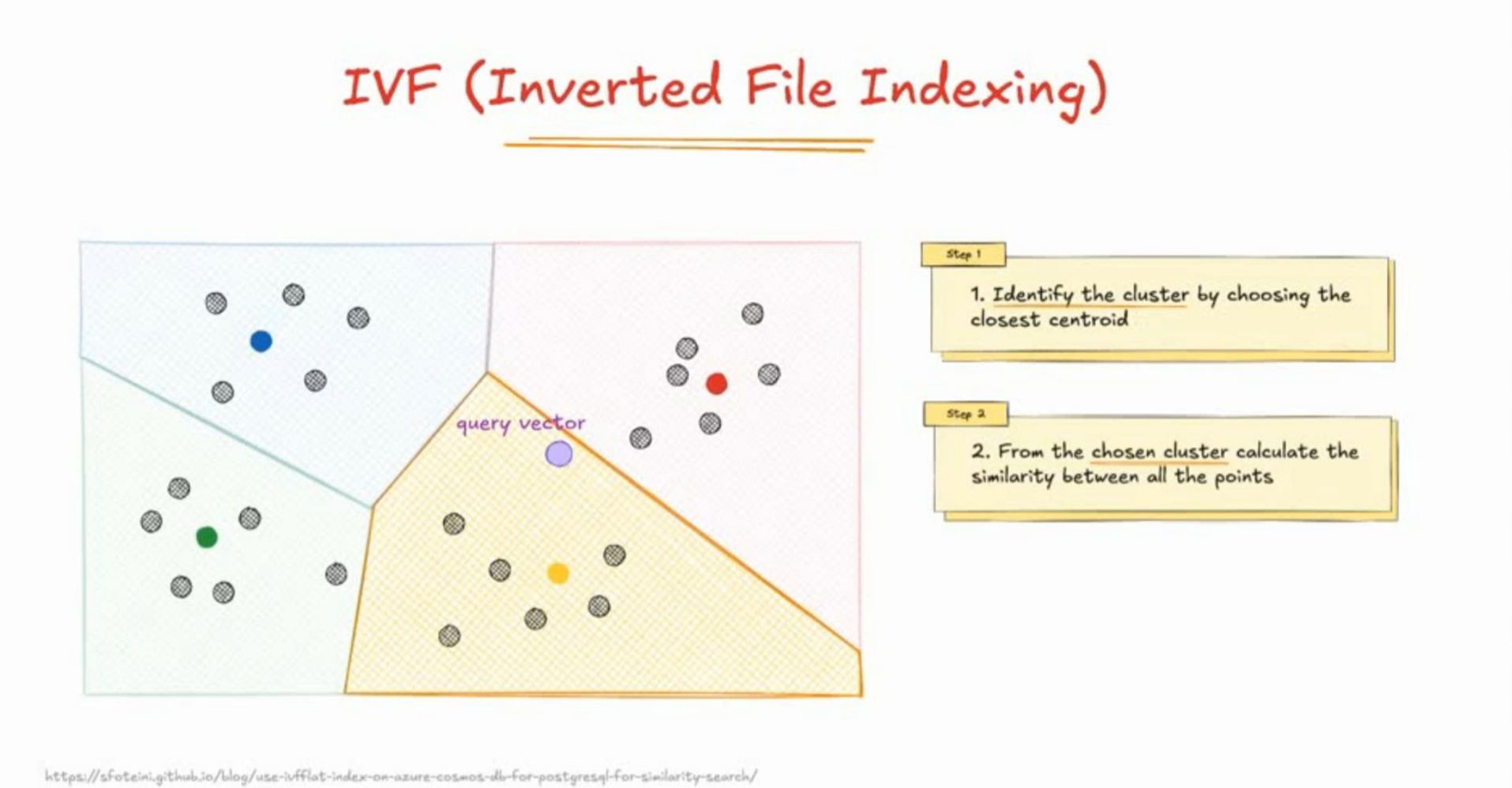
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: IVF indexing



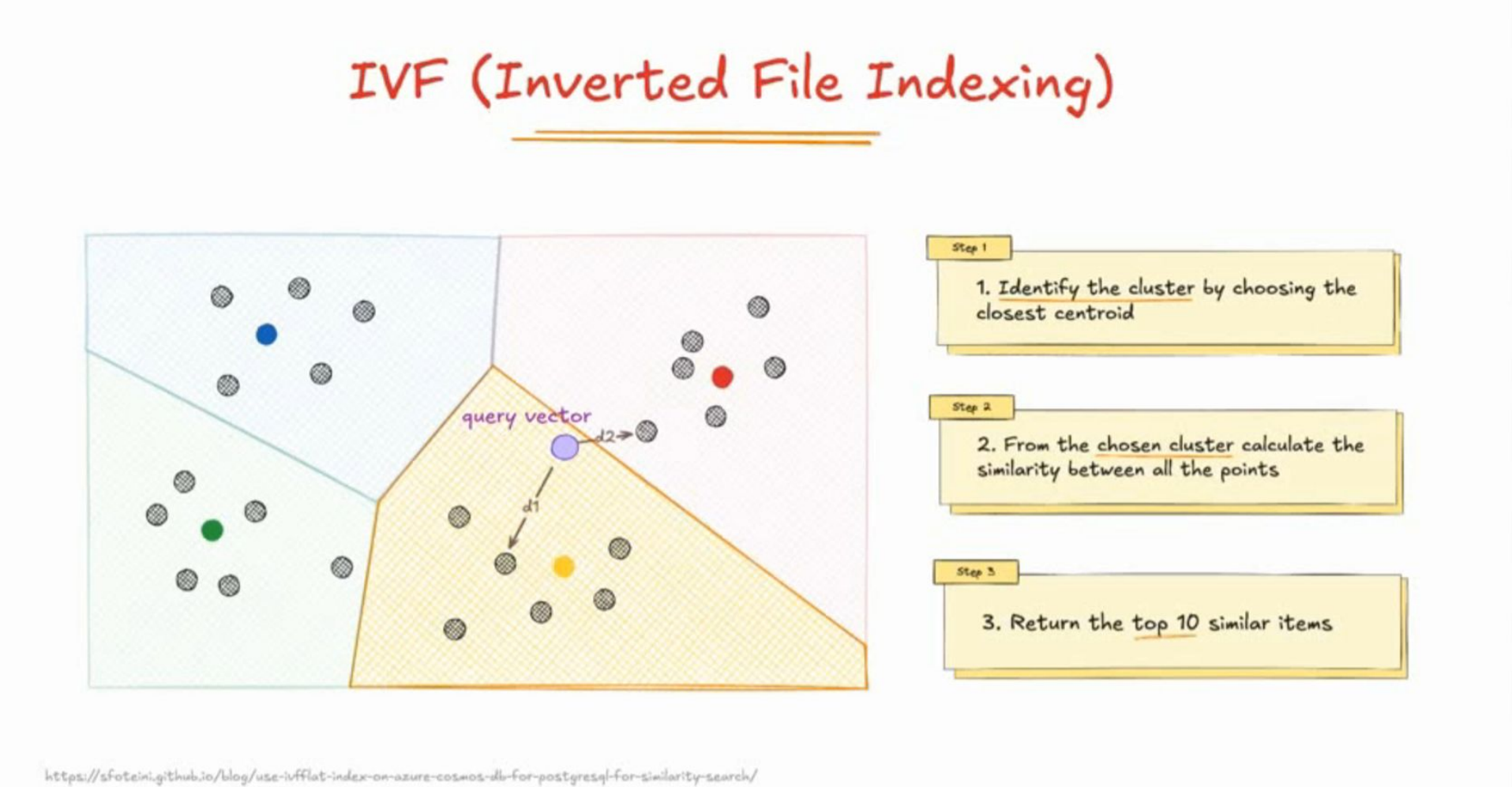
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: IVF indexing



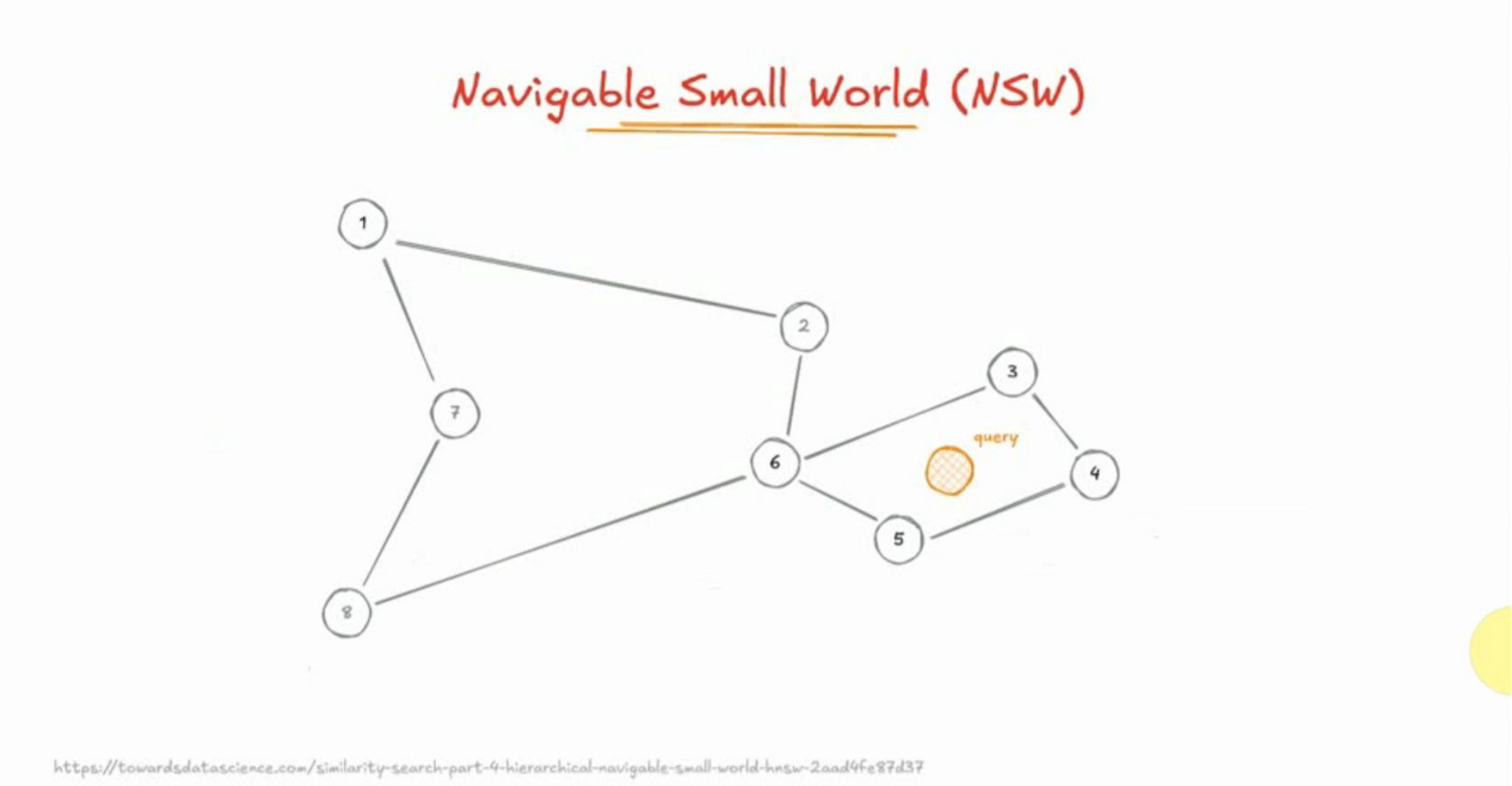
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: IVF indexing



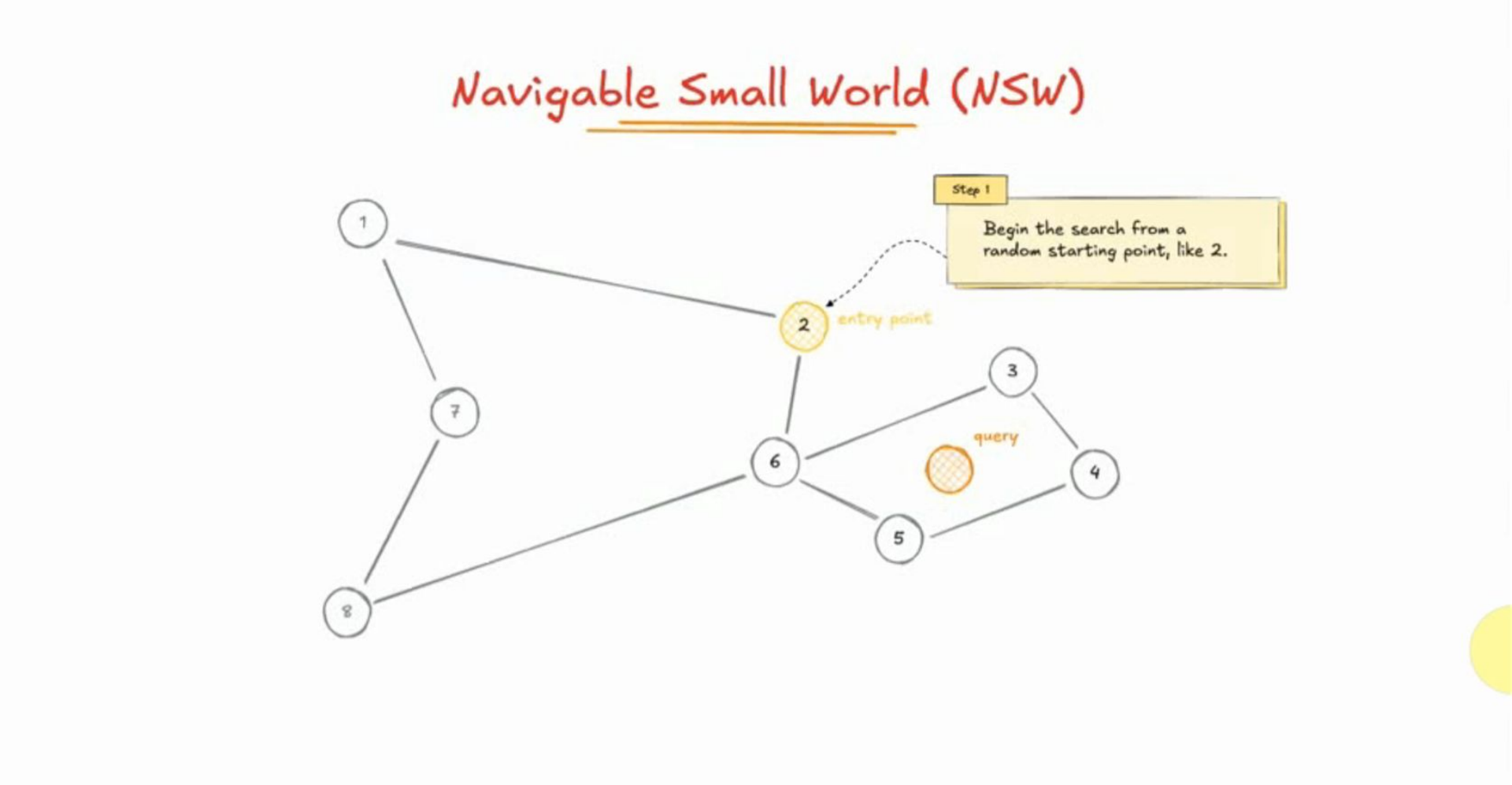
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



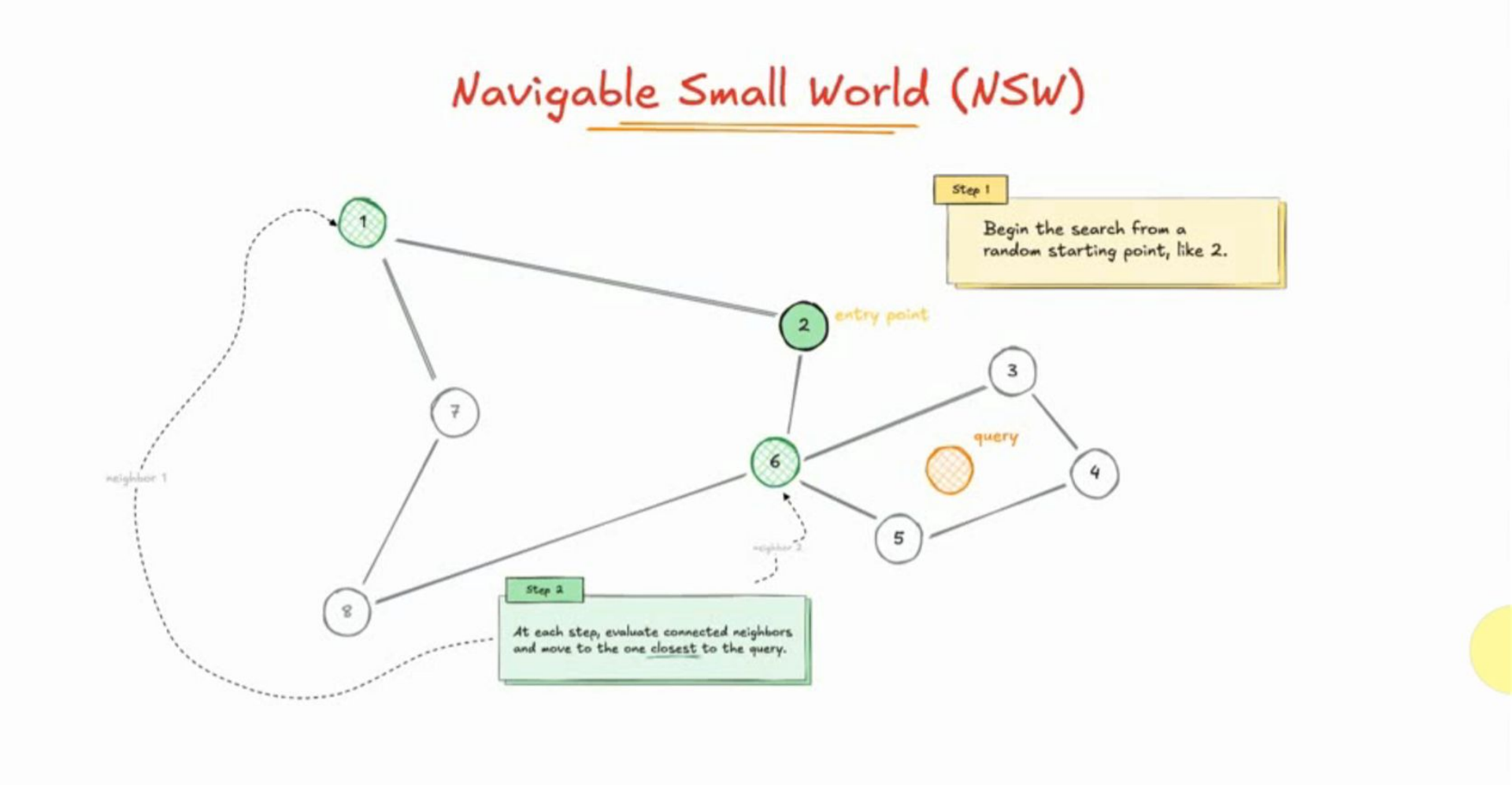
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



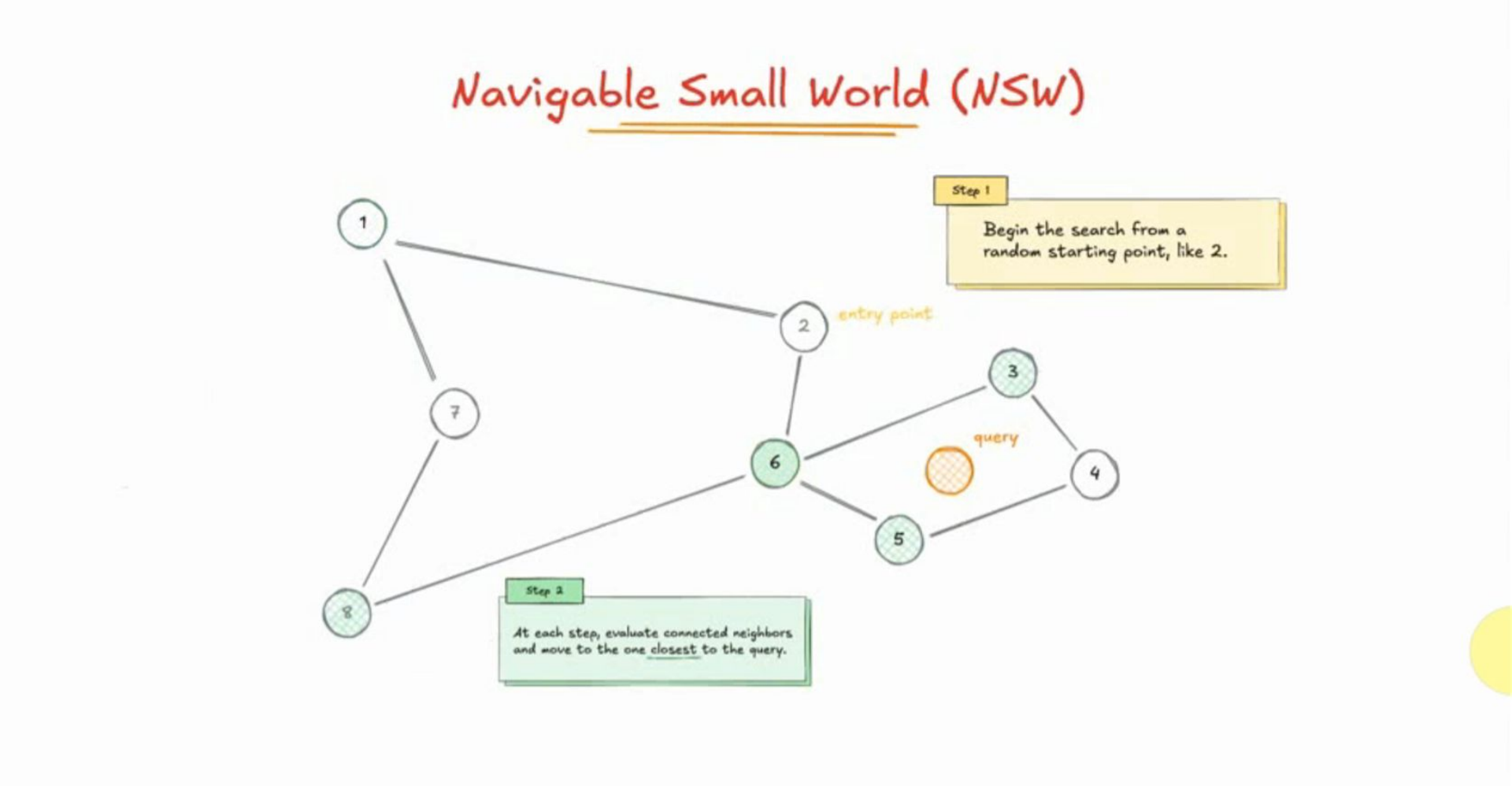
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



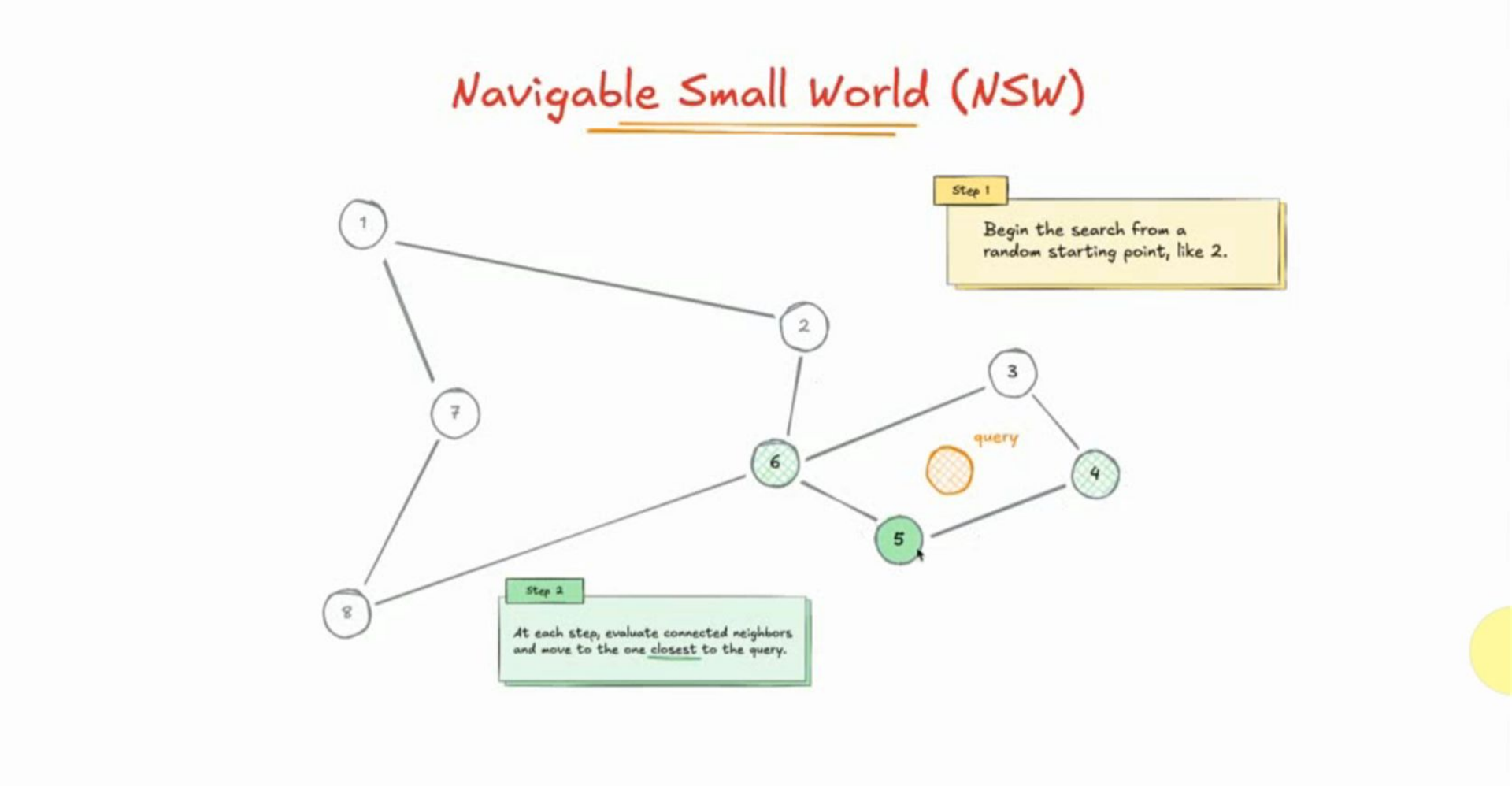
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



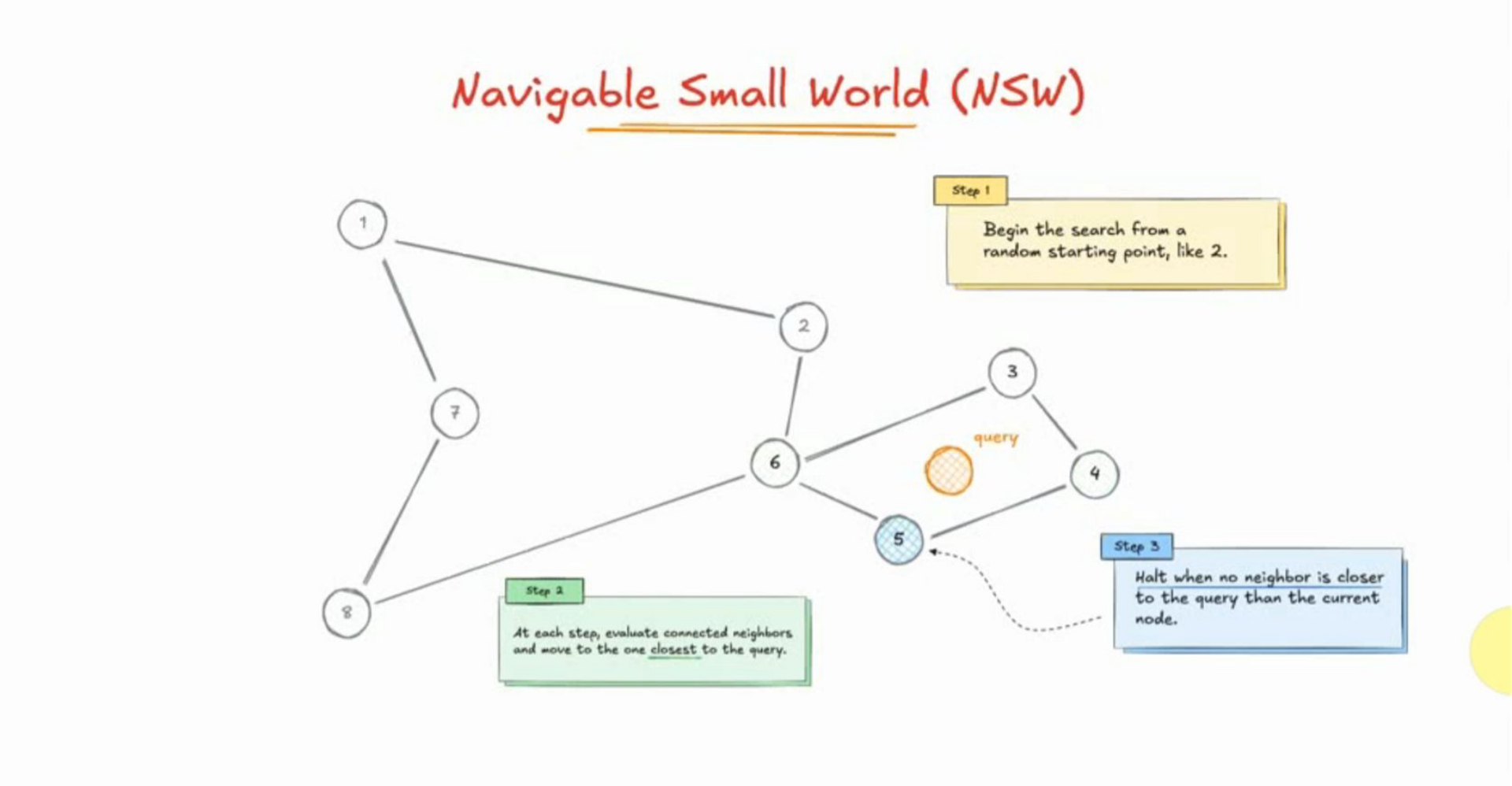
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



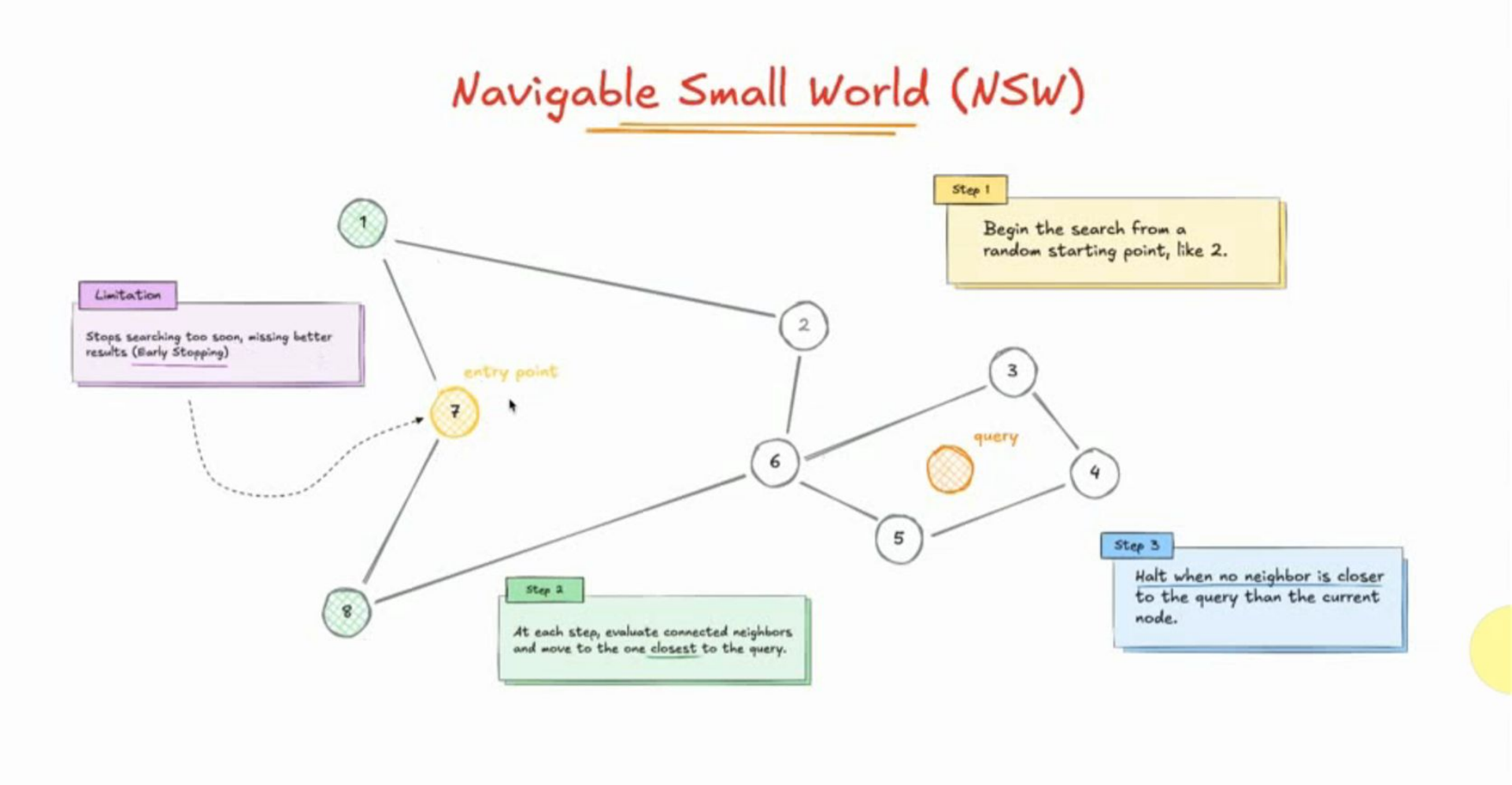
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



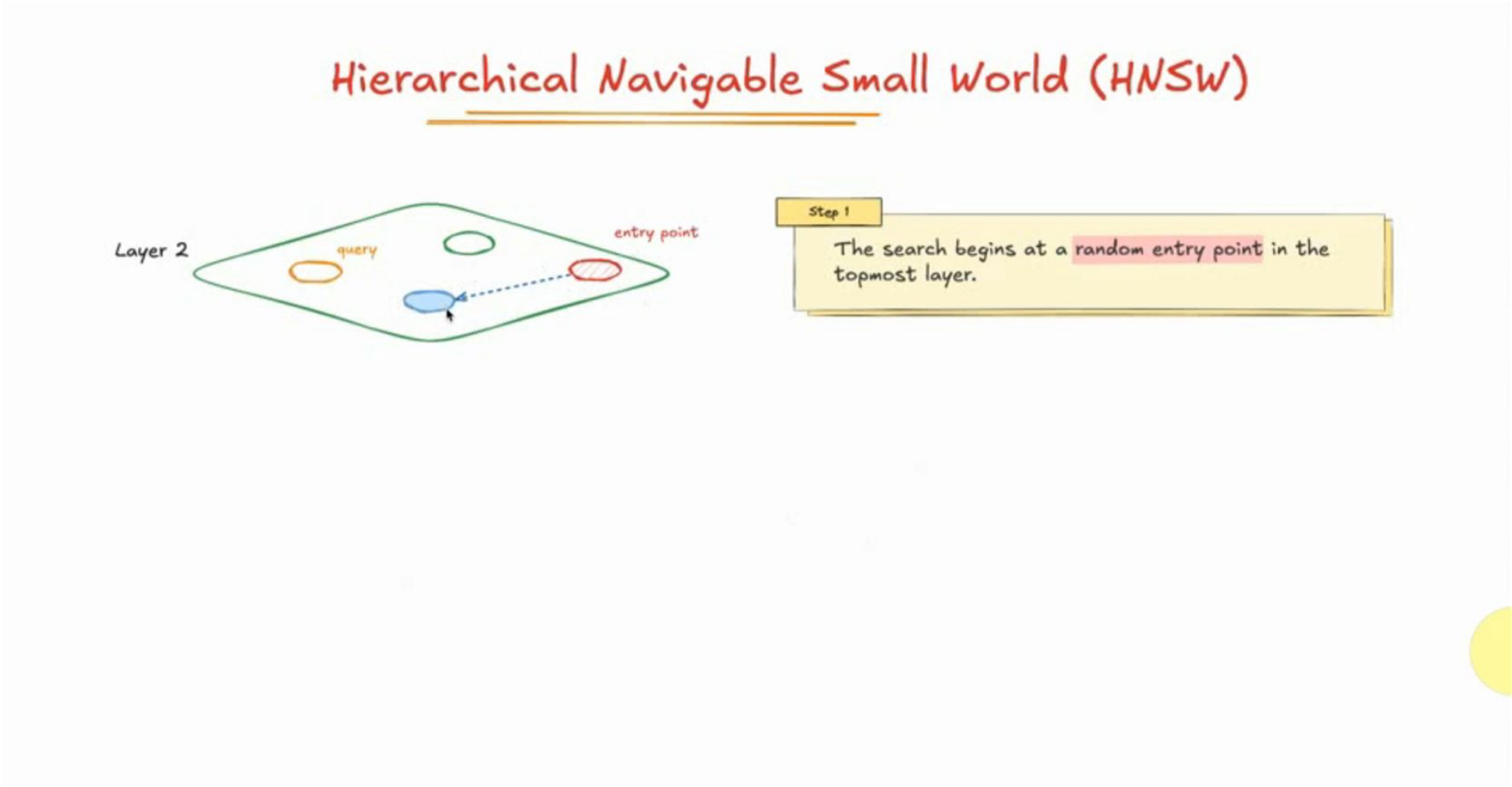
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



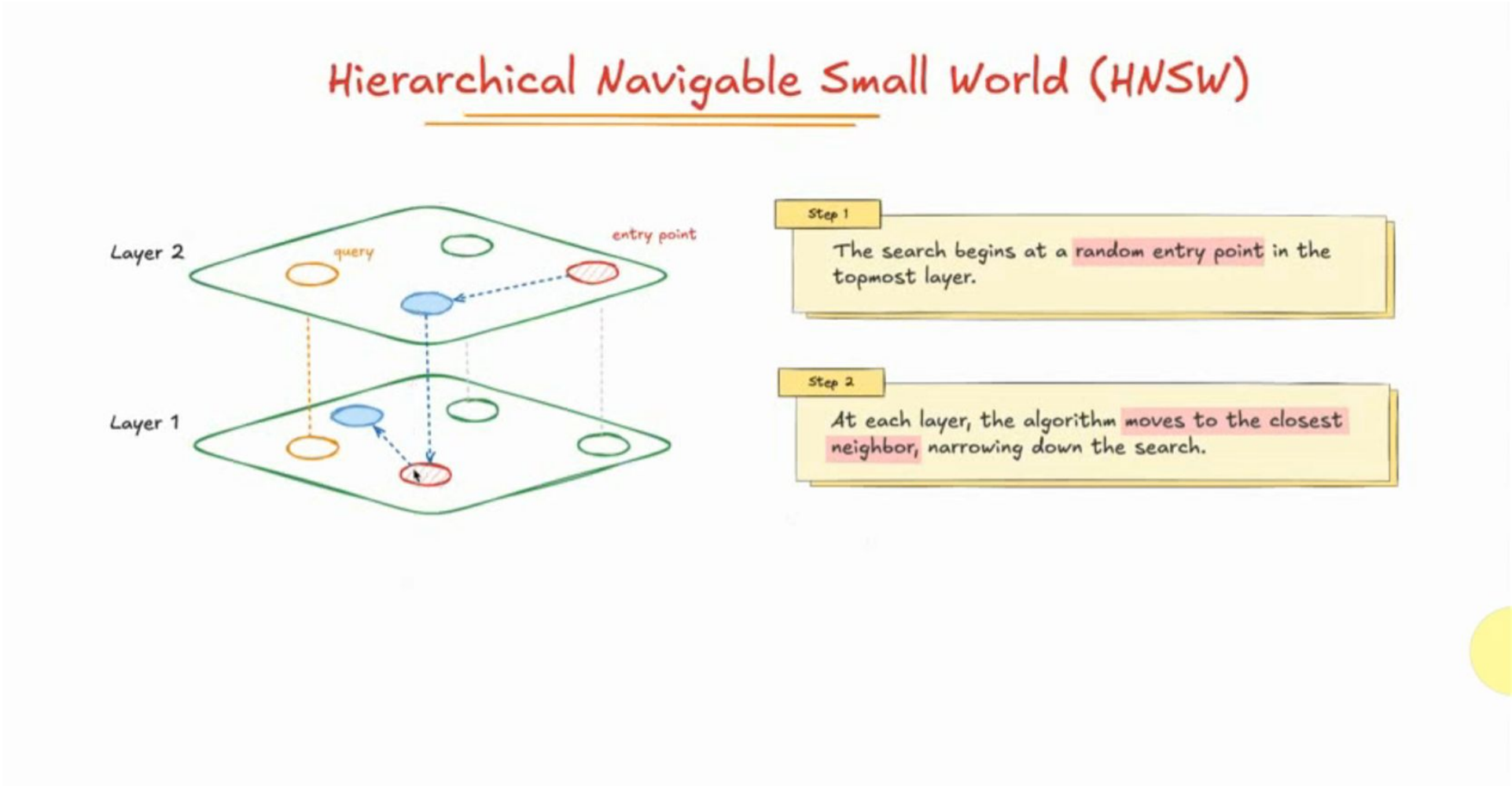
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



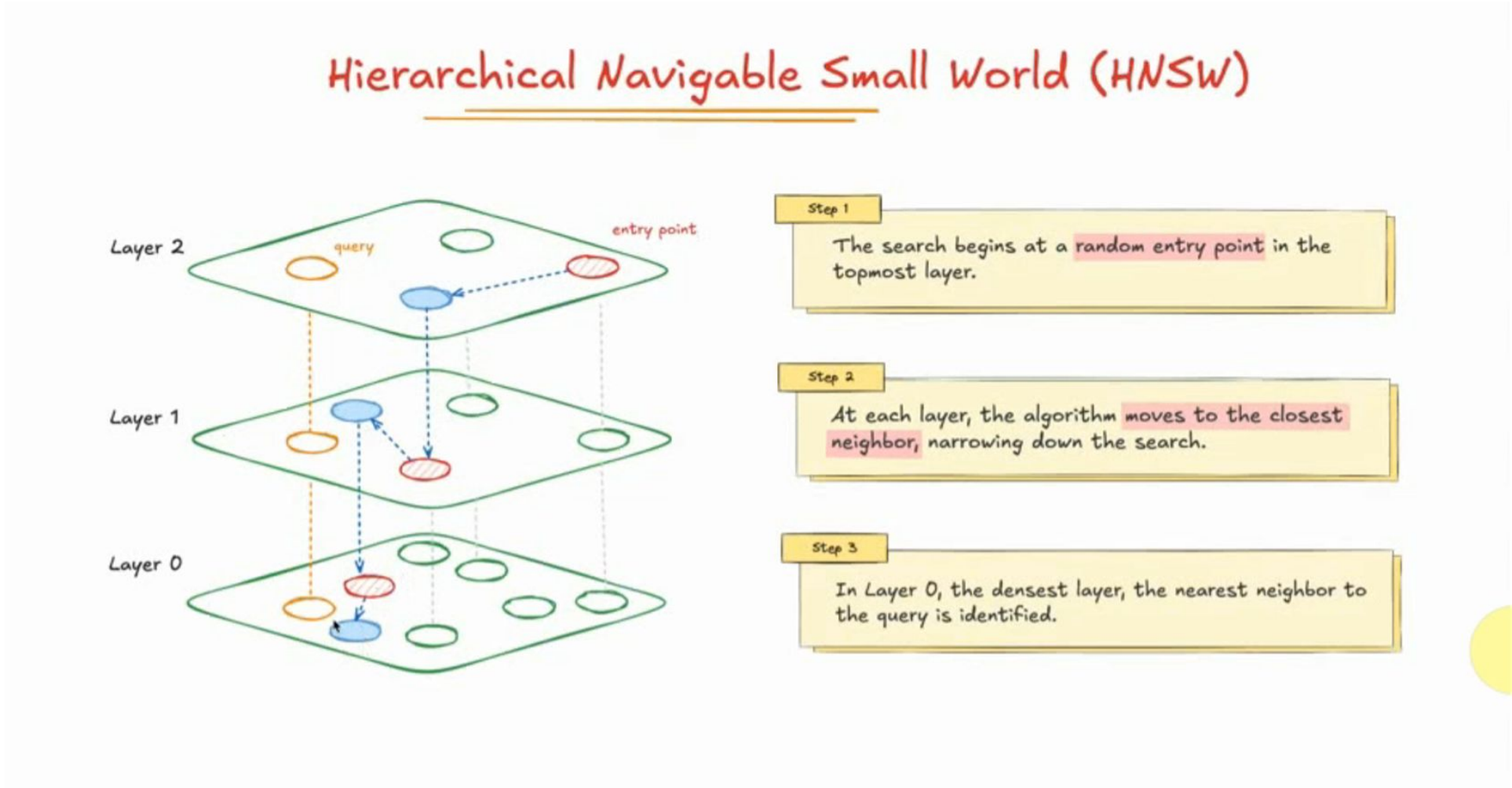
Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

Vector Space Search: Hierarchical Navigable Small Worlds



Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

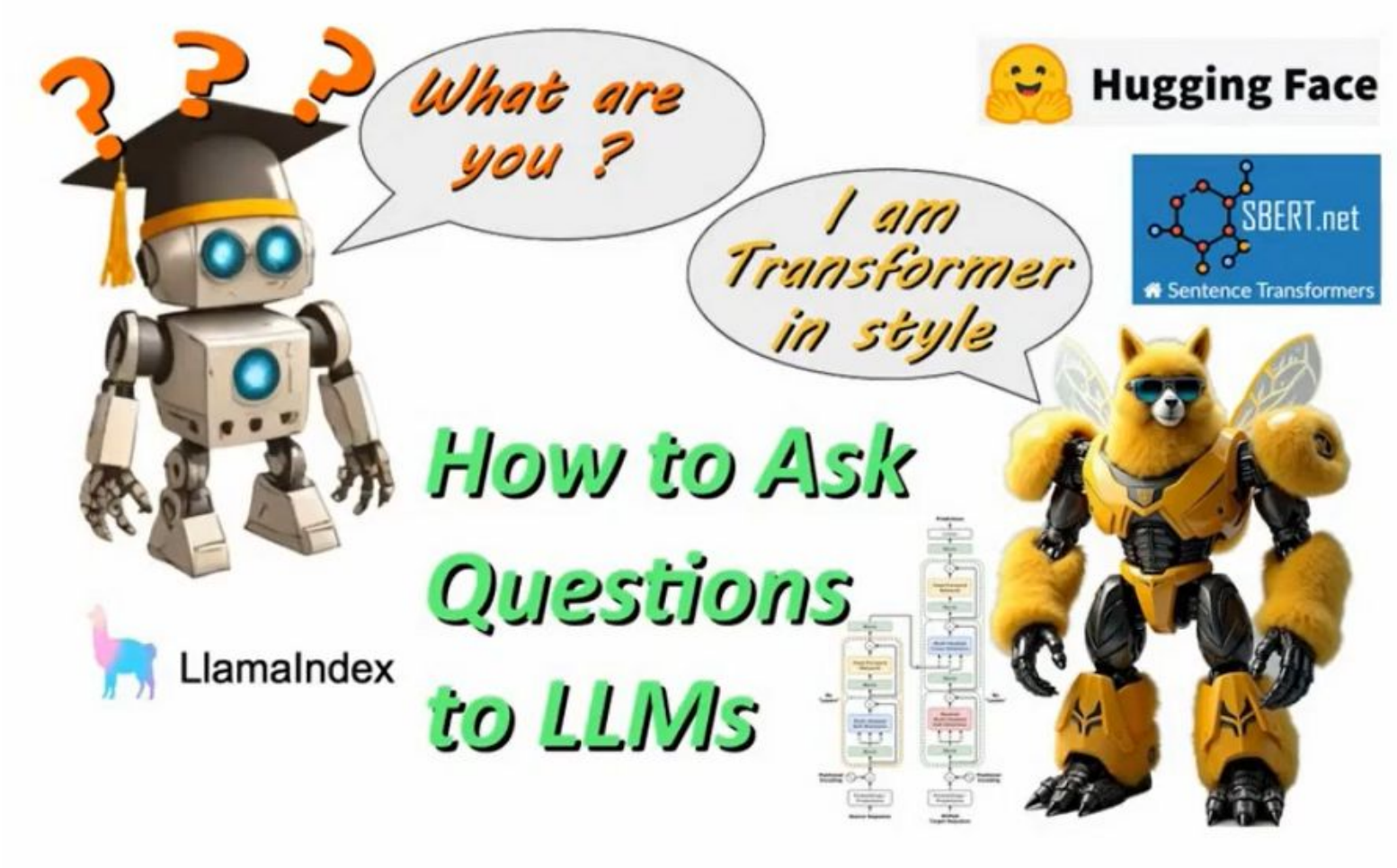
Vector Space Search: Hierarchical Navigable Small Worlds



Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF)
https://www.youtube.com/watch?v=chz74Mtd1AAutube.com/watch?v=chz74Mtd1AA

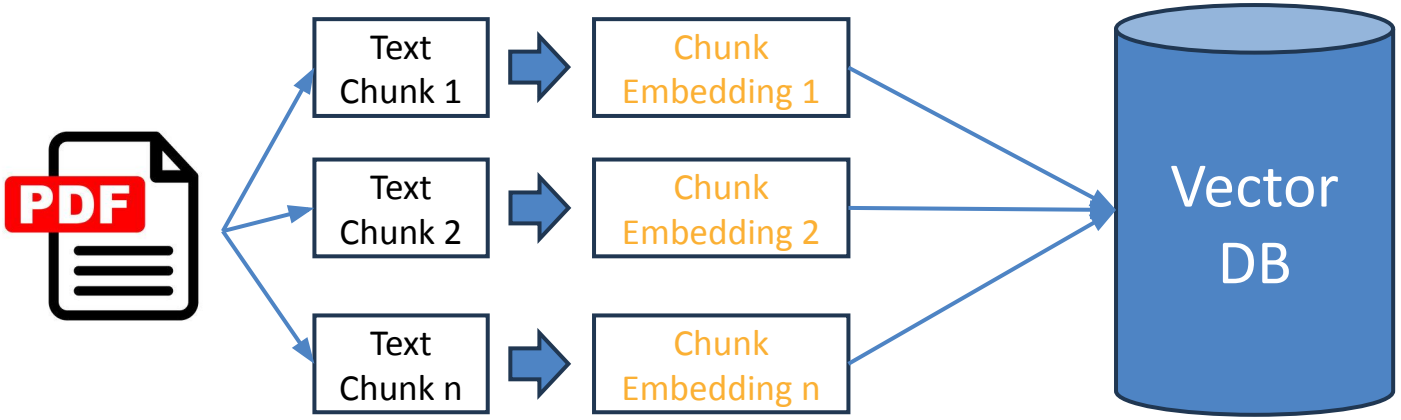
Document Chunking

Semantic Search



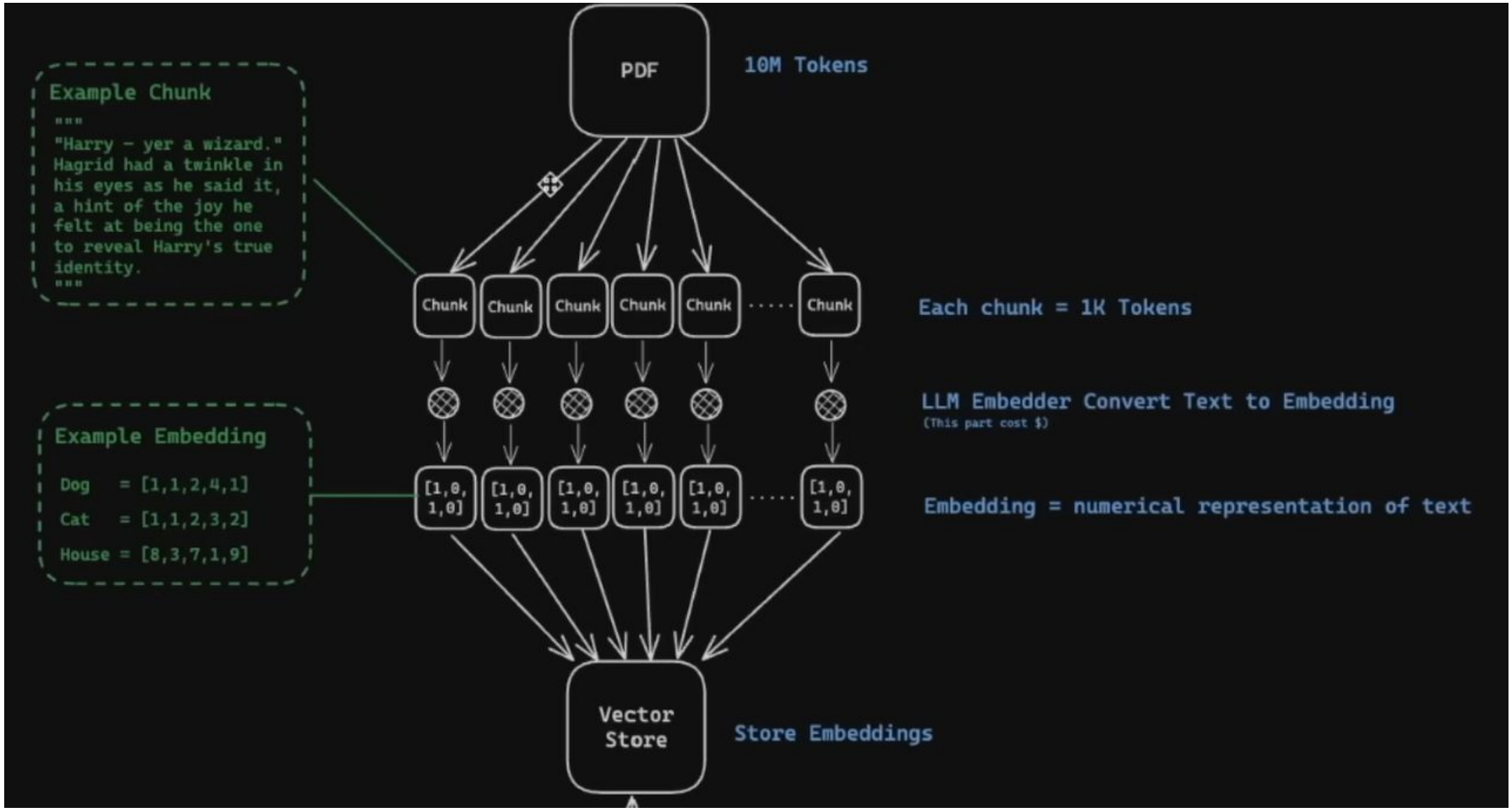
How to ask Questions to LLMs
Github:
https://github.com/enoten/huggingface_llms_apps/blob/main/ask_questions_about_research_papers%20to%20LLMs.ipynb
Youtube: https://www.youtube.com/watch?v=E4k1qGFemSE

Embeddings & Vector DB



Vector DB

Vector DB: Generic Stru

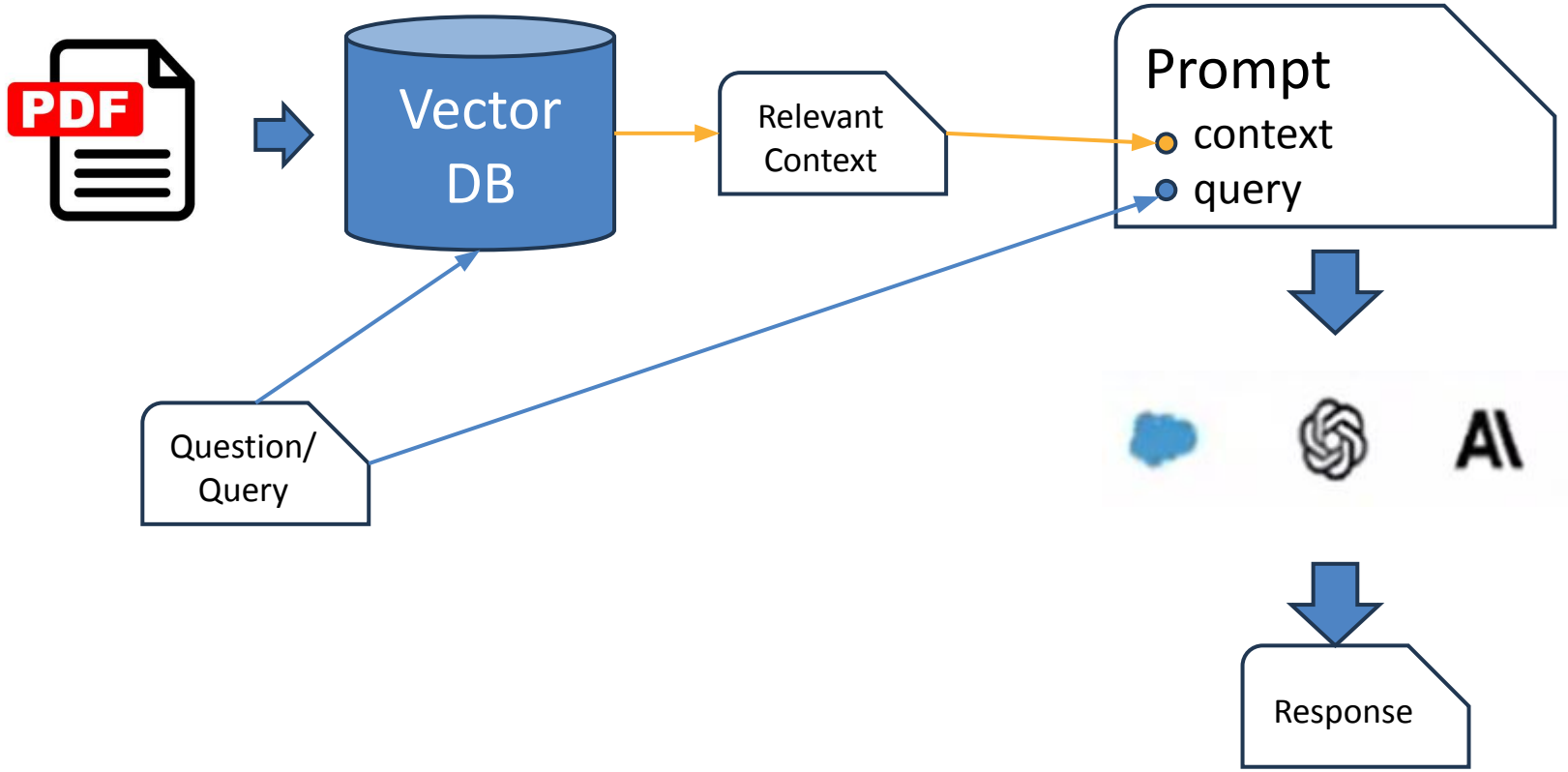


Vector DB: Examples

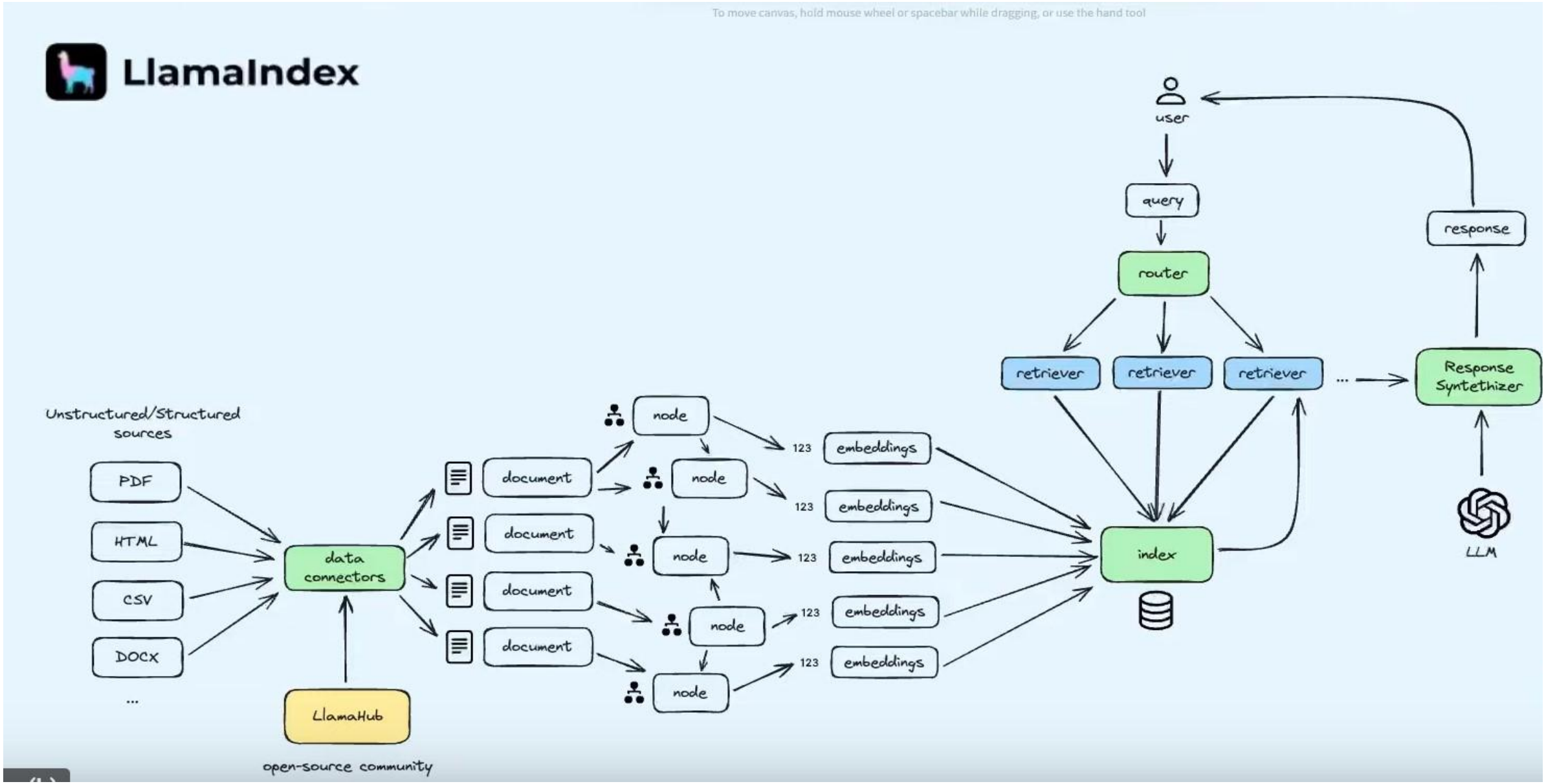
Database	Strengths	Use Cases	Type	Vendor
Pinecone	- Fully managed - High scalability - Real-time updates - Advanced filtering	- Production-grade RAG - Semantic & hybrid search - Personalization engines	Proprietary (SaaS, Cloud)	Pinecone, Inc.
Milvus	- High throughput - Designed for billion-scale vectors - GPU acceleration - Rich indexing options	- Image/video search - Large-scale AI pipelines - IoT sensor data retrieval	Open-source + Managed (Zilliz Cloud)	Zilliz
FAISS	- Extremely fast - Fine-tuned indexing control - Good for offline/local search	- Research and prototyping - On-device vector search - Embedding similarity testing	Library (C++/Python, Local)	Meta (Facebook AI)
Chroma DB	- Simple developer UX - Built-in embedding storage - Tight integration with LangChain - Lightweight	- Prototyping RAG apps - Local vector stores - Rapid LLM app development	Open-source (local-first)	Chroma (open community project)

Vector DB and RAG

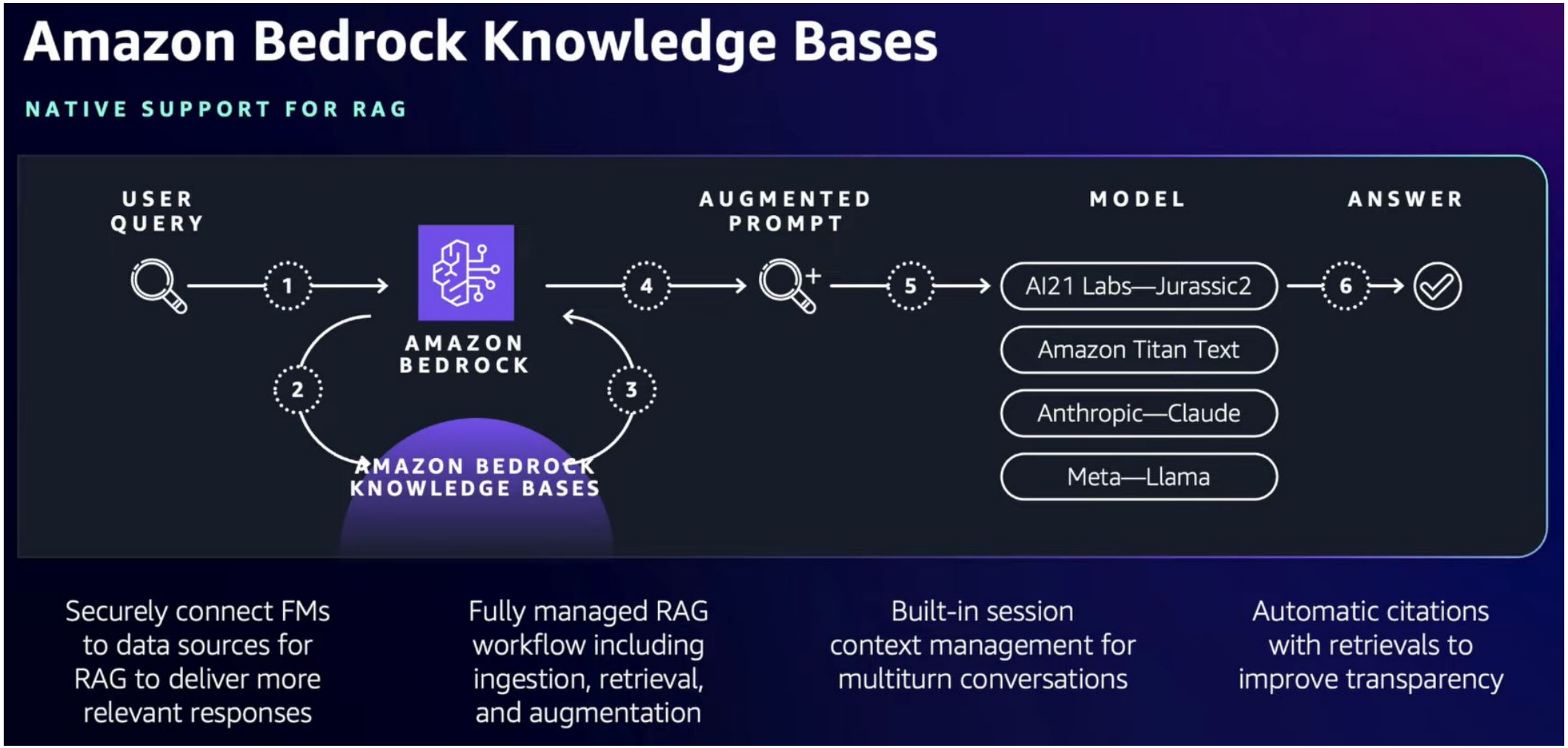
Retrieval Augmented Generation (RAG)



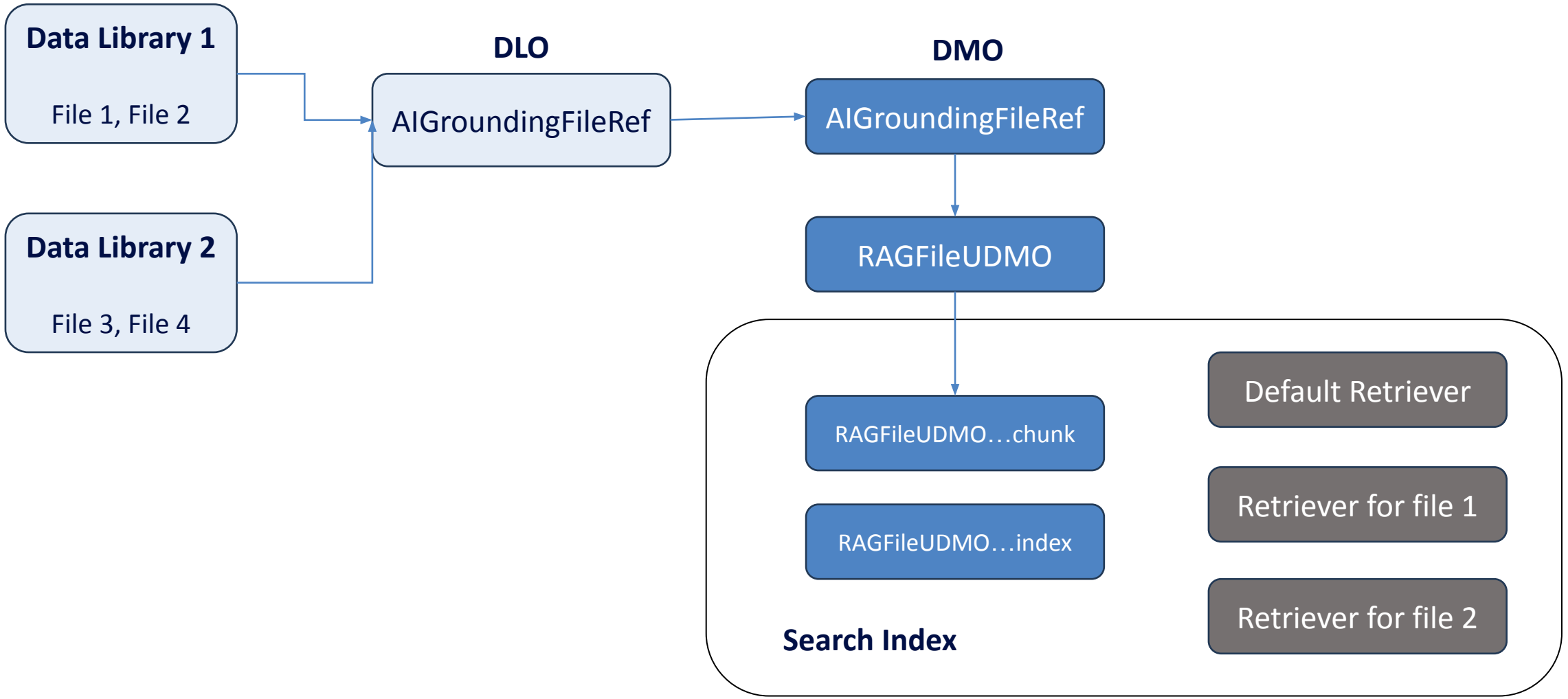
Retrieval Augmented Generation: LlamaIndex



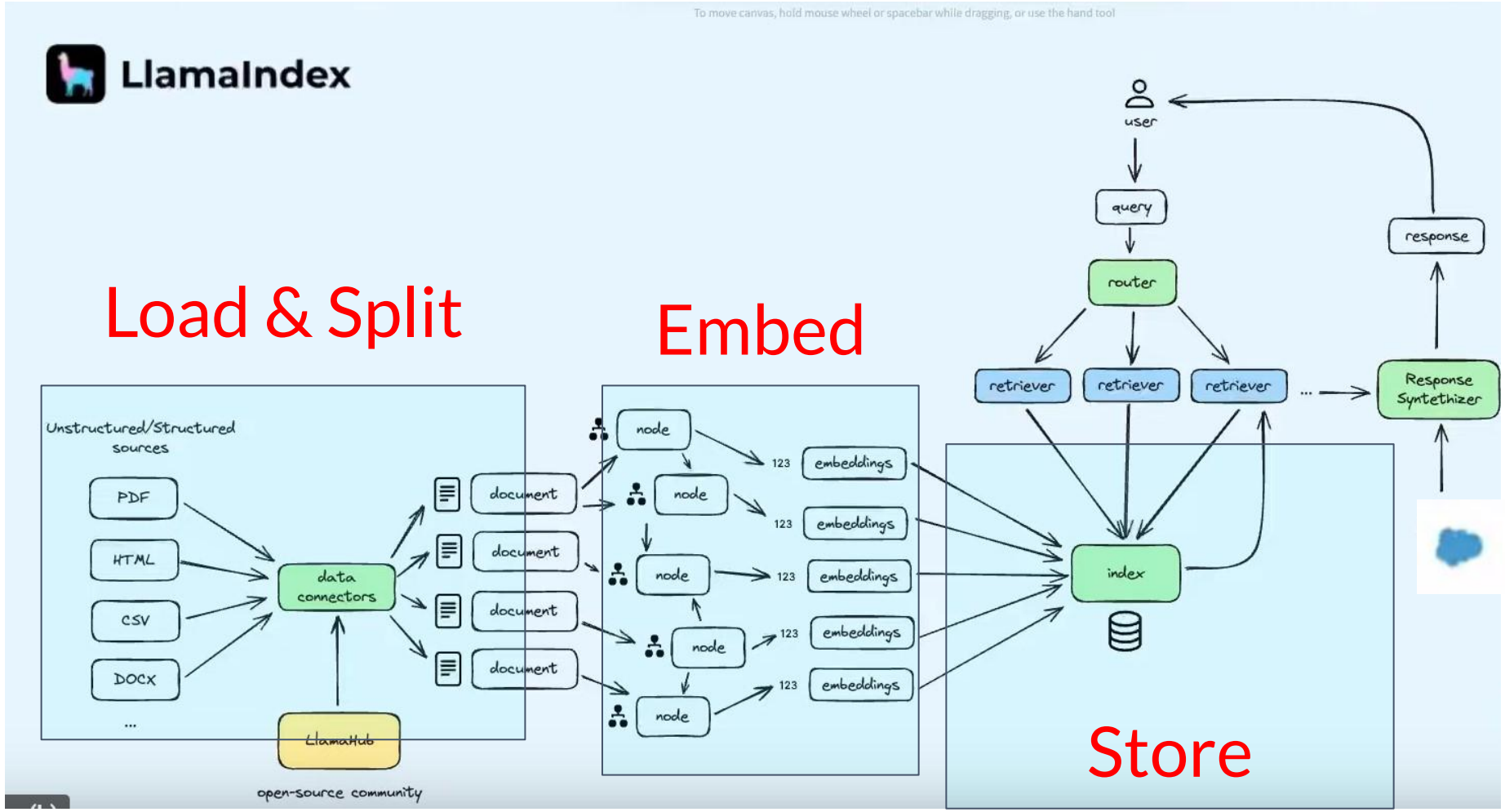
Retrieval Augmented Generation (RAG)



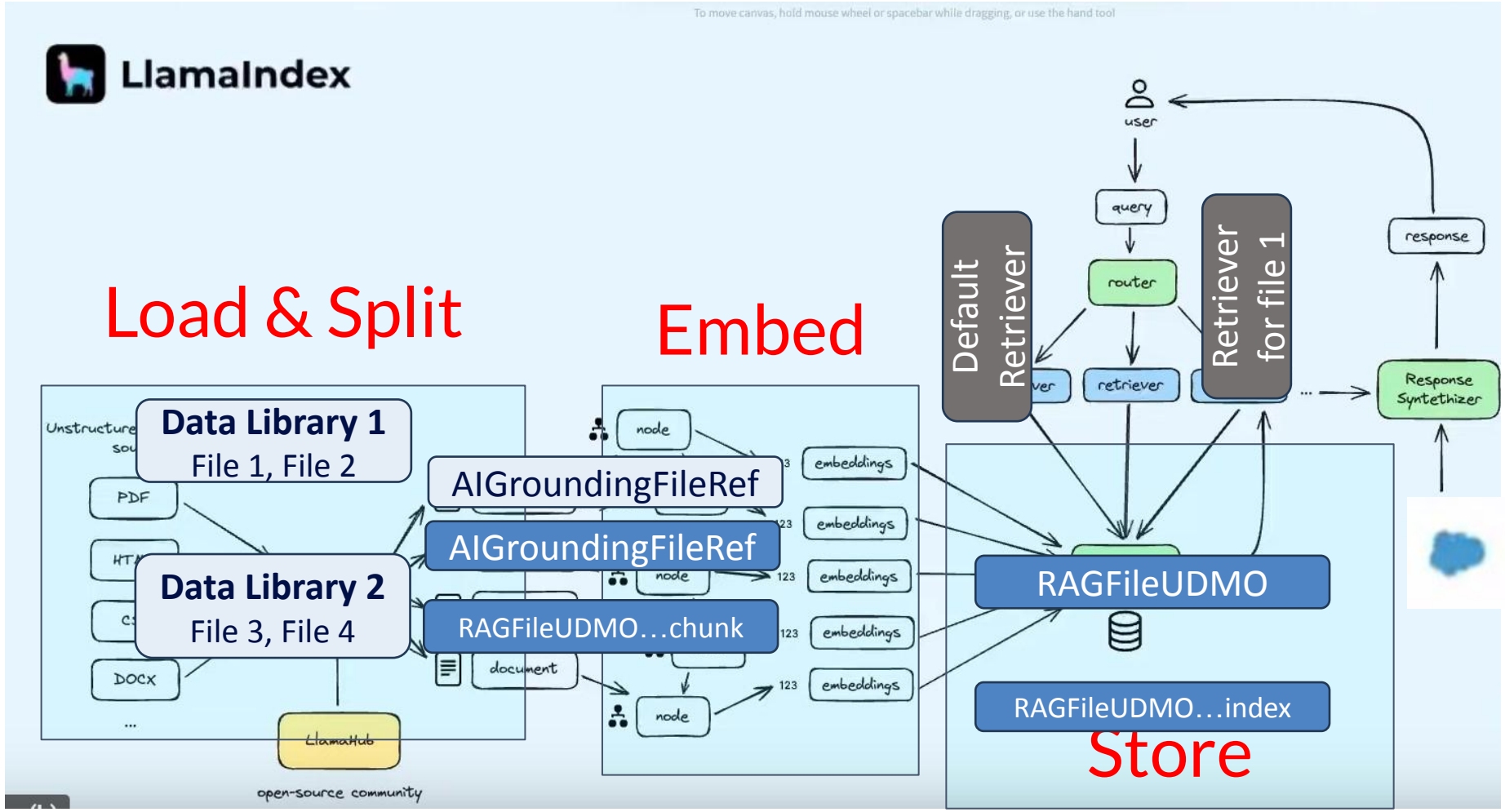
RAG in Agentforce Salesforce



Retrieval Augmented Generation (RAG)



Retrieval Augmented Generation



Useful Links

https://www.youtube.com/@phiai1618

Embedding & Support Ticket Classification

https://www.youtube.com/watch?v=NlJ1yS0F03M

https://github.com/enoten/support_ticket_analysis/blob/main/

How To Ask Questions to LLMs

https://www.youtube.com/watch?v=E4k1qGFemSE

https://github.com/enoten/huggingface_llms_apps/blob/main/