Esprits numeriques

Séance 11

Contexte, attention et modèles génératifs

Jonathan Simon

programme

- 1) Apprentissage non supervisé
- 2) Modèles génératifs et modèles discriminatifs
- 3) Vectorisations de mots (et de jetons) / plongement lexical : une revue
- 4) Le défi du context
- 5) Mécanismes attentionnels
- 6) Transformeurs
- 7) Capacités émergentes

Apprentissage non supervisé

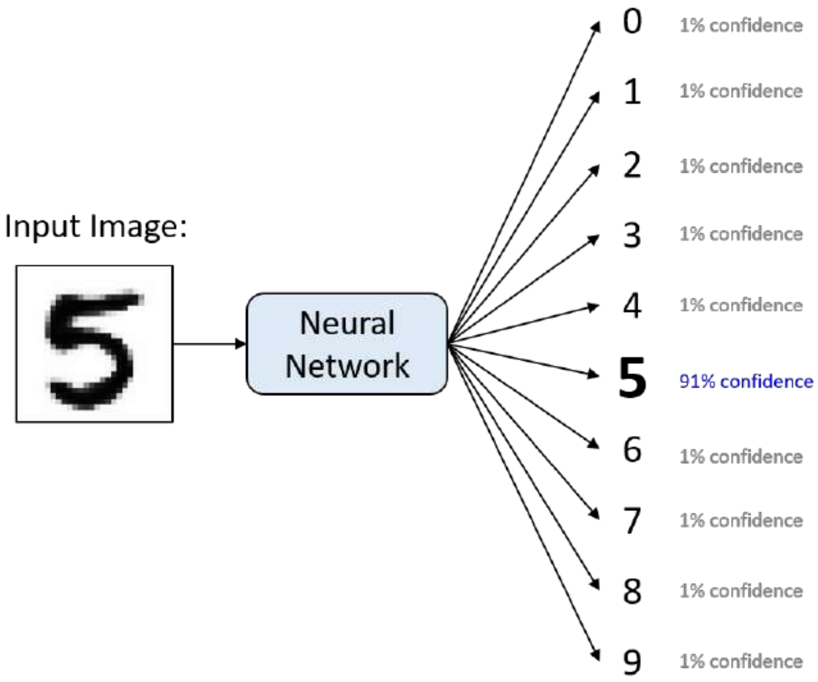
Apprentissage supervisé

- Nous avons principalement parlé des systèmes d'apprentissage supervisé (et de renforcement)
- Apprentissage supervisé : la fonction de coût est définie par les données étiquetées

-- c'est-à-dire que pour un système séparant les photos de chats des photos de chiens, si l'étiquette pour le chat est {0,1} et l'étiquette pour le chien est {1,0}, alors un échantillon d'apprentissage sera, par exemple, (cat-pic-7, {0,1})

Apprentissage supervisé

- *Le système apprend ensuite à estimer la probabilité conditionelle qu'un objet soit un chat (0,1), compte tenu de ses autres caractéristiques : P(chat | chat-pic-7)*
- Cette méthode nécessite que l'on dispose de données étiquetées : un goulot d'étranglement important.



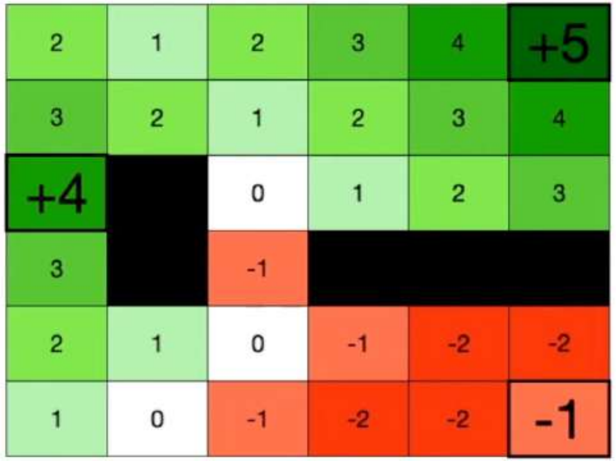
Input Image:
Neural Network
0
1% confidence
1
1% confidence
2
1% confidence
3
1% confidence
4
1% confidence
5
91% confidence
6
1% confidence
7
1% confidence
8
1% confidence
9
1% confidence

Apprentissage par renforcement

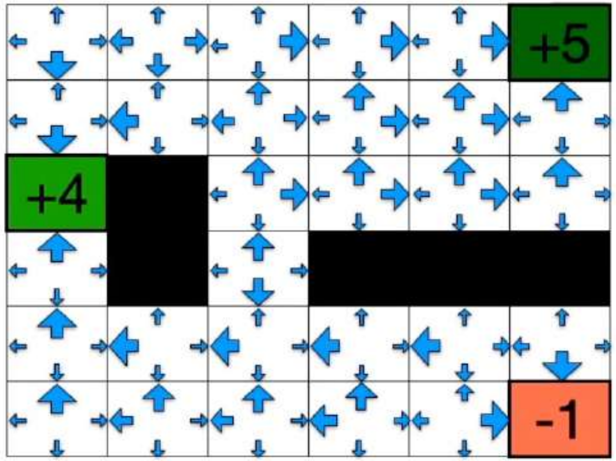
- Dans l'apprentissage par renforcement, on a besoin de moins de données, le système «crée» ses propres données à partir des indices épars de «recompenses» placés dans l'environnement : un échantillon d'entraînement sera un chemin (plus ou moins aléatoire) qu'il emprunte dans son environnement jusqu'à un état final, et la valeur qu'il attribue à ce chemin par la suite.

Two neural networks

Value neural network



Policy neural network



Luis Serrano

Apprentissage non supervisé

- Mais ces deux méthodes sont limitées : elles correspondent à l'apprentissage humain par instruction ou par essais et erreurs, mais pas à l'apprentissage par interaction constante et continue avec le monde

Y. LeCun

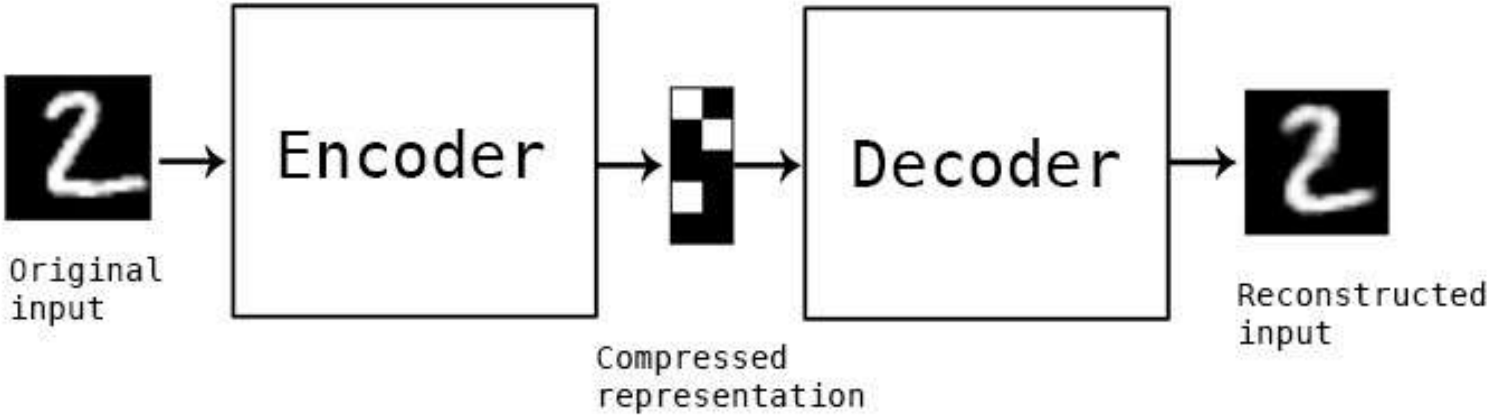
How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Yann Lecun

Apprentissage non supervisé

- L'apprentissage non supervisé consiste à apprendre plus directement à partir d'un environnement (une distribution d'exemples)
- Un exemple représentatif : les auto-encodeurs


Encoder
Decoder
Original input
Compressed representation
Reconstructed input

Apprentissage non supervisé

- Ici, l'objectif est d'entraîner la couche cachée de manière à ce qu'elle "corresponde" et soit capable de reconstruire l'entrée : en fait, l'entrée sert de sa propre étiquette

Apprentissage non supervisé

- De nombreuses variations de cette idée où la tâche consiste à reconstruire ou à deviner un aspect des données originales (et à être capable de le faire pour toutes les données de l'ensemble de formation).

- En général, cela revient à apprendre la distribution des données elles-mêmes, plutôt que d'apprendre seulement la probabilité conditionnelle qu'une certaine étiquette s'applique étant donné les caractéristiques d'un échantillon

Apprentissage non supervisé

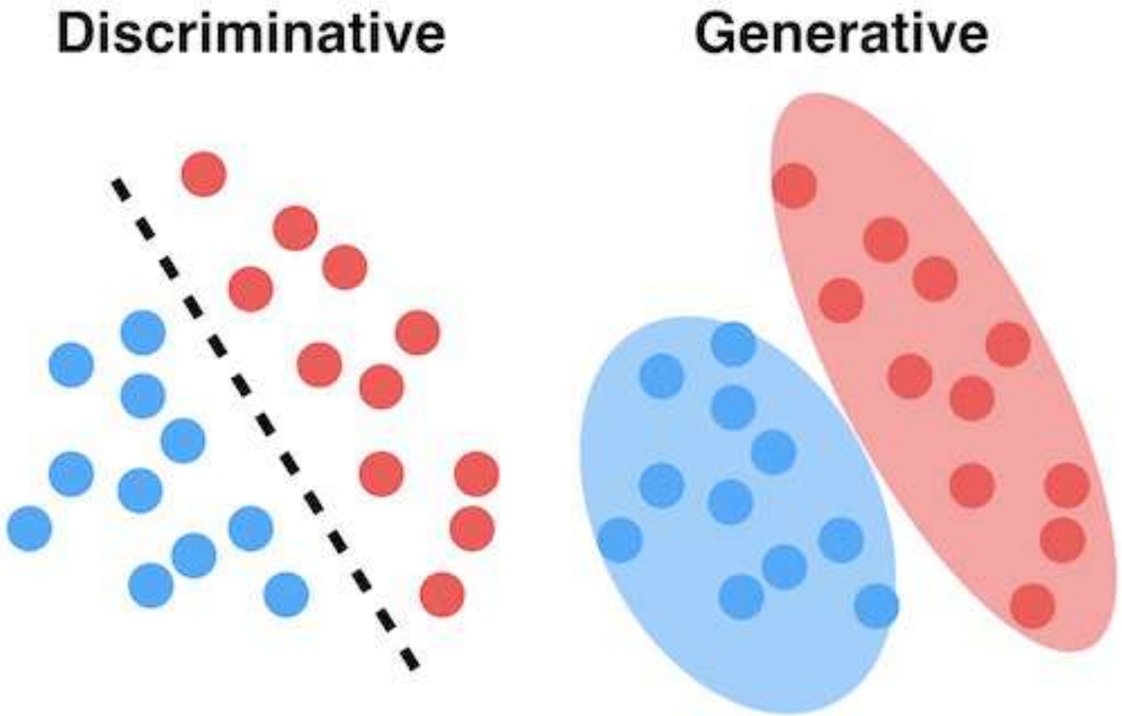
- Pour les grands modèles de langage (LLM, transformeurs), l'objectif est de deviner le mot suivant (ce qu'il peut vérifier lui-même, puisqu'il dispose de l'ensemble du texte)

Modèles génératifs et modèles discriminatifs

Génératif vs discriminatif

- Les **modèles génératifs** capturent la probabilité conjointe $p(X, Y)$, ou simplement $p(X)$ s'il n'y a pas d'étiquettes.

- Les **modèles discriminatifs** capturent la probabilité conditionnelle $p(Y \mid X)$.


Discriminative
Generative

Génératif vs discriminatif

- Les modèles les plus importants d'aujourd'hui (par exemple, les modèles basés sur des transformateurs comme ChatGPT ou DALLE) sont non supervisés (plutôt que supervisés, au moins pour l'étape de «pré-entraînement»), et génératifs (plutôt que discriminatifs - encore une fois, au moins pour cette étape de «pré-entraînement»).

Vectorisation de mots (et de jetons) / plongement lexical : une revue

Plongement lexical (Embeddings)

- Le vectorisation d'un mot (ou d'un jeton, en fait un phonème) est un vecteur qui exprime, sous forme compressée, des informations sur les statistiques d'apparition de ce mot ou de ce jeton dans les données de manière plus générale

Plongement lexical (Embeddings)

- Le résultat fascinant est qu'en utilisant cette méthode (en supposant que tu disposes d'un ensemble de textes suffisamment riche pour t'entraîner), tu arrives à des vecteurs qui capturent beaucoup de nos intuitions sémantiques sur les similitudes (sémantiques) entre les mots

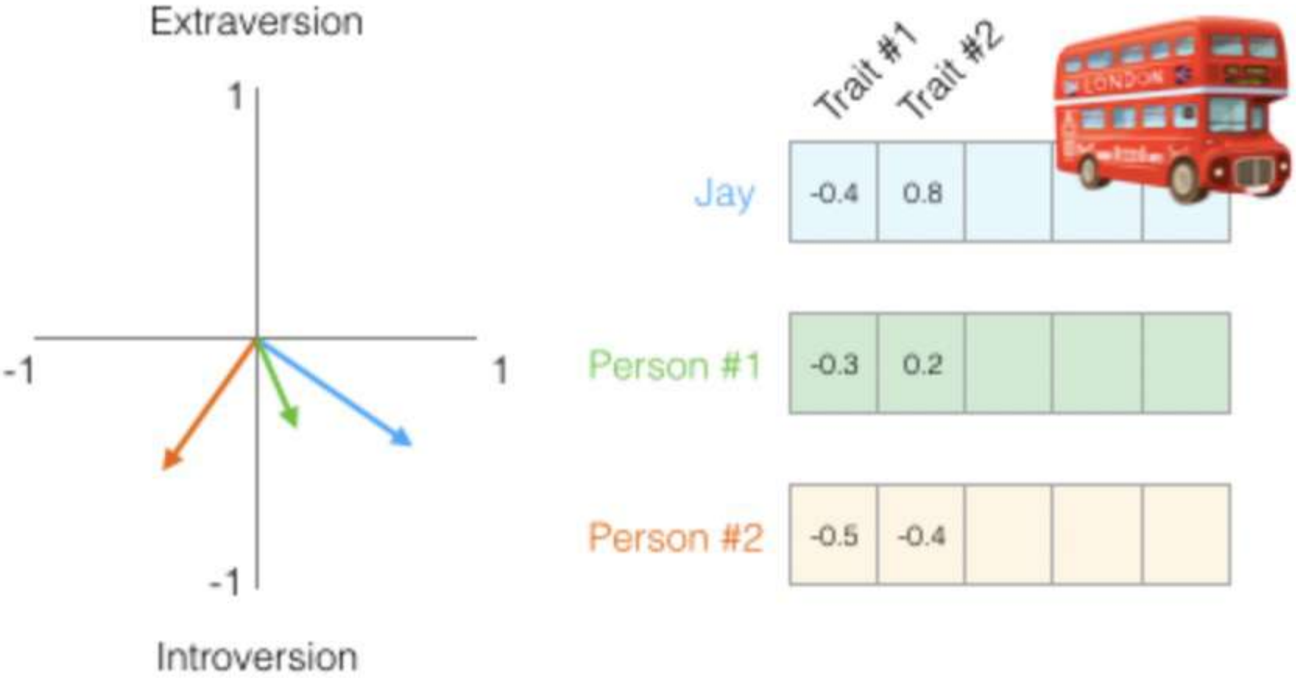
Personality Embeddings: What are you like?

> *"I give you the desert chameleon, whose ability to blend itself into the background tells you all you need to know about the roots of ecology and the foundations of a personal identity" ~Children of Dune*

On a scale of 0 to 100, how introverted/extraverted are you (where 0 is the most introverted, and 100 is the most extraverted)? Have you ever taken a personality test like MBTI – or even better, the Big Five Personality Traits test? If you haven't, these are tests that ask you a list of questions, then score you on a number of axes, introversion/extraversion being one of them.

Openness to experience	79 out of 100
Agreeableness	75 out of 100
Conscientiousness	42 out of 100
Negative emotionality	50 out of 100
Extraversion	58 out of 100

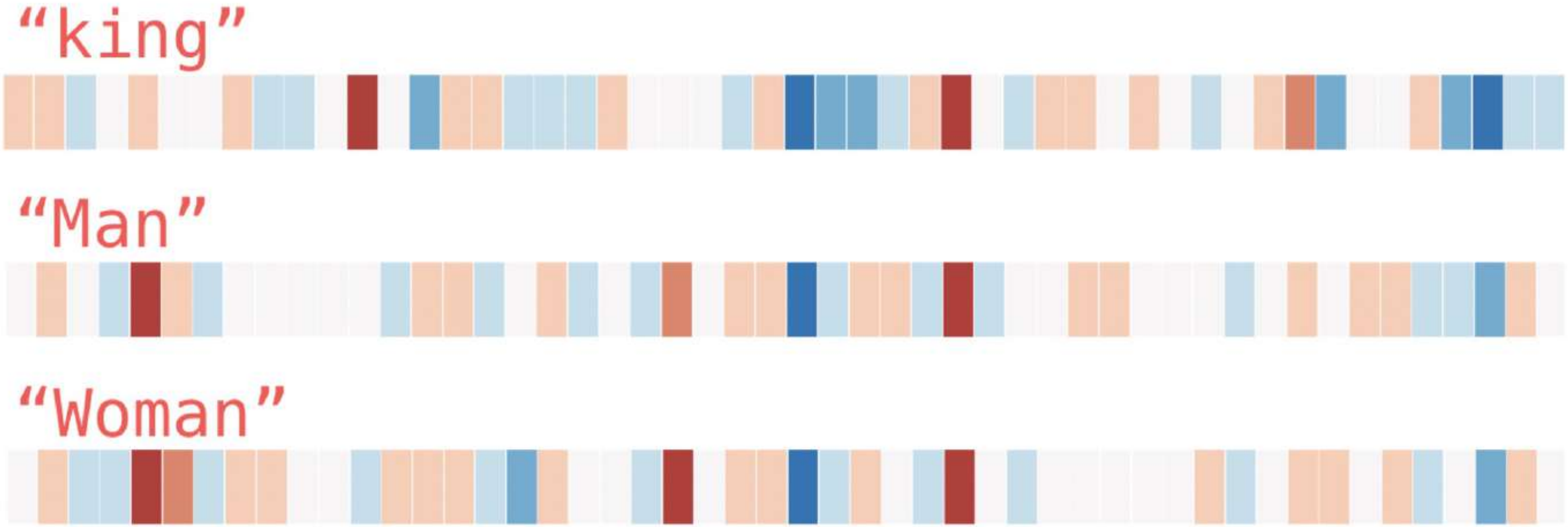
Jay Alammar


Extraversion
1
-1
1
-1
Introversion
Trait #1
Trait #2
Jay
-0.4
0.8
Person #1
-0.3
0.2
Person #2
-0.5
-0.4

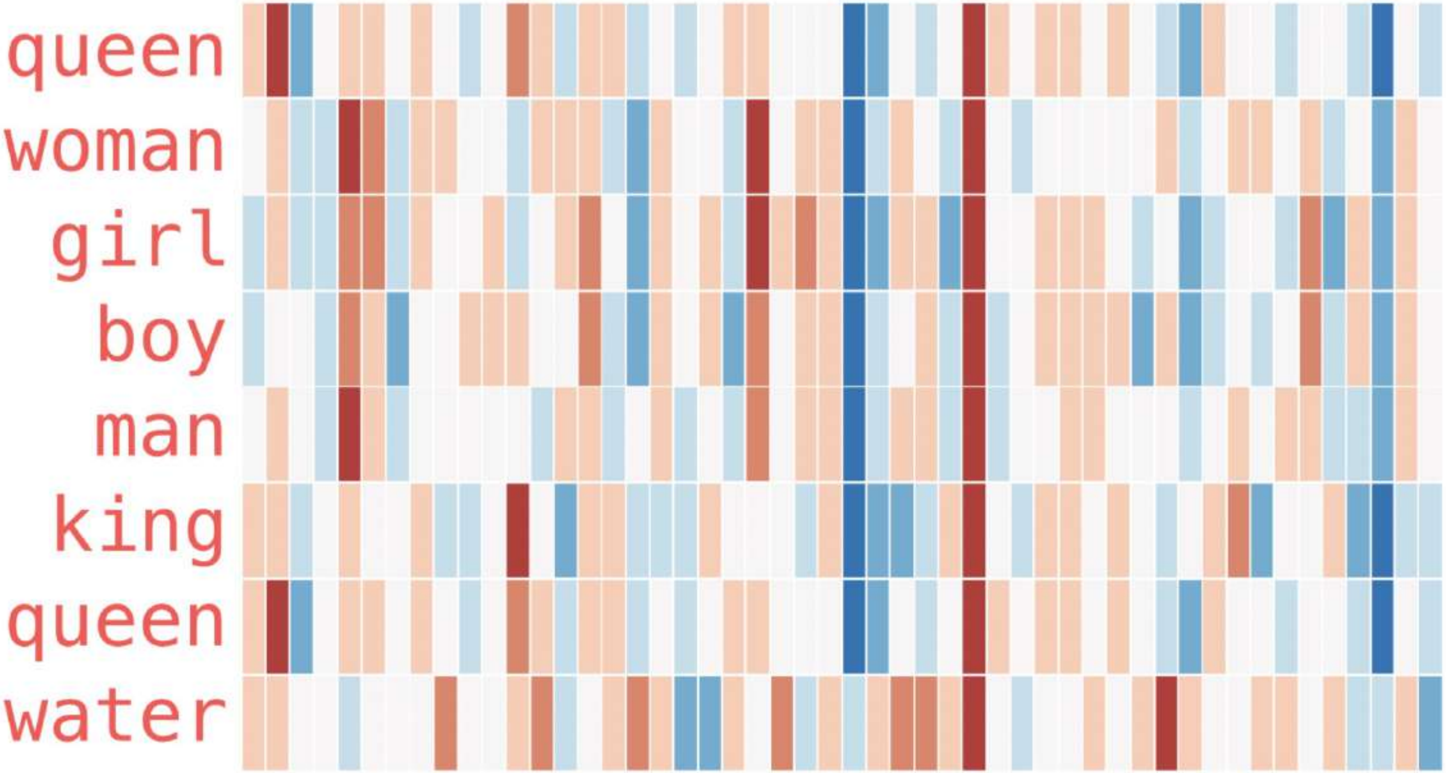
Jay Alammar

	Trait #1	Trait #2	Trait #3	Trait #4	Trait #5
Jay	-0.4	0.8	0.5	-0.2	0.3
Person #1	-0.3	0.2	0.3	-0.4	0.9
Person #2	-0.5	-0.4	-0.2	0.7	-0.1

Jay Alammar

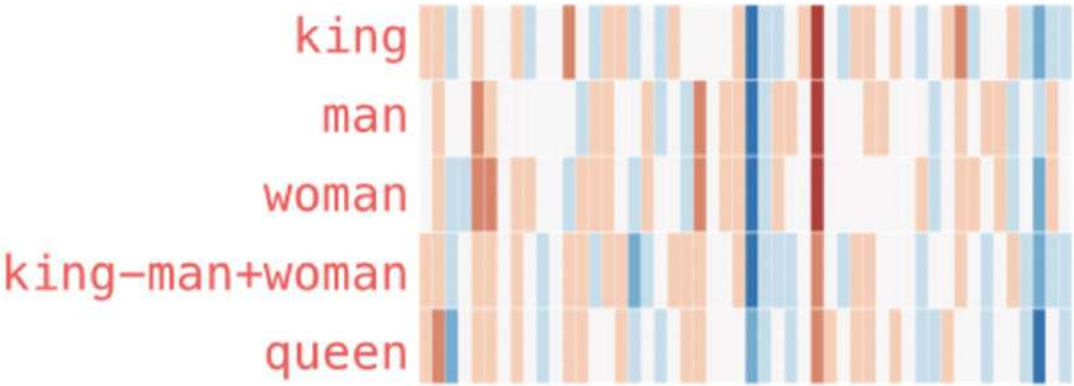


Jay Alammar



Jay Alammar

king − man + woman ~= queen

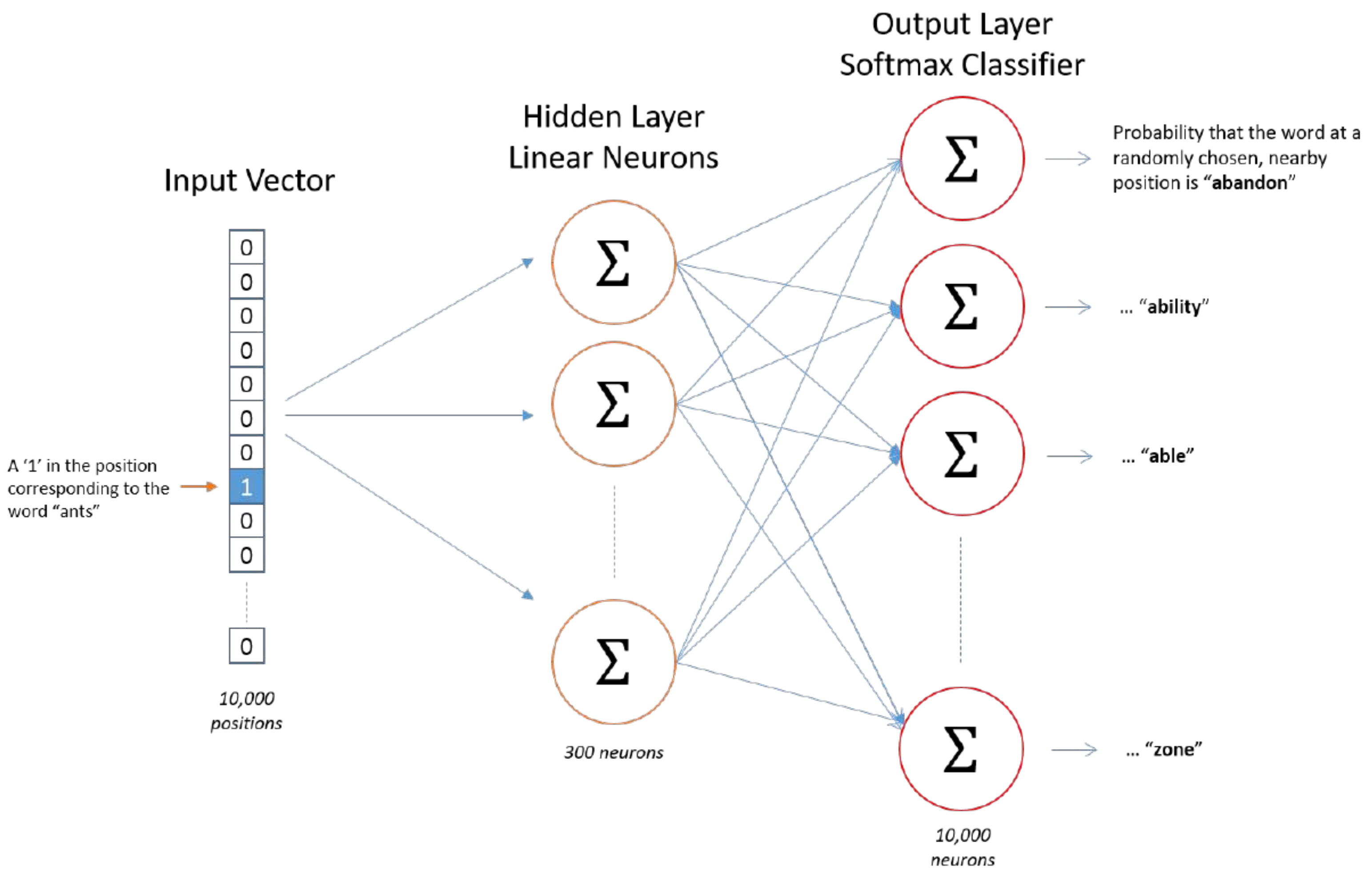


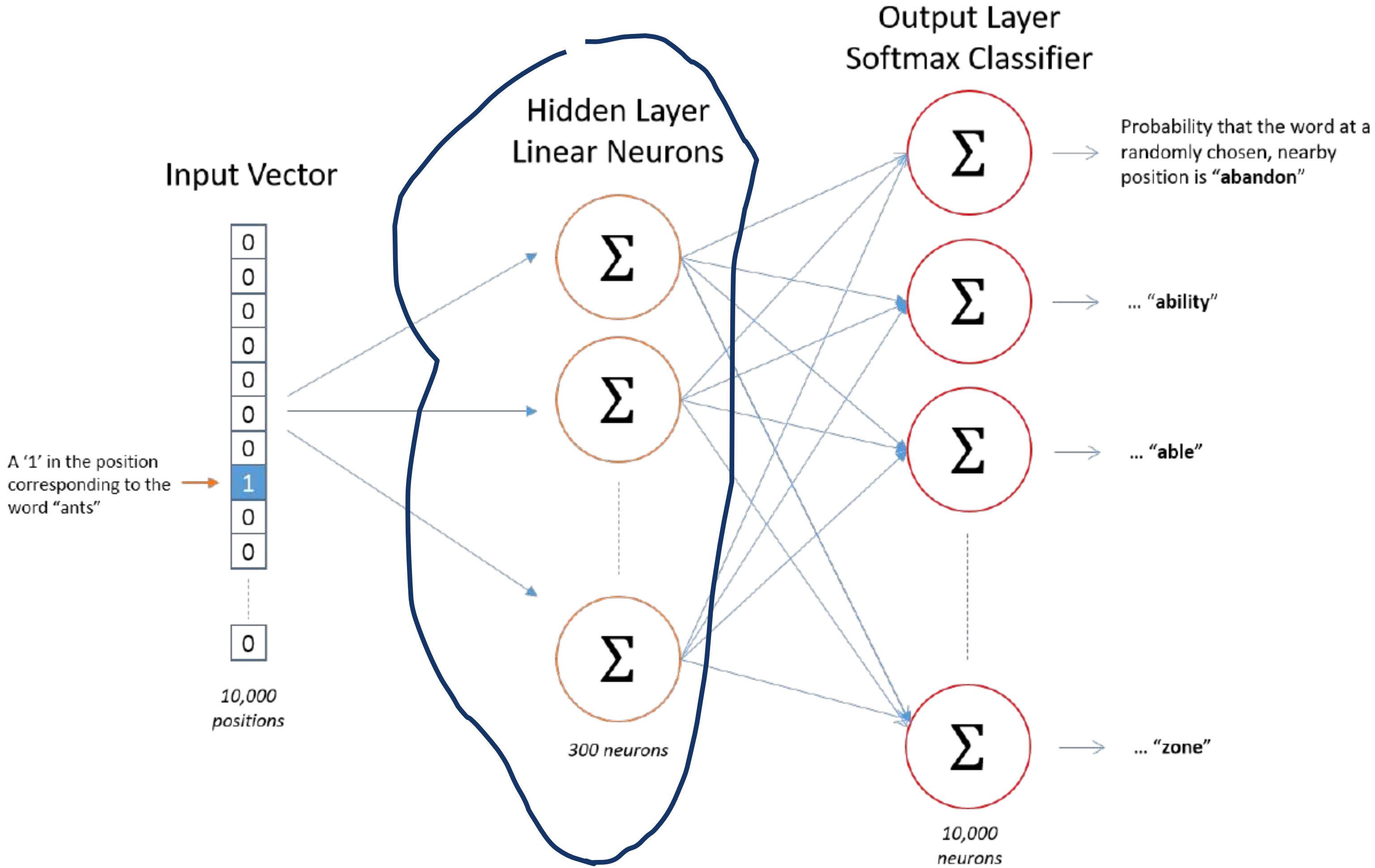
The resulting vector from "king-man+woman" doesn't exactly equal "queen", but "queen" is the closest word to it from the 400,000 word embeddings we have in this collection.

Jay Alammar

Plongement lexical (Embeddings)

- Comment les déduire ?
- En utilisant un autre réseau neuronal génératif :


Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
300 neurons
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "**abandon**"
... "**ability**"
... "**able**"
... "**zone**"



Input Vector
0
0
0
0
0
0
0
1
0
0
0
10,000 positions
A '1' in the position corresponding to the word "ants"
Hidden Layer Linear Neurons
300 neurons
Output Layer Softmax Classifier
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
10,000 neurons

Plongement lexical (Embedding)

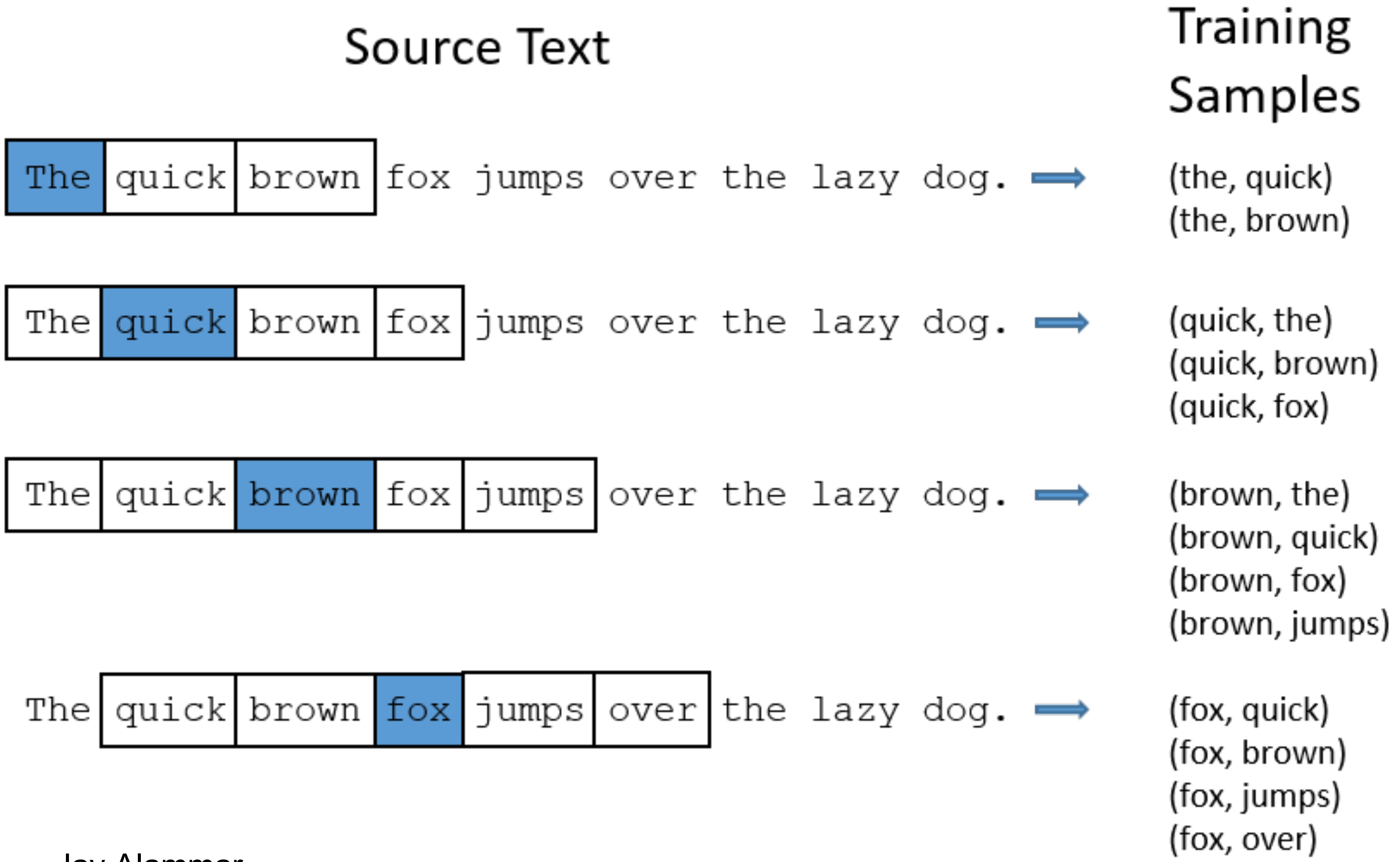
Entrée : un mot (représenté comme un vecteur "one-hot", essentiellement un index), [0,0,0,0,1,0,0...] si c'est le 5ieme mot

Sortie: une probabilité, pour tous les autres mots de l'index, de l'occurrence d'un mot à proximité.

Plongement lexical

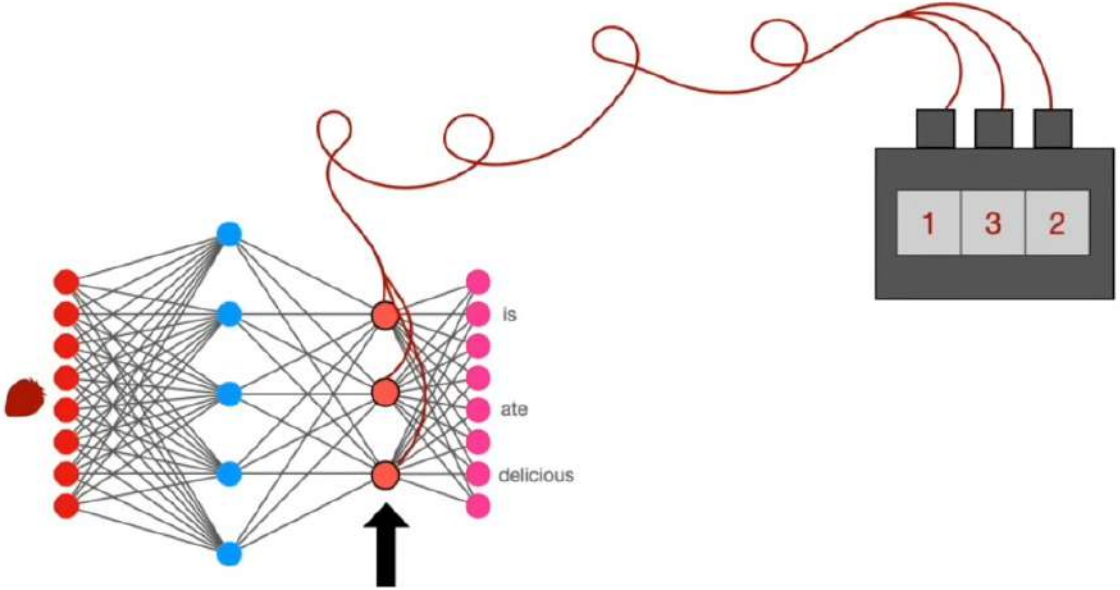
Données d'entraînement pour la fonction de coût :

« contexts »: séquences de texte courtes, par exemple de 2 à 5 mots


Source Text
Training Samples
The quick brown fox jumps over the lazy dog.
(the, quick)
(the, brown)
The quick brown fox jumps over the lazy dog.
(quick, the)
(quick, brown)
(quick, fox)
The quick brown fox jumps over the lazy dog.
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
The quick brown fox jumps over the lazy dog.
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

Jay Alammar

Thought experiment



Word	Numbers		
Strawberry	1	3	2
Apple	1.1	2.9	2.2

Luis Serrano

Le défi du context

Le défi du context

- Comme tout linguiste te le dira, le langage est ambigu et dépend du contexte
- Comment un ordinateur pourrait-il saisir les subtilités du contexte (cf. l'article de Landgrebe et Smith) ?

J'ai mangé une **baguette** avec du fromage.

La fée a jeté un sort avec sa **baguette**.

Baguette

Les musiciens suivent la **baguette** du chef d'orchestre.

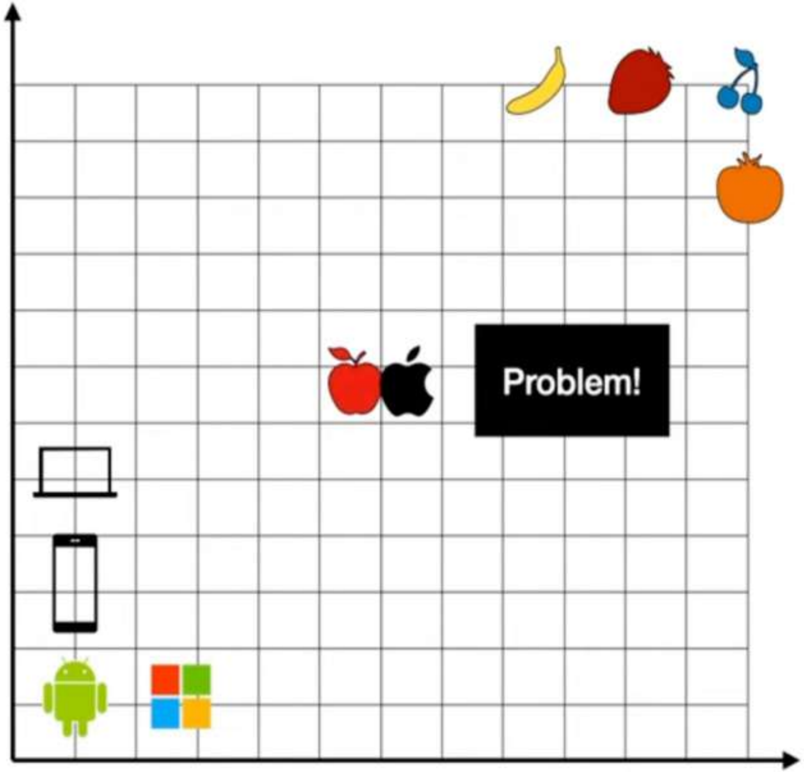
Dans la forêt, le Petit Chaperon Rouge a croisé le loup.

Au Carnaval, j'ai mis un loup pour ne pas qu'on me reconnaisse.

Loup

Le loup nage dans la mer.

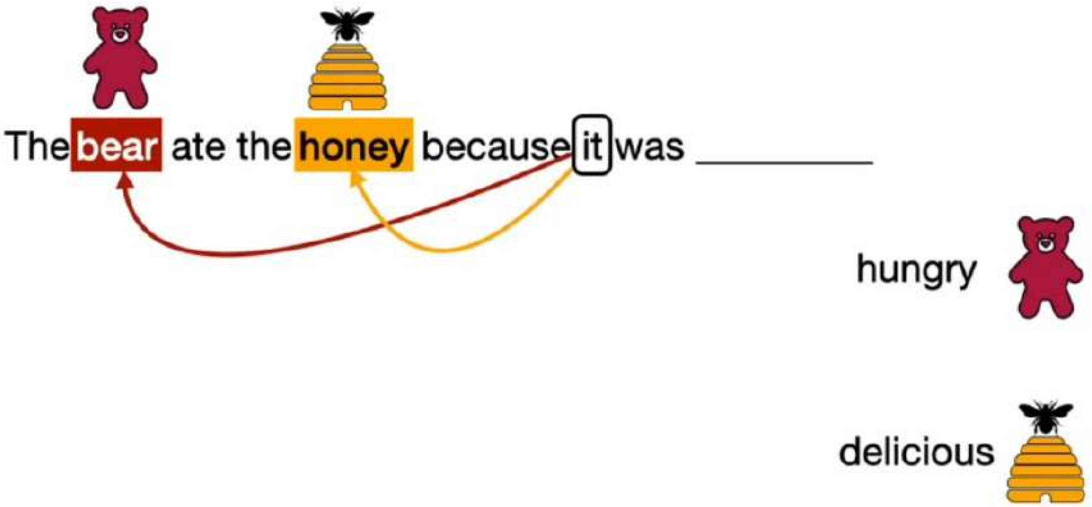
Embeddings Quiz 2



Top right or bottom left?

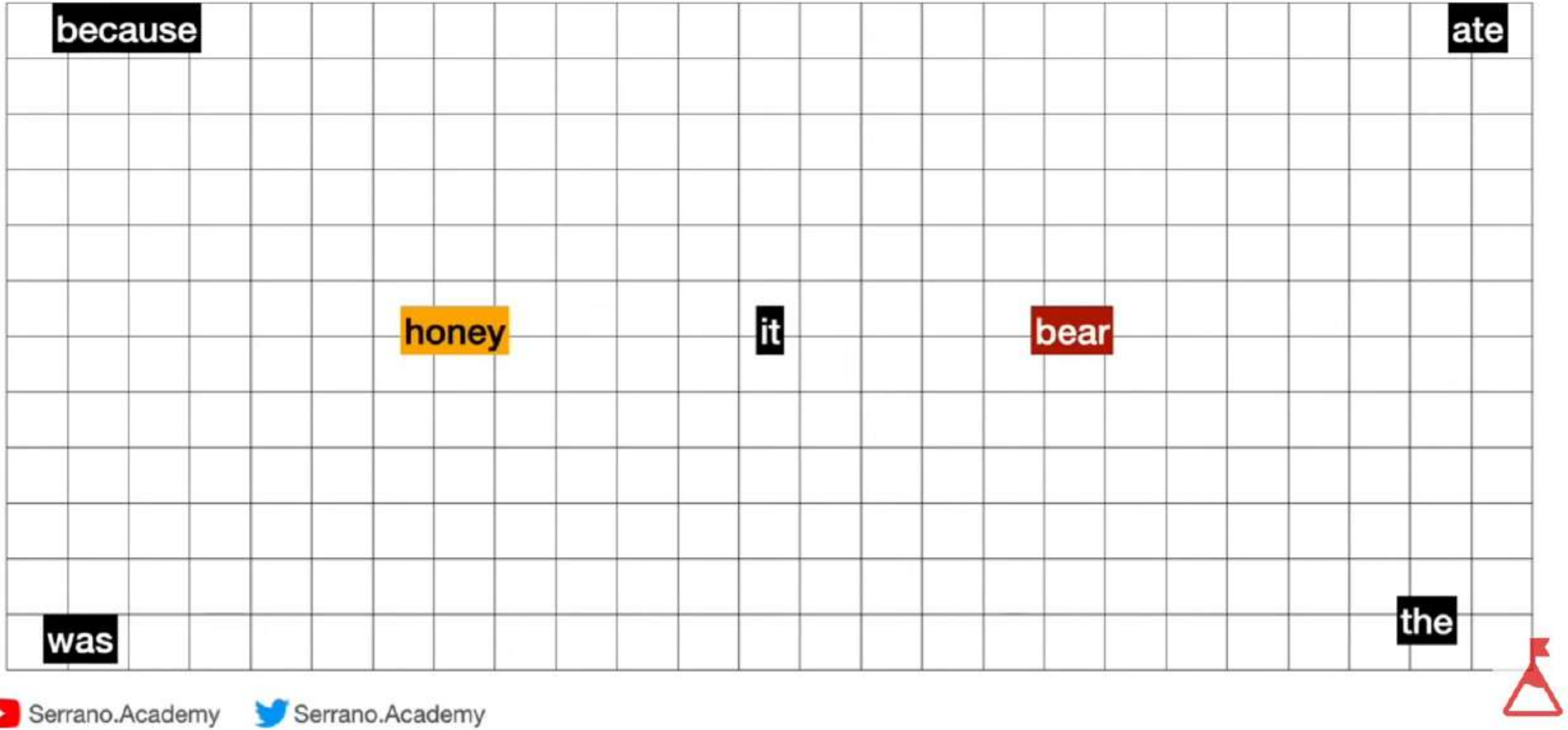
Luis Serrano

Using context



Luis Serrano

The bear ate the honey because it was _________



Mécanismes attentionnels

Mécanismes attentionnels

- L'objectif principal des mécanismes attentionnels est de modifier "temporairement" les encastrements des mots (pour désambiguïsation contextuelle).

Mécanismes attentionnels

- Compare : l'idée générale de l'attention sensorielle est d'augmenter le volume des neurones dont tu fais attention

- (rappelle que les encastrements sont en fait définis par les poids neuronaux)

Mécanismes attentionnels

- différences possibles avec attention sensoriel:

- 1) Les mécanismes attentionnels de l’IA pour LLM agissent par paire, par exemple, ils ressemblent davantage à un champ gravitationnel, où tout agit sur tout... il est possible que l'attention sensorielle ne fonctionne pas de cette façon.

Mécanismes attentionnels

- différences possibles avec attention sensoriel:

- 1) En particulier, l'attention sensorielle met généralement en avant une chose et fait passer les autres au second plan. L'attention de l’IA (pour les LLM) ne fait que modifier la proximité des choses les unes par rapport aux autres

Mécanismes attentionnels

- différences possibles avec attention sensoriel:

- 2) L'attention sensorielle peut réellement modifier les poids neuronaux, tandis que l'attention de l'IA produit simplement une nouvelle couche pour simuler ce que la sortie aurait été si les poids avaient été modifiés

Mécanismes attentionnels

- différences possibles avec attention sensoriel:

- 3) L'attention sensorielle peut être top-down, c'est-à-dire dirigée par ce que tu penses ou ressens, plutôt que par ce que tu vois ou entends. Dans l'IA, l'attention est entièrement fonction de ce que tu « vois »

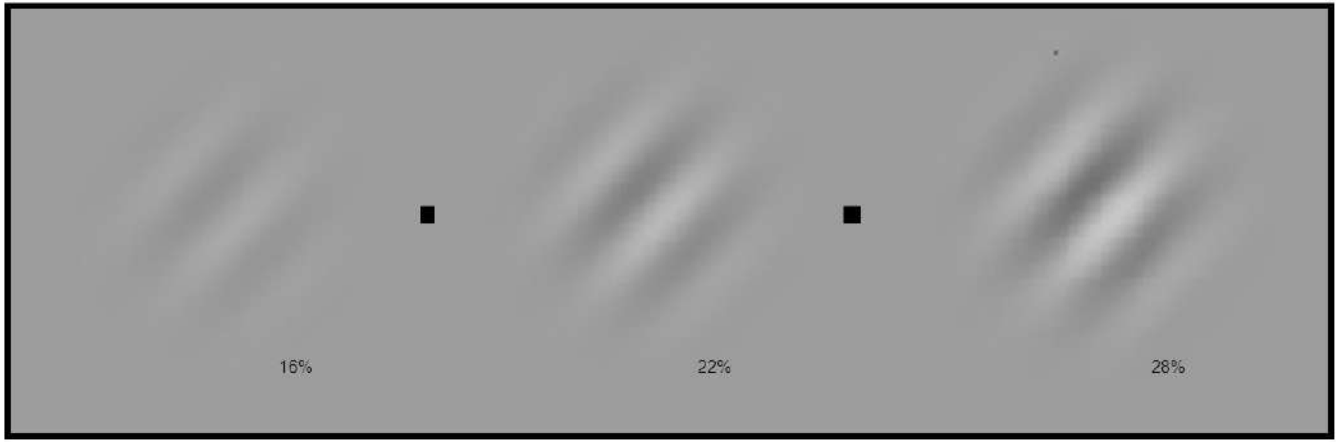
Mécanismes attentionnels

- différences possibles avec attention sensoriel:

- 4) Intuitivement, l'attention sensorielle ne modifie pas le contenu de ta représentation, elle se contente de «zoomer» - en revanche l'attention en IA modifie l'encastrement qui donne le sens lui-même
- Mais …

Attention alters contrast appearance

Test Cued | Neutral | Standard Cued



Carrasco, Ling & Read
Nature Neurosci, 2004

D’ici

- A partir d’ici j’utilize beaucoup des diapos tirée des videos de….

Serrano.Academy

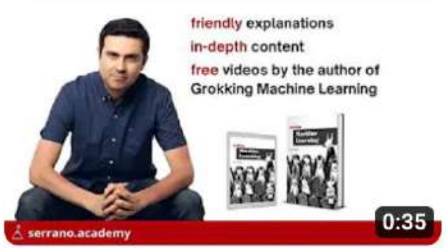
@SerranoAcademy · 127K subscribers · 46 videos

Welcome to Serrano.Academy! I'm Luis Serrano and I love demystifying concepts, capturin... >

twitter.com/SerranoAcademy **and 5 more links**

Subscribed

Home Videos Shorts Live Playlists Community



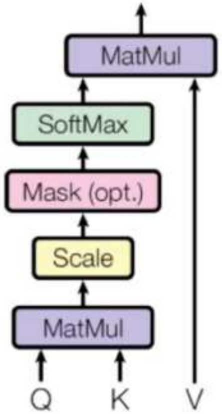
Serrano.Academy

Serrano.Academy • 11K views • 1 year ago

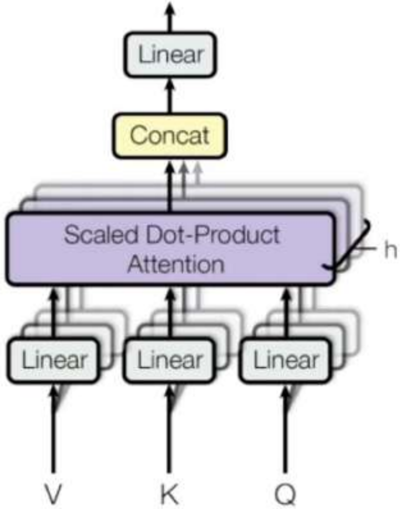
Learn machine learning and data science the simple way. Free videos and tutorials on deep learning, reinforcement learning, probability, statistics, and much more. Luis Serrano, the author...

Attention

Scaled Dot-Product Attention



Multi-Head Attention



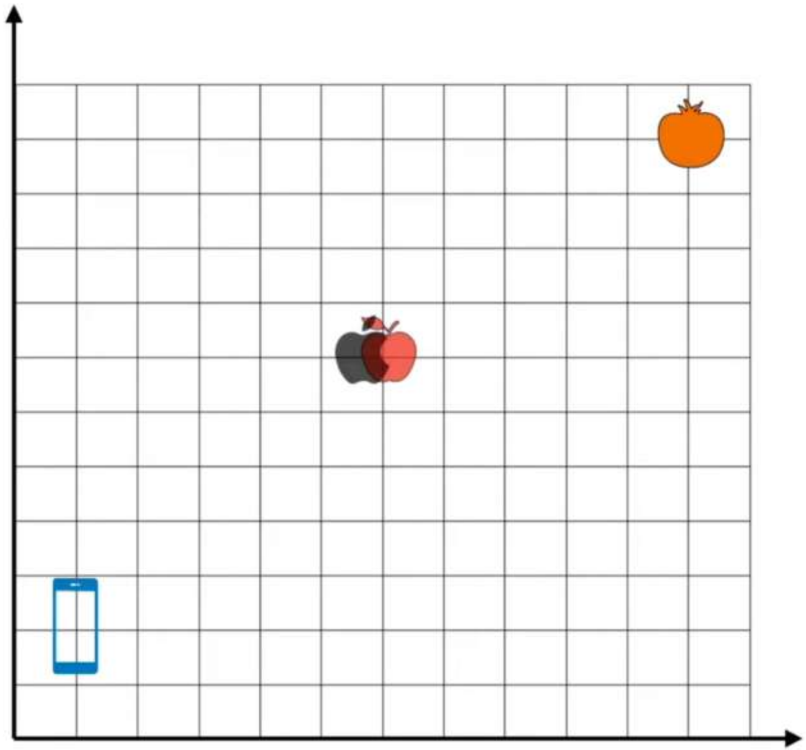
$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Vaswani et al 2017

Attention

please buy an apple and an $\boxed{\text{orange}}$



apple unveiled the new phone

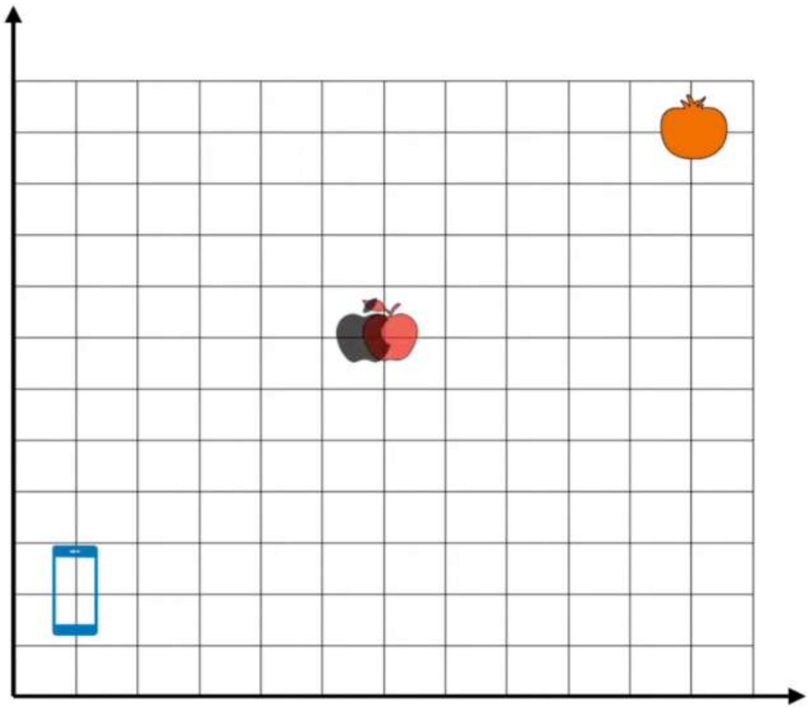
Luis Serrano

Attention

please buy an apple and an orange

apple unveiled the new phone

Attention



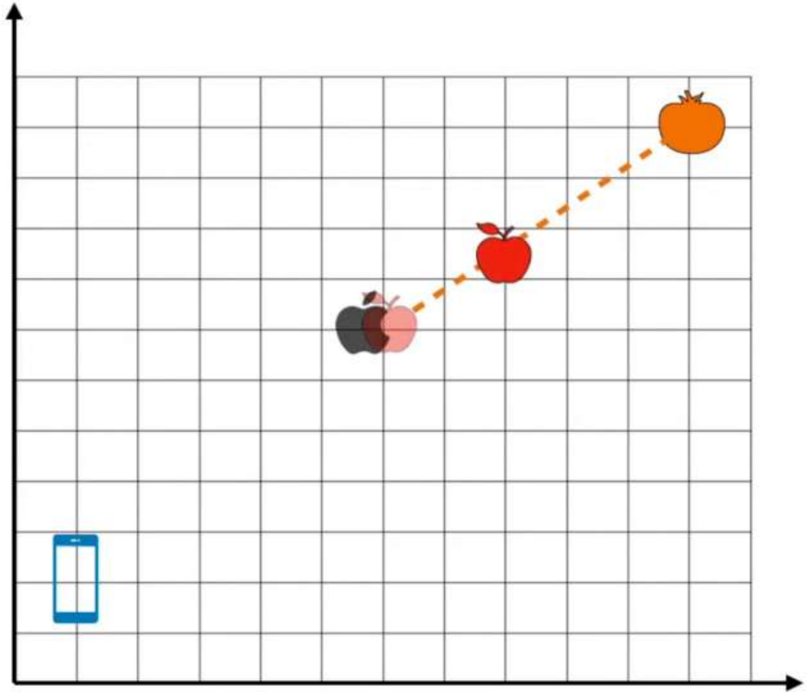
please buy an apple and an orange

apple unveiled the new phone

Luis Serrano

Attention

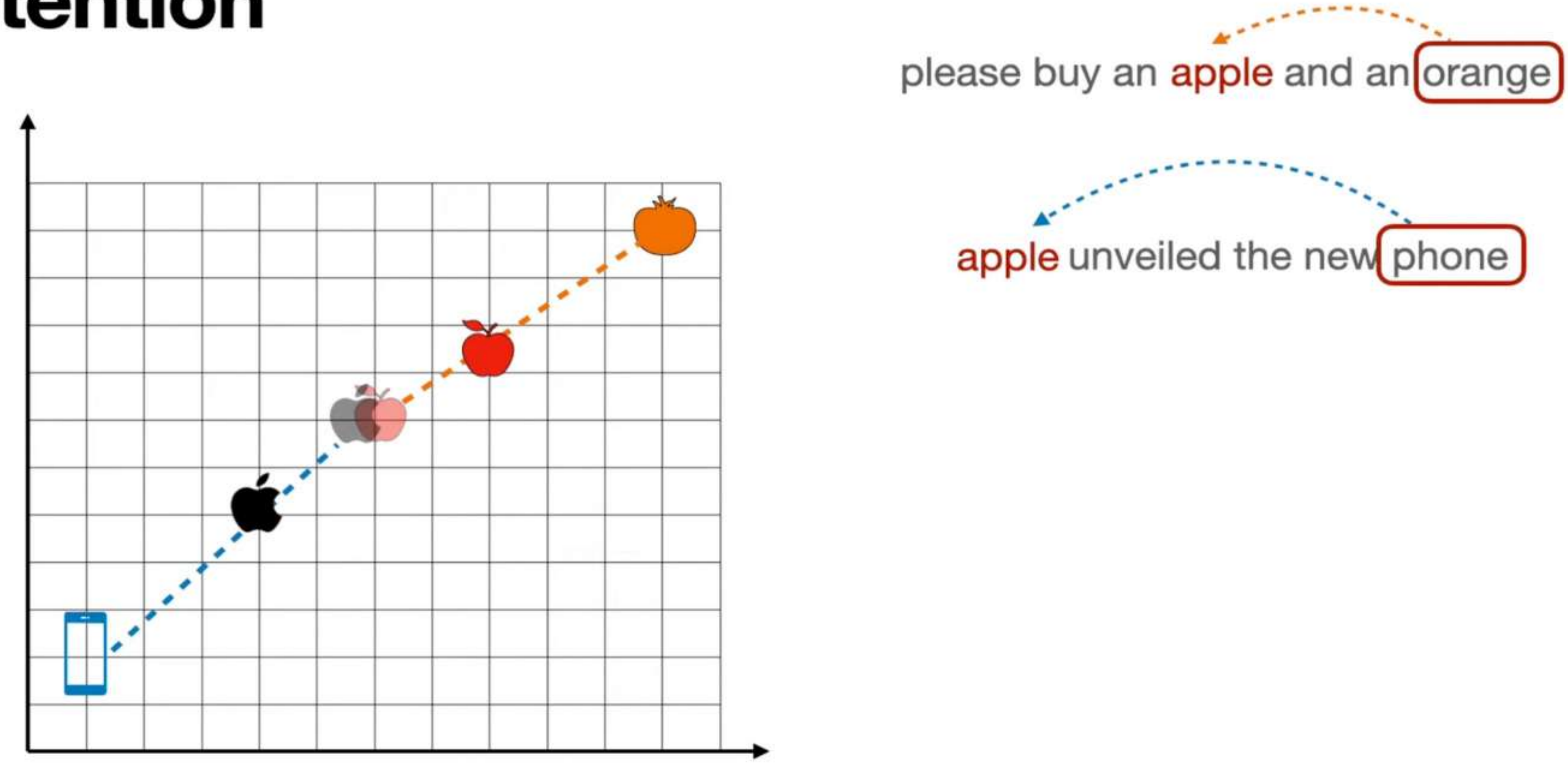
please buy an apple and an orange



apple unveiled the new phone

Luis Serrano

Attention

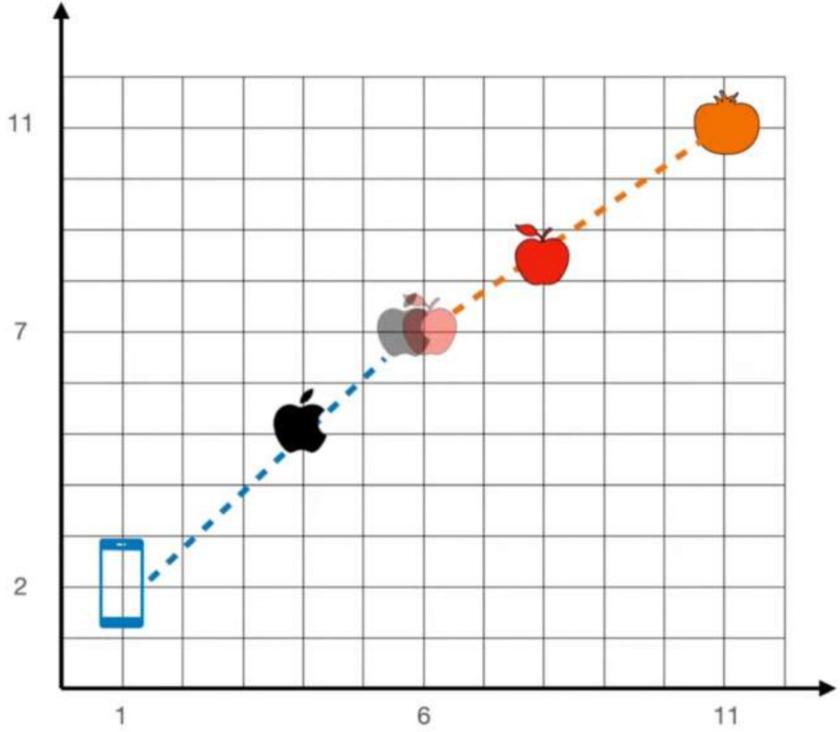


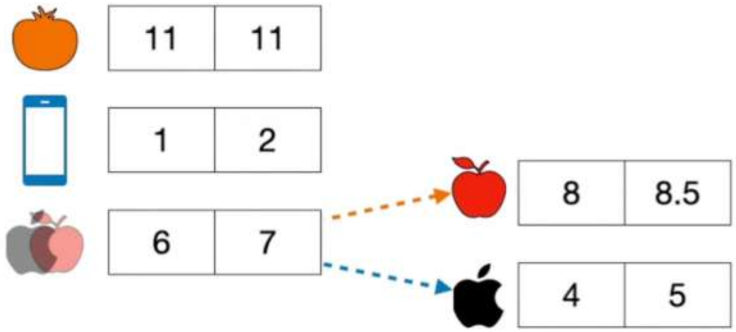
Luis Serrano

Attention

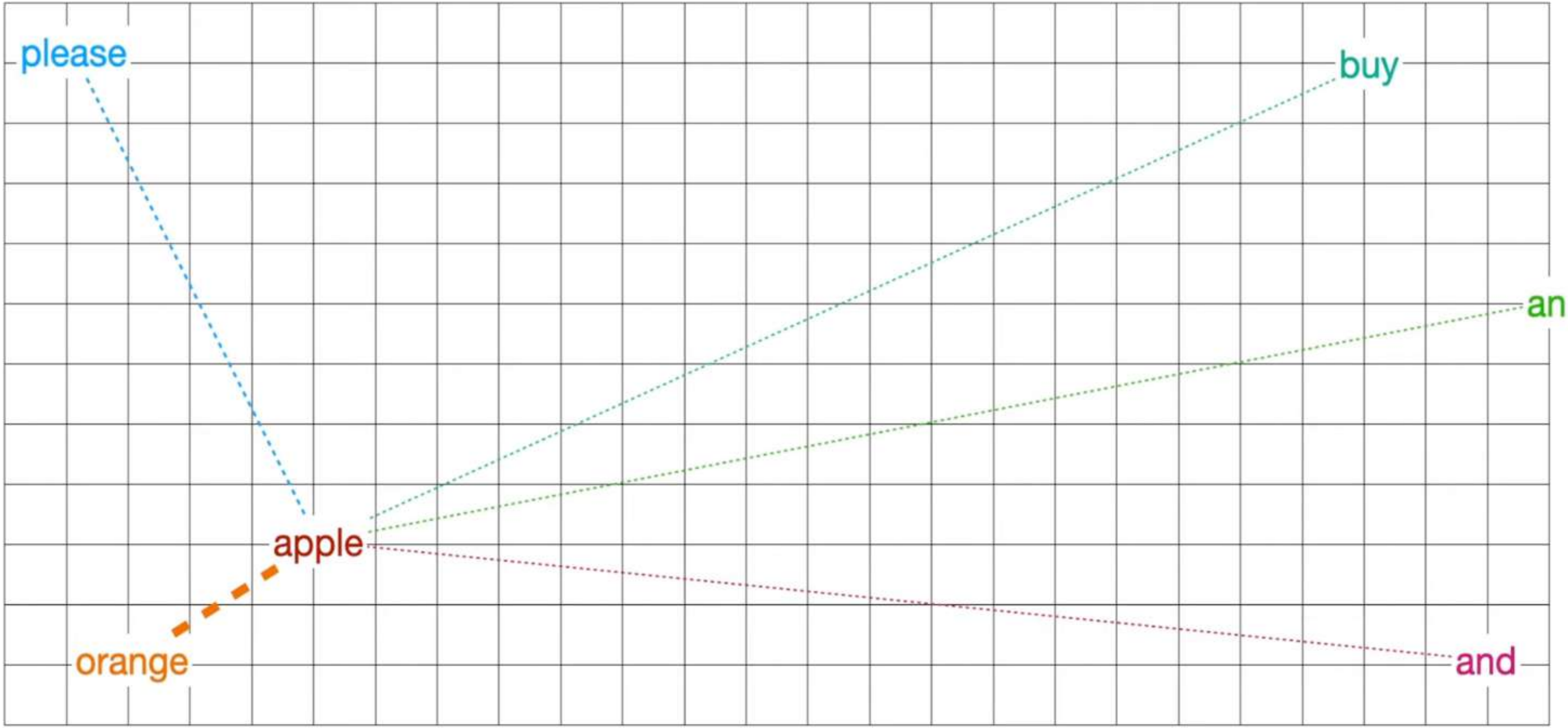
please buy an apple and an orange

apple unveiled the new phone



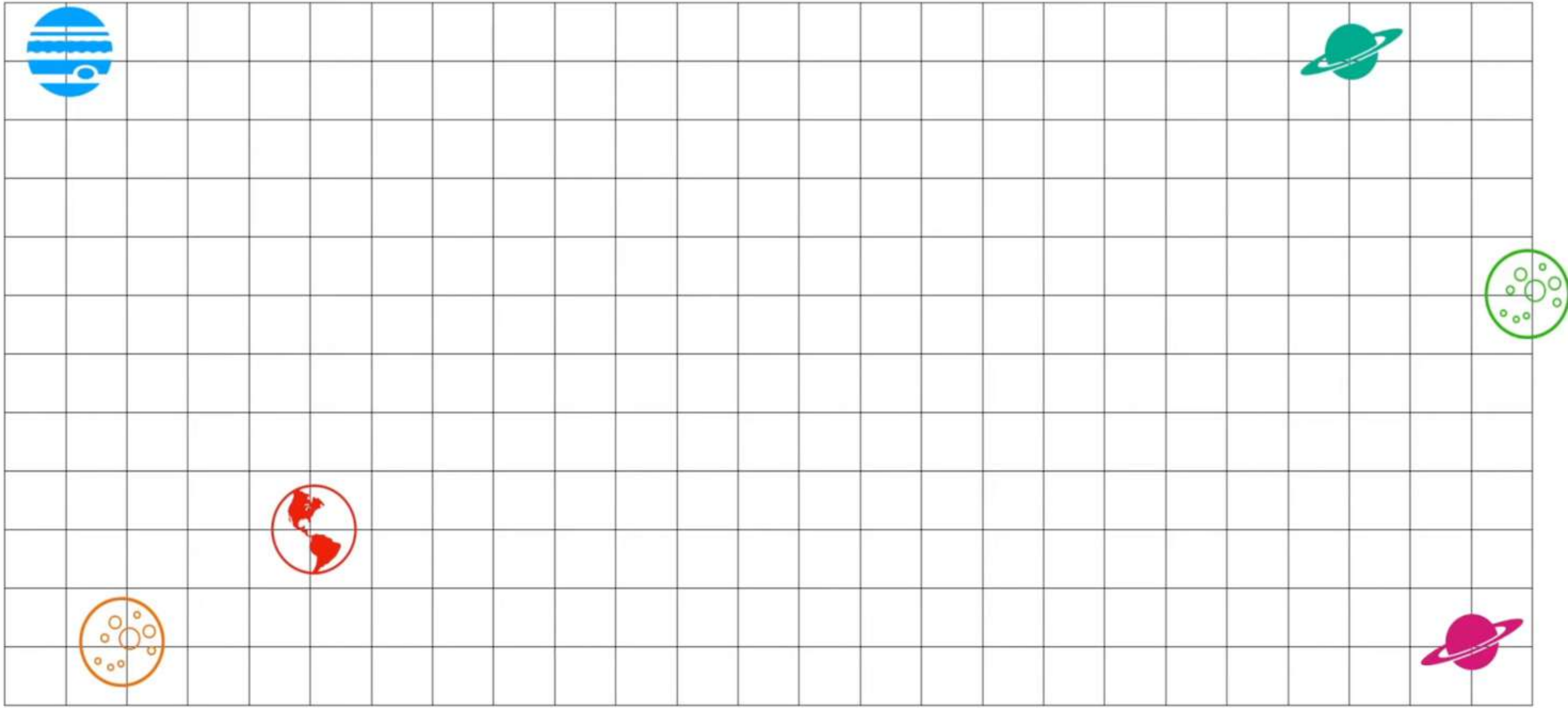


What about the other words?



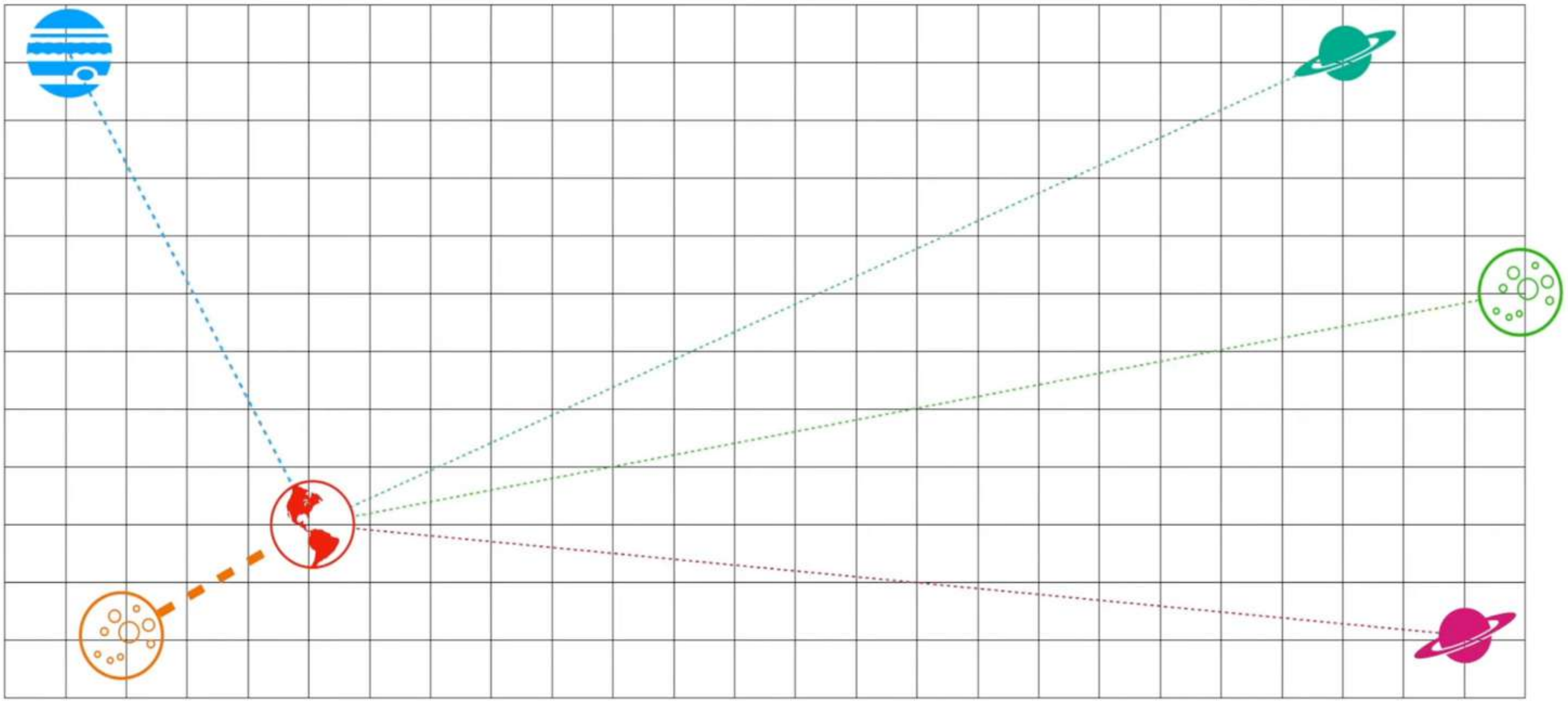
please buy an apple and an orange

It’s kind of like gravity…

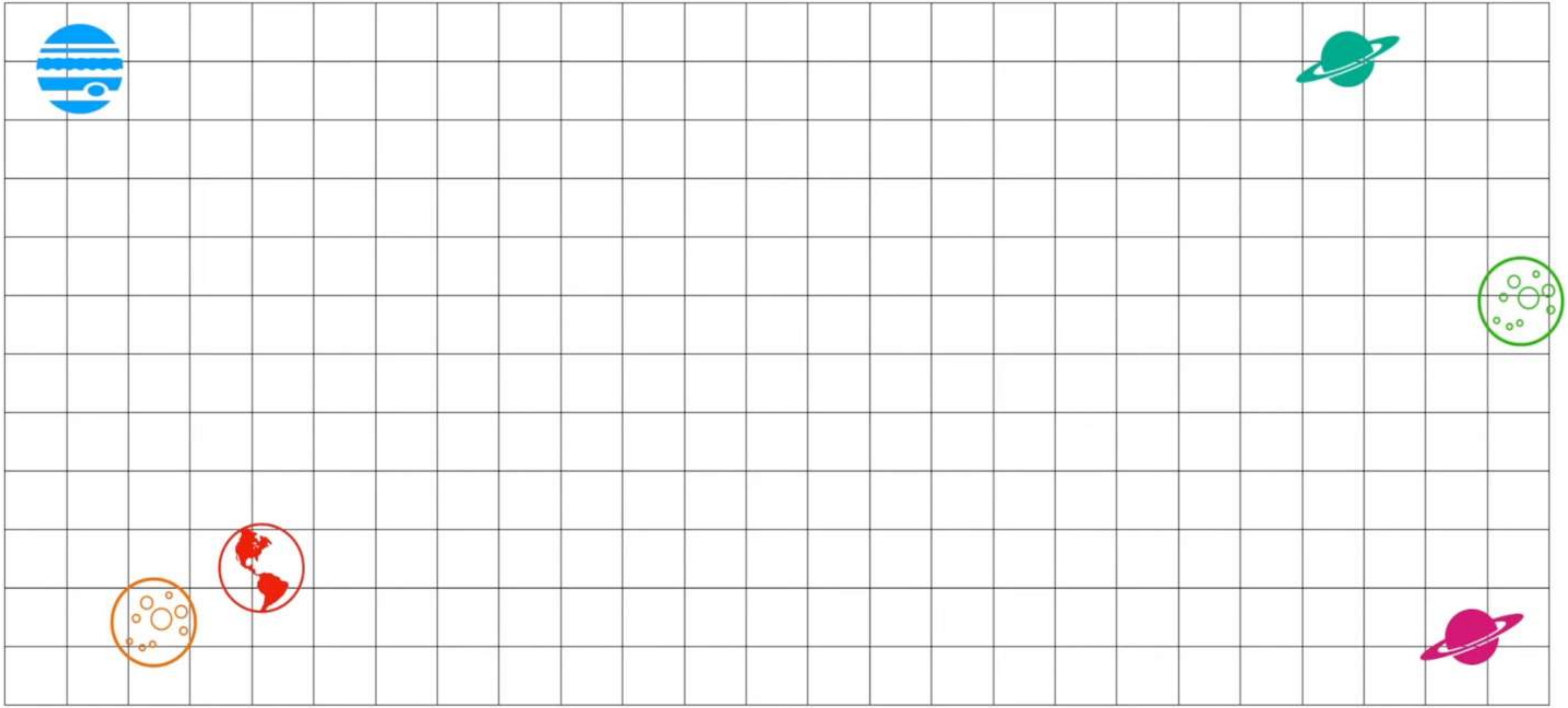


please buy an apple and an orange

It’s kind of like gravity…

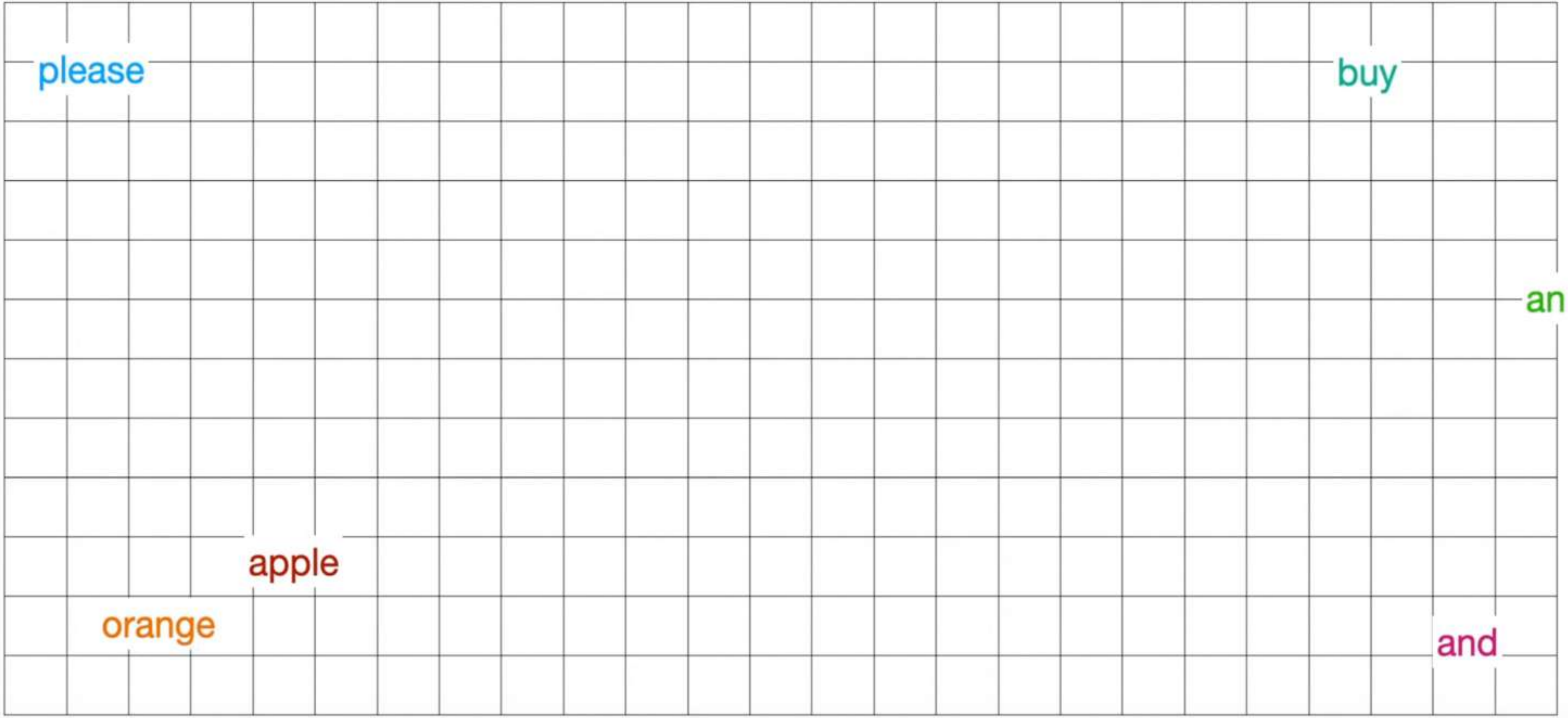


It’s kind of like gravity…



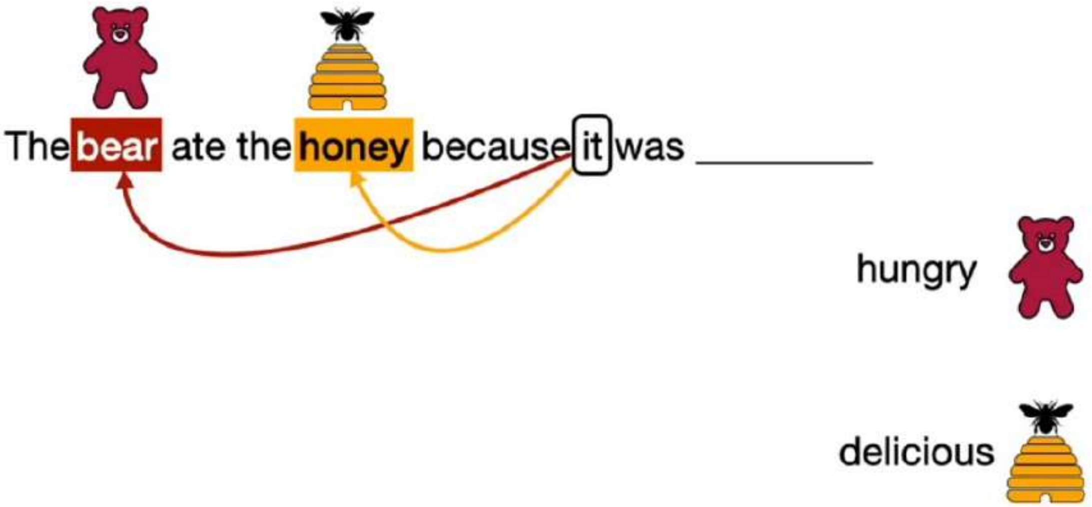
please buy an apple and an orange

You apply attention to all the words



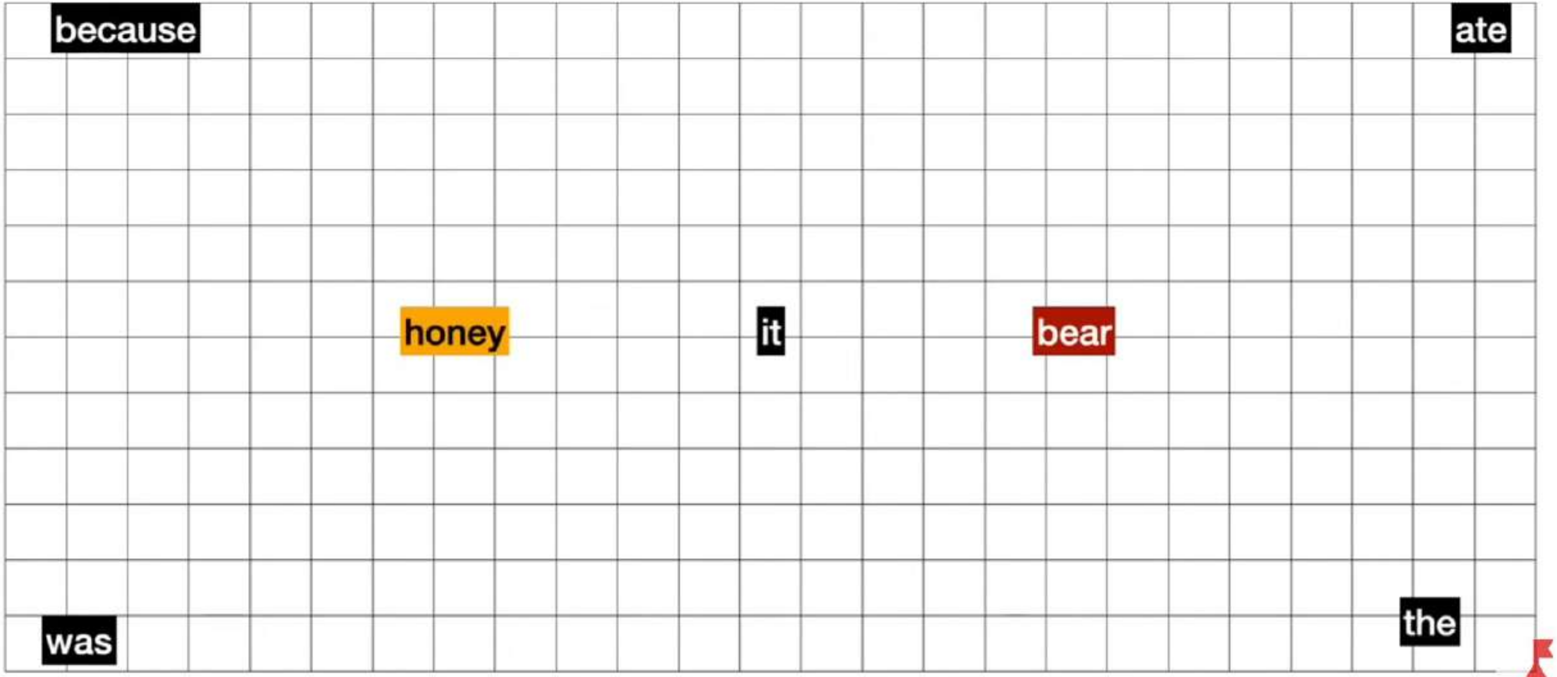
please buy an apple and an orange

Using context

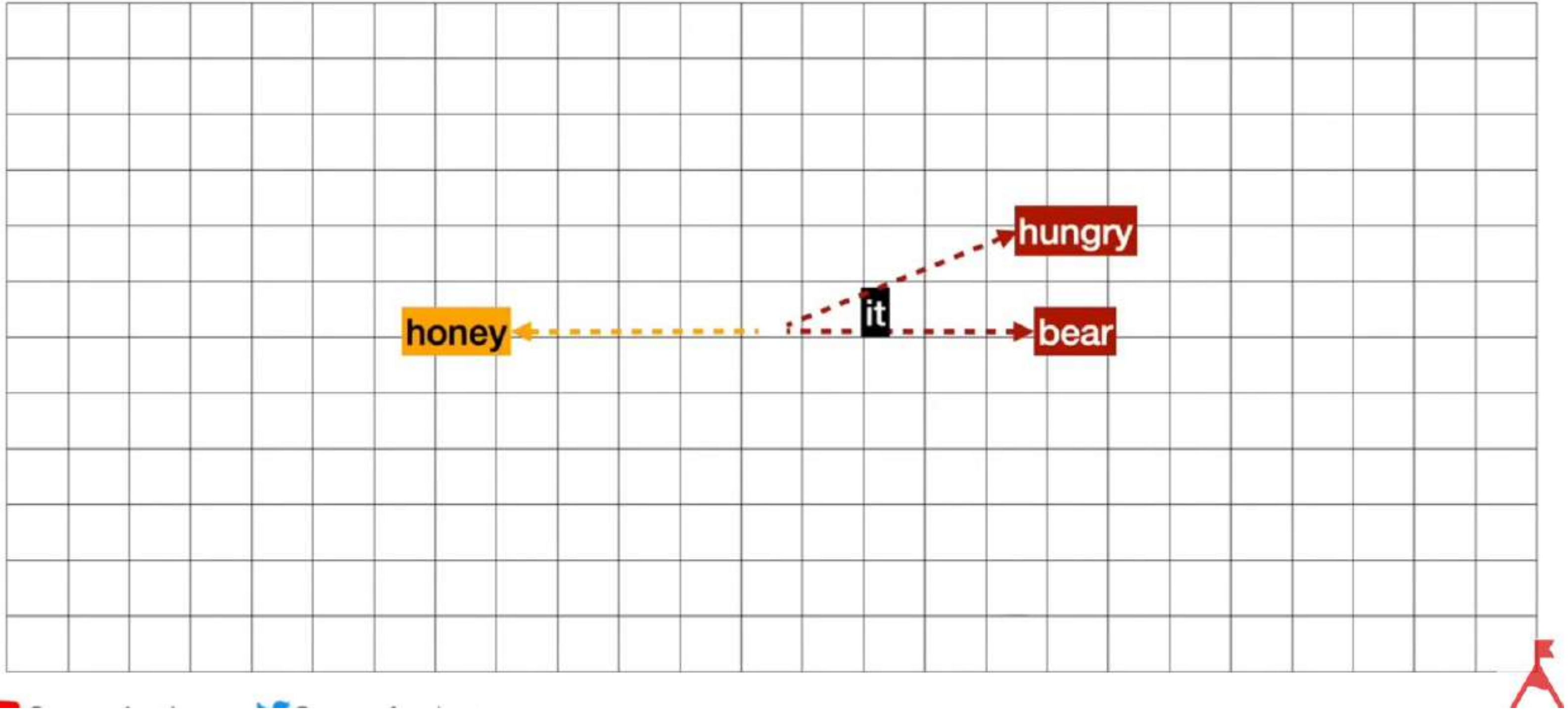


Luis Serrano

The bear ate the honey because it was ________

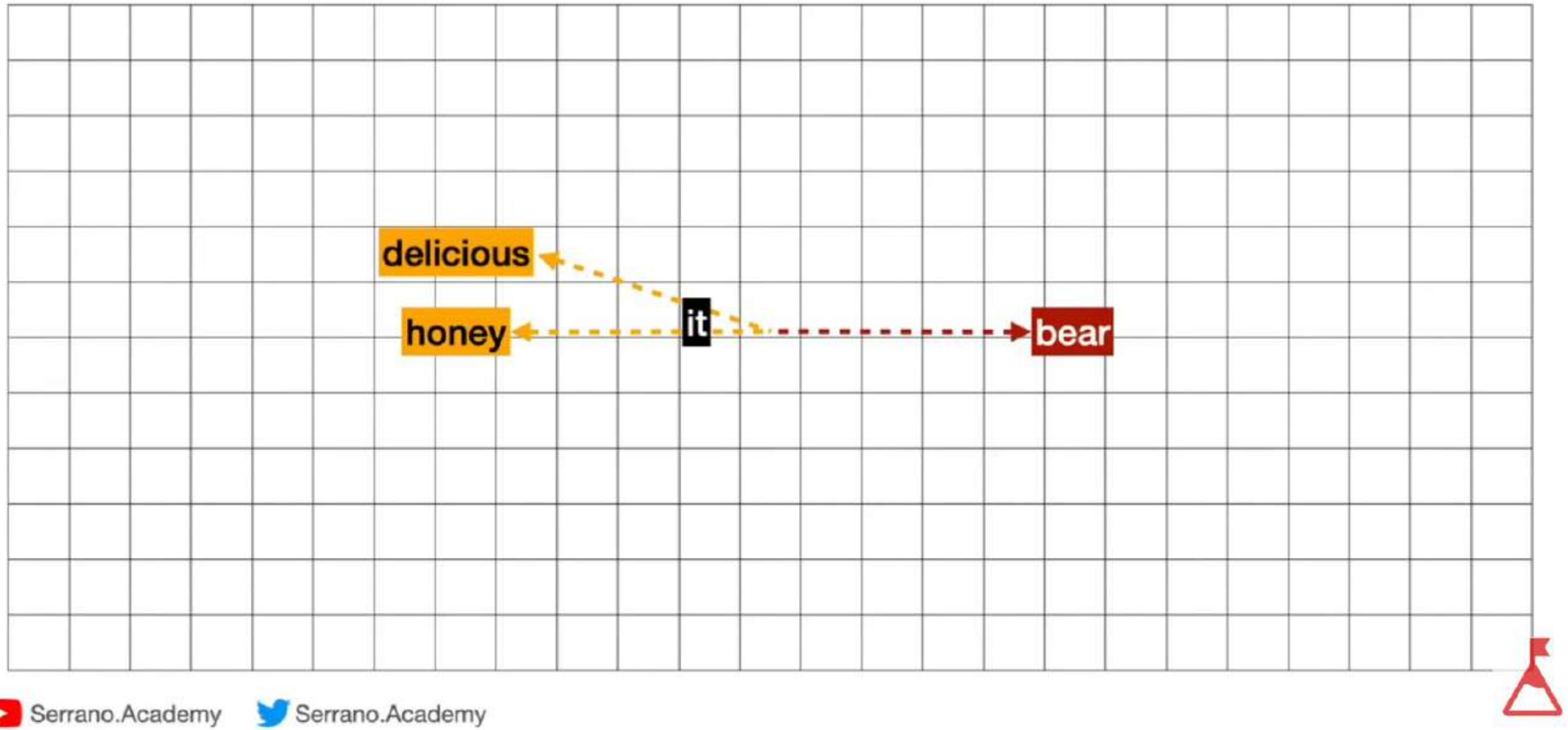


The bear ate the honey because it was hungry

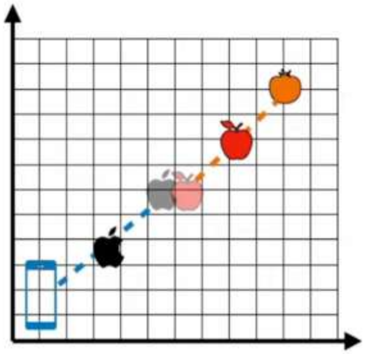


Luis Serrano

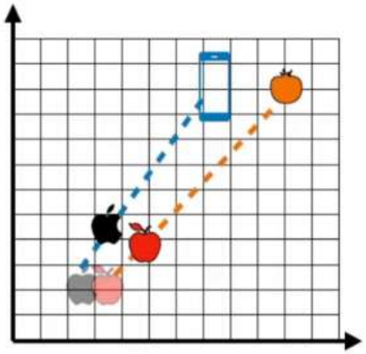
The bear ate the honey because it was delicious



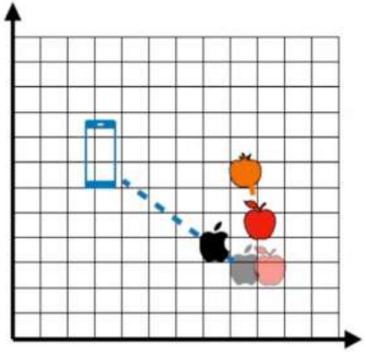
Ideally, we'd like to have lots of embeddings



Good



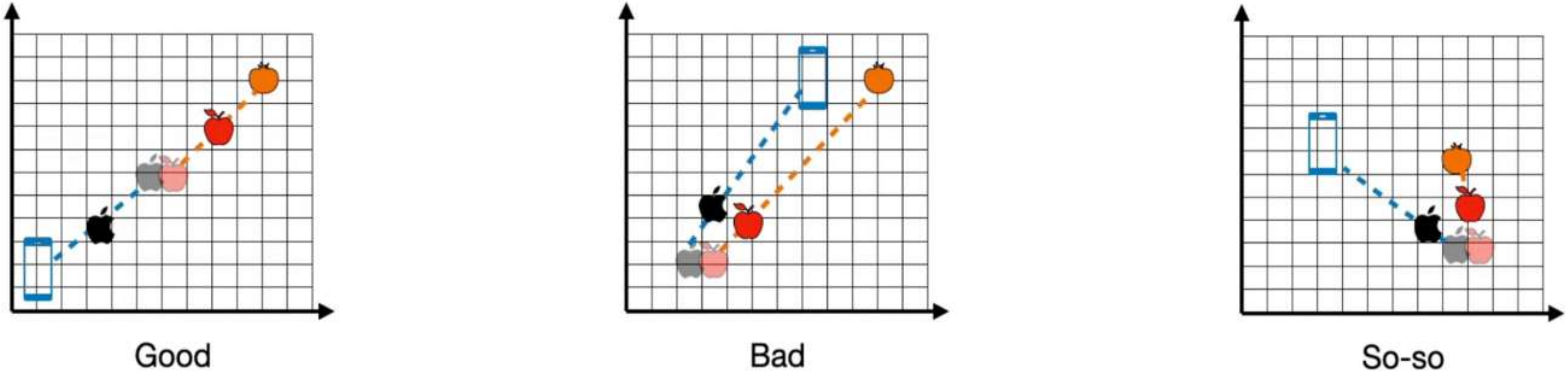
Bad



So-so

Luis Serrano

Ideally, we'd like to have lots of embeddings

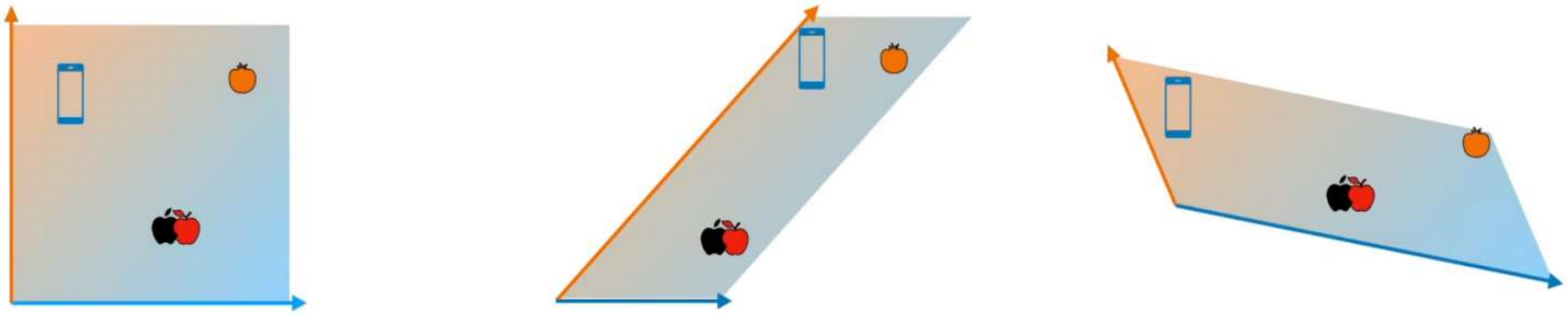


Problem: Building many embeddings is a lot of work!

Solution: We'll build embeddings by modifying existing embeddings

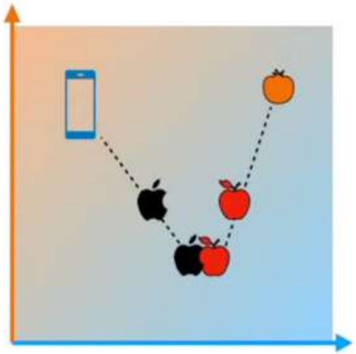
Luis Serrano

Get new embeddings from existing ones

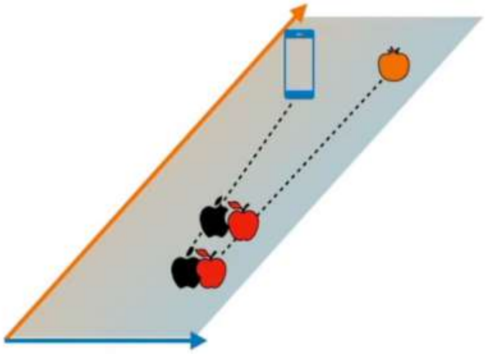


Luis Serrano

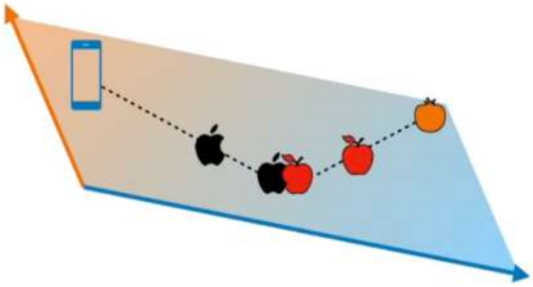
Get new embeddings from existing ones



Okay



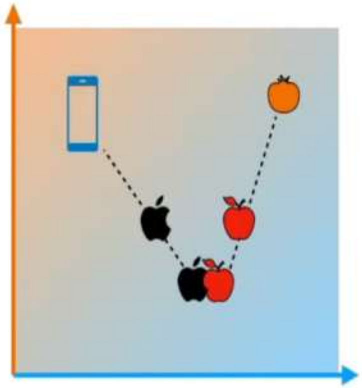
Bad



Good

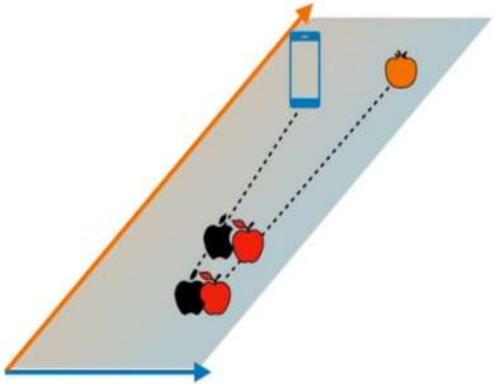
Luis Serrano

Get new embeddings from existing ones



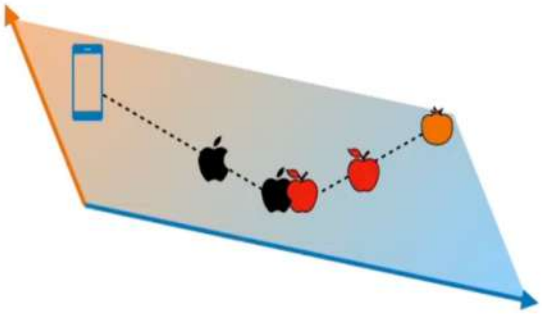
Okay

Score: 1



Bad

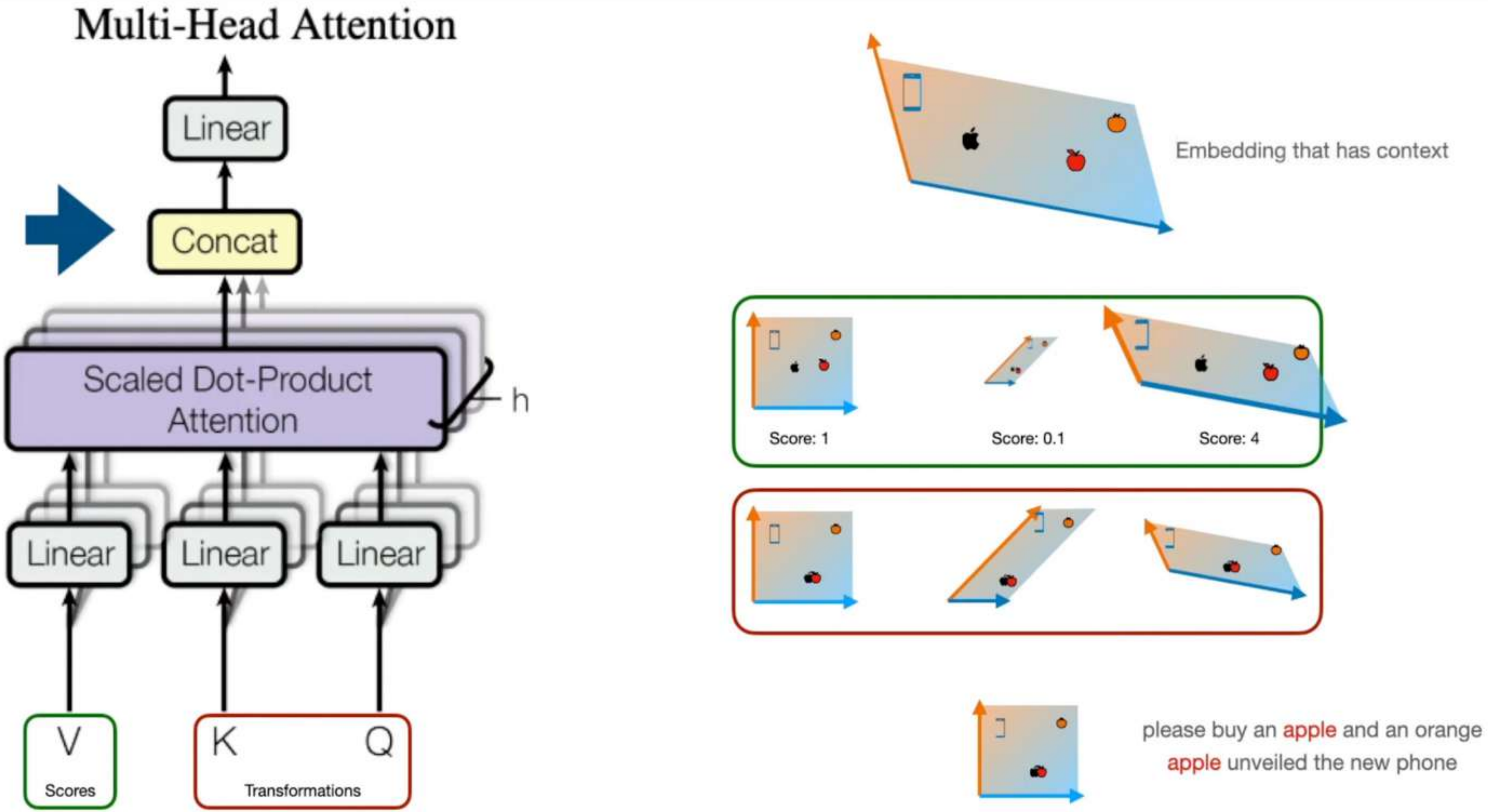
Score: 0.1



Good

Score: 4

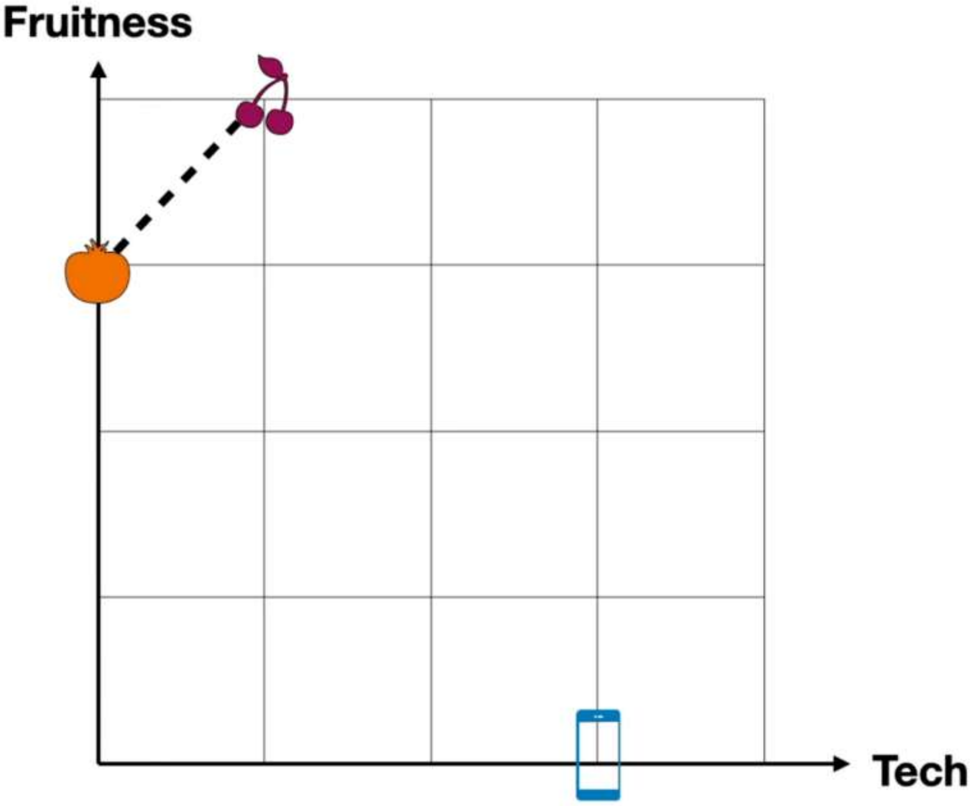
Luis Serrano



Luis Serrano

Comment faire le calcul?

Measure 1: Dot product



Sim

1	4
0	3

Measure 1: Dot product

Sim

	Tech	Fruitness
	1	4
	0	3

Tech		Fruitness	

$$1 \cdot 0 \; + \; 4 \cdot 3 \; = 12$$

Measure 1: Dot product

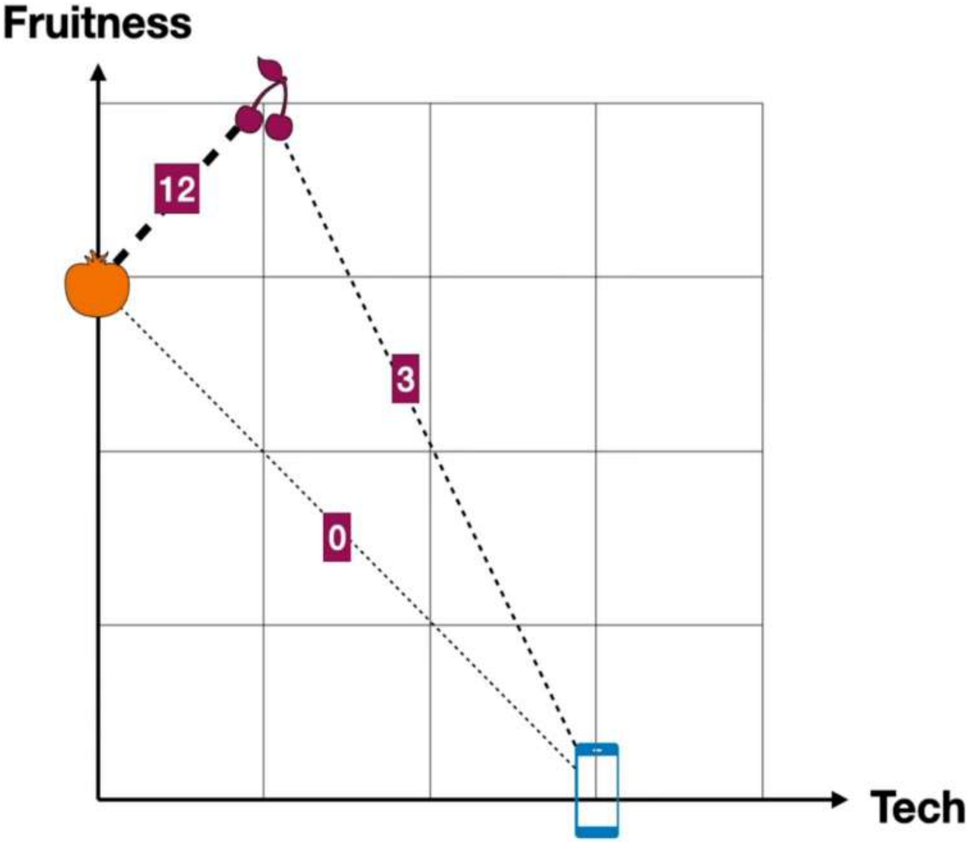
	Tech	Fruitness
Sim (cherry)	1	4
(orange)	0	3

$$1 \cdot 0 + 4 \cdot 3 = 12$$

	Tech	Fruitness
Sim (cherry)	1	4
(phone)	3	0

$$1 \cdot 3 + 4 \cdot 0 = 3$$

Measure 1: Dot product



Sim

Tech	Fruitness
1	4
0	3

$$1 \cdot 0 + 4 \cdot 3 = 12$$

Sim

Tech	Fruitness
1	4
3	0

$$1 \cdot 3 + 4 \cdot 0 = 3$$

Sim

Tech	Fruitness
0	3
3	0

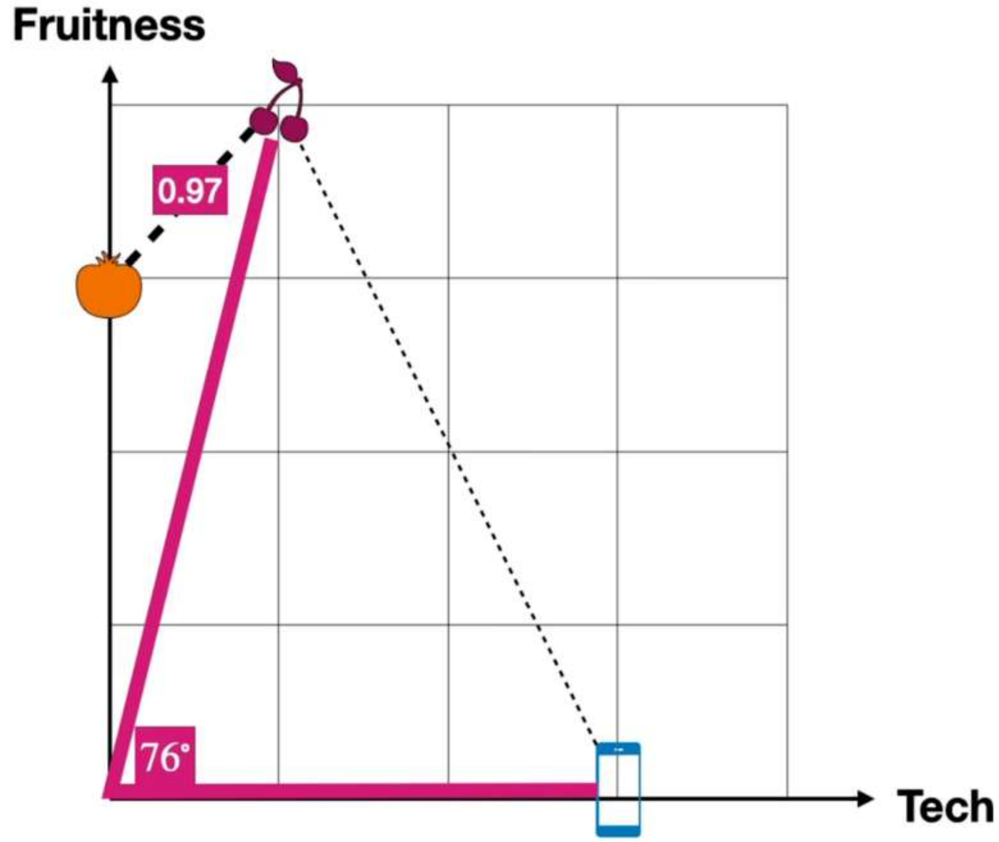
$$0 \cdot 3 + 3 \cdot 0 = 0$$

Measure 2: Cosine similarity

Measure 2: Cosine similarity

$cos(14^{\circ}) = 0.97$

Measure 2: Cosine similarity



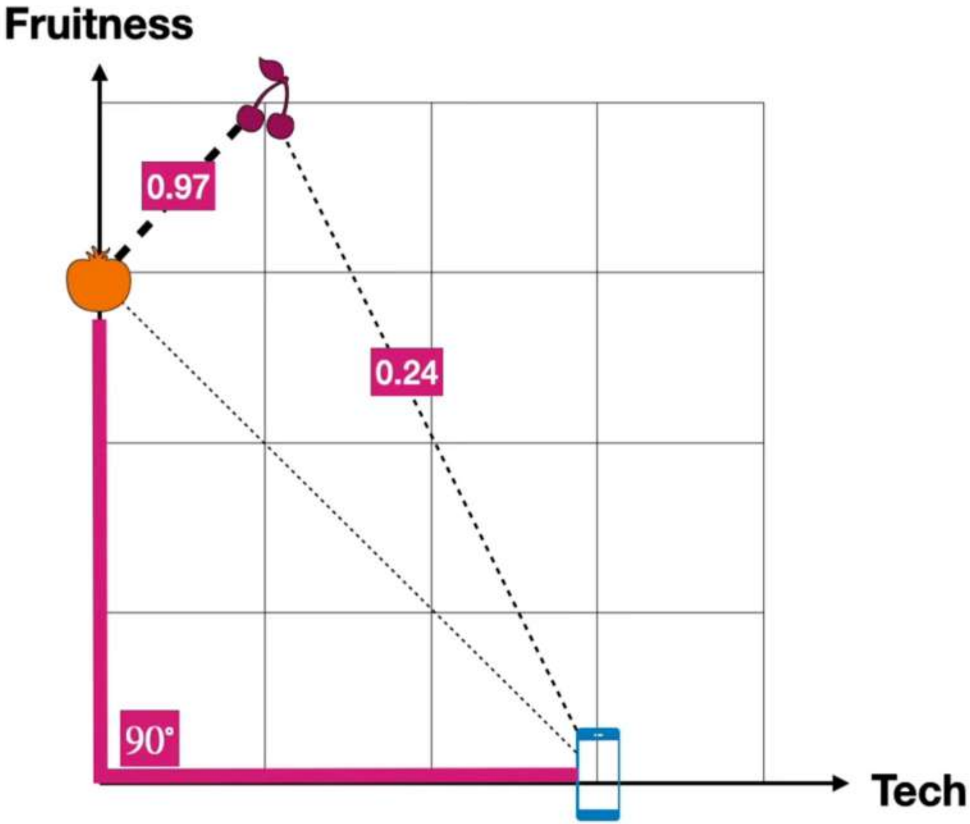
Sim

$cos(14^\circ) = 0.97$

Sim

$cos(76^\circ) = 0.24$

Measure 2: Cosine similarity

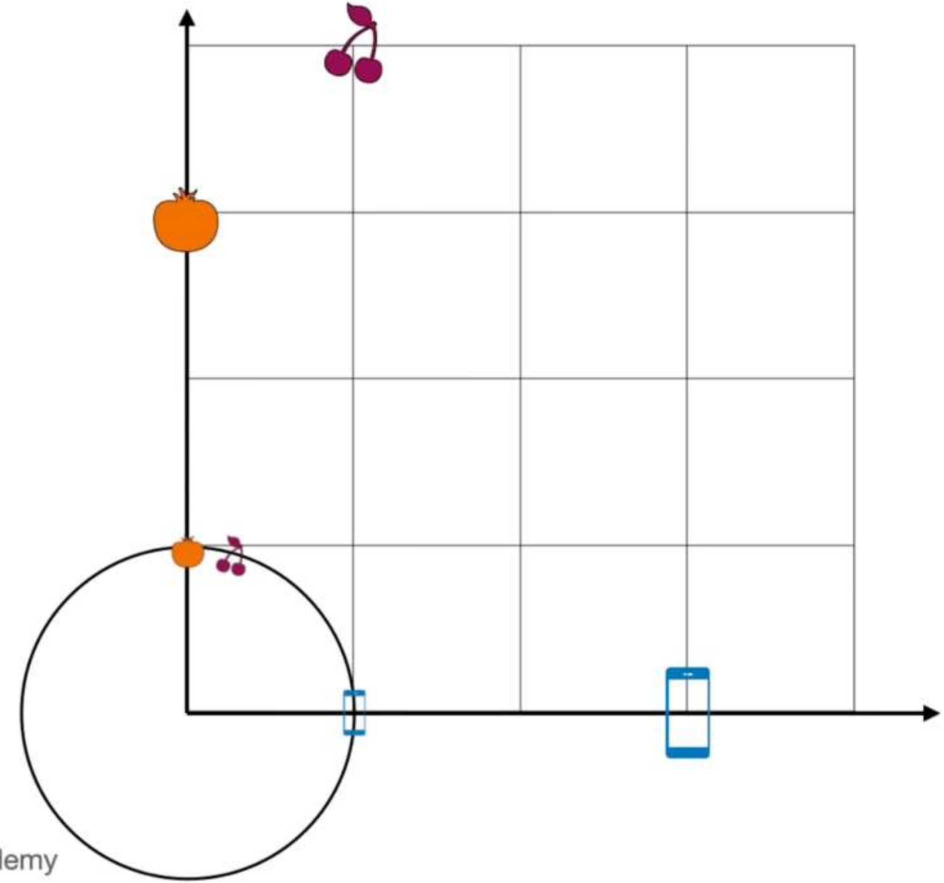


Sim $cos(14°) = 0.97$

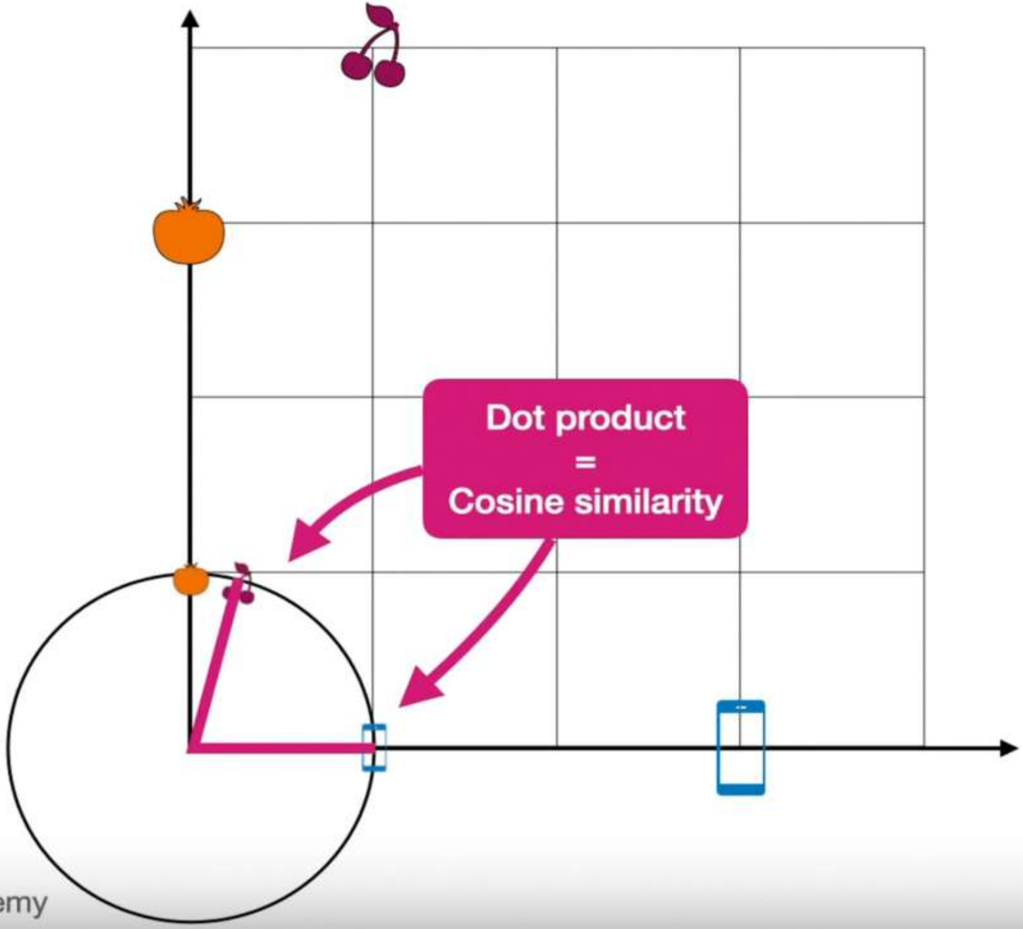
Sim $cos(76°) = 0.24$

Sim $cos(90°) = 0$

Dot product and cosine similarity



Dot product and cosine similarity



Measure 3: Scaled dot product

Dot product divided by the square root of the length of the vector

Sim

1	4

0	3

$$1 \cdot 0 \; + \; 4 \cdot 3 \; = 12 \longrightarrow \frac{12}{\sqrt{2}} = 8.49$$

Measure 3: Scaled dot product

Dot product divided by the square root of the length of the vector

Sim

1	4

0	3

$$1 \cdot 0 + 4 \cdot 3 = 12 \longrightarrow \frac{12}{\sqrt{2}} = 8.49$$

Sim

1	4

3	0

$$1 \cdot 3 + 4 \cdot 0 = 3 \longrightarrow \frac{3}{\sqrt{2}} = 2.12$$

Sim

0	3

3	0

$$0 \cdot 3 + 3 \cdot 0 = 0 \longrightarrow \frac{0}{\sqrt{2}} = 0$$

Measure 3: Scaled dot product

Dot product divided by the square root of the length of the vector

Sim

1	4
0	3

$$1 \cdot 0 + 4 \cdot 3 = 12 \longrightarrow \frac{12}{\boxed{\sqrt{2}}} = 8.49$$

Sim

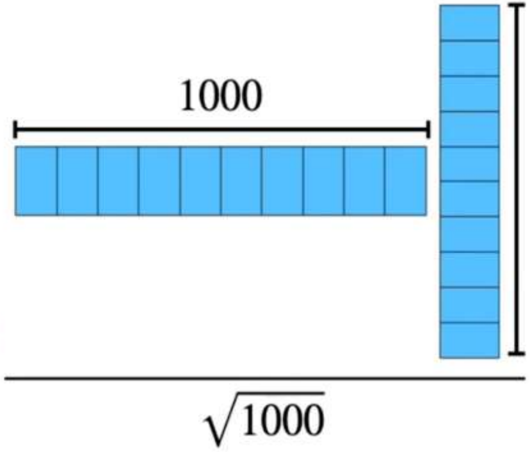
1	4
3	0

$$1 \cdot 3 + 4 \cdot 0 = 3 \longrightarrow \frac{3}{\boxed{\sqrt{2}}} = 2.12$$

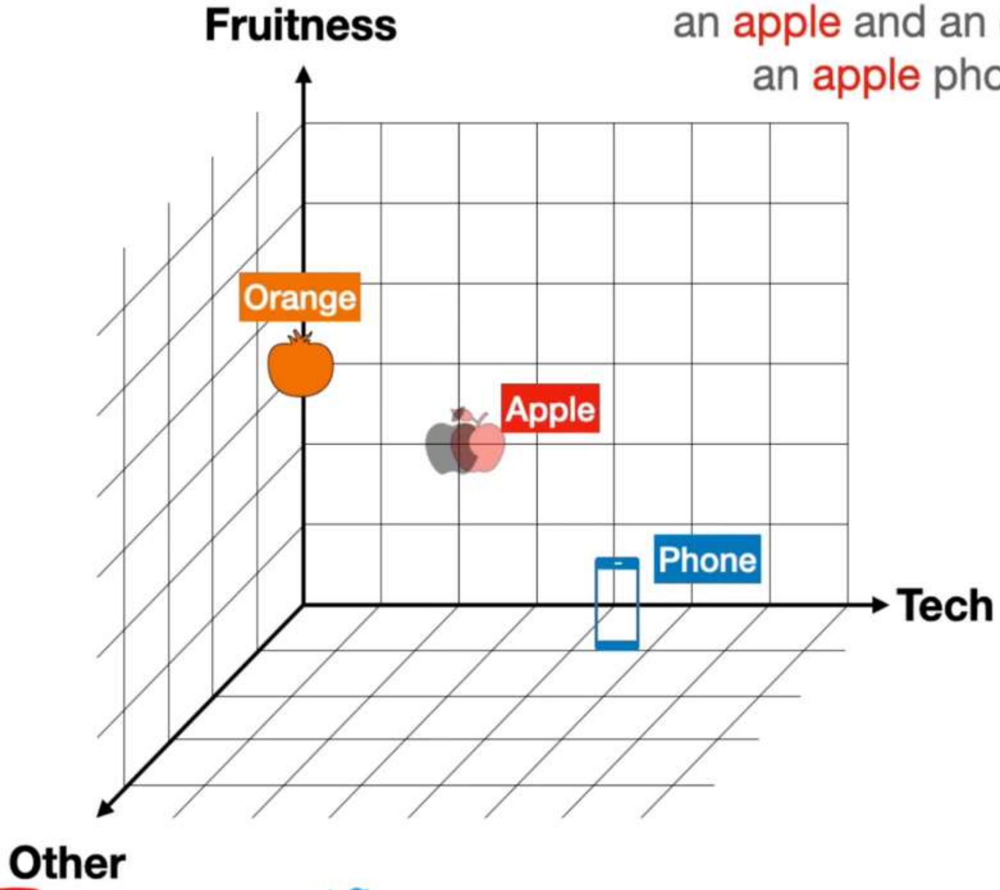
Sim

0	3
3	0

$$0 \cdot 3 + 3 \cdot 0 = 0 \longrightarrow \frac{0}{\boxed{\sqrt{2}}} = 0$$



Cosine similarity

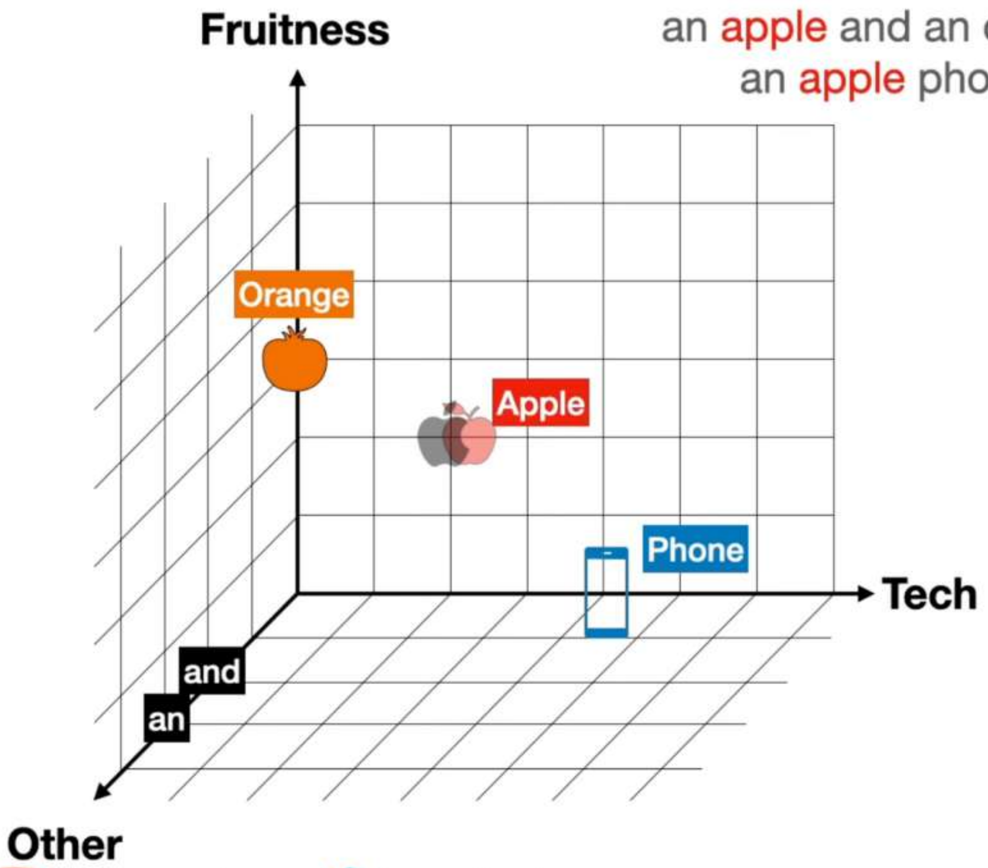


an apple and an orange
an apple phone

	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0

Luis Serrano

Cosine similarity

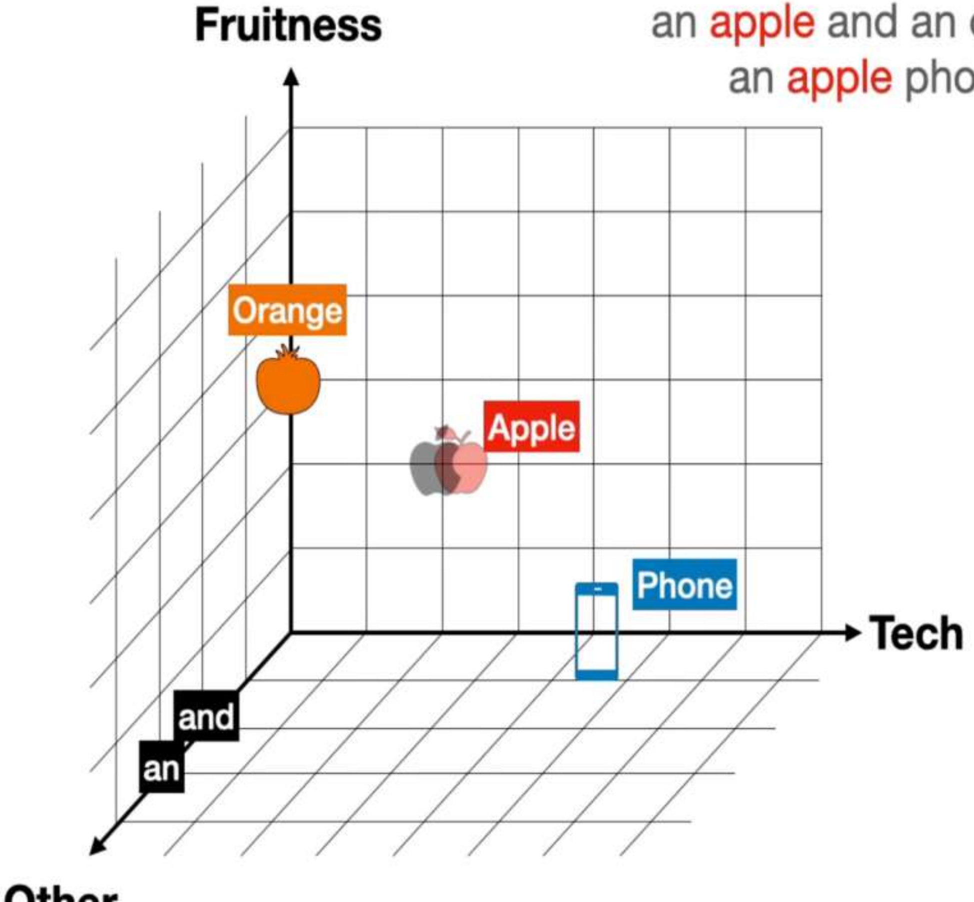


an apple and an orange
an apple phone

	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

Cosine similarity

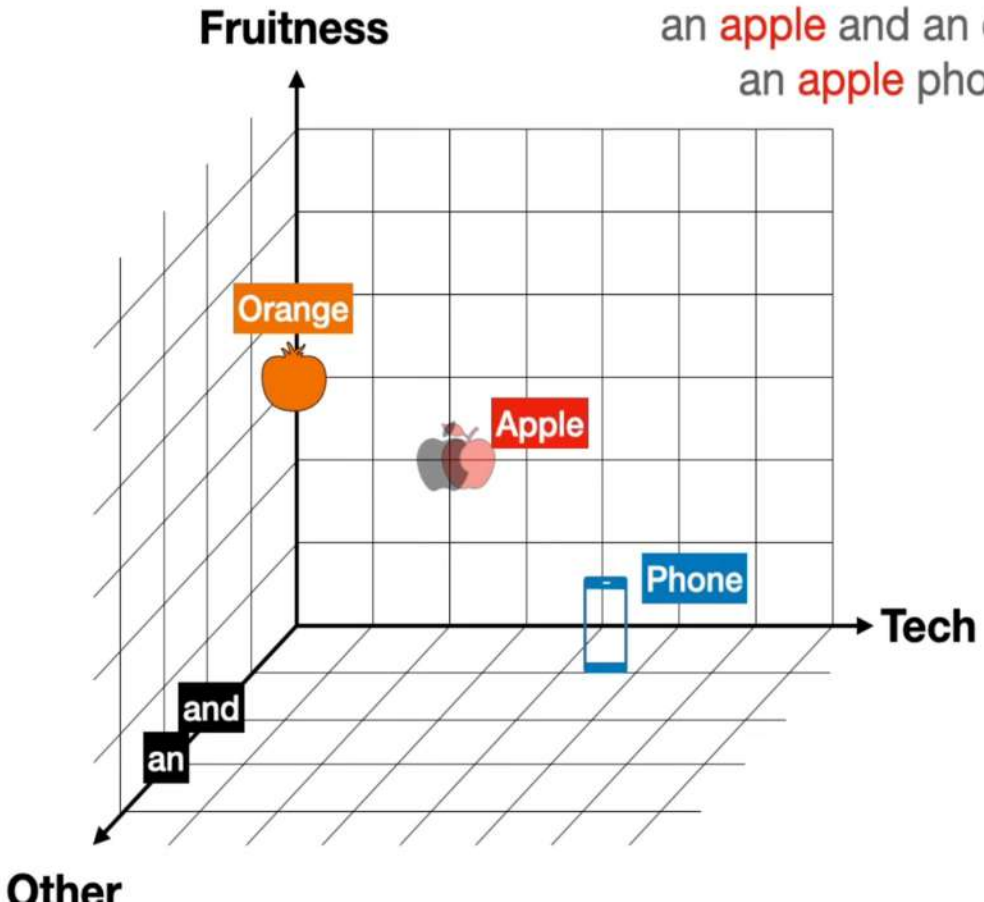
an apple and an orange
an apple phone



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

	Orange	Phone	Apple	And	An
Orange	1				
Phone		1			
Apple			1		
And				1	
An					1

Cosine similarity



	Tech	Fruitness	Other
Orange	0	3	0
Phone	4	0	0
Apple	2	2	0
And	0	0	2
An	0	0	3

	Orange	Phone	Apple	And	An
Orange	1	0	0.71	0	0
Phone	0	1	0.71	0	0
Apple	0.71	0.71	1	0	0
And	0	0	0	1	1
An	0	0	0	1	1

Word math

an apple and an orange

	Orange	Apple	And	An
Orange	1	0.71	0	0
Apple	0.71	1	0	0
And	0	0	1	1
An	0	0	1	1

Orange $\longrightarrow$ 1 Orange + 0.71 Apple

Apple $\longrightarrow$ 0.71 Orange

And $\longrightarrow$

An $\longrightarrow$

Word math

an apple and an orange

	Orange	Apple	And	An
Orange	1	0.71	0	0
Apple	0.71	1	0	0
And	0	0	1	1
An	0	0	1	1

Orange $\longrightarrow$ 1 Orange + 0.71 Apple

Apple $\longrightarrow$ 0.71 Orange + 1 Apple

And $\longrightarrow$

An $\longrightarrow$

Luis Serrano

Word math

an apple and an orange

	Orange	Apple	And	An
Orange	1	0.71	0	0
Apple	0.71	1	0	0
And	0	0	1	1
An	0	0	1	1

Orange $\longrightarrow$ 1 Orange + 0.71 Apple

Apple $\longrightarrow$ 0.71 Orange + 1 Apple

And $\longrightarrow$ 1 And + 1 An

An $\longrightarrow$

Luis Serrano

Word math

an apple and an orange

	Orange	Apple	And	An
Orange	1	0.71	0	0
Apple	0.71	1	0	0
And	0	0	1	1
An	0	0	1	1

Orange $\longrightarrow$ 1 Orange + 0.71 Apple

Apple $\longrightarrow$ 0.71 Orange + 1 Apple

And $\longrightarrow$ 1 And + 1 An

An $\longrightarrow$ 1 An

Word math

an apple phone

	Phone	Apple	An
Phone	1	0.71	0
Apple	0.71	1	0
An	0	0	1

Phone $\longrightarrow$ 1 Phone + 0.71 Apple

Apple $\longrightarrow$ 0.71 Phone + 1 Apple

An $\longrightarrow$ 1 An

Luis Serrano

Normalization

Want coefficients to add to 1

Orange $\longrightarrow$ 1 Orange + 0.71 Apple

Normalization

Want coefficients to add to 1

$$\text{Orange} \longrightarrow \frac{1\ \text{Orange} + 0.71\ \text{Apple}}{1 + 0.71} = 0.58\ \text{Orange} + 0.42\ \text{Apple}$$

Luis Serrano

Normalization

Want coefficients to add to 1

$$\text{Orange} \longrightarrow \frac{1\ \text{Orange} + 0.71\ \text{Apple}}{1 + 0.71} = 0.58\ \text{Orange} + 0.42\ \text{Apple}$$

Need coefficients to be positive

$$\text{Orange} \longrightarrow \frac{1\ \text{Orange} - 1\ \text{Motorcycle}}{1 - 1} =$$

Normalization

Want coefficients to add to 1

$$\text{Orange} \longrightarrow \frac{1\ \text{Orange} + 0.71\ \text{Apple}}{1 + 0.71} = 0.58\ \text{Orange} + 0.42\ \text{Apple}$$

Need coefficients to be positive

! $$\text{Orange} \longrightarrow \frac{1\ \text{Orange} - 1\ \text{Motorcycle}}{1 - 1} = \text{✗}$$

Solution? $x \longrightarrow e^{x}$

Luis Serrano

Softmax

$$x \longrightarrow e^x$$

$$\text{Orange} \longrightarrow \frac{e^{1}\,\text{Orange} + e^{0.71}\,\text{Apple}}{e^{1} + e^{0.71}} = 0.58\ \text{Orange} + 0.42\ \text{Apple}$$

$$\text{Orange} \longrightarrow \frac{1\ \text{Orange} - 1\ \text{Motorcycle}}{1 - 1} =$$

Luis Serrano

Softmax

$$x \longrightarrow e^x$$

$$\text{Orange} \longrightarrow \frac{e^1\,\text{Orange} + e^{0.71}\,\text{Apple}}{e^1 + e^{0.71}} = 0.57\,\text{Orange} + 0.43\,\text{Apple}$$

$$\text{Orange} \longrightarrow \frac{1\,\text{Orange} - 1\,\text{Motorcycle}}{1 - 1} =$$

Luis Serrano

Softmax

$$x \longrightarrow e^{x}$$

$$\text{Orange} \longrightarrow \frac{e^{1}\,\text{Orange} + e^{0.71}\,\text{Apple}}{e^{1} + e^{0.71}} = 0.57\,\text{Orange} + 0.43\,\text{Apple}$$

$$\text{Orange} \longrightarrow \frac{e^{1}\,\text{Orange} + e^{-1}\,\text{Motorcycle}}{e^{1} + e^{-1}} = 0.88\,\text{Orange} + 0.12\,\text{Motorcycle}$$

an apple and an orange

	Orange	Apple	And	An
Orange	1	0.71	0	0
Apple	0.71	1	0	0
And	0	0	1	1
An	0	0	1	1

Orange $\longrightarrow$ 0.57 Orange + 0.43 Apple

Apple $\longrightarrow$ 0.43 Orange + 0.57 Apple

And $\longrightarrow$ 0.5 And + 0.5 An

An $\longrightarrow$ 0.5 An + 0.5 And

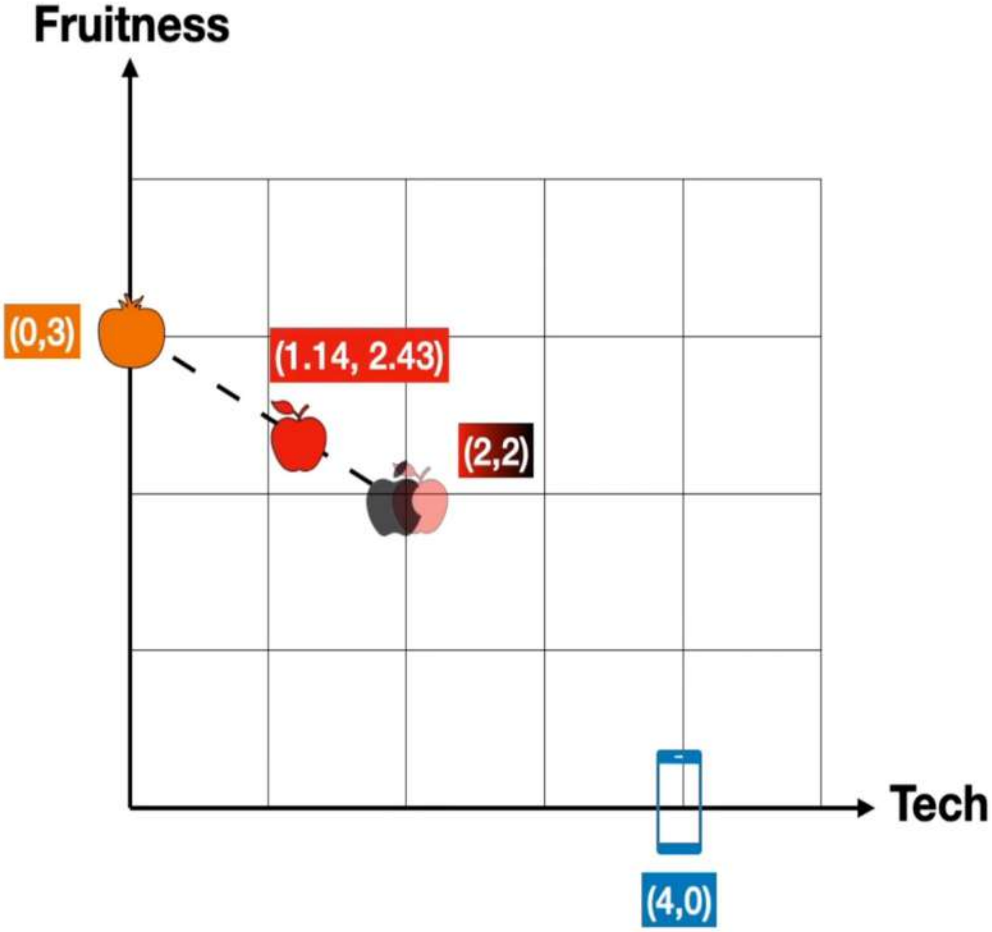
an apple phone

	Phone	Apple	An
Phone	1	0.71	0
Apple	0.71	1	0
An	0	0	1

Phone $\longrightarrow$ 0.57 Phone + 0.43 Apple

Apple $\longrightarrow$ 0.43 Phone + 0.57 Apple

An $\longrightarrow$ 1 An

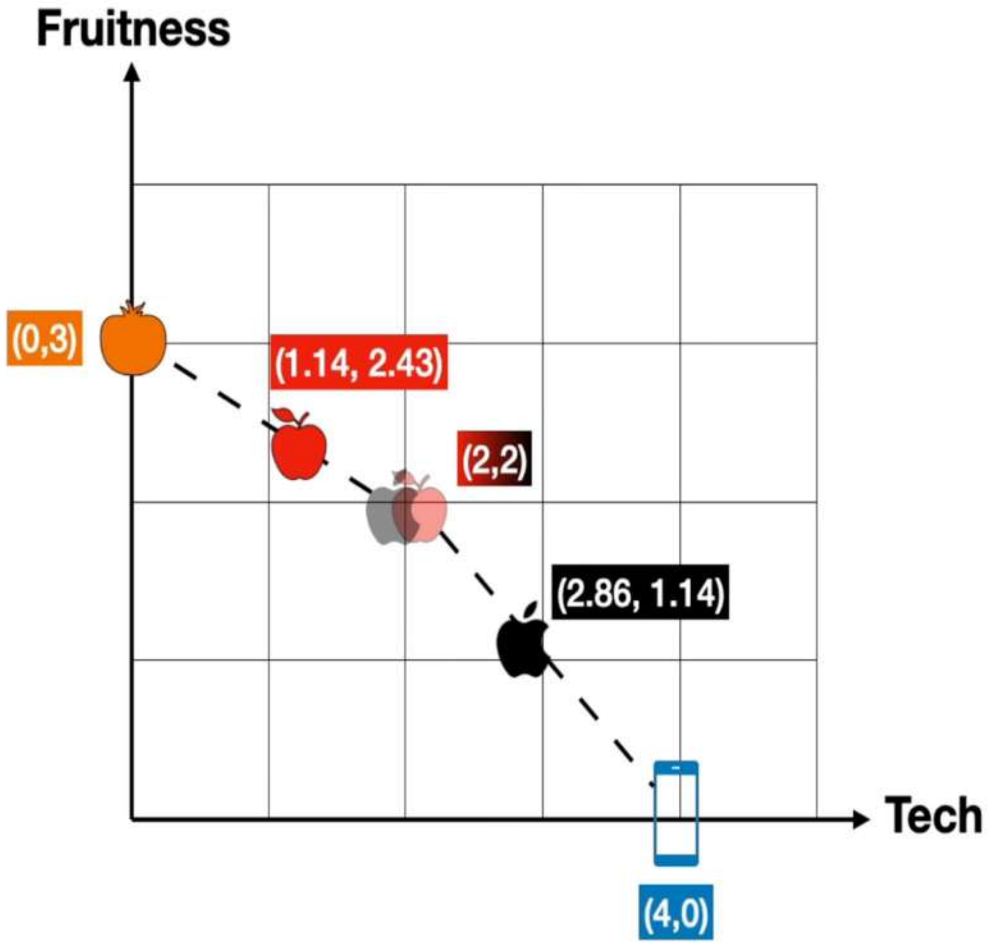


an apple and an orange

Apple $\longrightarrow$ 0.43 Orange + 0.57 Apple

an apple phone

Luis Serrano



an apple and an orange

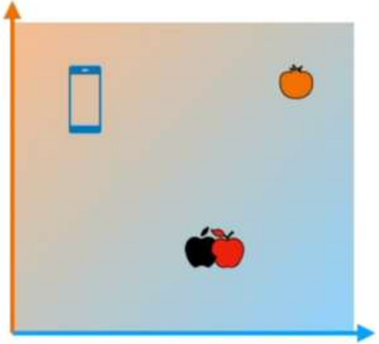
Apple $\longrightarrow$ 0.43 Orange + 0.57 Apple

an apple phone

Apple $\longrightarrow$ 0.43 Phone + 0.57 Apple

Luis Serrano

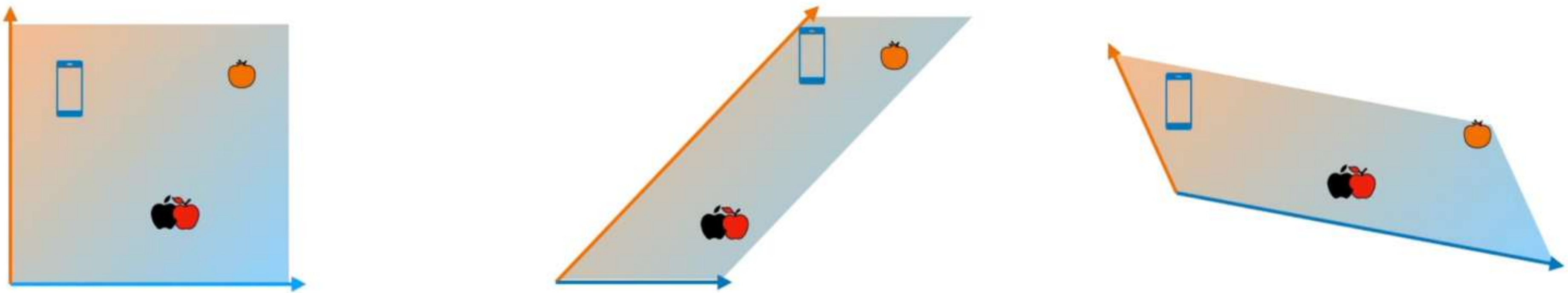
Get new embeddings from existing ones

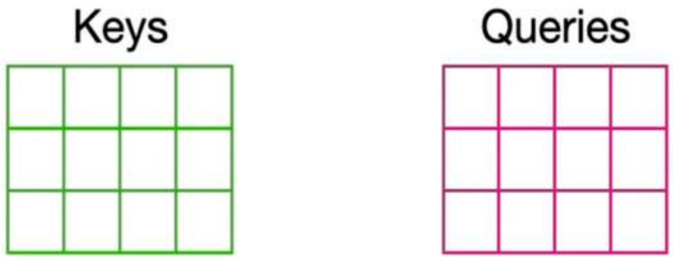


Keys Queries

Luis Serrano

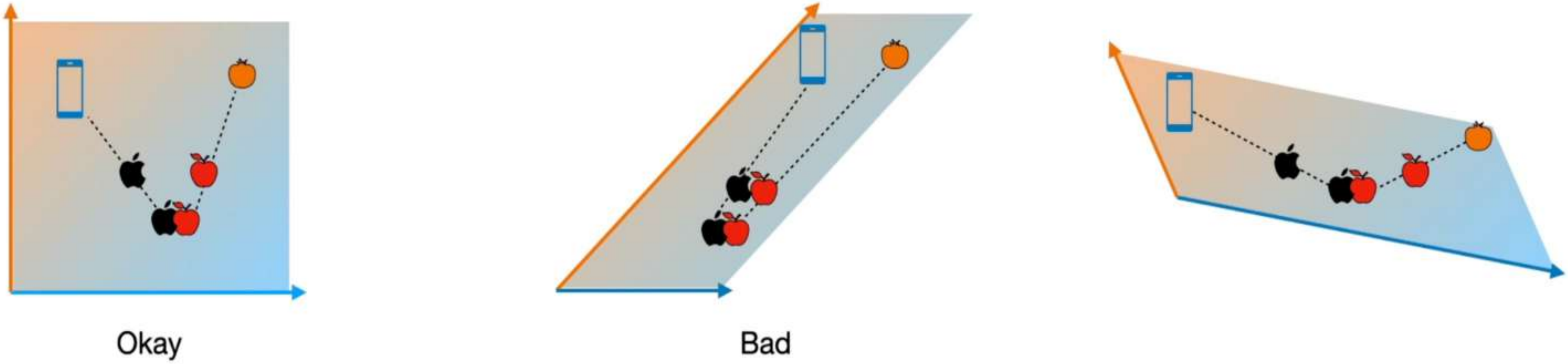
Get new embeddings from existing ones

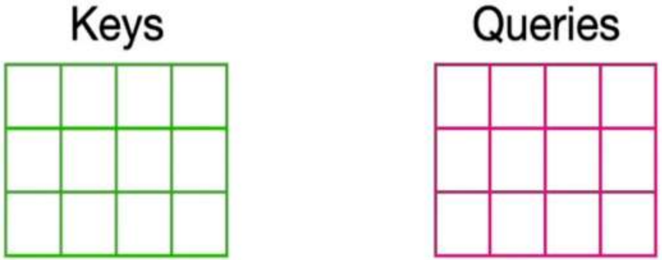




Luis Serrano

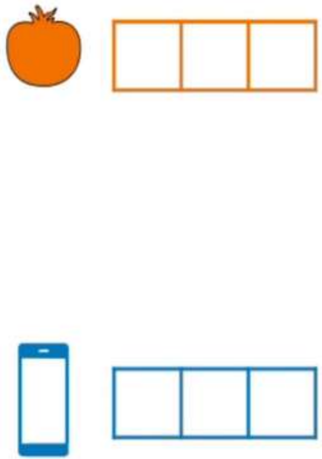
Get new embeddings from existing ones

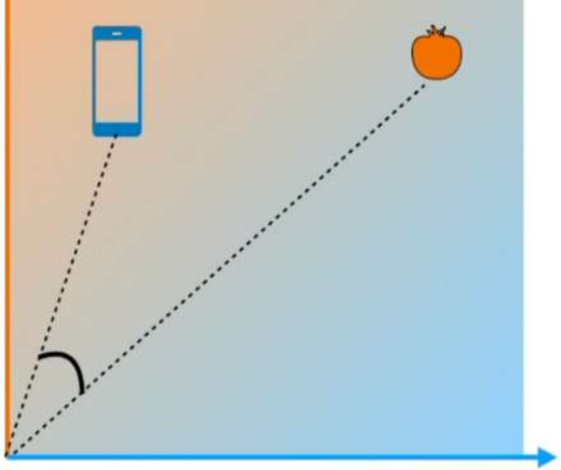


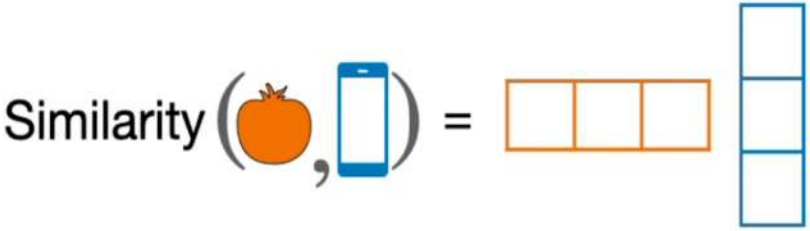


Luis Serrano

Similarity

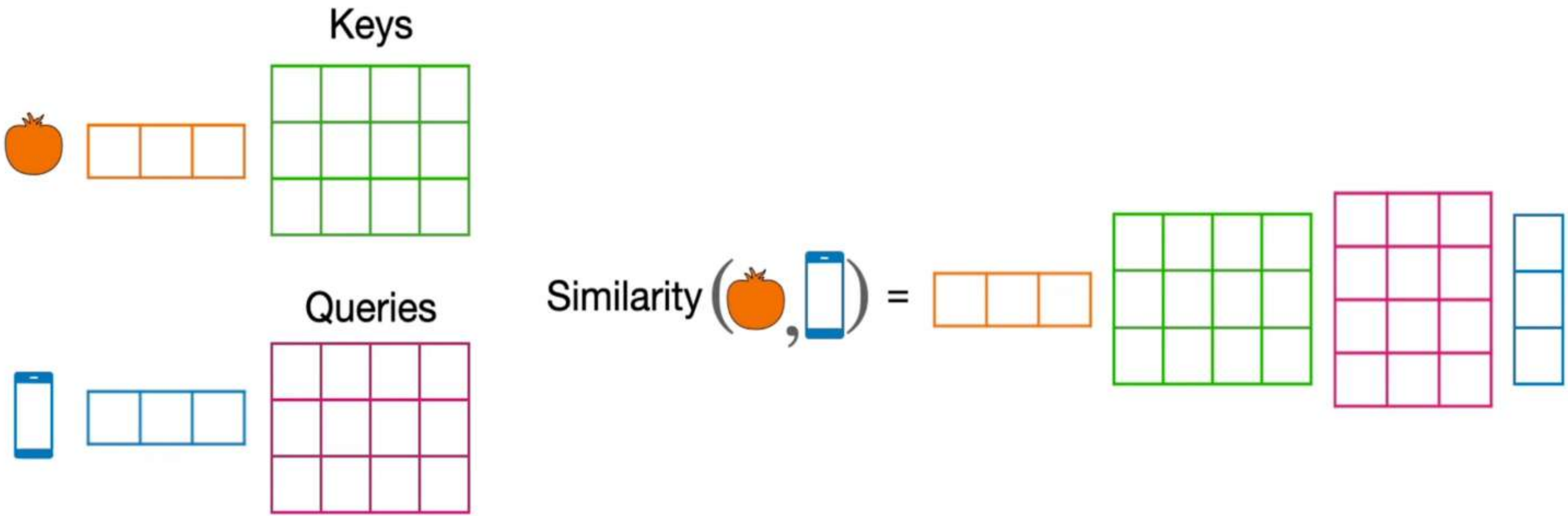






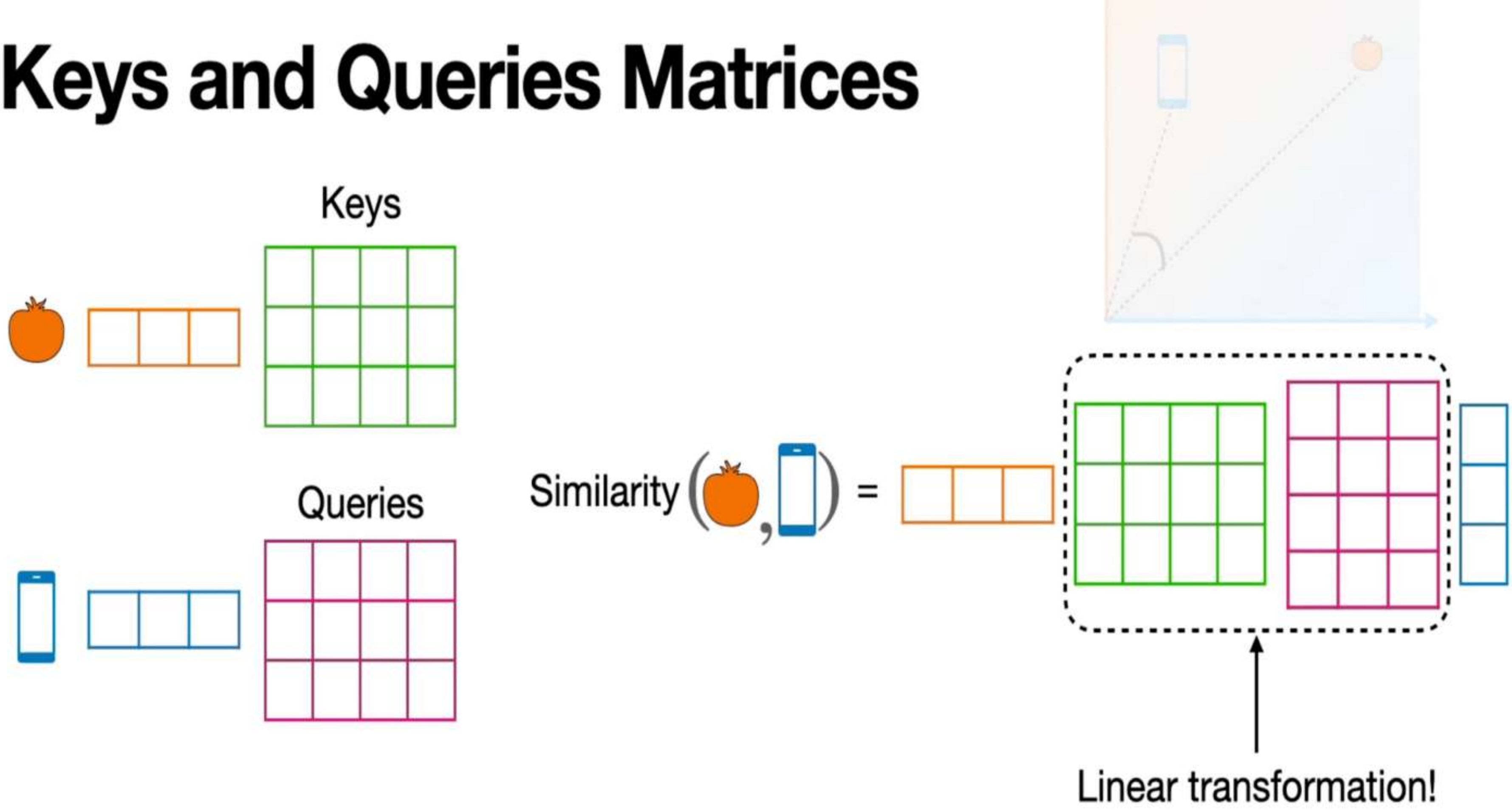
Luis Serrano

Keys and Queries Matrices



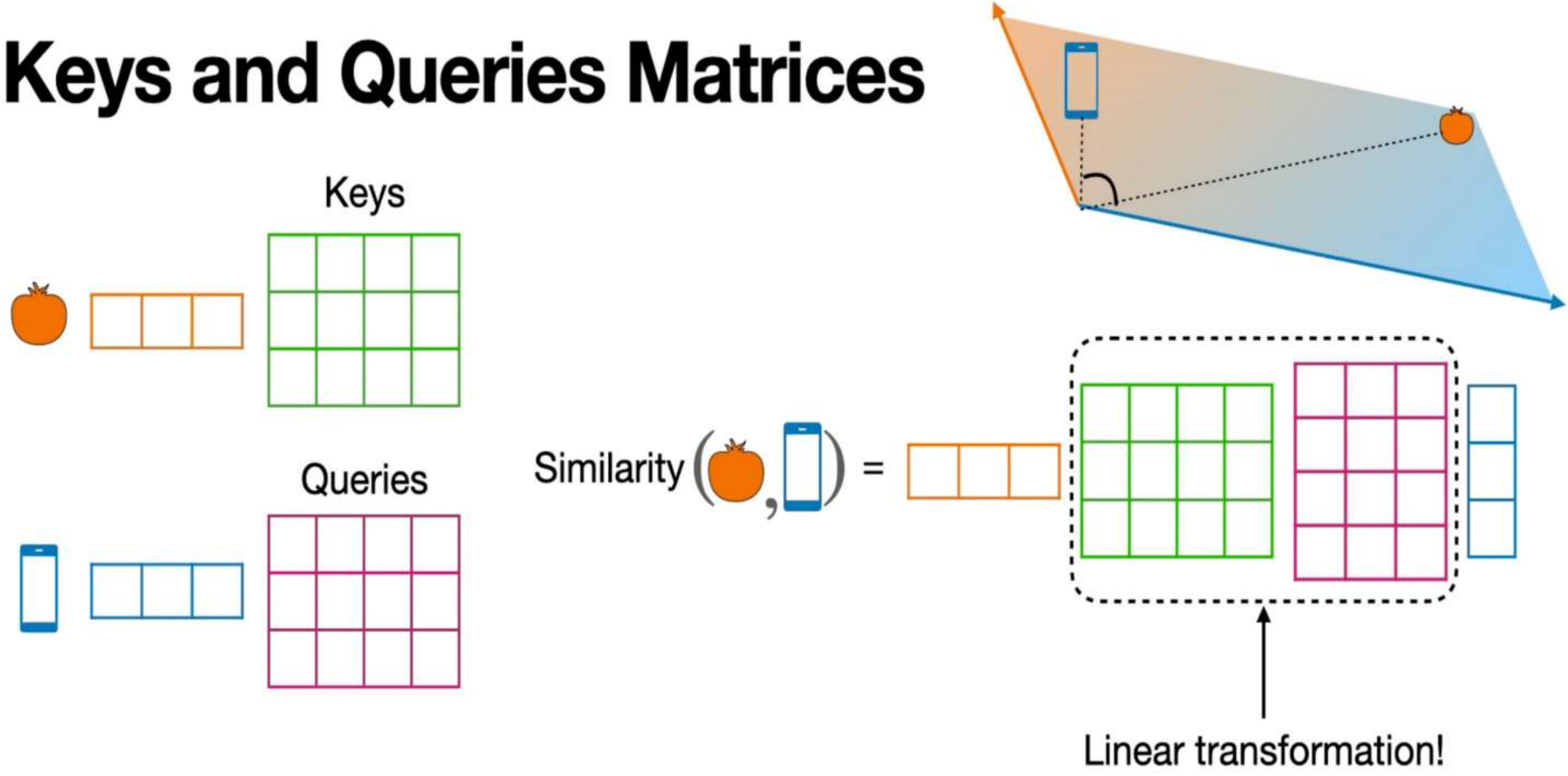
Luis Serrano

Keys and Queries Matrices



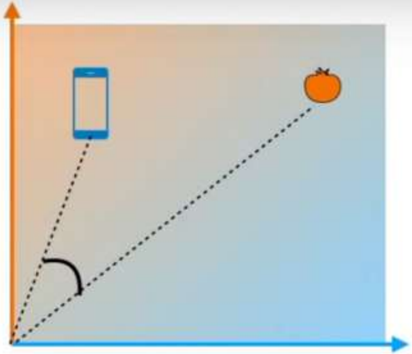
Luis Serrano

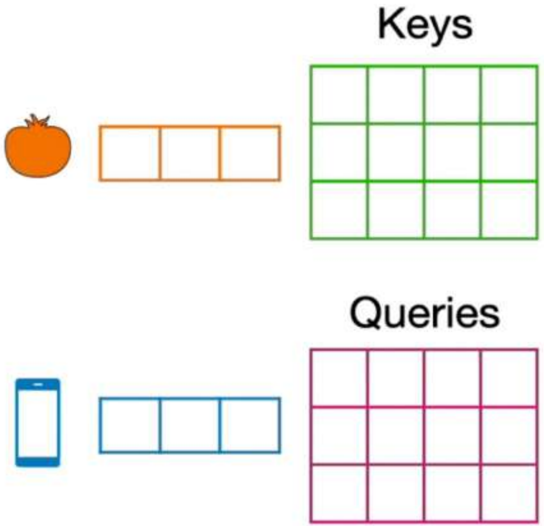
Keys and Queries Matrices

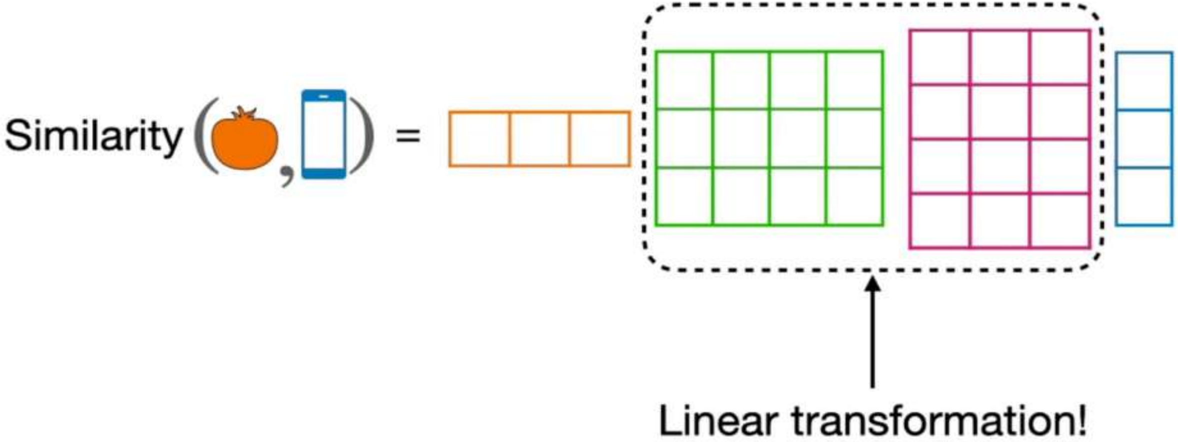


Luis Serrano

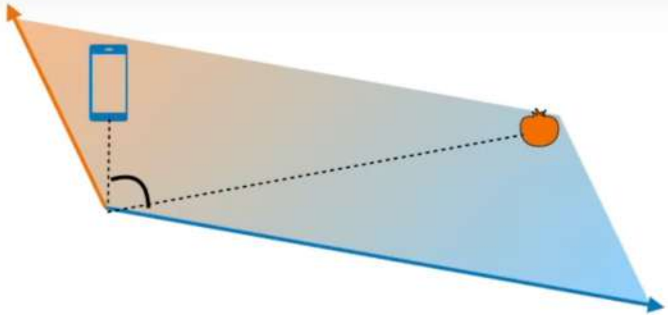
Keys and Queries Matrices

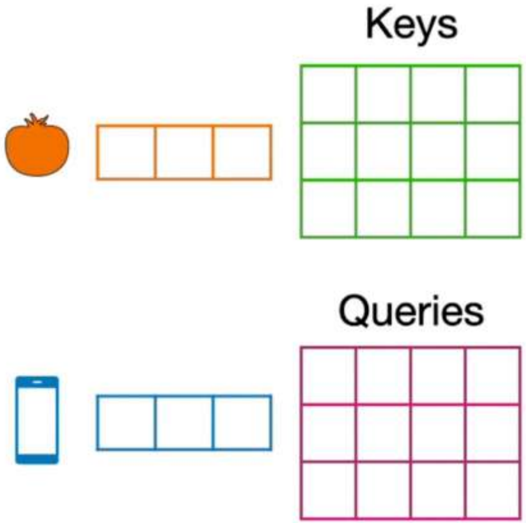


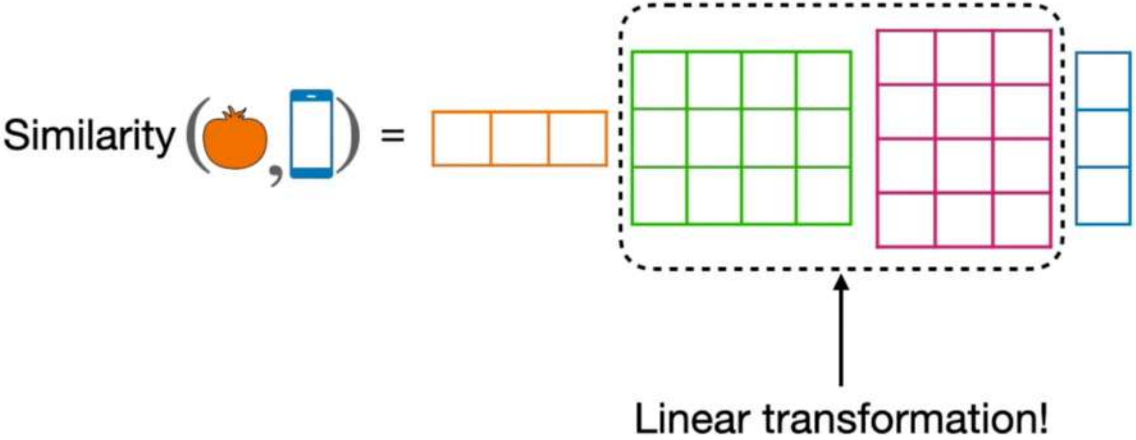




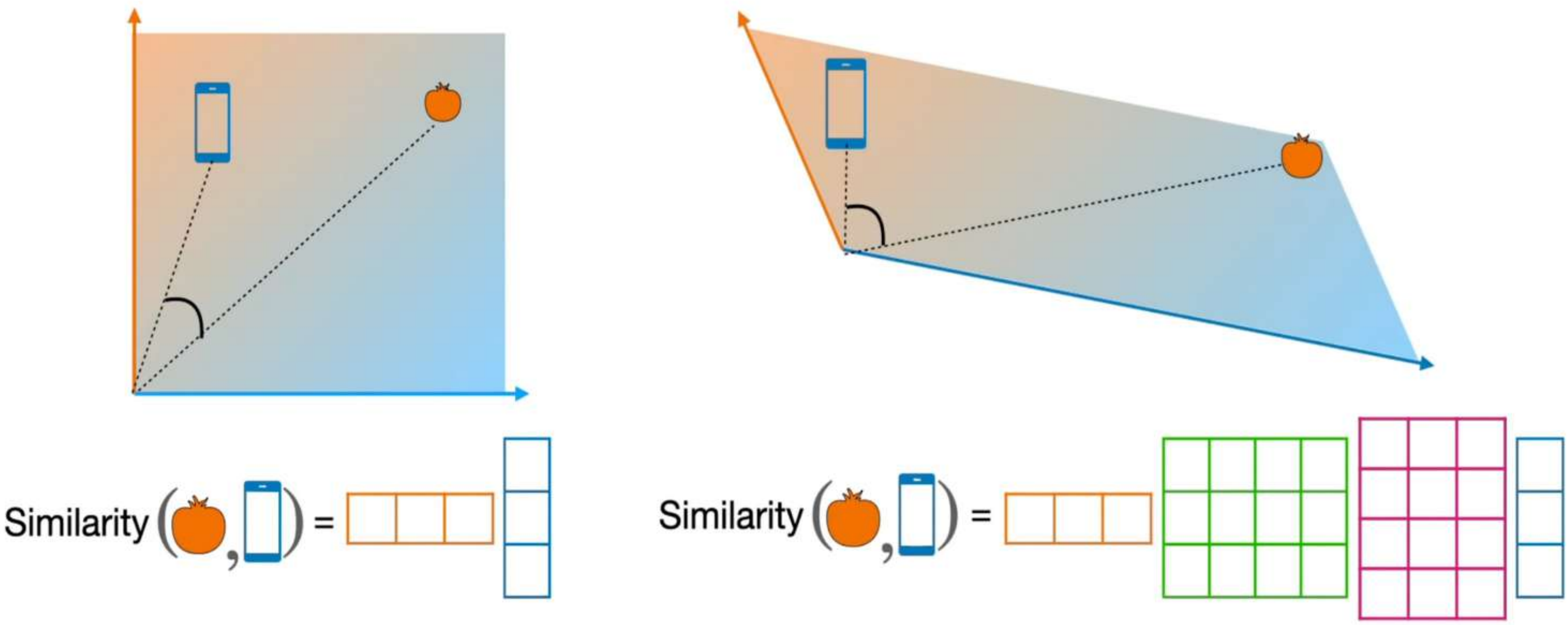
Keys and Queries Matrices





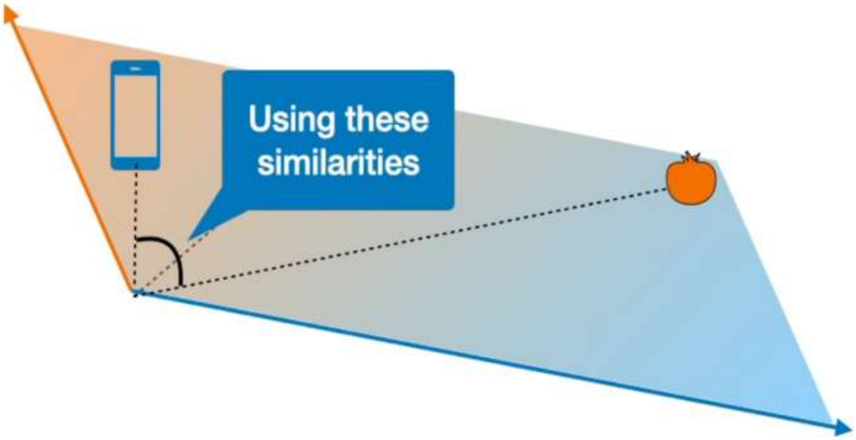


Similarity on a transformed embedding



Luis Serrano

Values matrix

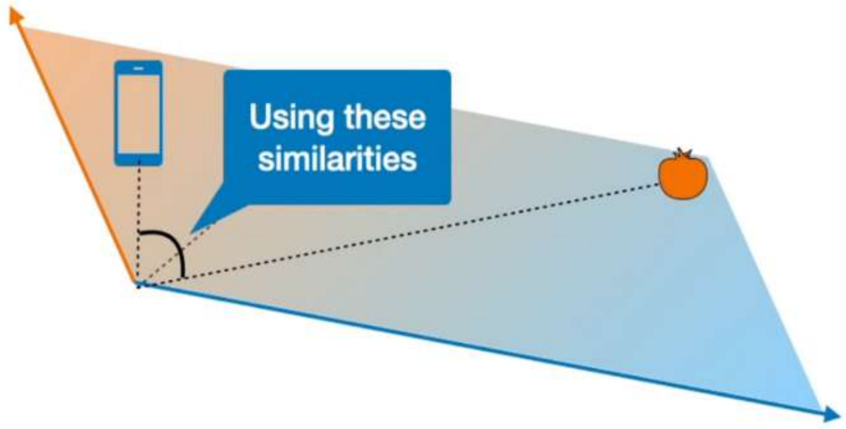


Best embedding for finding similarities

Best embedding for finding the next word

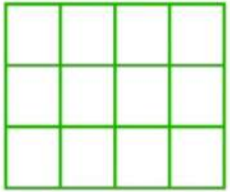
Luis Serrano

Values matrix

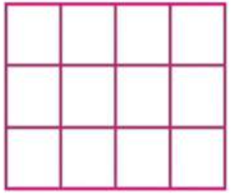


Best embedding for finding similarities

Keys

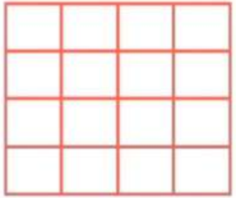


Queries

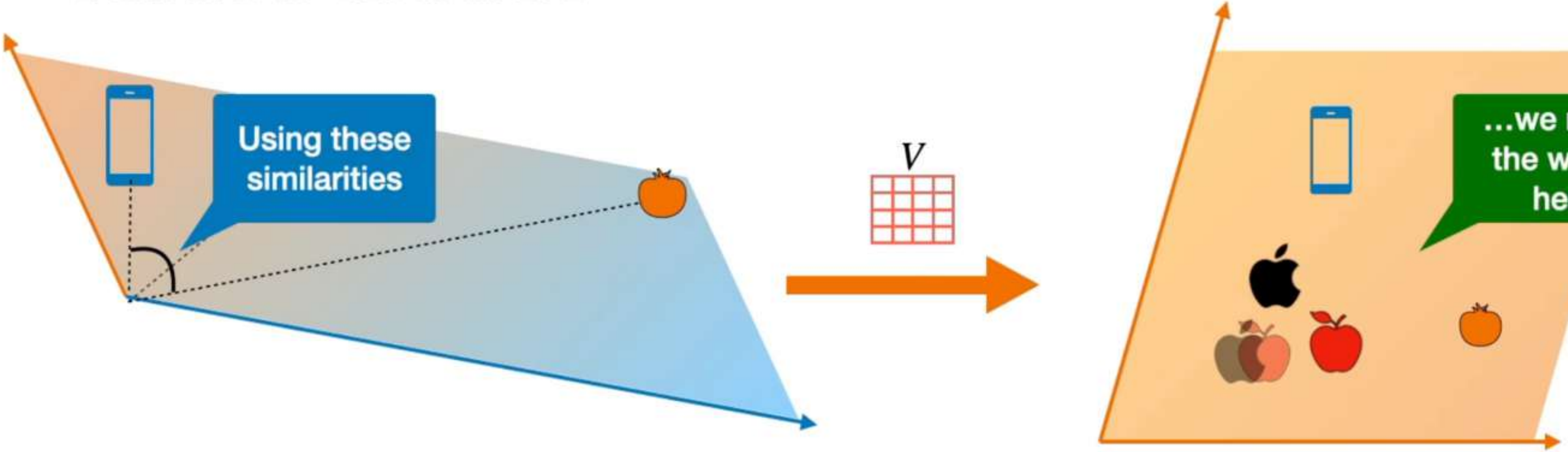


Best embedding for finding the next word

Values



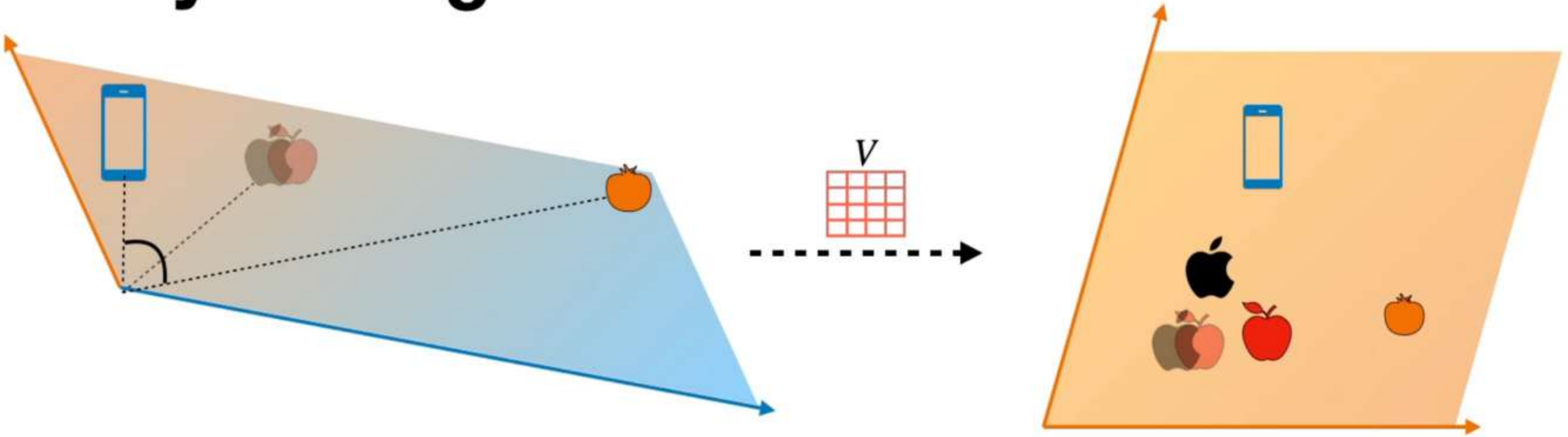
Values matrix



Best embedding for finding similarities

Best embedding for finding the next word

Why moving words on a different embedding?



Best embedding for finding similarities

Best embedding for finding the next word

This embedding(s) know features of the words

- Color
- Size
- Fruitness
- Technology

This embedding knows when two words could appear in the same context

I want to buy a _____

- car
- apple
- phone

Value matrix

an apple and an orange

	Orange	Apple	And	An
Orange	0.4	0.3	0.15	0.15
Apple	0.3	0.4	0.15	0.15
And	0.15	0.15	0.5	0.5
An	0.15	0.15	0.5	0.5

Value matrix

=

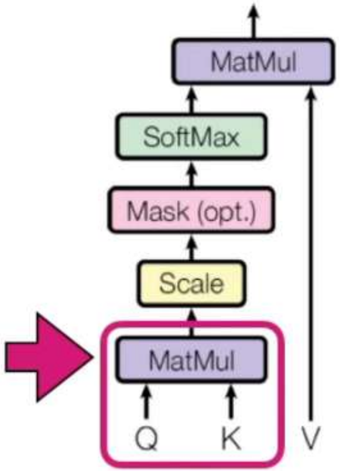
	Orange	Apple	And	An
Orange	v_{11}	v_{12}	v_{13}	v_{14}
Apple	v_{21}	v_{22}	v_{23}	v_{24}
And	v_{31}	v_{32}	v_{33}	v_{34}
An	v_{41}	v_{42}	v_{43}	v_{44}

$$\text{apple} \longrightarrow 0.3 \cdot \text{orange} + 0.4 \cdot \text{apple} + 0.15 \cdot \text{and} + 0.15 \cdot \text{an}$$

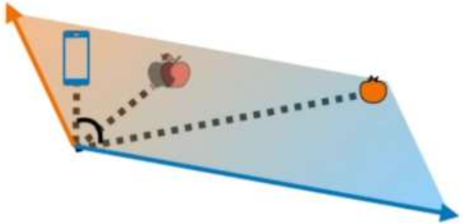
$$\text{apple} \longrightarrow v_{21} \cdot \text{orange} + v_{22} \cdot \text{apple} + v_{23} \cdot \text{and} + v_{24} \cdot \text{an}$$

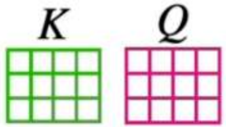
Self-attention

Scaled Dot-Product Attention

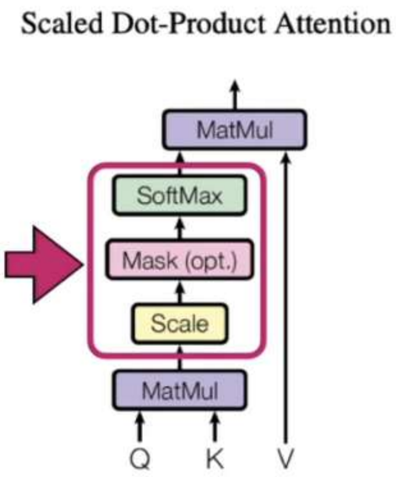


$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\boxed{\frac{QK^T}{\sqrt{d_k}}}\right)V$$

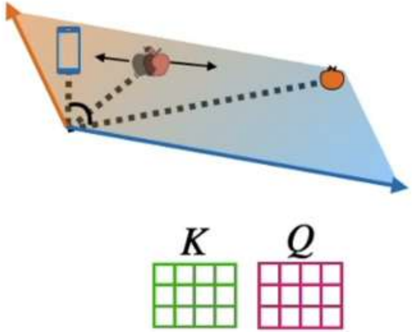




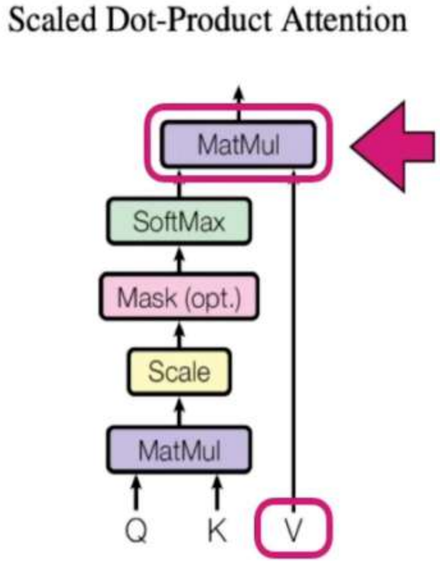
Self-attention



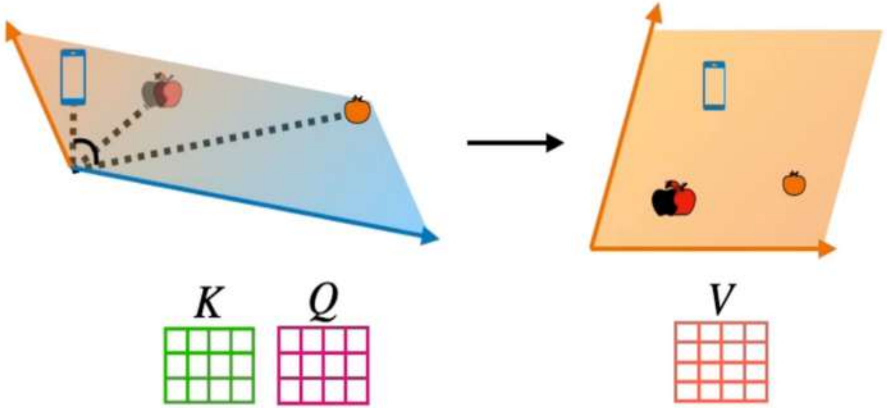
$$\text{Attention}(Q, K, V) = \boxed{\text{softmax}}(\frac{QK^T}{\sqrt{d_k}})V$$



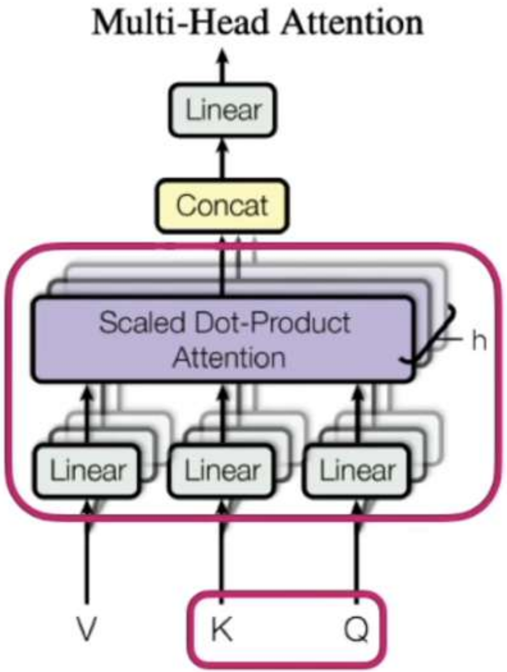
Self-attention



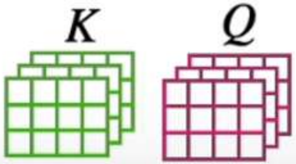
$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})\boxed{V}$$



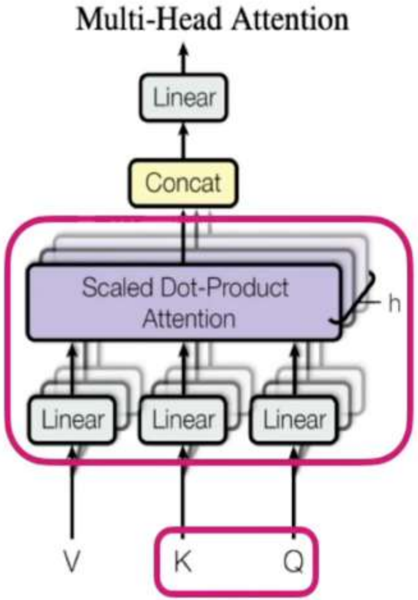
Multi-head attention



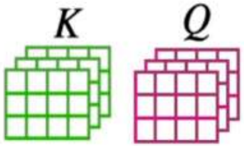
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

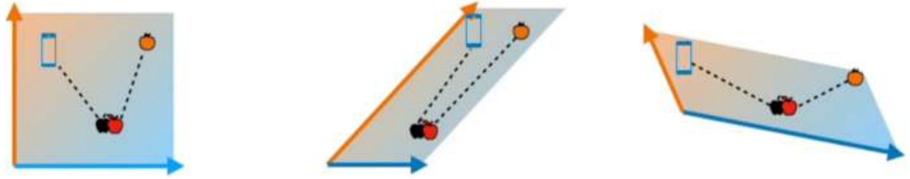


Multi-head attention

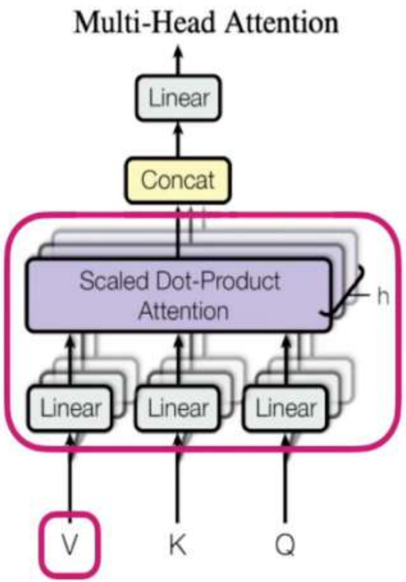


$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

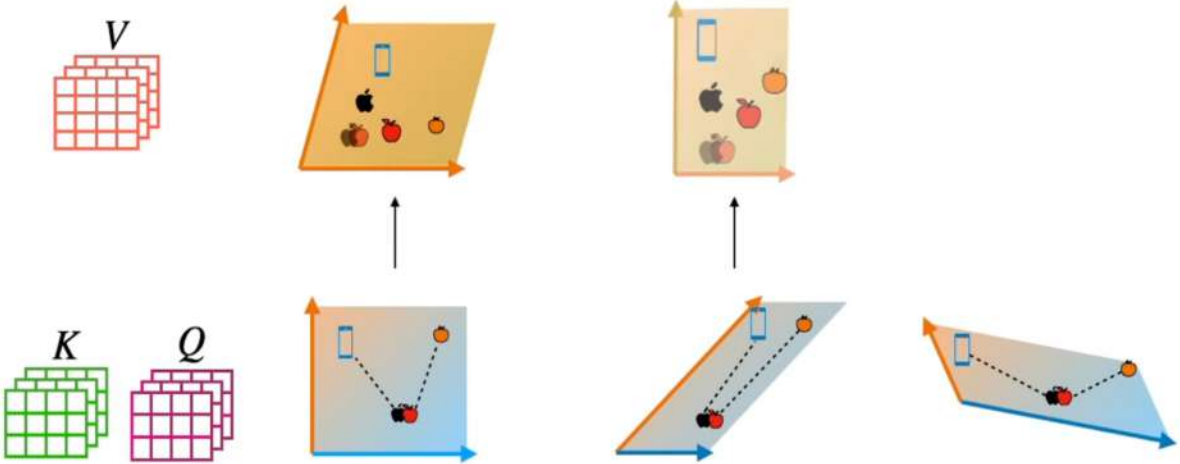




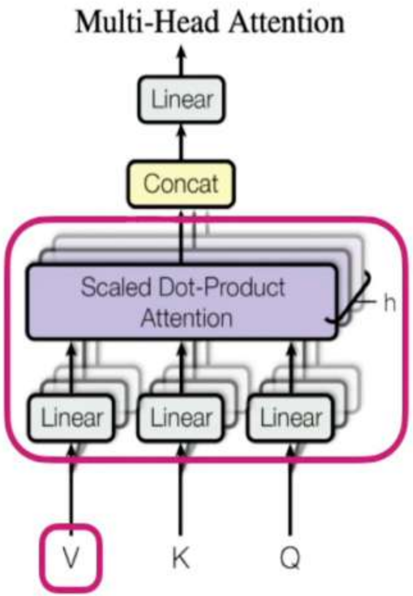
Multi-head attention



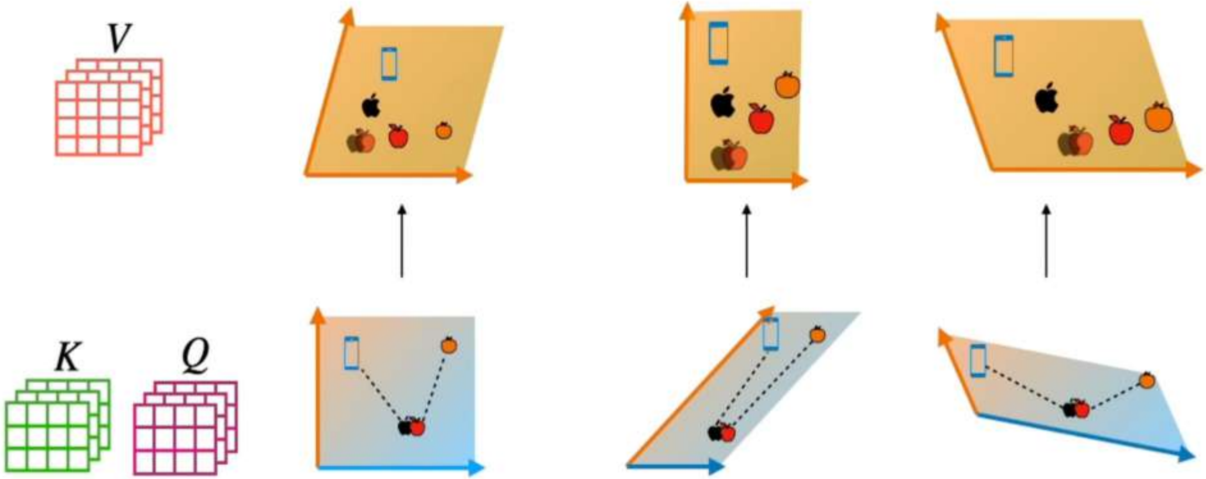
$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



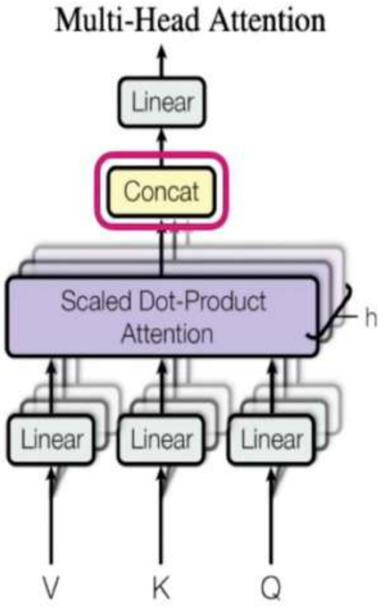
Multi-head attention



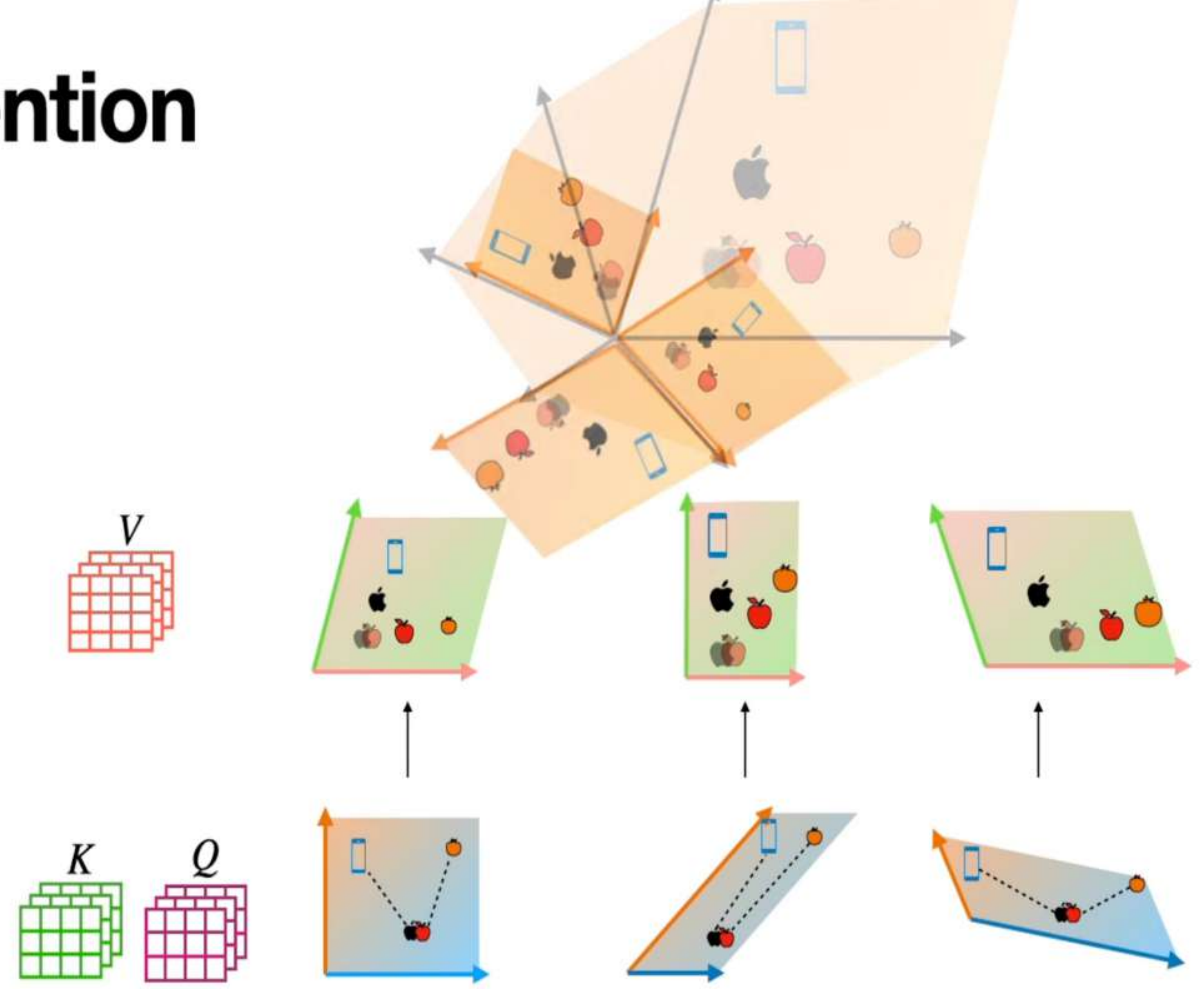
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



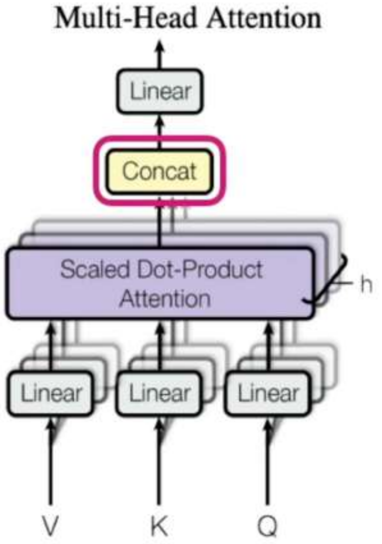
Multi-head attention



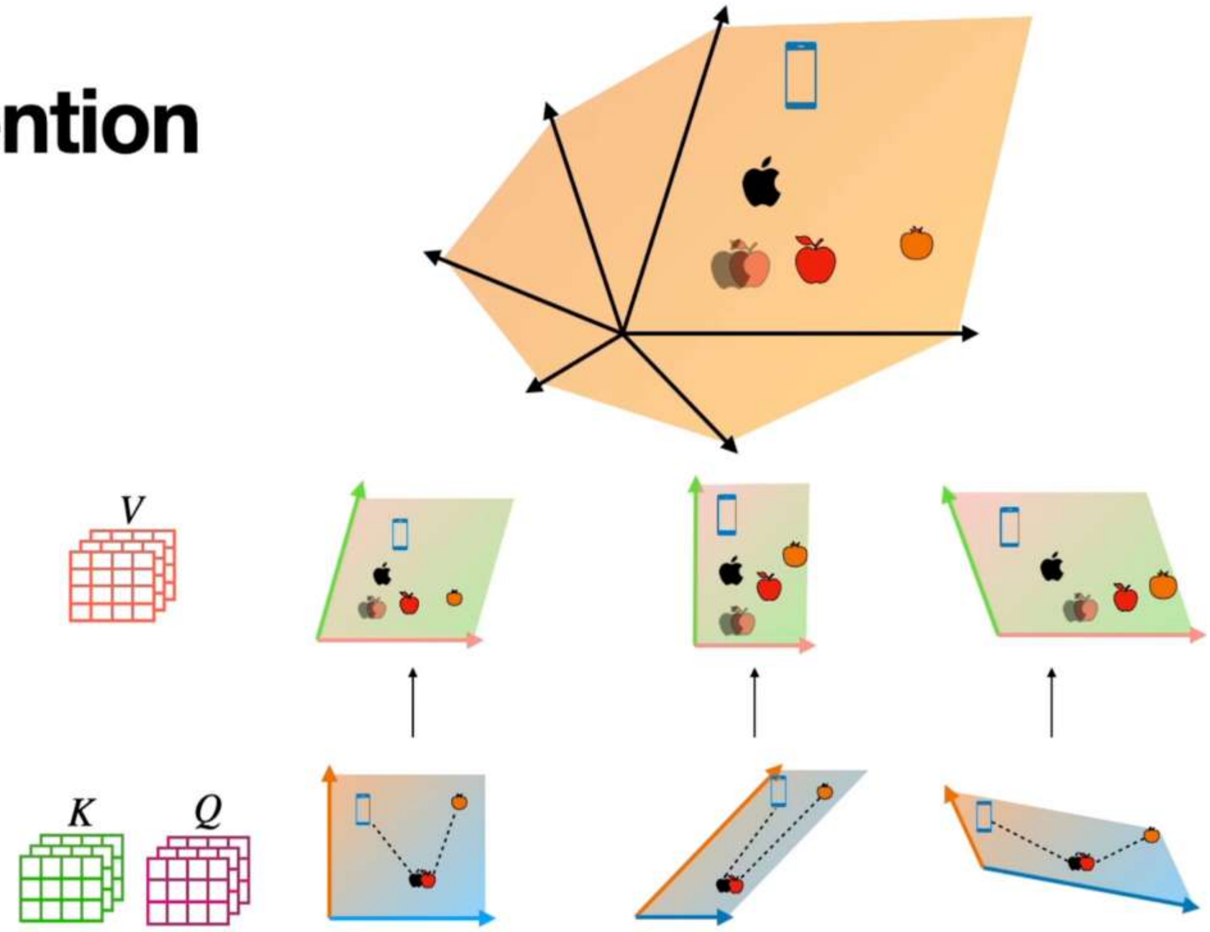
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



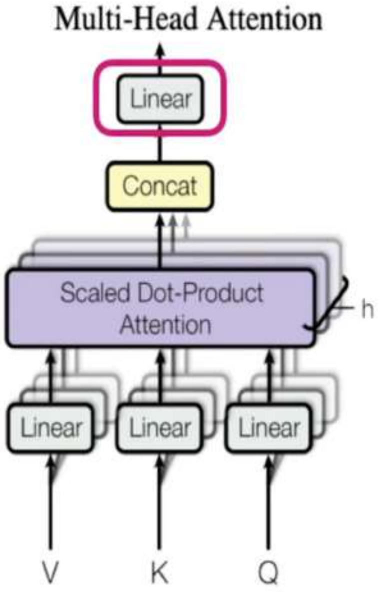
Multi-head attention



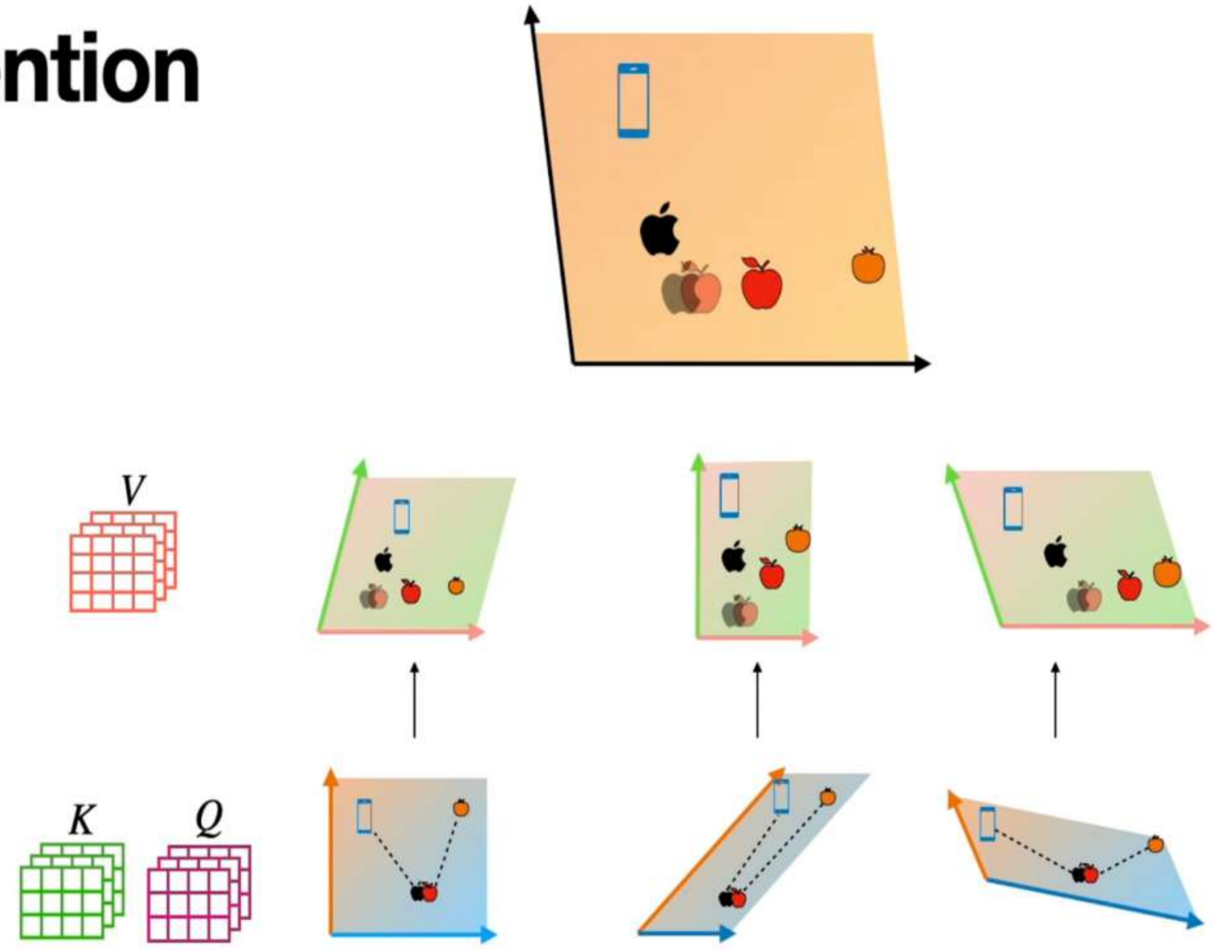
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



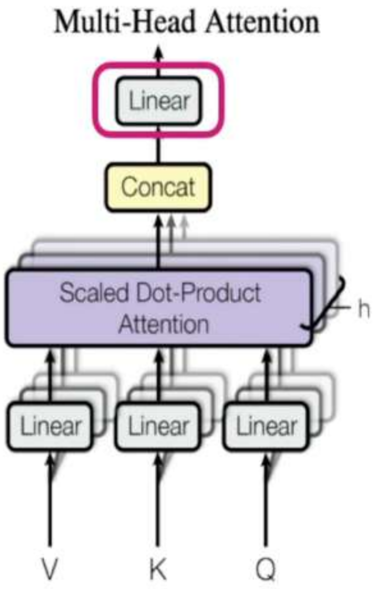
Multi-head attention



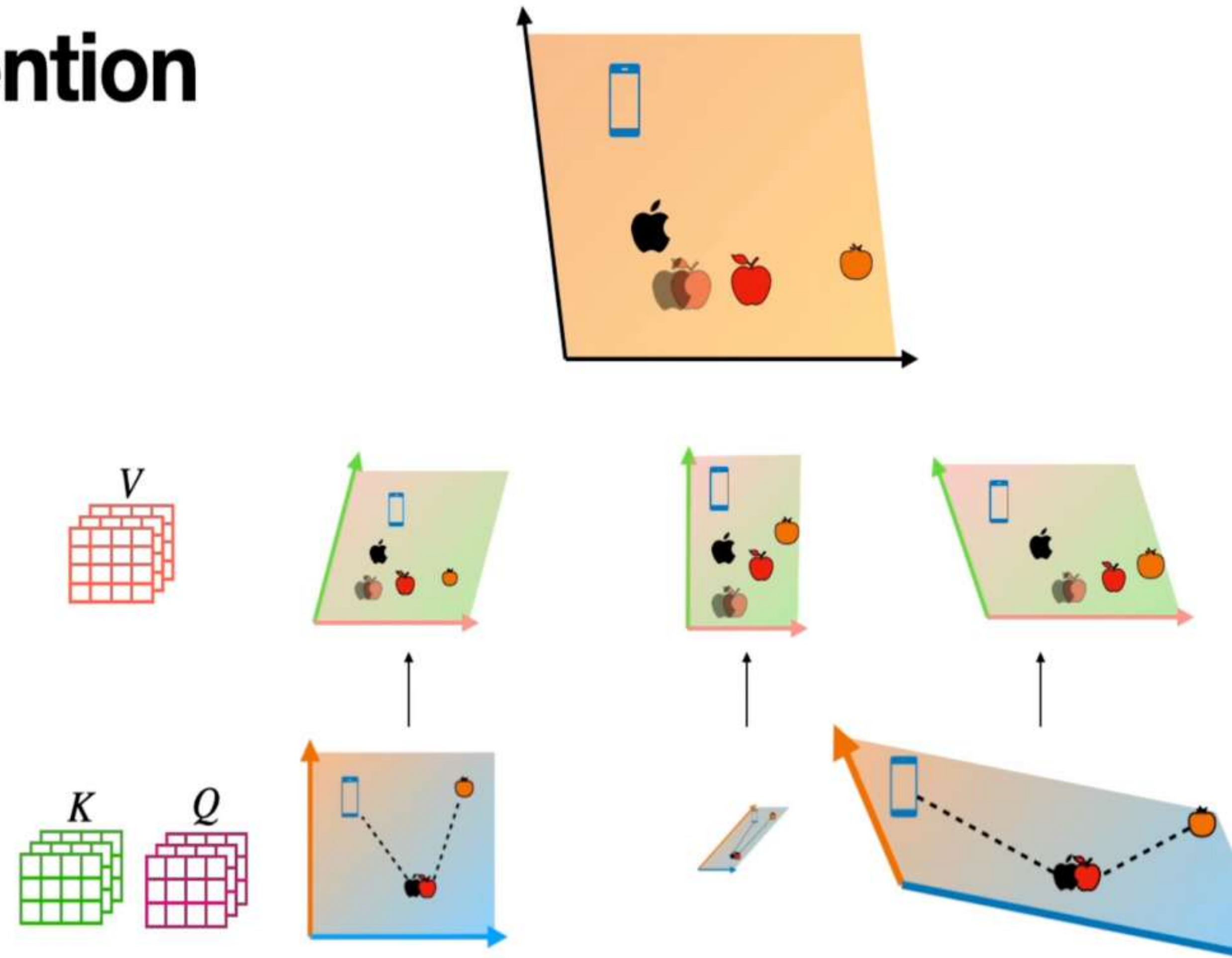
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_\mathrm{h})W^O$$
$$\text{where } \mathrm{head}_\mathrm{i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



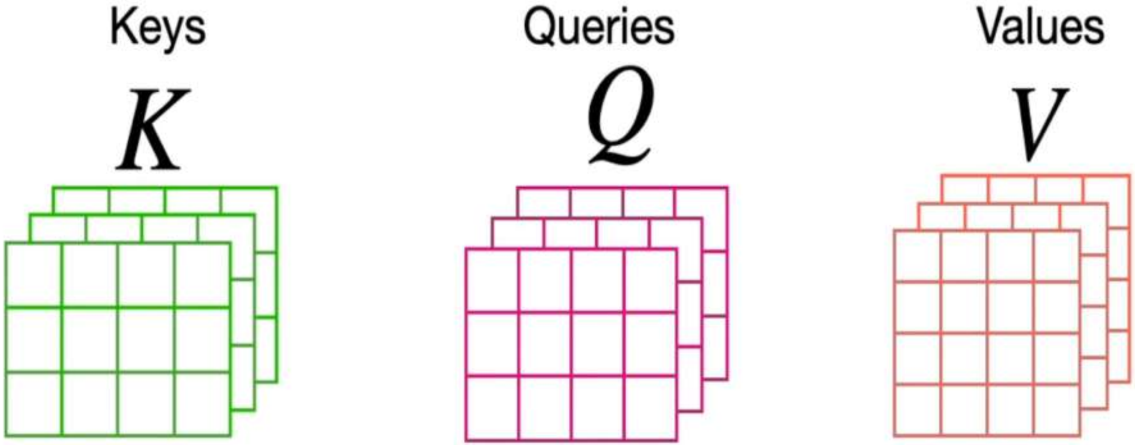
Multi-head attention



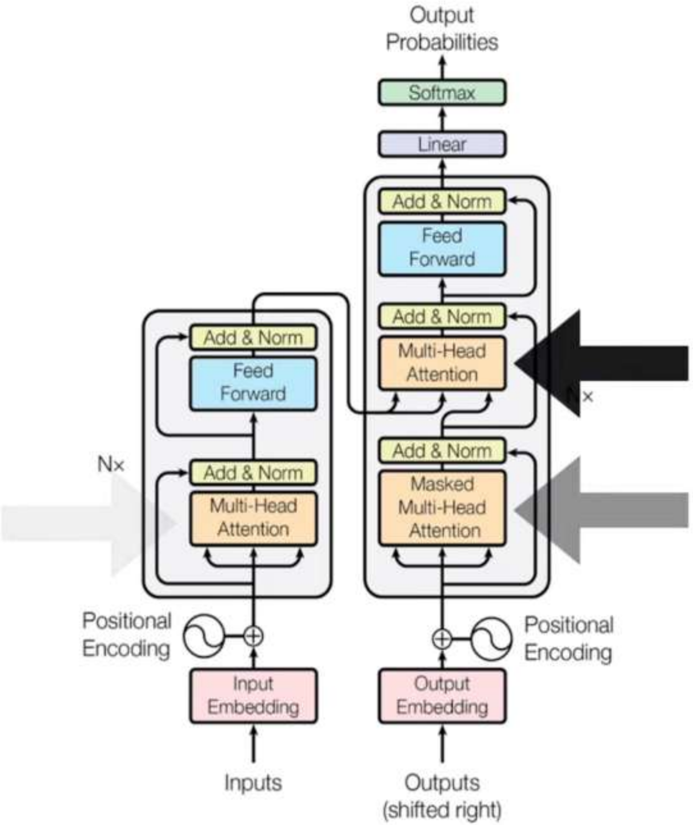
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



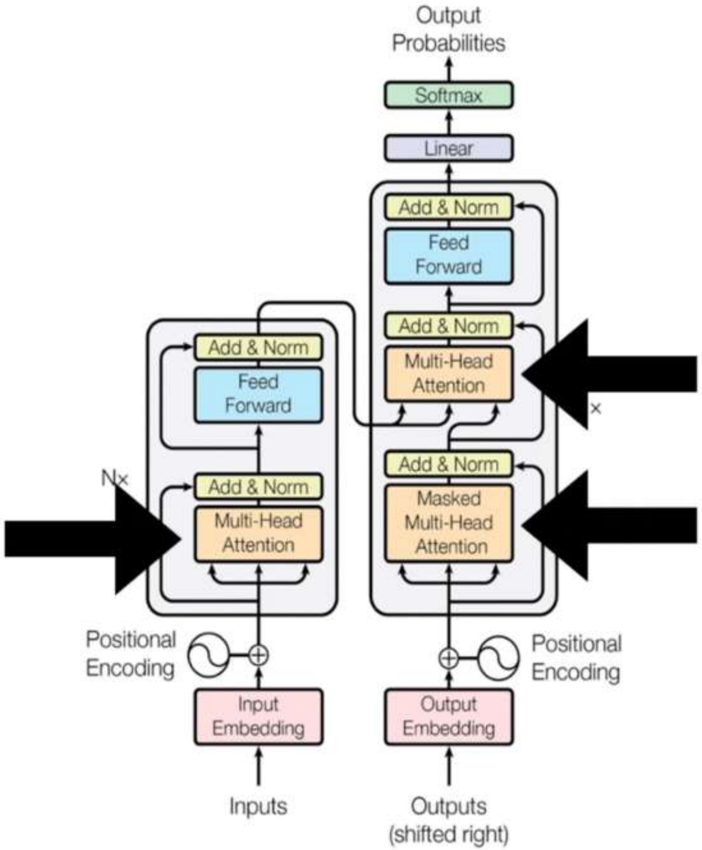
How to get these matrices?



Weights get trained with the transformer model

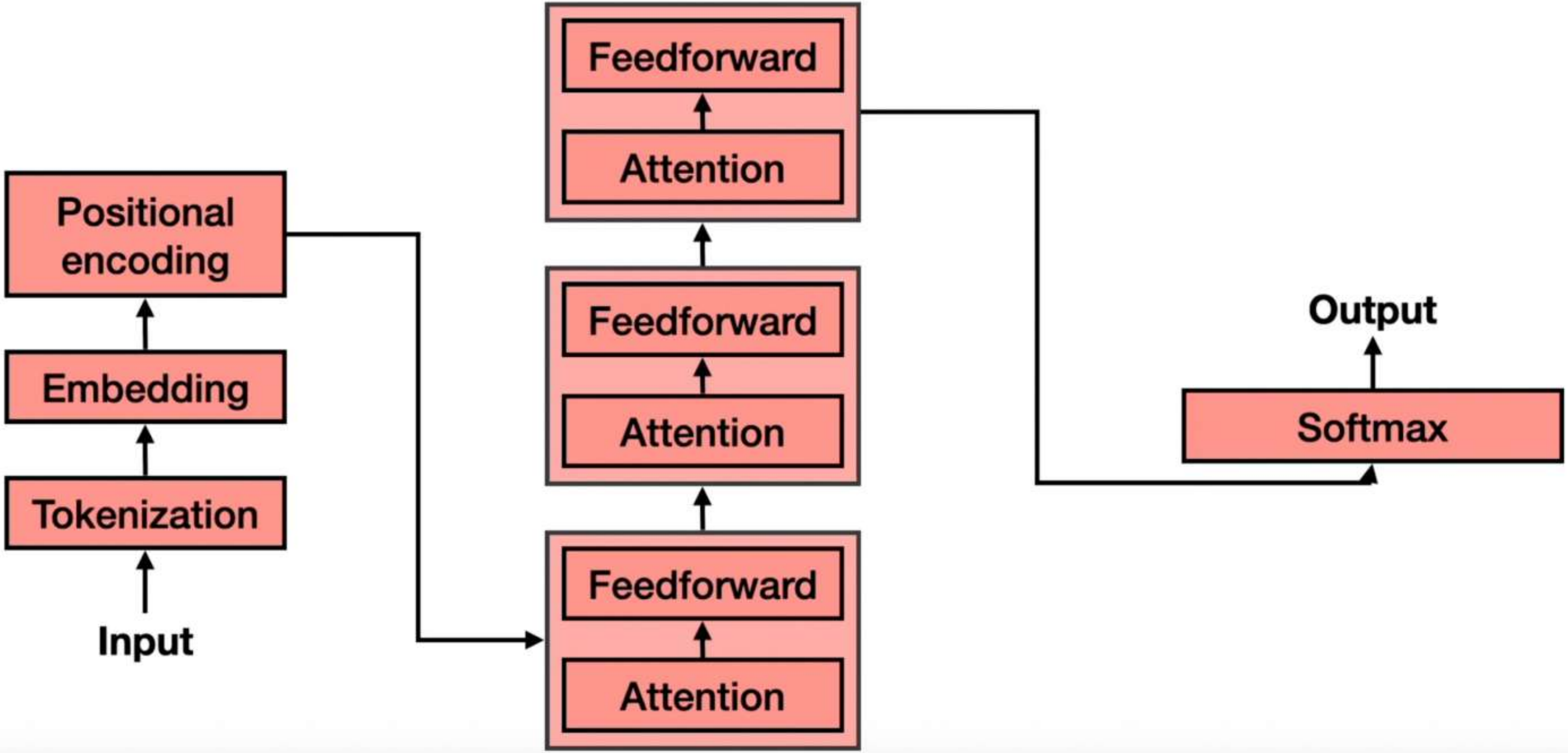


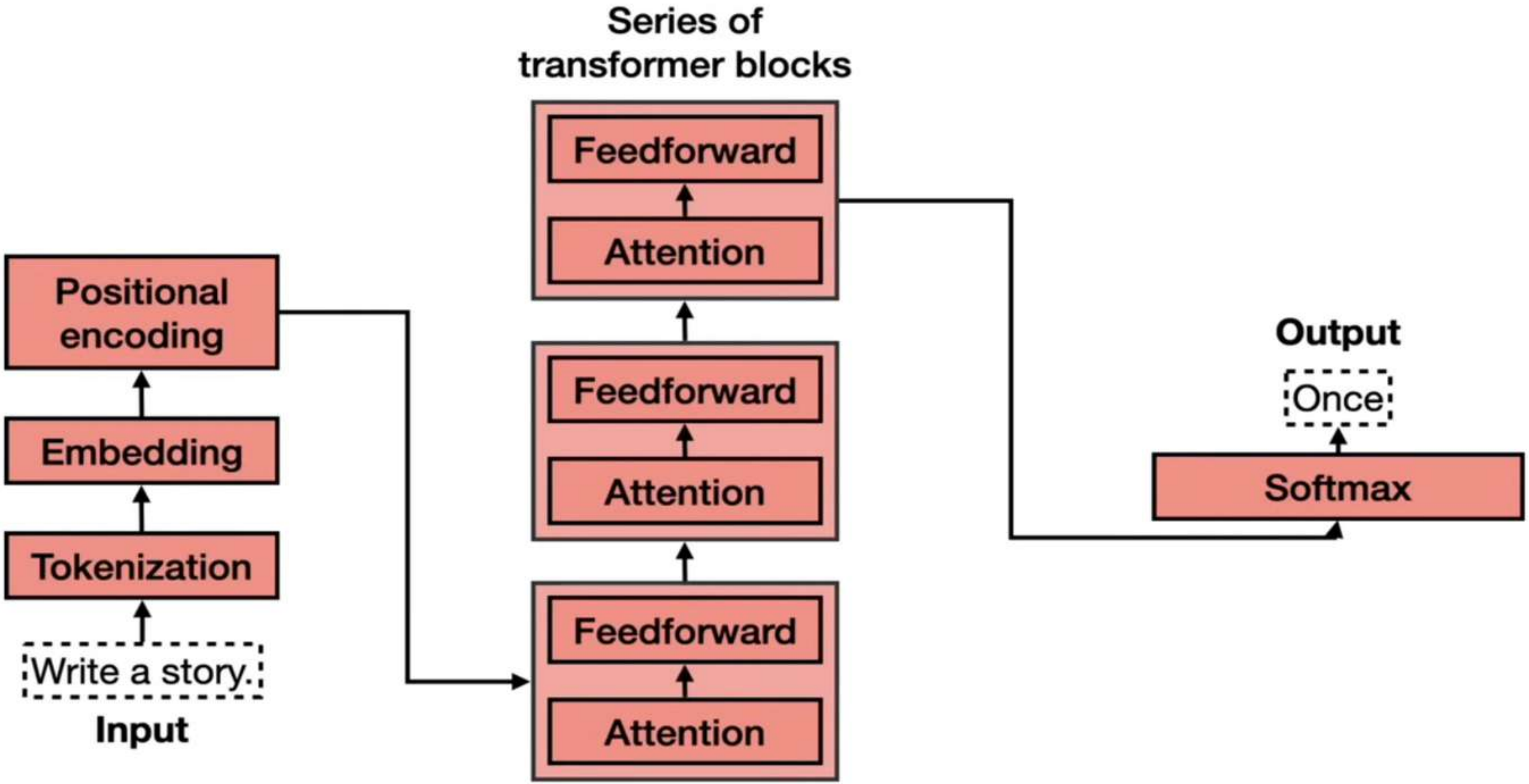
Weights get trained with the transformer model



Transformeurs

Weights get trained with the transformer model




Series of
transformer blocks
Feedforward
Attention
Feedforward
Attention
Feedforward
Attention
Positional
encoding
Embedding
Tokenization
Write a story.
Input
Output
Once
Softmax

Transformeurs

- 1) Tokenization
- 2) Encastrement
- 3) Encodage positionnel
- 4) Mécanismes d'attention
- 5) Softmax vers la sortie
- 6) Rince et répète *pour chaque mot*

1) Tokenization

- Word2vec permet de trouver des encastrements au niveau des mots. D'autres algorithmes d'intégration les trouvent à un niveau plus fin, pour les segments de mots, la ponctuation, etc. (cette méthode est plus générale, il est difficile de traiter la ponctuation par le biais des intégrations de mots).

1) Tokenization

- Aujourd'hui, les principaux modèles utilisent des enchâssements de jetons, par exemple, « doesn’t » sera composé de deux jetons, « does » et « n’t ».
- «.» sera son propre jeton, etc

2) Encastrement

- Nous commençons par des représentations simples de chaque token, par exemple des vecteurs «one-hot» (un vecteur de la forme [0,0,0,0,1,0,0,00], avec un emplacement pour chaque token du vocabulaire).

- Nous appliquons ensuite un algorithme d’encastrement (déjà entraîné) pour rappeler l’encastrement de ce jeton.

- Nous allons maintenant commencer à modifier l’encastrement de défaut pour gérer le contexte

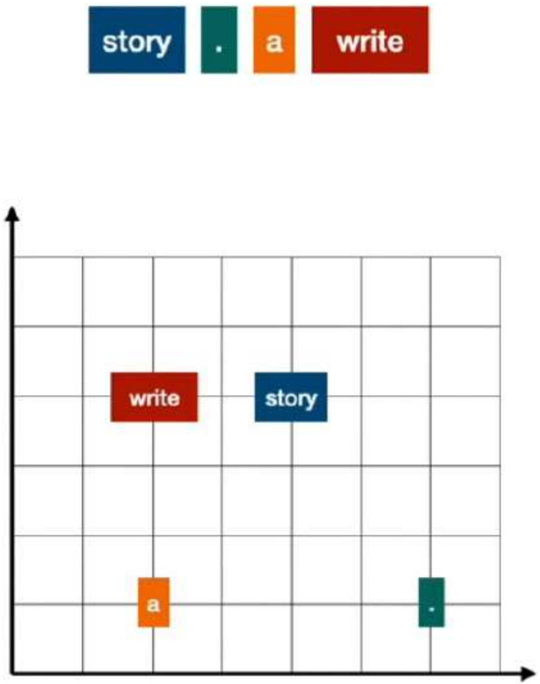
3) Encodage positionnel

- Tout d'abord, nous devons saisir le fait que l'ordre a de l'importance.
- Les réseaux neuronaux récurrents disposent d'un mécanisme pour ce faire (mais il s'est avéré moins efficace pour les séquences plus longues)
- Les transformateurs utilisent le codage positionnel : en fait, ils indexent chaque élément de la séquence avec un numéro, par exemple 1) j’ 2) ai 3) faim, qui sera différent de 1) faim 2) ai 3) je

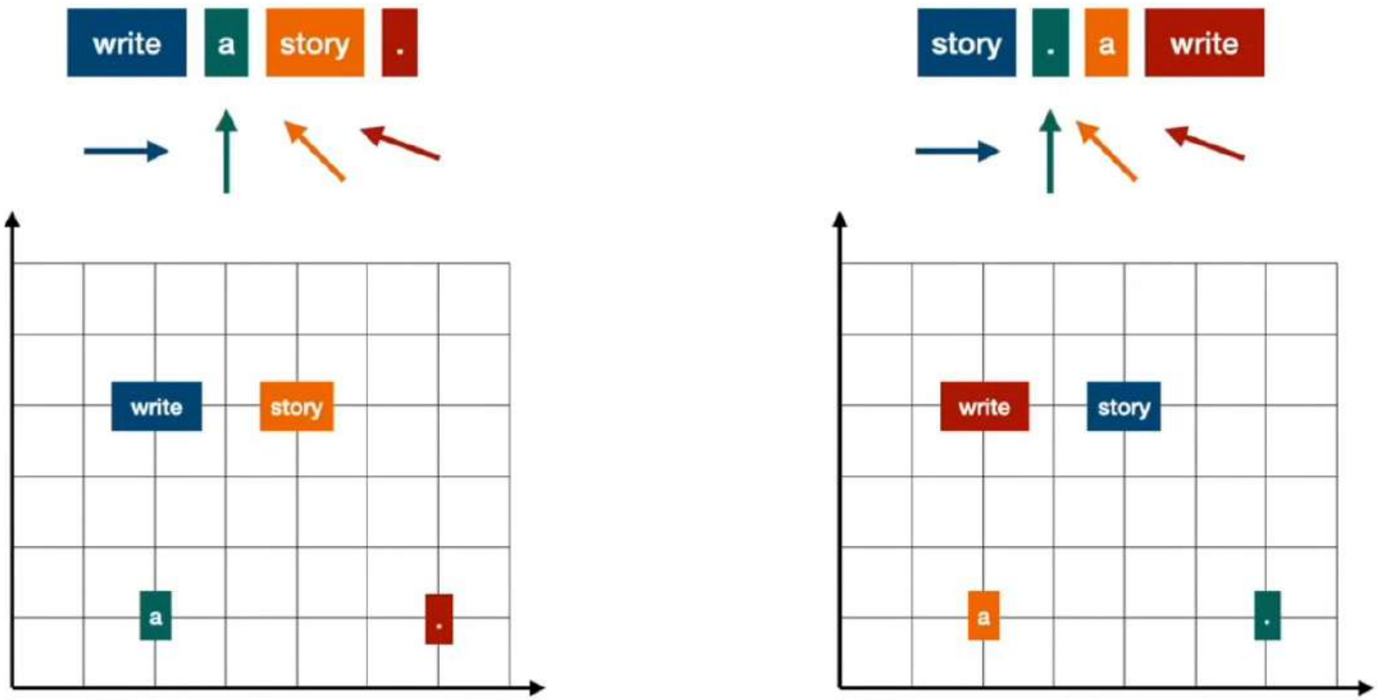
3) Encodage positionnel

- En fait, tu le fais par une première modification de l'intégration : au lieu d'un index avec des nombres, il s'agit d'un index avec des directions.

Positional encoding

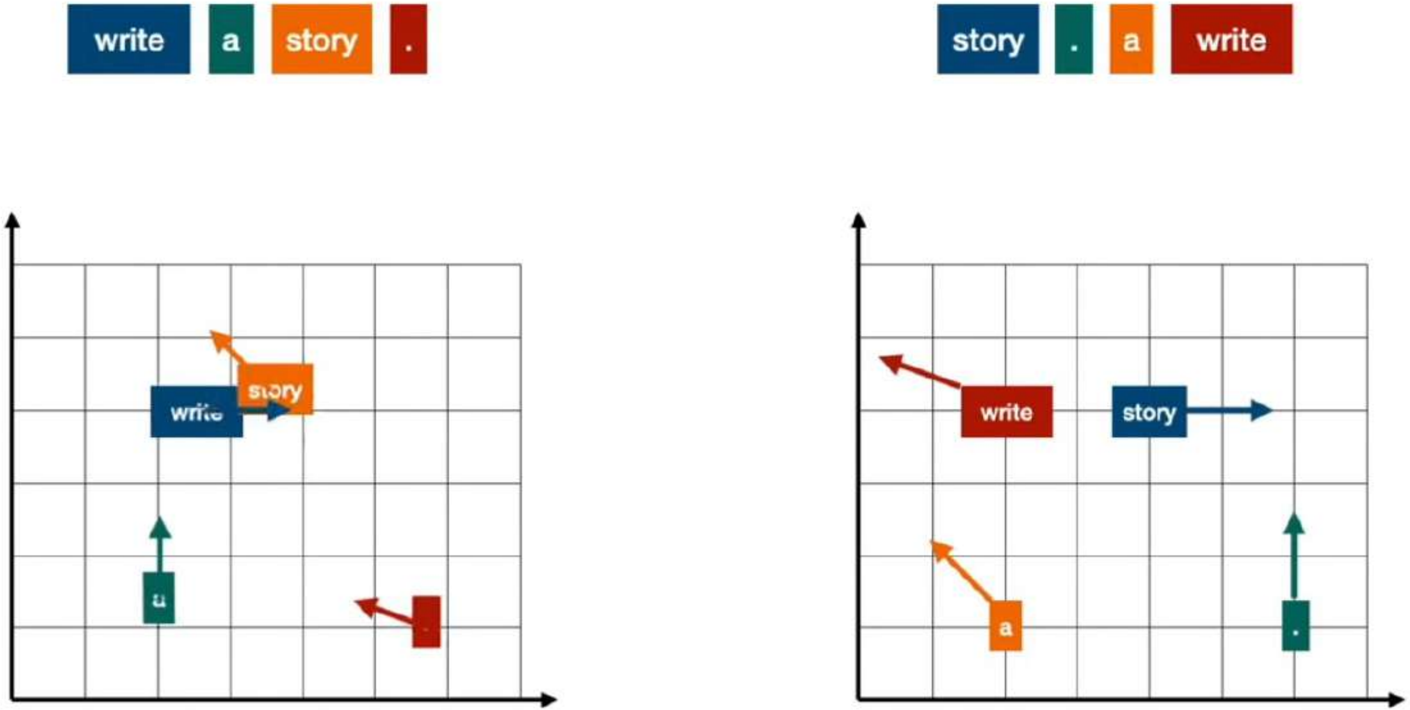


Positional encoding



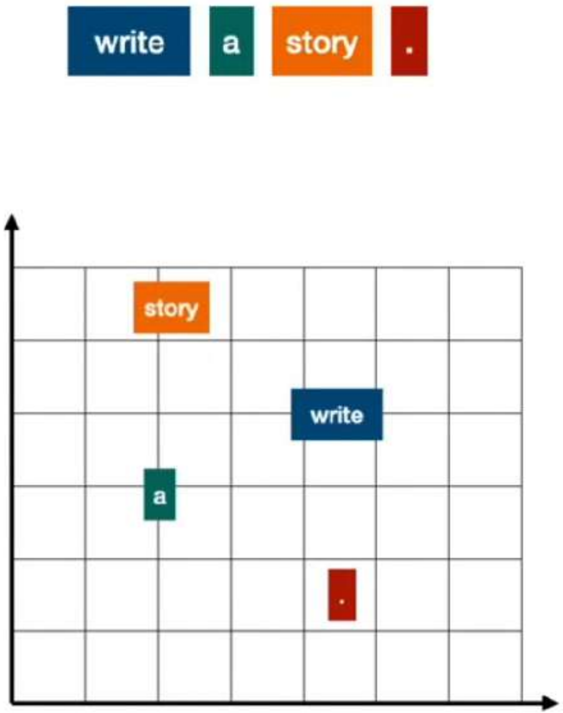
Luis Serrano

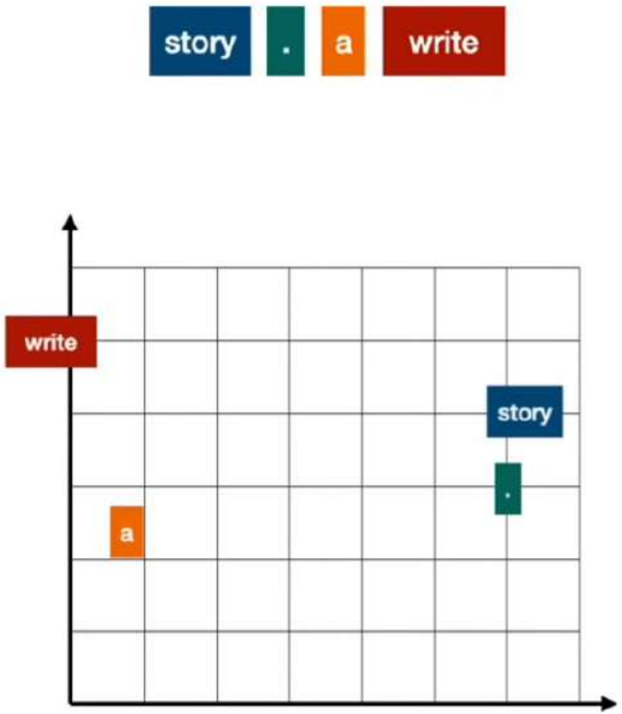
Positional encoding



Luis Serrano

Positional encoding





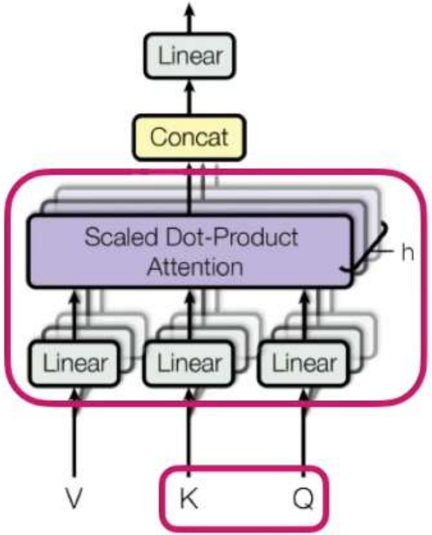
Luis Serrano

4) Attention

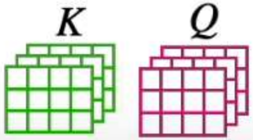
- Ainsi, lorsque le mécanisme attentionnel reçoit l'entrée, les enregistrements ont déjà été modifiés...
- Le mécanisme agit ensuite sur ces encastrements modifiés en position, comme nous l'avons vu plus haut.
- Le processus peut être itéré : imagine plusieurs pas de temps d'un système dans un champ gravitationnel.

Multi-head attention

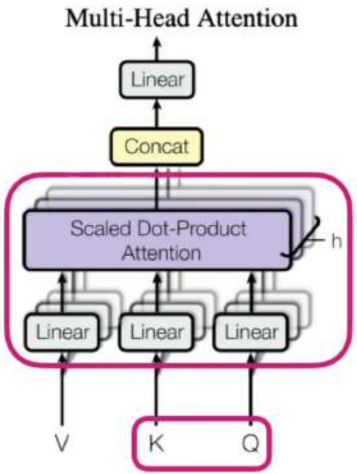
Multi-Head Attention



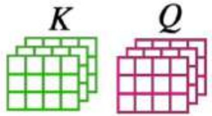
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_\mathrm{h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

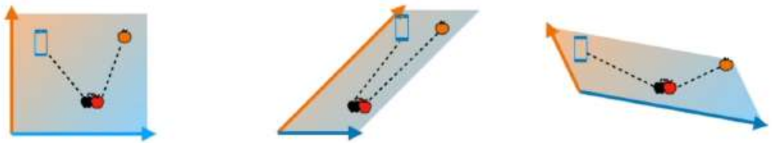


Multi-head attention



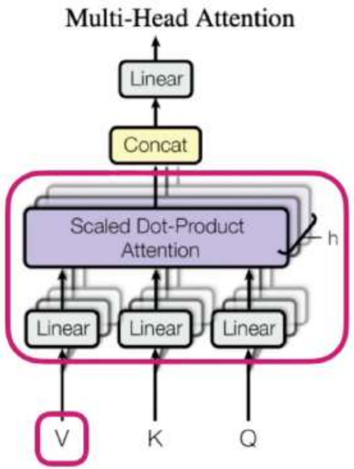
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$





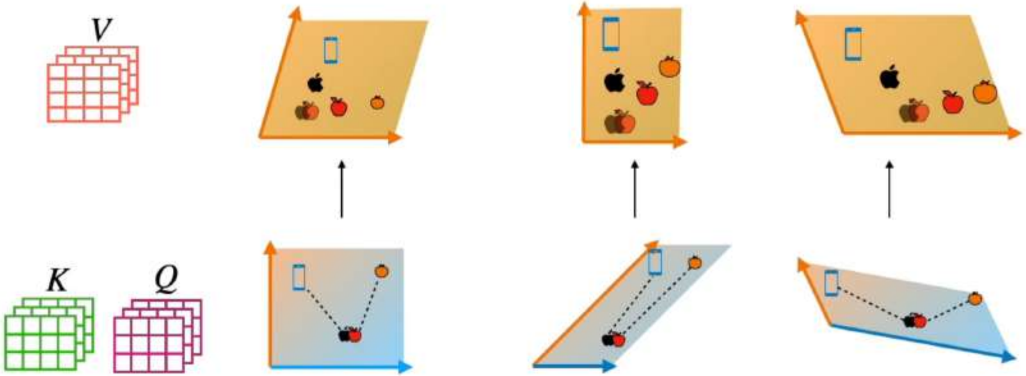
Luis Serrano

Multi-head attention



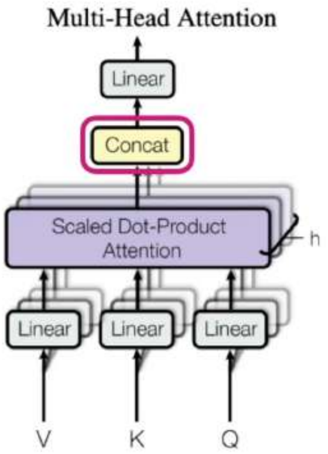
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$

$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

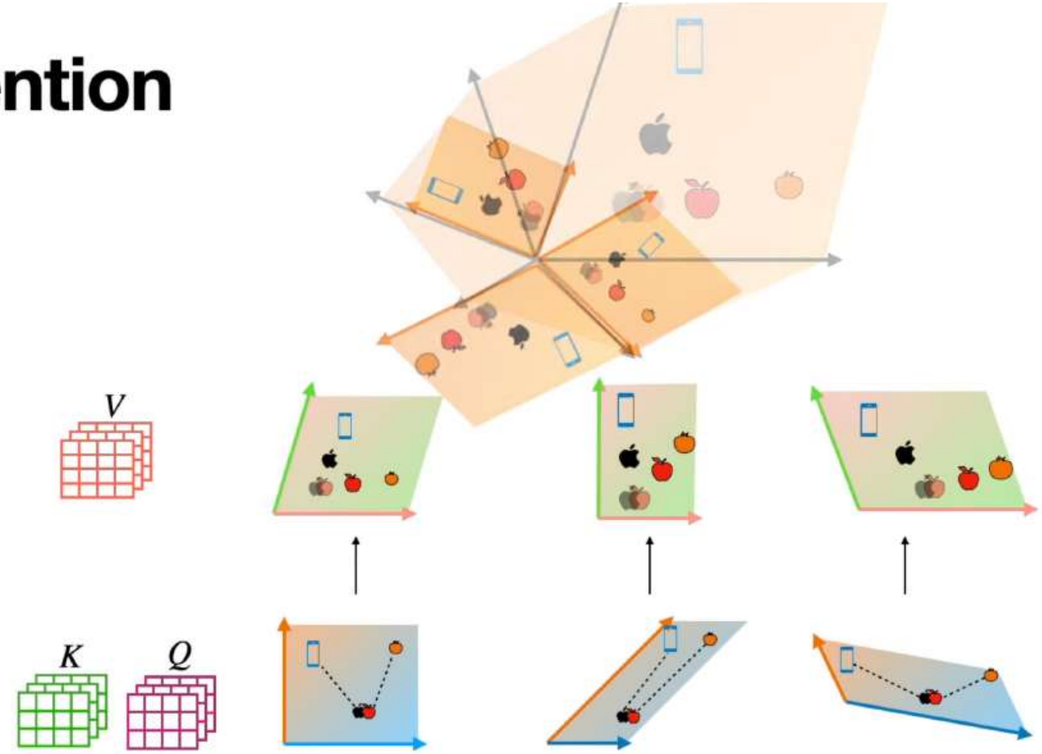


Luis Serrano

Multi-head attention

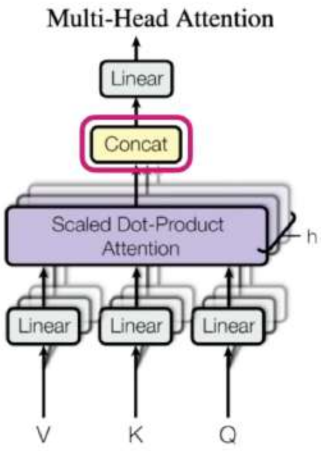


$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, \ldots, \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

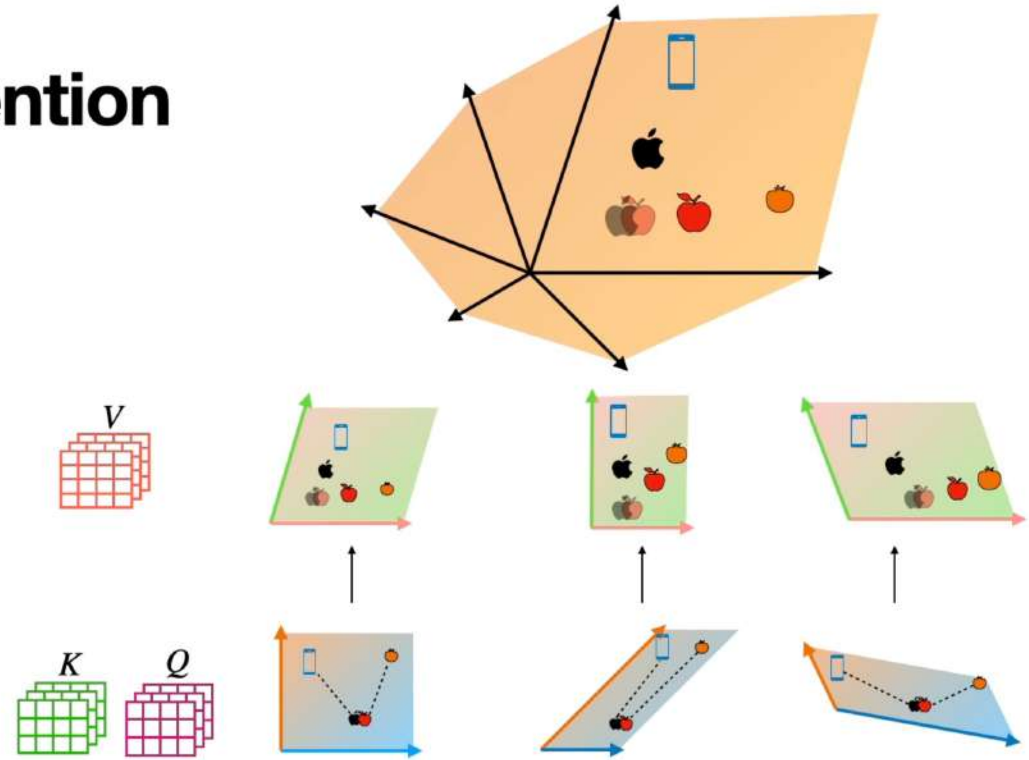


Luis Serrano

Multi-head attention

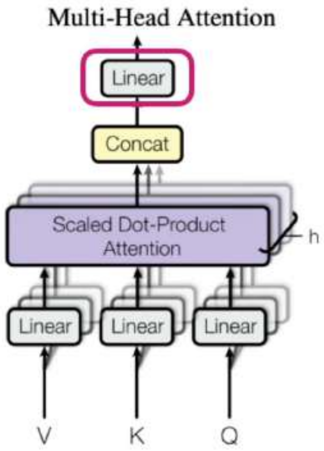


$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$
$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

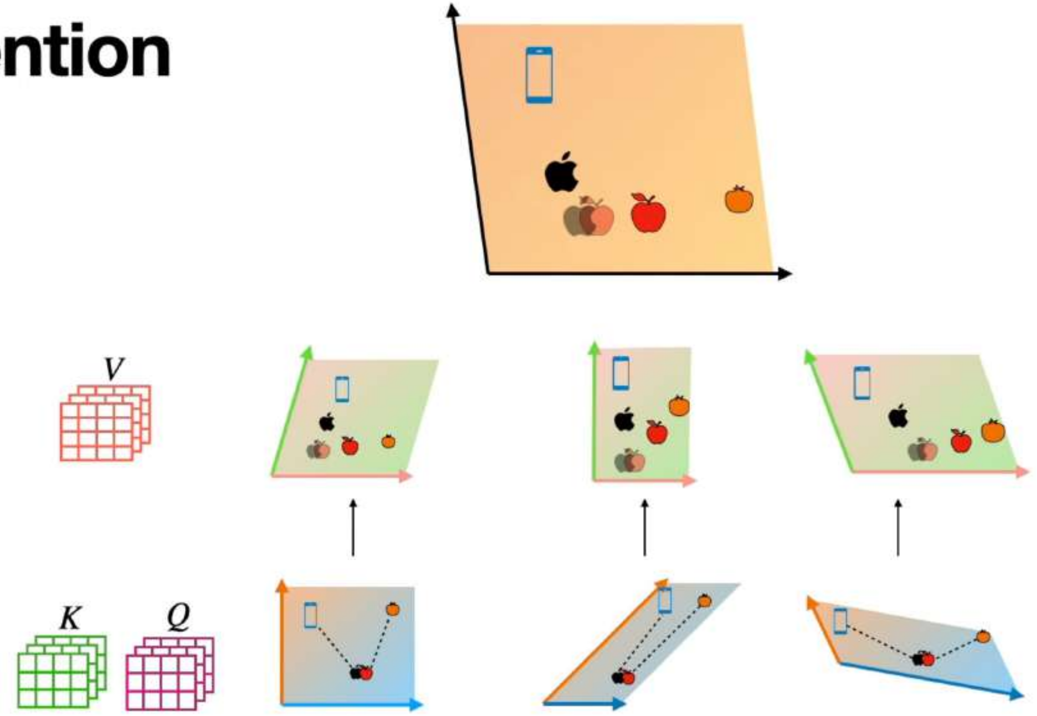


Luis Serrano

Multi-head attention

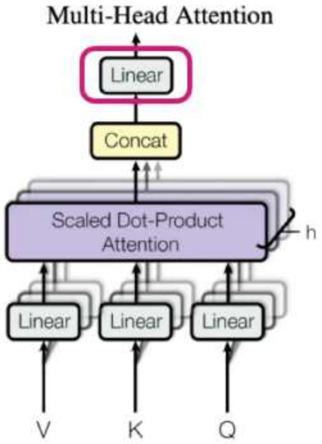


$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)W^O$$
$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

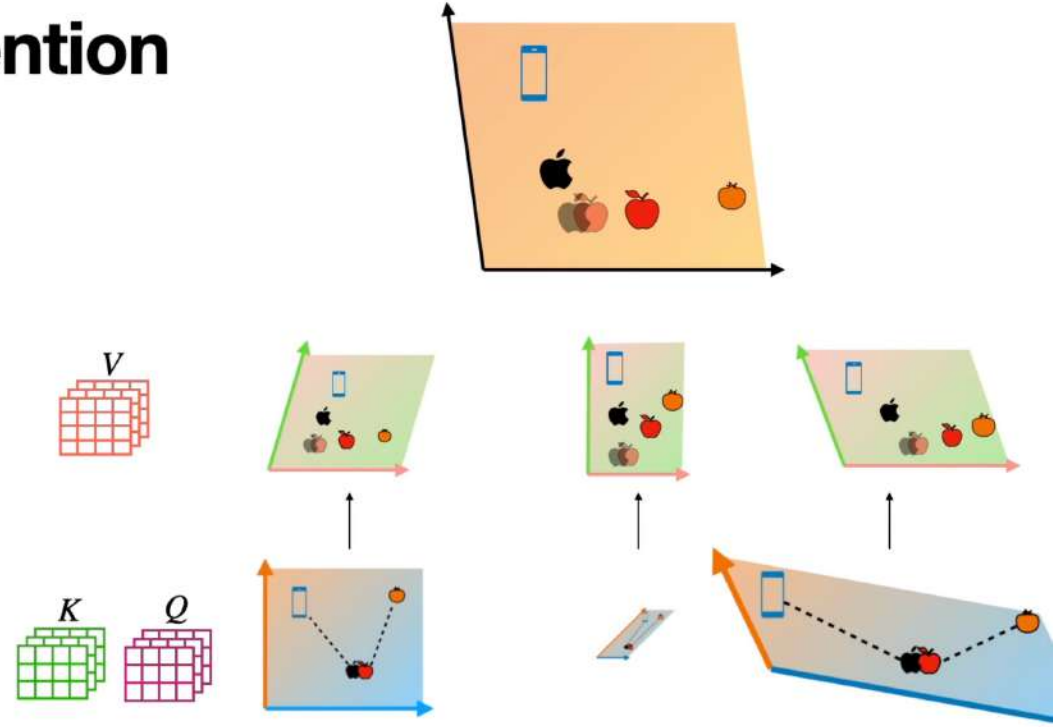


Luis Serrano

Multi-head attention



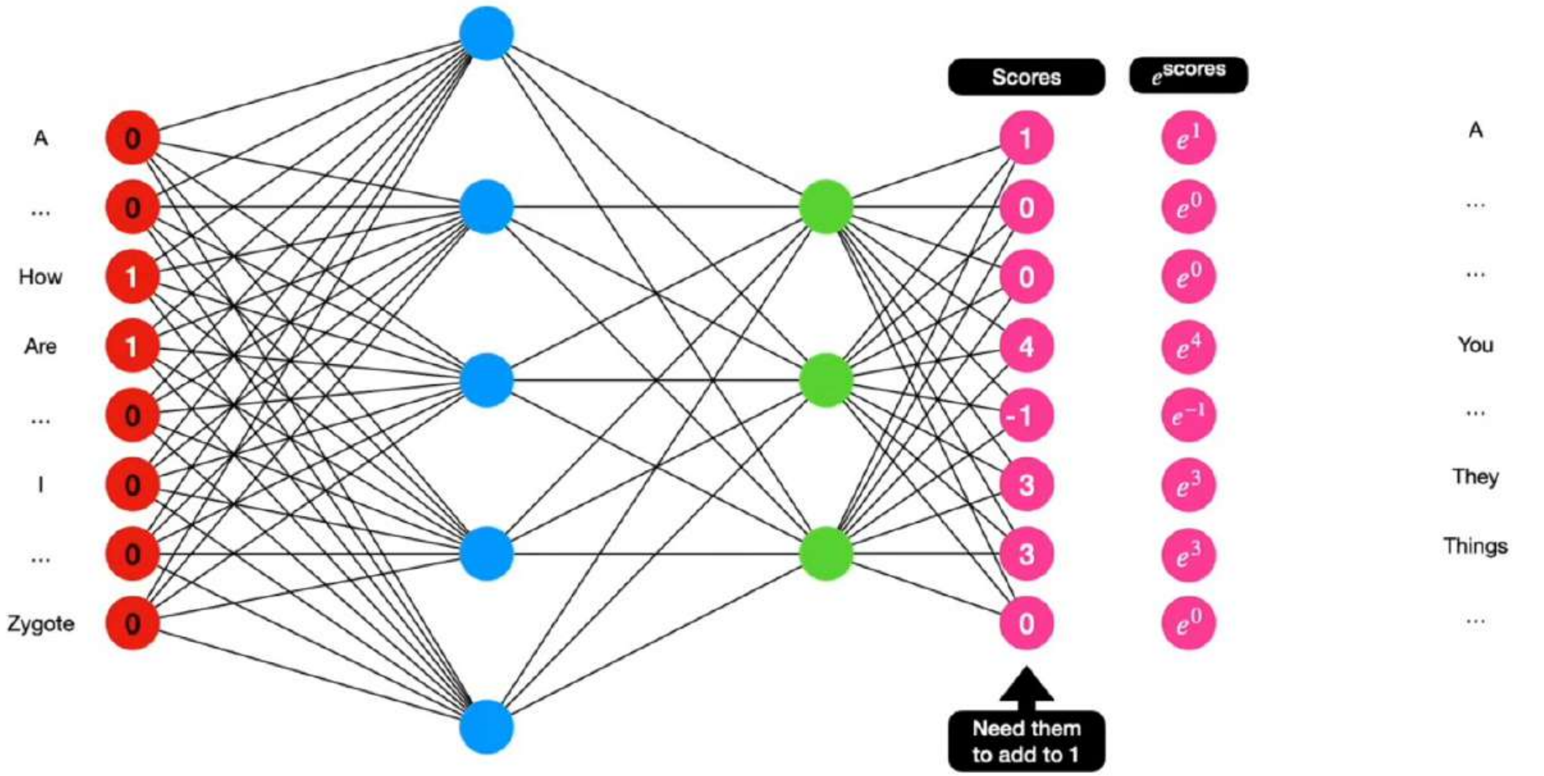
$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)W^O$$
$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



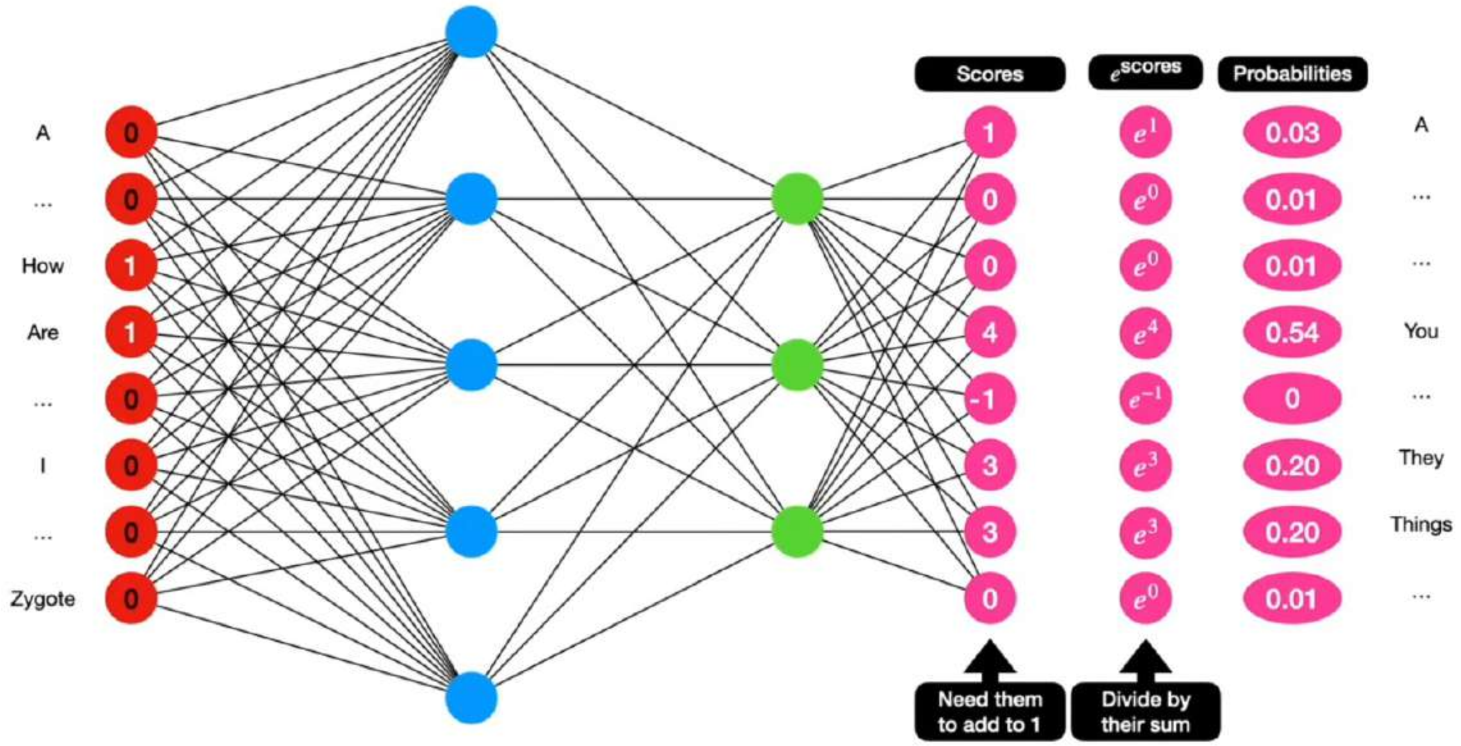
Luis Serrano

5) Softmax

- Enfin, le système émettra des scores, plus il est positif plus il pense que ce mot est approprié, plus il est négatif plus il pense que ce mot est inapproprié
- Tu veux normaliser ces données en probabilités : softmax est un bon moyen de le faire.

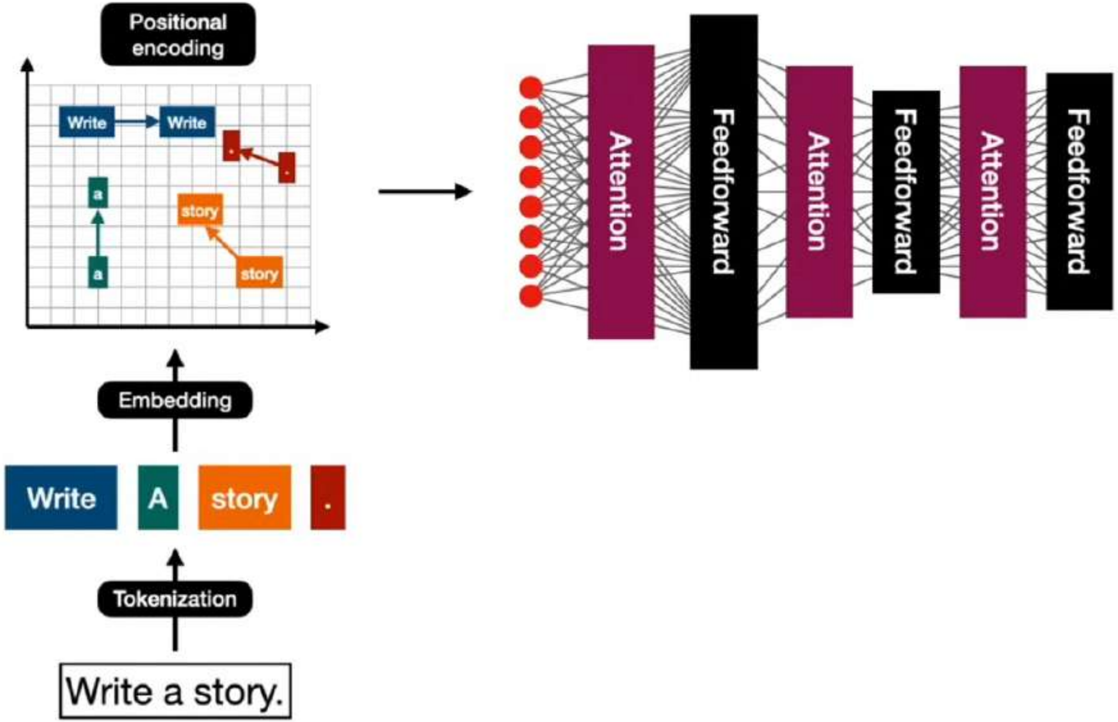


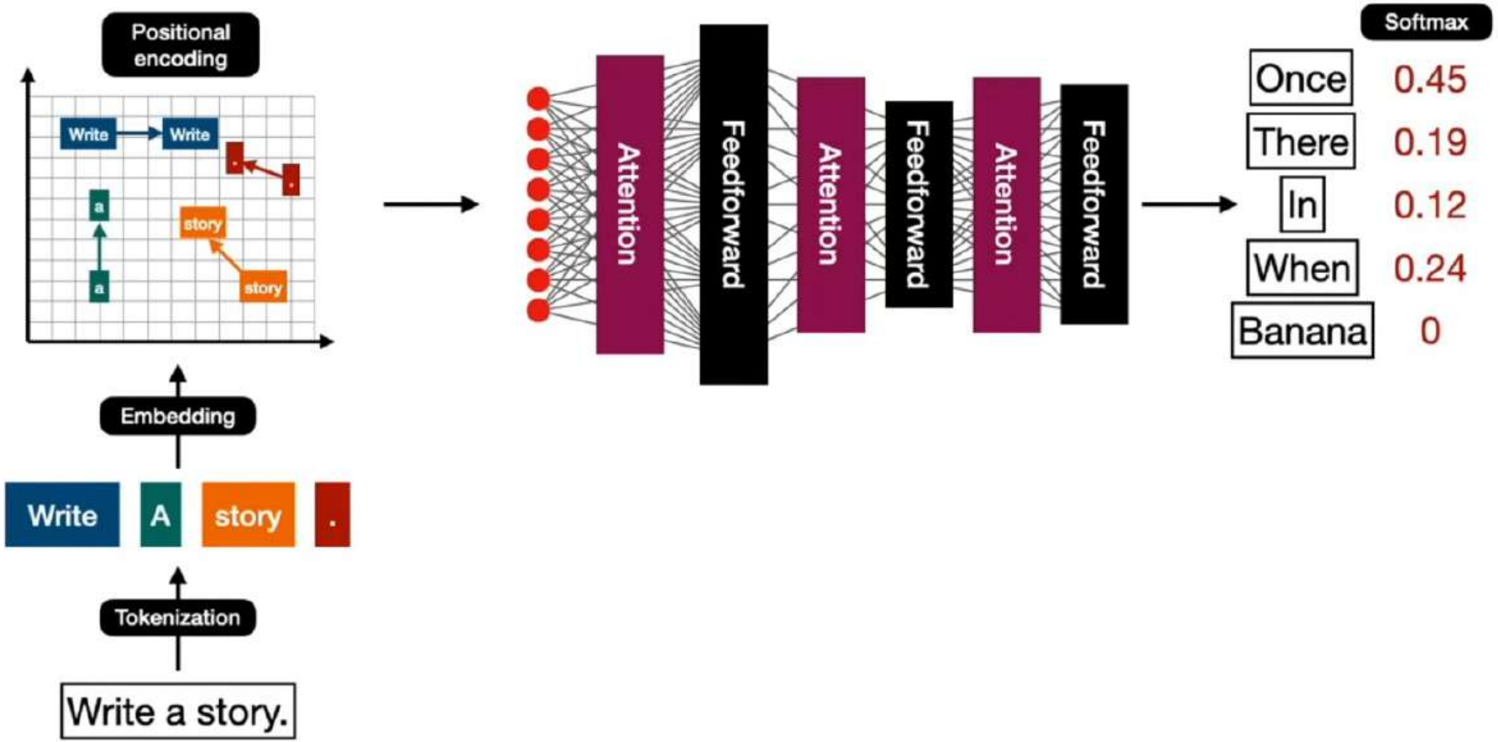
Luis Serrano



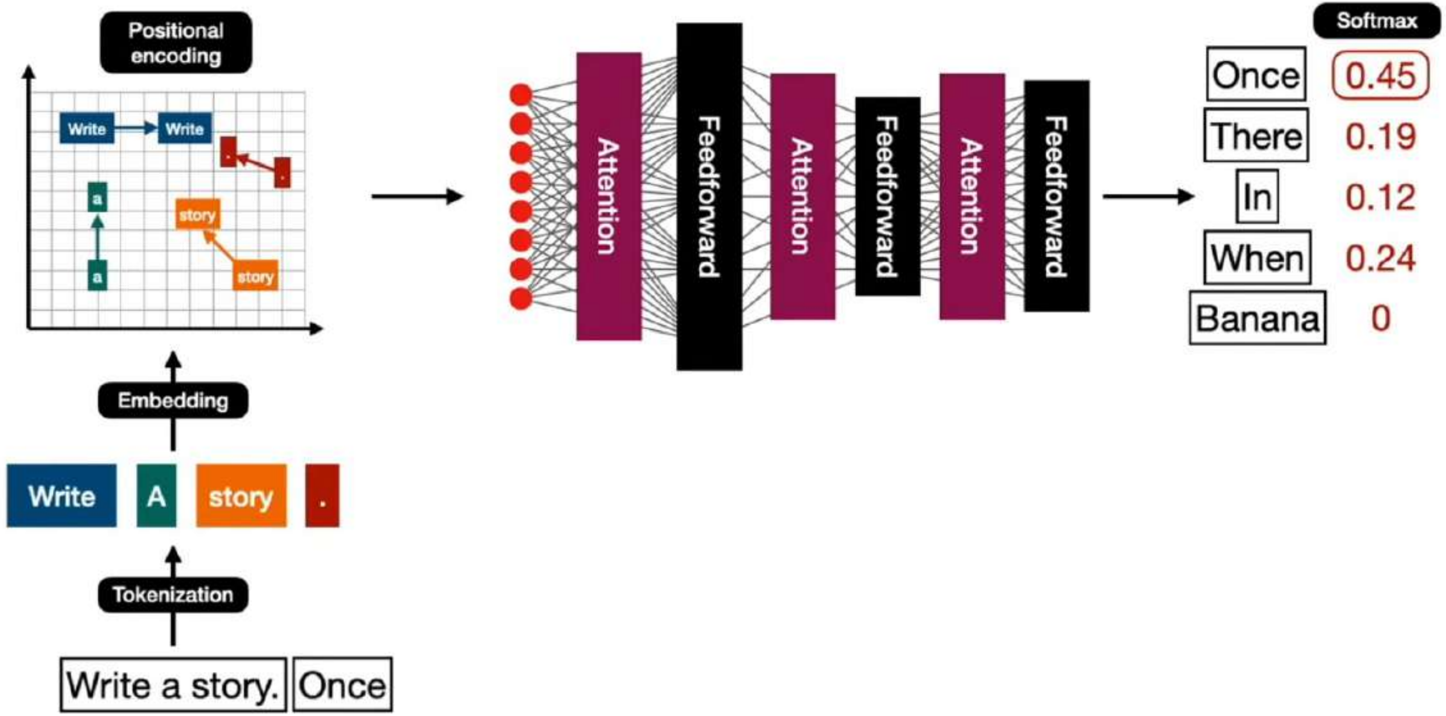
Luis Serrano

Tout ensemble:


Positional encoding
Write
Write
a
a
story
story
Embedding
Write
A
story
.
Tokenization
Write a story.
Attention
Feedforward
Attention
Feedforward
Attention
Feedforward



Luis Serrano

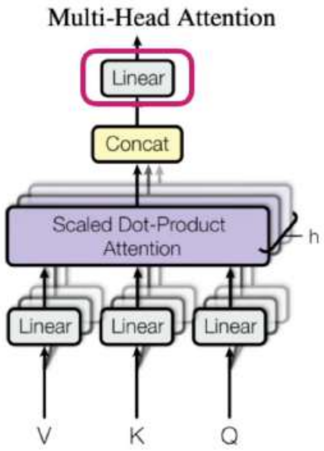


Luis Serrano

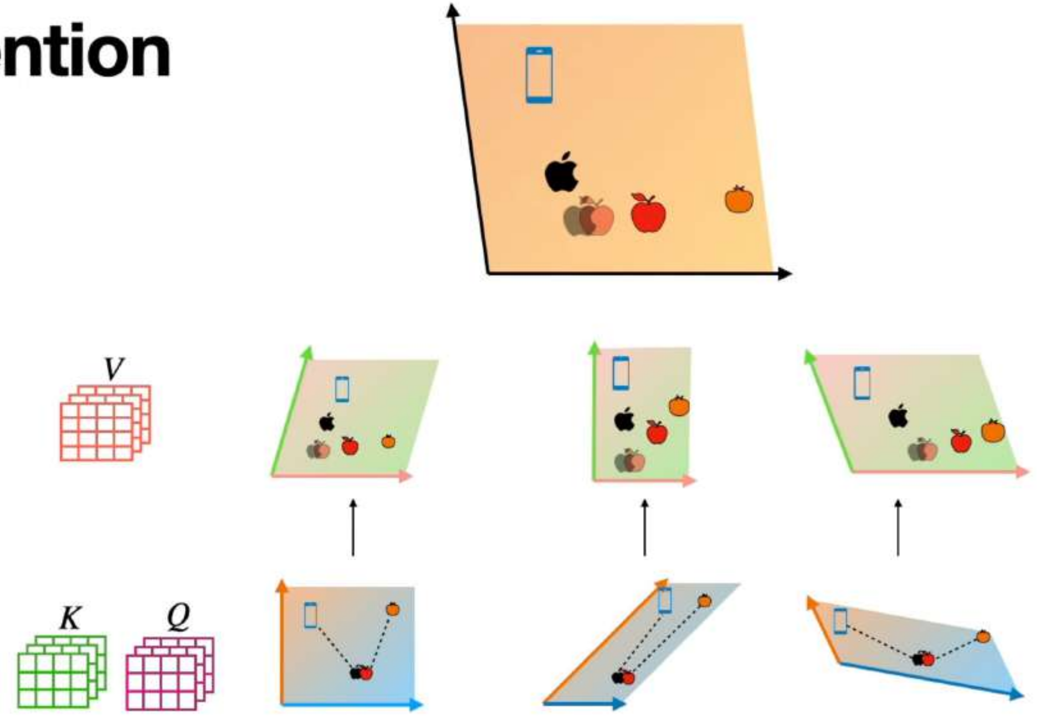
Entrainement / Apprentissage

- 1) Pre-training (la parti génératif)

Multi-head attention

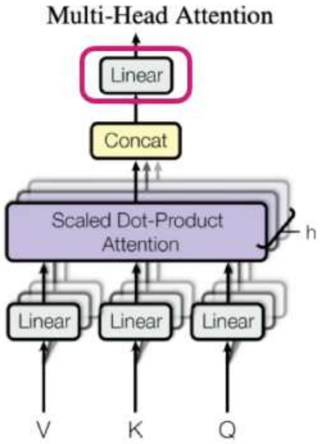


$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$
$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

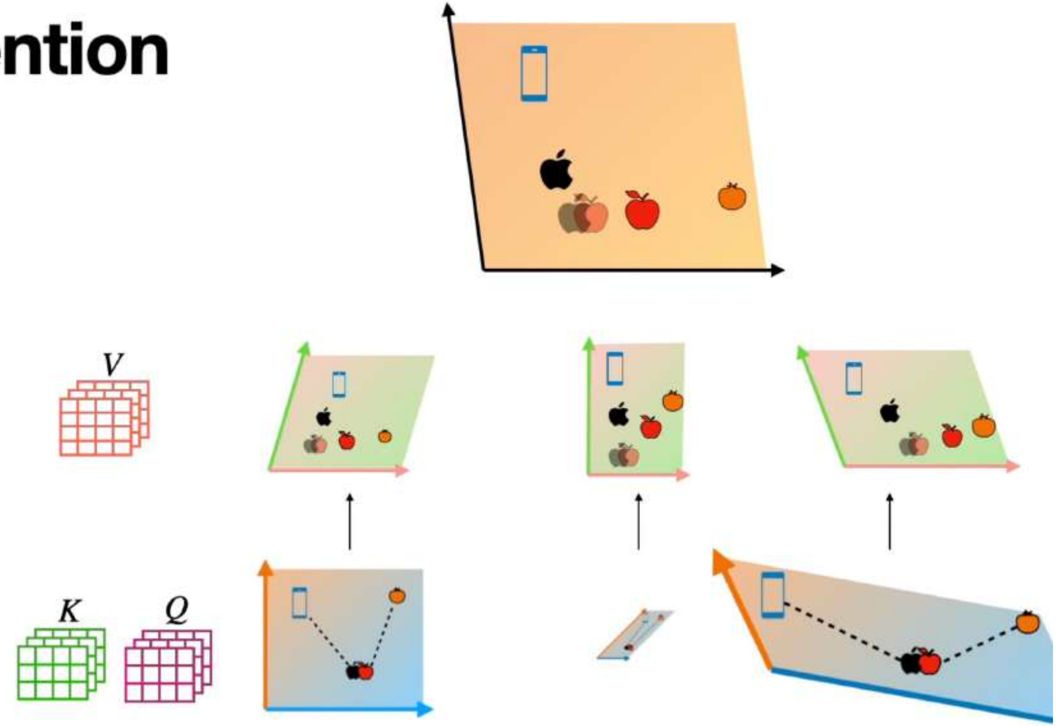


Luis Serrano

Multi-head attention



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



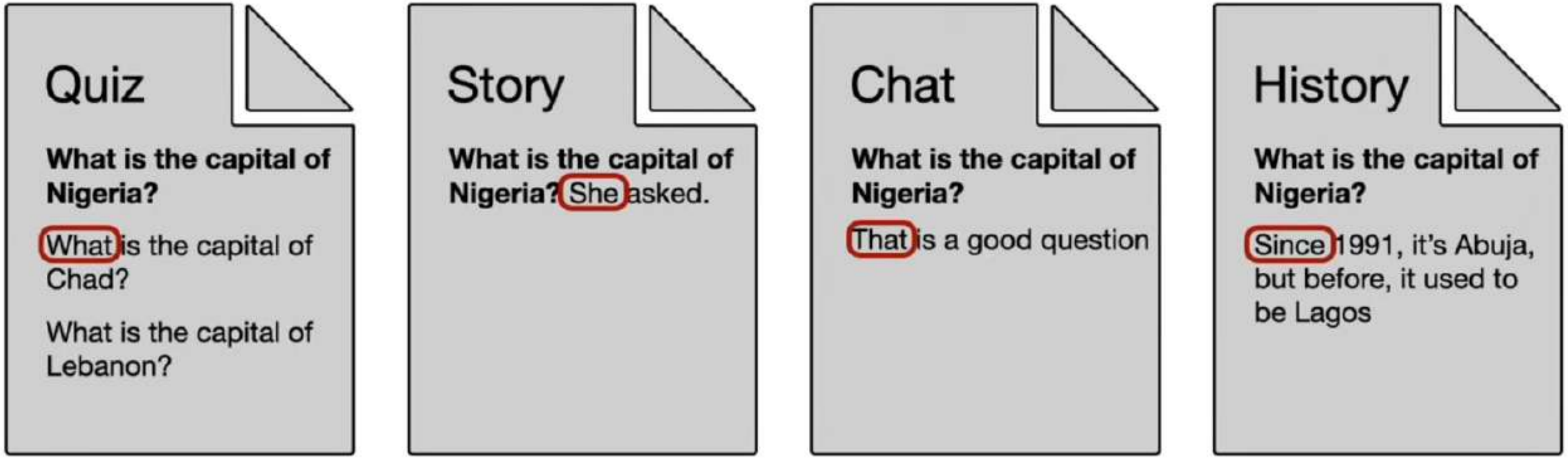
Luis Serrano

Entrainement / Apprentissage

- 2) Fine-tuning (étapes d'apprentissage supervisé ou par renforcement après pre-training

The internet is not a question/answer repository

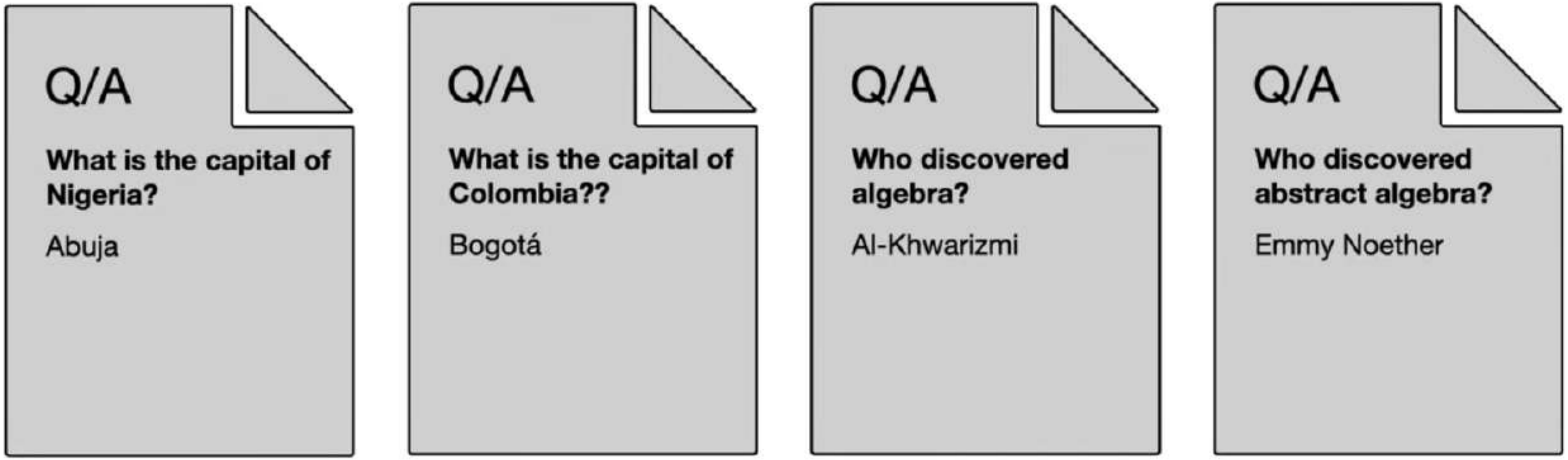
What is the capital of Nigeria? Abuja



Luis Serrano

Solution: Post-train it with Q/A datasets

What is the capital of Nigeria? Abuja



Luis Serrano

For chat: Post-train it with chats

Hello, how are you?

Good, and you?

...

Chat

Hello, how are you?

I'm good, and you?

Great, thank you!

Chat

Good morning, how can I help you?

Thank you, can you connect me with…

Chat

Hi mom!

Hello dear!

Chat

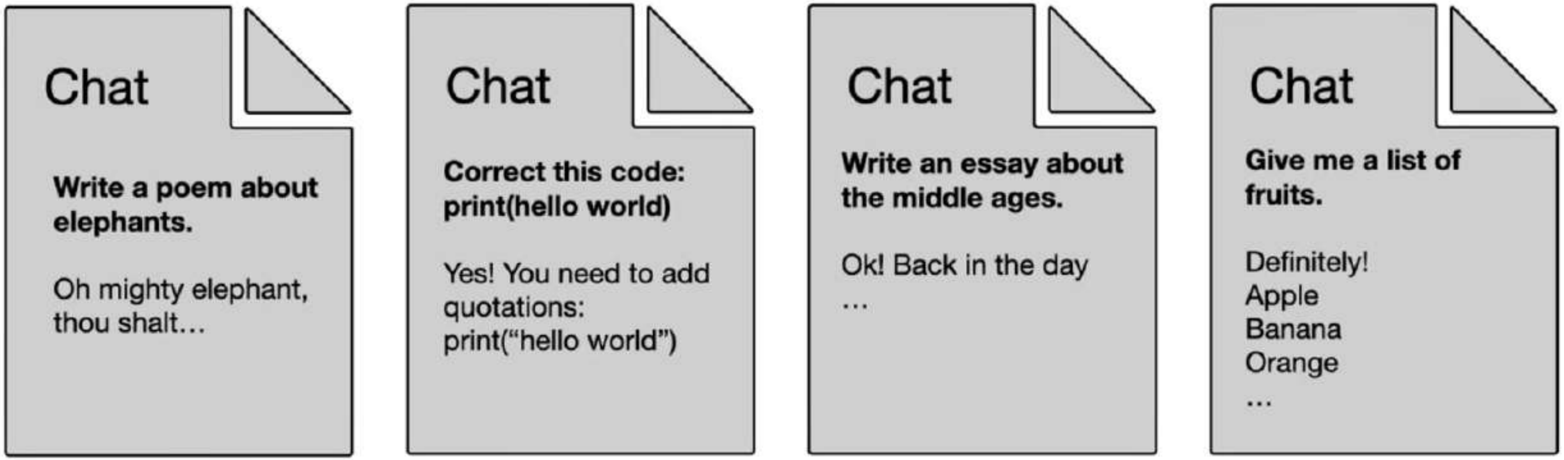
Hello, please connect me with customer support.

Of course!

Thank you!

For commands: Post-train it with command/action pairs

Do this!
Ok, boss!



Capacités émergentes

- **Neurons responding to specific words which are split into multiple tokens:** "Bank|ing", "word|ing", "Ch|olesterol", "Libert|arian", "Civil|ian", "Sh|anghai", "Not|withstanding"...
- **Neurons responding to the names of famous people:** "Martin| Luther| King", "Donald| Trump", "Lyndon| Johnson", "George| Orwell", "Ernest| Hemingway", "Muhammad| Ali", "Oprah| Winfrey"... (cf. [17])
- **Neurons responding to other nouns:** "Human| Rights| Watch", "International| Monetary| Fund", "Hurricane| Matthew", "Real| Madrid"...
- **Neurons responding to compound words:** "book| club", "social| security", "computer| vision", "organized| crime", "birthday| party", "heart| attack"...
- **Neurons responding to LaTeX "\" commands:** "\|left", "\|frac|{", "\|begin"...

Elhage et al 2022

A huge variety of interesting neurons can be found in these layers. Some common categories we observed include:

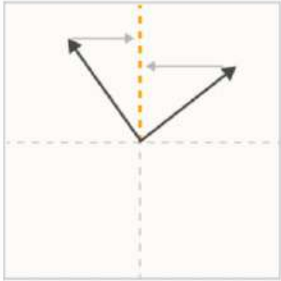
- **Neurons which fire on particular types of descriptive clauses:** a neuron which fires on a clause describing a sound, a neuron for clauses describing clothing, a neuron for musical descriptive clauses (e.g. "in the key of C major"), a neuron for clauses describing text written on an object, ...
- **Neurons which respond to discourse markers:** a neuron which responds to markers emphasizing the importance of something (e.g. "the amazing thing is"), a neuron which responds to hedging (e.g. "it seems to me that..."), ...
- **Neurons which disambiguate a special interpretation of a token:** a neuron which responds to A/B/C/D when used as grades, a neuron which responds to the "day" portion of a date, a neuron which responds to numbers when they're a quantity in a recipe, a neuron which responds to C-style format specifiers (e.g. "%s" or "%d") in strings, ...

Elhage et al 2022

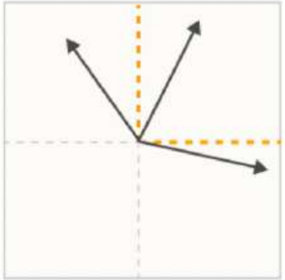
3.4 The Superposition Hypothesis

Roughly, the idea behind the superposition hypothesis is that neural networks "want to represent more features than they have neurons," so they exploit a property of high-dimensional spaces to simulate a model with many more neurons. (Note that as a matter of terminology we use "polysemanticity" to refer to the empirical phenomenon of neurons responding to multiple features, and "superposition" to refer to the hypothesis described here.)

If true, the superposition hypothesis means there is *no basis* in which activations are interpretable: searching for an interpretable basis is fundamentally the wrong framing. Especially important features might get dedicated neurons, but most features don't align with neurons because they need to share and *can't* have a dedicated one.



Polysemanticity is what we'd expect to observe if features were not aligned with a neuron, despite incentives to align with the privileged basis.



In the **superposition hypothesis**, features can't align with the basis because the model embeds more features than there are neurons. Polysemanticity is inevitable if this happens.

Elhage et al 2022