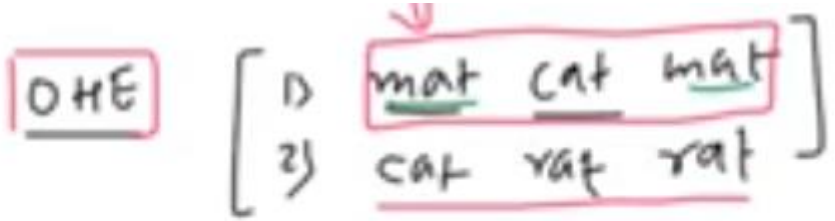
Self-Attention

Dr. Muhammad Safyan

- For any NLP task , we give the input a set of words →numbers.
- How to convert words in number called vectorization
- First step in NLP
  - Convert the word into number
    - One-hot-encoding
- Problem:

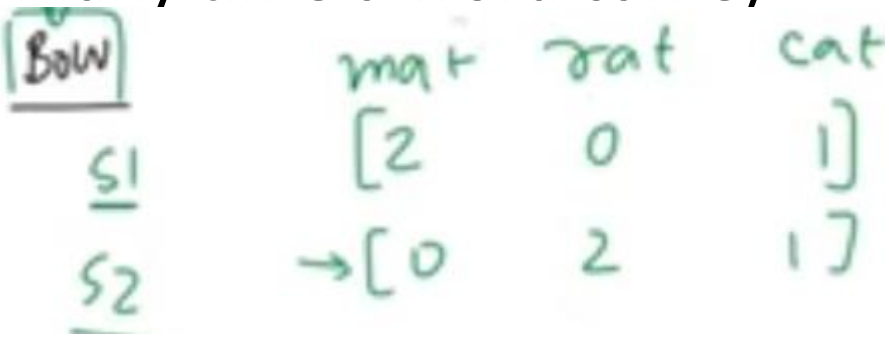


num → [1 0 0] [0 1 0] [1 0 0]

king → [0.6 0.1 1 0 0.9]

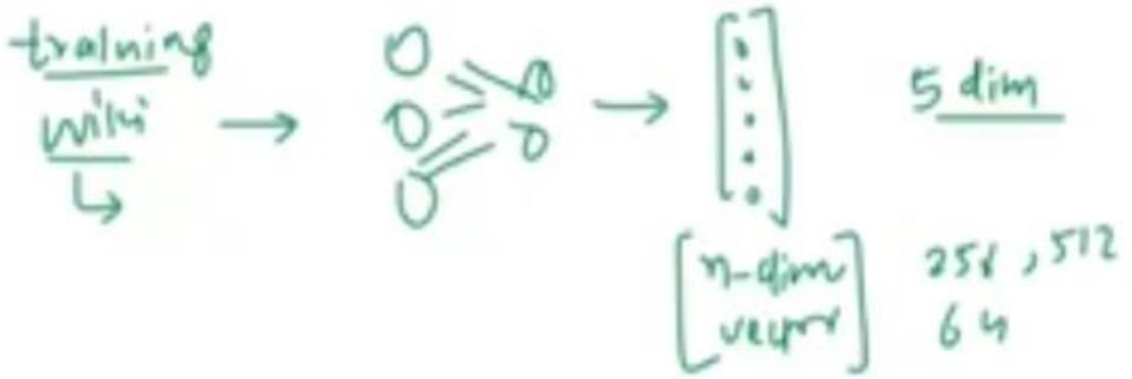
queen → [0.3 0.2 0.4 1 0]

- BoW(how many time a word came)



- Tf-Idf

- Word Embedding (Most Powerful having Semantics)



- Pick the data send them in NN and convert it into n-dimension vector, let say 5 dimension
- Each dimension represent a factor. Each dimension has meaning.

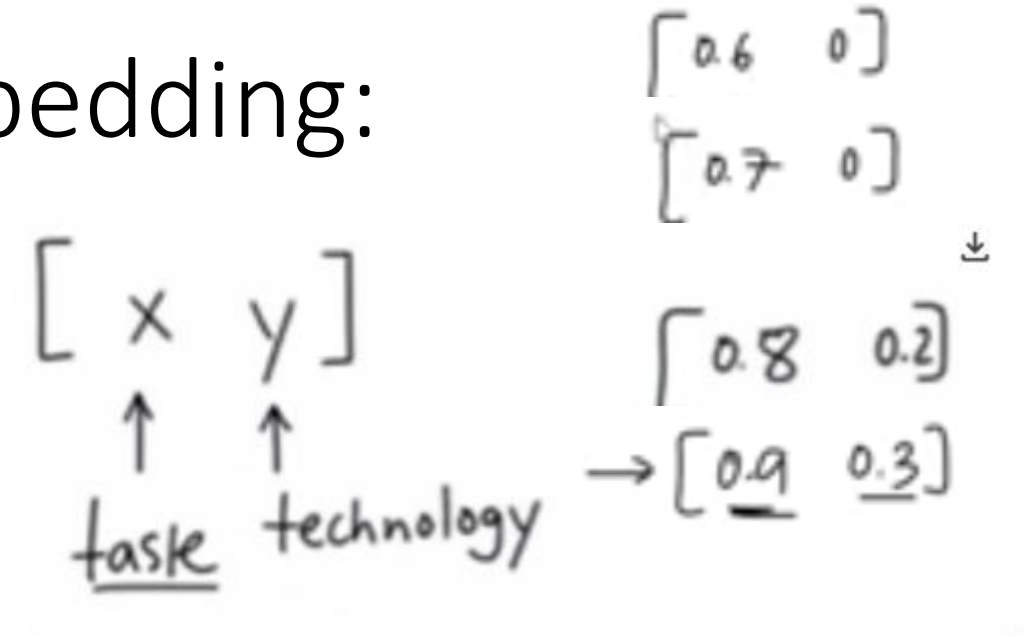
- king → [0.6 0.1 1 0 0.7]

queen → [0.3 0.2 0.4 1 0]

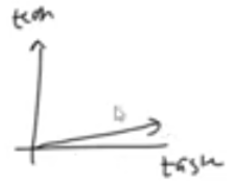
cricket [0.2 0.9 0.5 0.90

Problem with w2vec Embedding:

- Capture average Meaning:
  1. An apple a day keeps the doctor away
  2. Apple is healthy
  3. Apple is better than orange
  4. Apple moves great phones



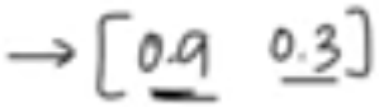
- Capture average meaning.
- Word Embedding →Static Make once and used again& again
- Make once and used again &again
- It should be dynamic



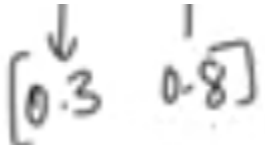
Fruit→ [**Apple Launched a new Phone while I was eating an orange**]
static → [0.9 0.3]

- Self-attention produce dynamic embedding from static embedding

- Static embedding is:

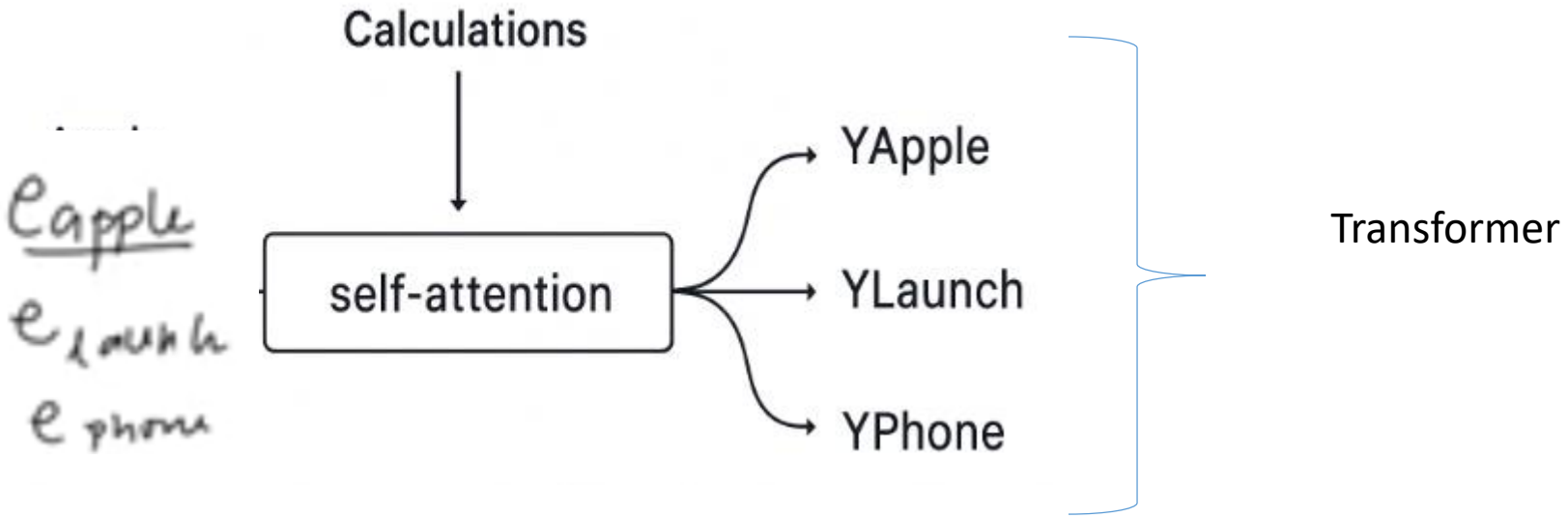


- Apple used as Technology but our static embedding tilted toward fruit.
- Need contextual embedding instead of static embedding.
- It should automatically increased value of .3, when sees the word phone and launched and should decrease weight of fruit



Self-Attention

- Self-attention is a mechanism that generate the smart contextual embedding from earlier generated static embedding

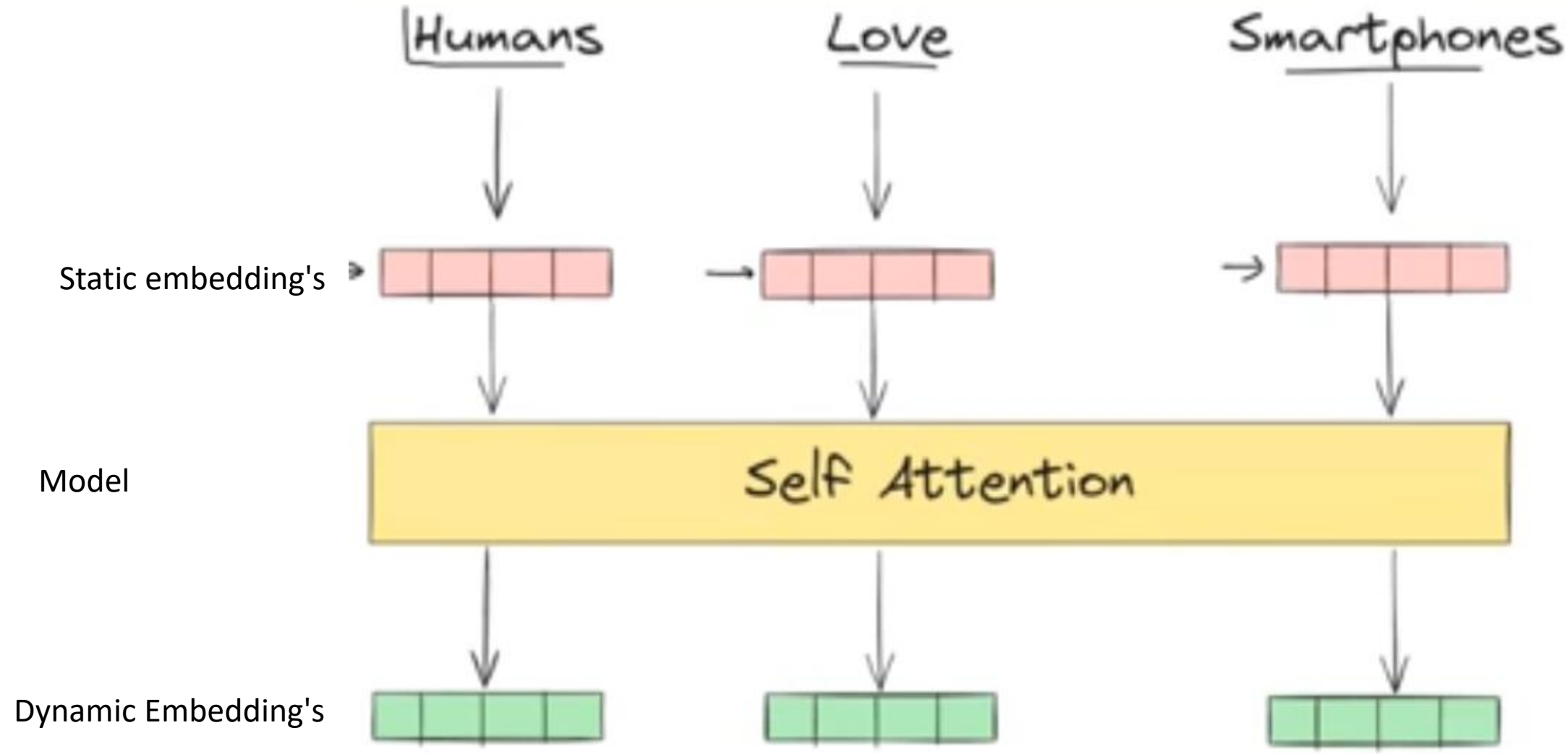


Transformer

- Gen-AI →center is Transformer→ center is Self-Attention.
- For any NLP application we represent word into numbers
  - Word embedding→static

  - Should be dynamic/contextual instead of static embedding's.

Self-Attention-Abstract Architecture



Self-attention-Box

Money Bank grows

bank → bank
bank → 0.3money + 0.7bank + 0.1grow

River Bank Flows

bank → bank
bank → 0.5river + 0.4bank + 0.1flows

money = 0.7 money + 0.2 bank + 0.1 grows
bank = 0.25 money + 0.7 bank + 0.05 grows
grows = 0.1 money + 0.2 bank + 0.7 grows

river = 0.8 river + 0.15 bank + 0.05 flows
bank = 0.2 river + 0.78 bank + 0.02 flows
flows = 0.4 river + 0.01 bank + 0.59 flows

Here words are used to represent. But machine only works with numbers

money = 0.7 money + 0.2 bank + 0.1 grows
bank = 0.25 money + 0.7 bank + 0.05 grows
grows = 0.1 money + 0.2 bank + 0.7 grows

river = 0.8 river + 0.15 bank + 0.05 flows
bank = 0.2 river + 0.78 bank + 0.02 flows
flows = 0.4 river + 0.01 bank + 0.59 flows

N-dimensions

N-dimensions

$$e_{\text{money}}^{(\text{new})} = 0.7\, e_{\text{money}} + 0.2\, e_{\text{bank}} + 0.1\, e_{\text{grows}}$$

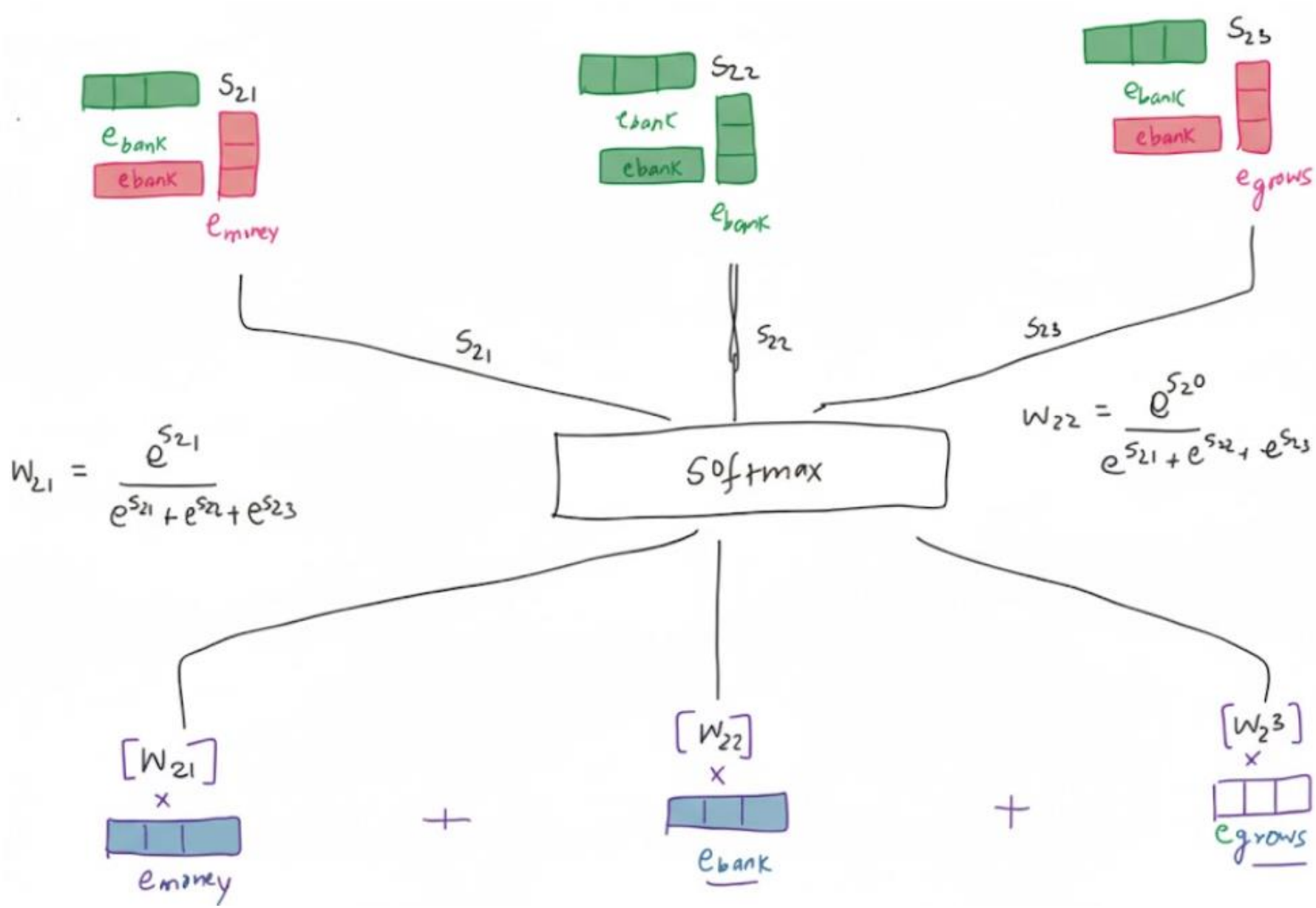
$$e_{\text{bank}}^{(\text{new})} = 0.25\, e_{\text{money}} + 0.7\, e_{\text{bank}} + 0.05\, e_{\text{grows}}$$

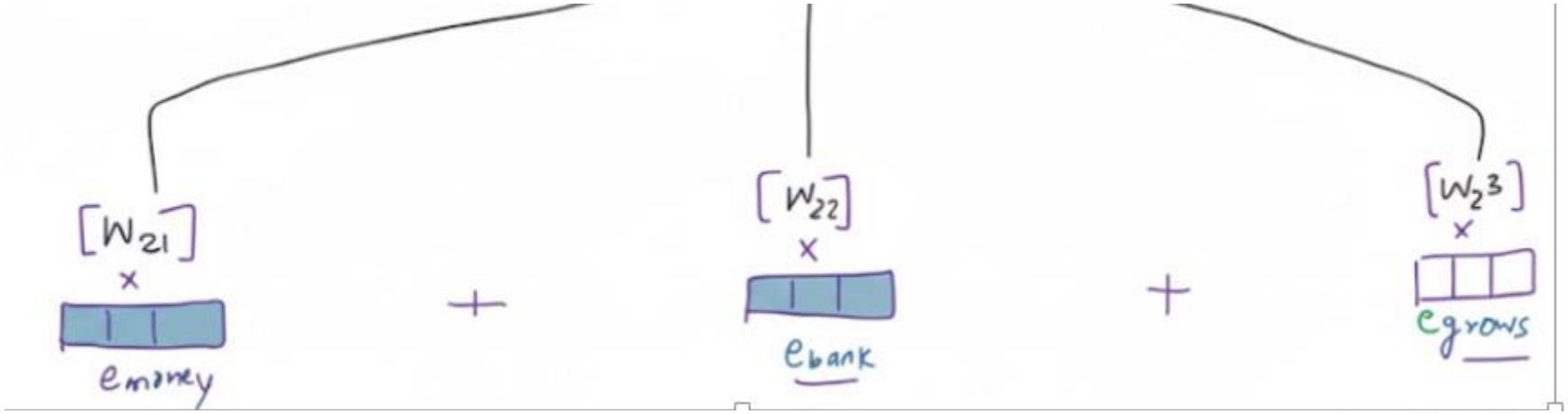
$$e_{\text{gross}}^{(\text{new})} = 0.1\, e_{\text{money}} + 0.2\, e_{\text{bank}} + 0.7\, e_{\text{gross}}$$

Numbers represent the similarity between the words
Similarity between the two vector can be calculated with dot product of two vectors.

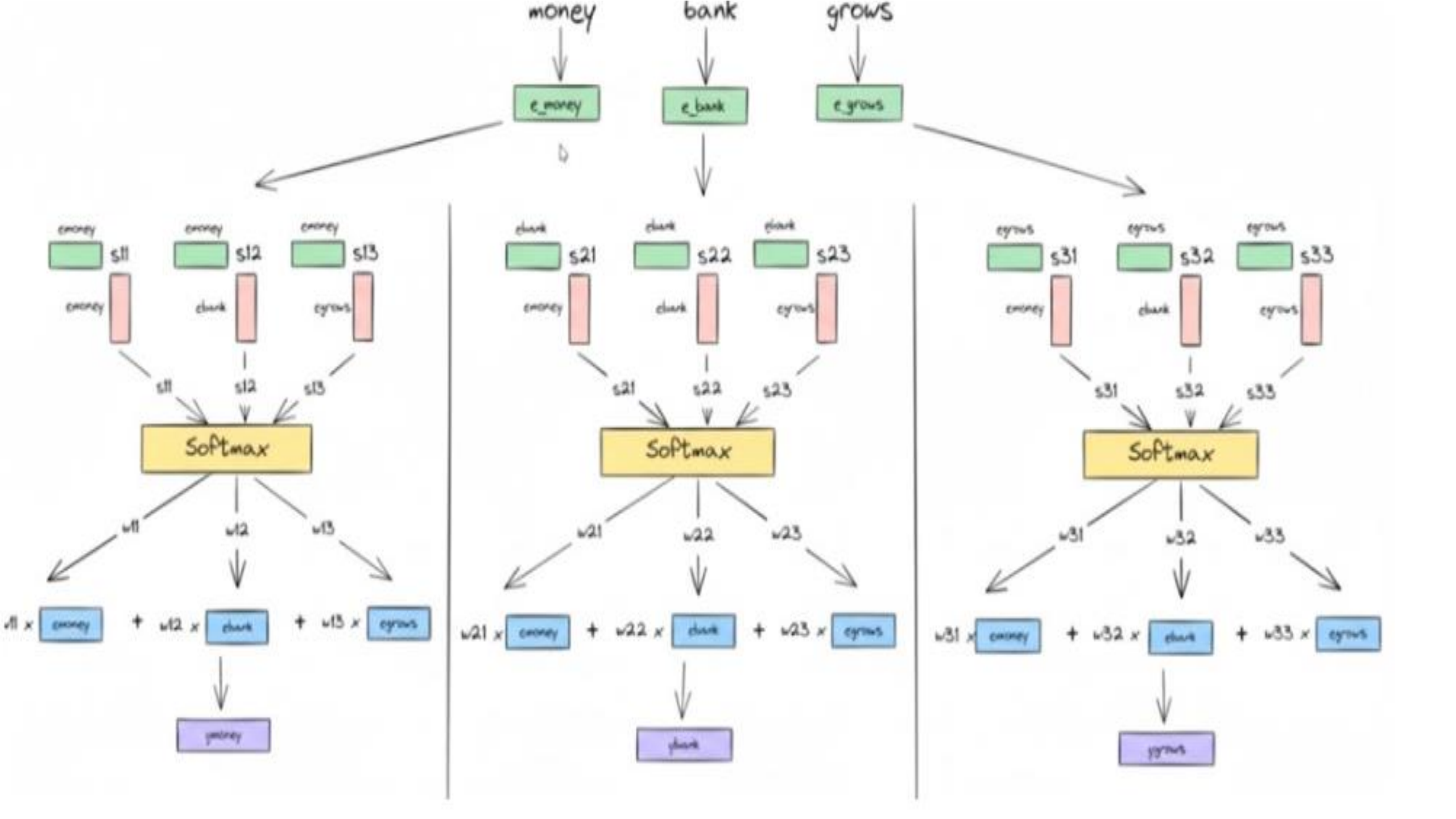
$$e_{bank}^{(new)} = (e_{bank} \cdot e_{money}^{T})e_{money} + (e_{bank} \cdot e_{bank}^{T})e_{bank} + (e_{bank} \cdot e_{grows}^{T})e_{grows}$$

$$e_{bank}^{(new)} = (e_{bank} \cdot e_{money}^T)e_{money} + (e_{bank} \cdot e_{bank}^T)e_{bank} + (e_{bank} \cdot e_{grows}^T)e_{grows}$$

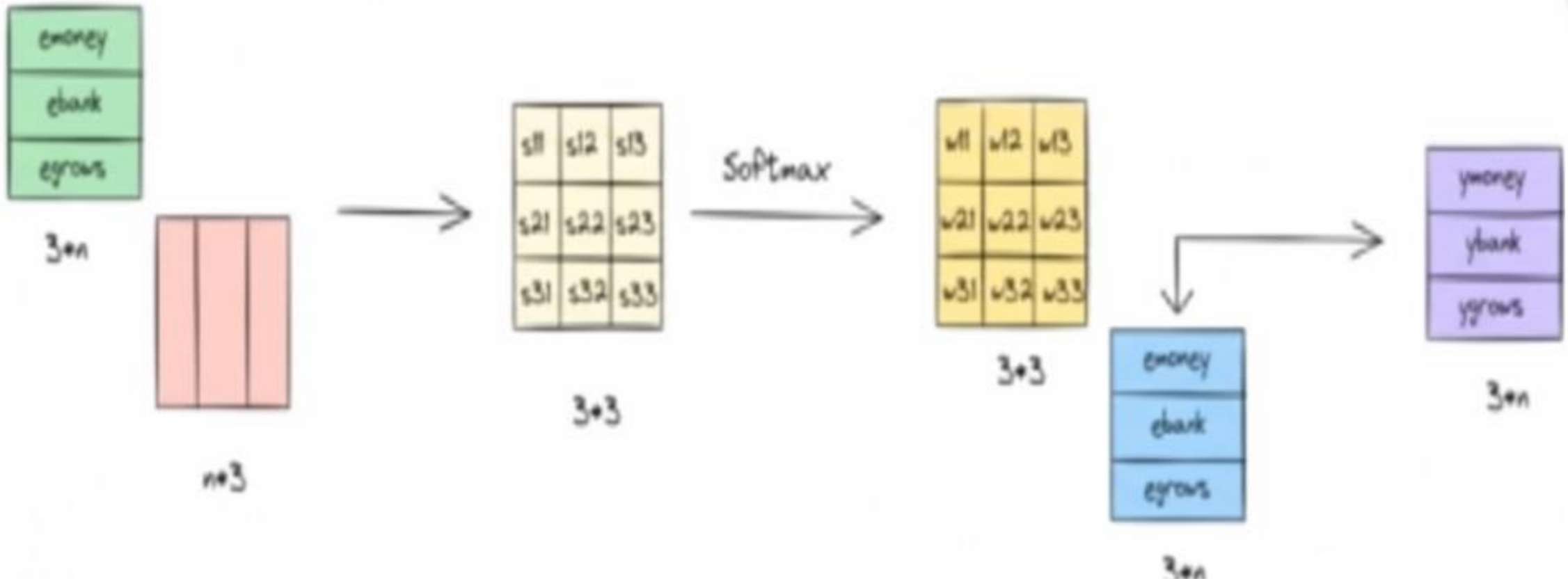




$$e_{bank}^{(new)} = Y_{bank} \rightarrow \text{Contextual word Embeding}$$

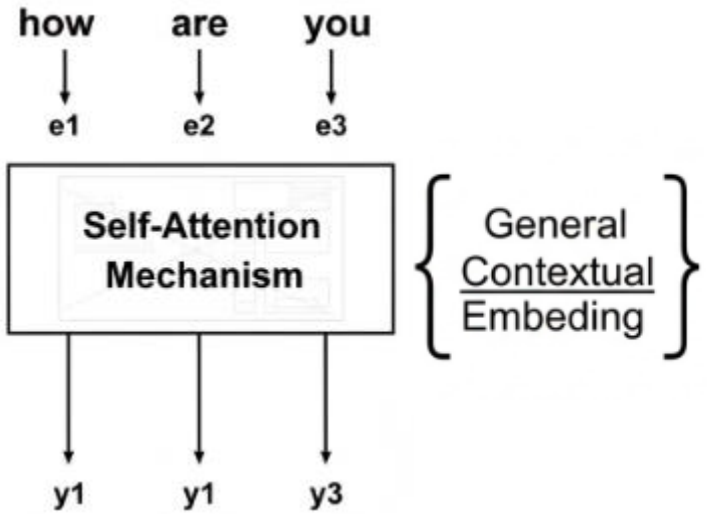

money
bank
grows
e_money
e_bank
e_grows
s11
s12
s13
s21
s22
s23
s31
s32
s33
Softmax
Softmax
Softmax
w11
w12
w13
w21
w22
w23
w31
w32
w33

- Points to Consider
- → This operation is a parallel operation
  → There are no parameters involved


emoney
ebank
egrows
3*n
n*3
s11 s12 s13
s21 s22 s23
s31 s32 s33
3*3
Softmax
v11 v12 v13
v21 v22 v23
v31 v32 v33
3*3
emoney
ebank
egrows
3*n
ymoney
ybank
ygrows
3*n

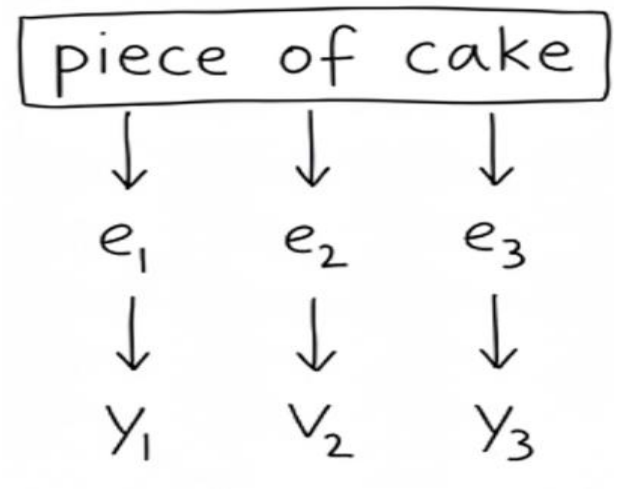
There is no parameter involved

- *Hello*→*e*1→*Y*1



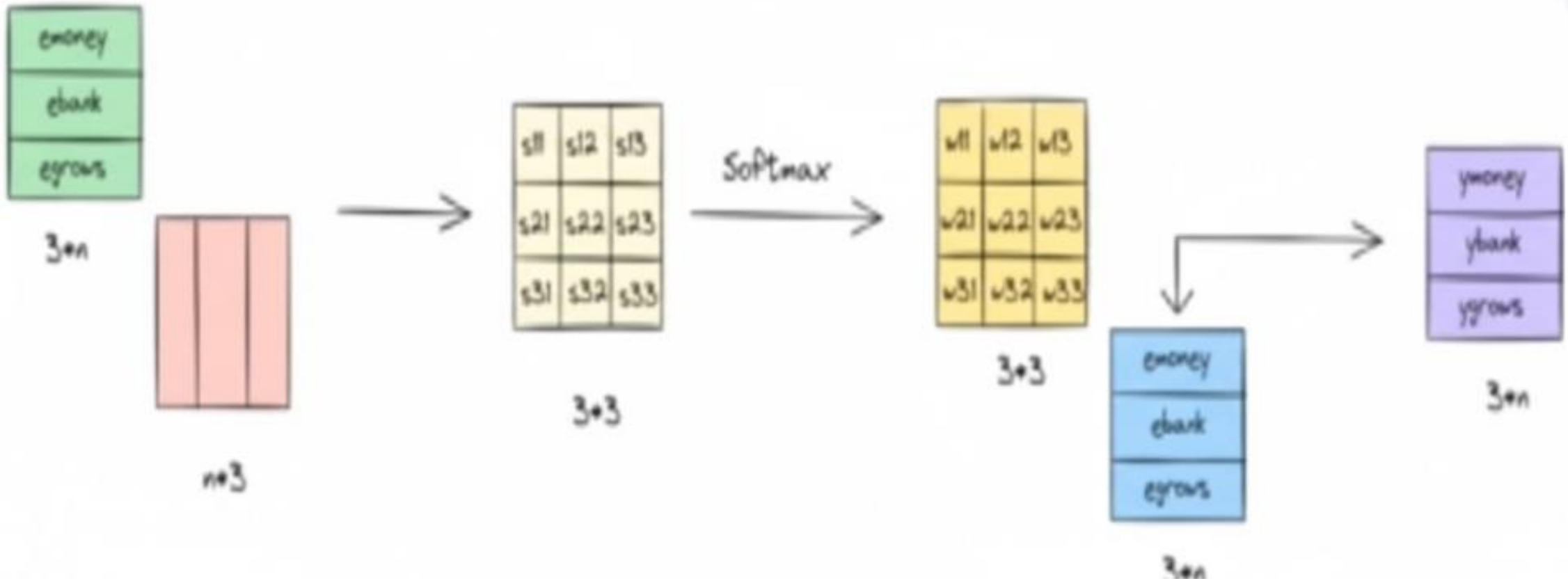
English	Urdu
How are you	کیسے ہو؟
I am good	میں ٹھیک ہوں
Piece of cake	بہت آسان کام

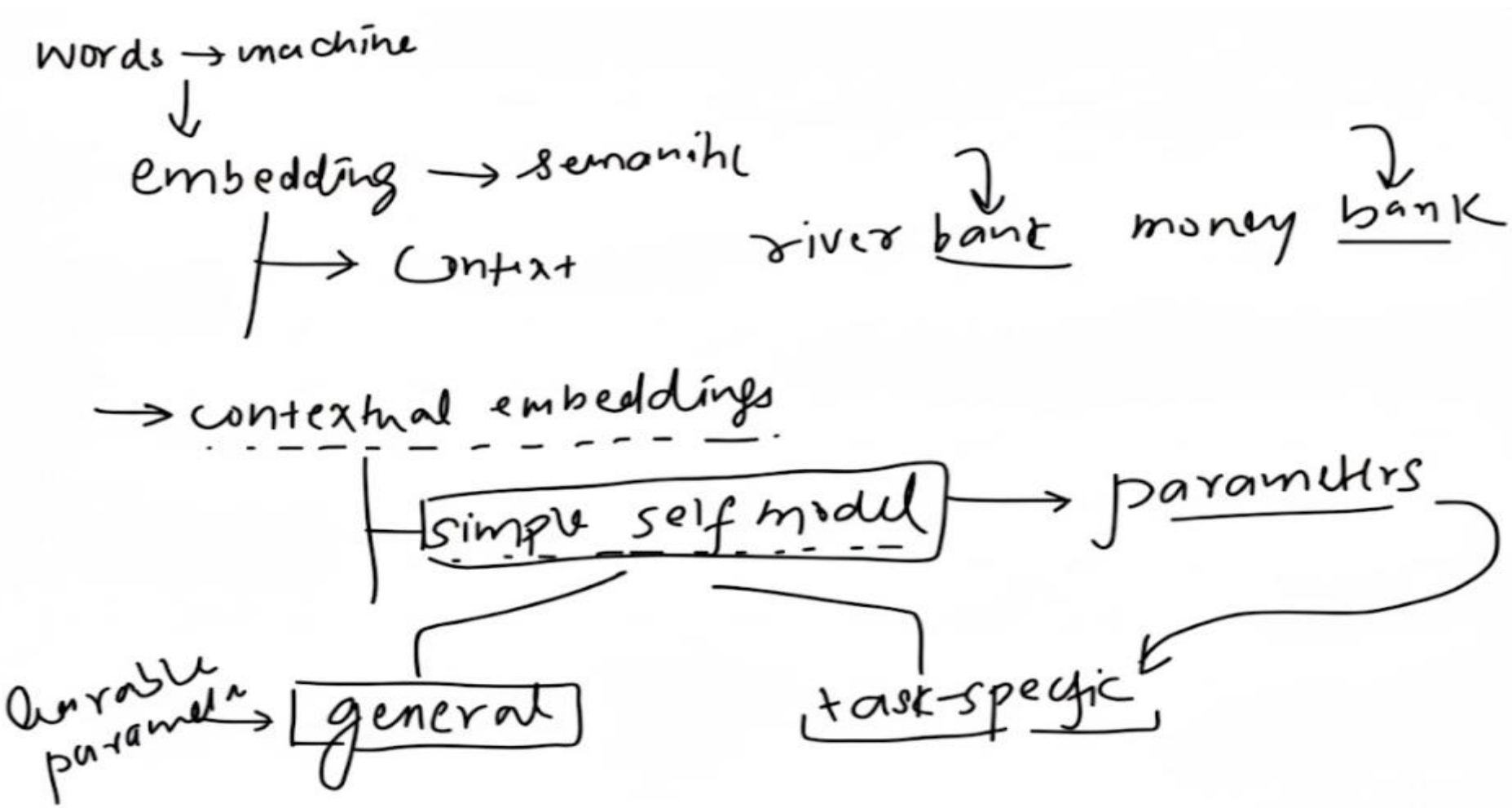
Not-Task Specific contextual Embedding



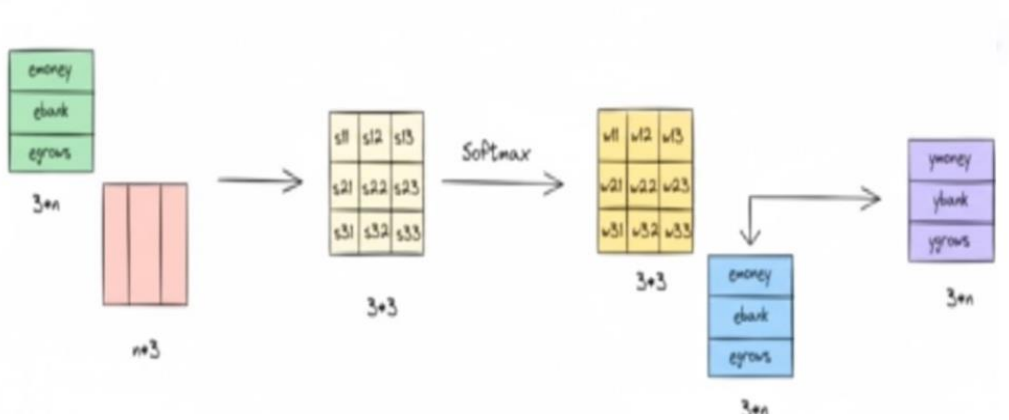
کیک کا ٹکڑا

break a leg کامیابی ہو

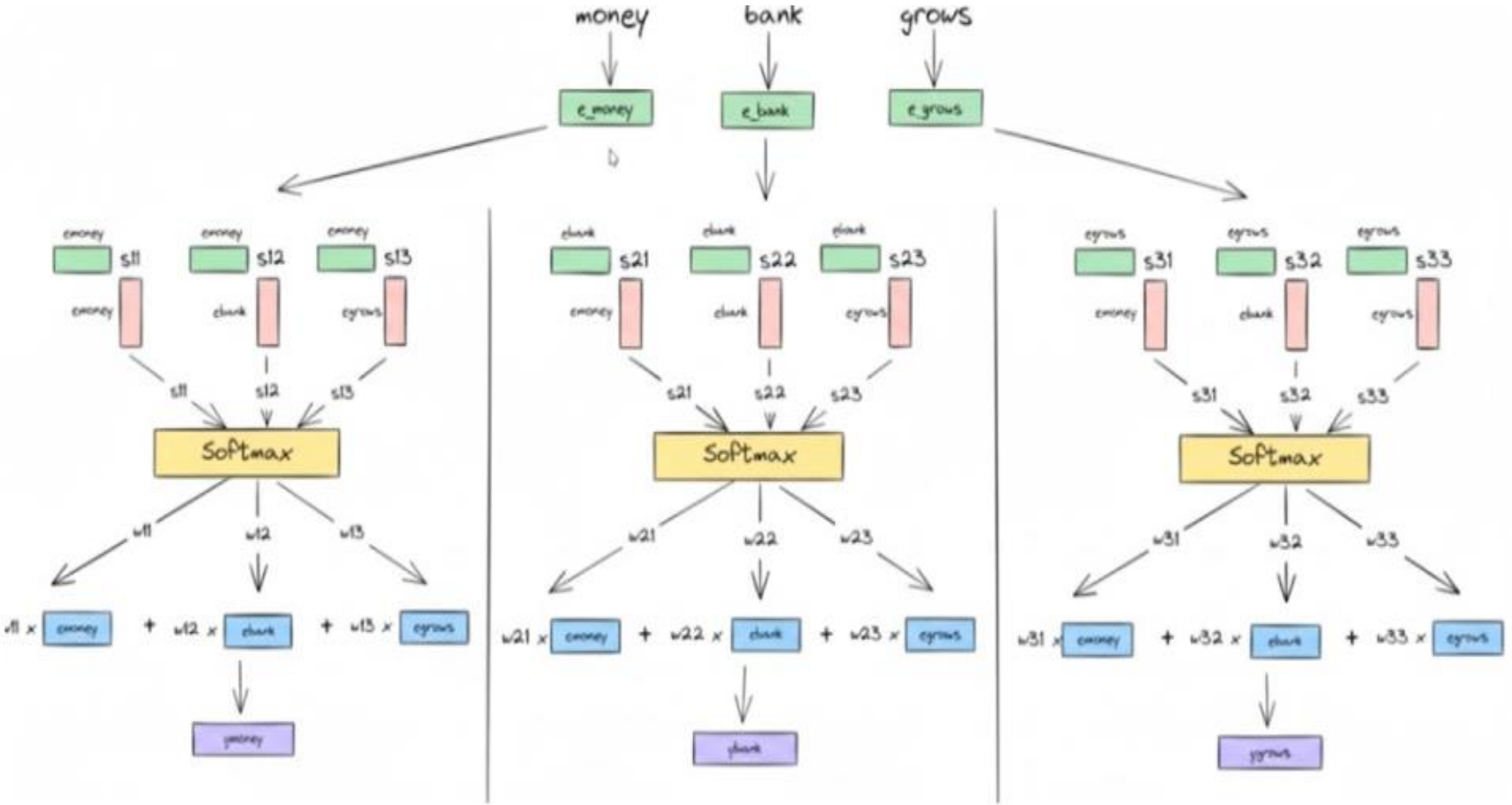

emoney
ebank
egrows
3*n
n*3
s11 s12 s13
s21 s22 s23
s31 s32 s33
3*3
Softmax
v11 v12 v13
v21 v22 v23
v31 v32 v33
3*3
emoney
ebank
egrows
3*n
ymoney
ybank
ygrows
3*n



To make it Task specific , we have to add some learnable parameters

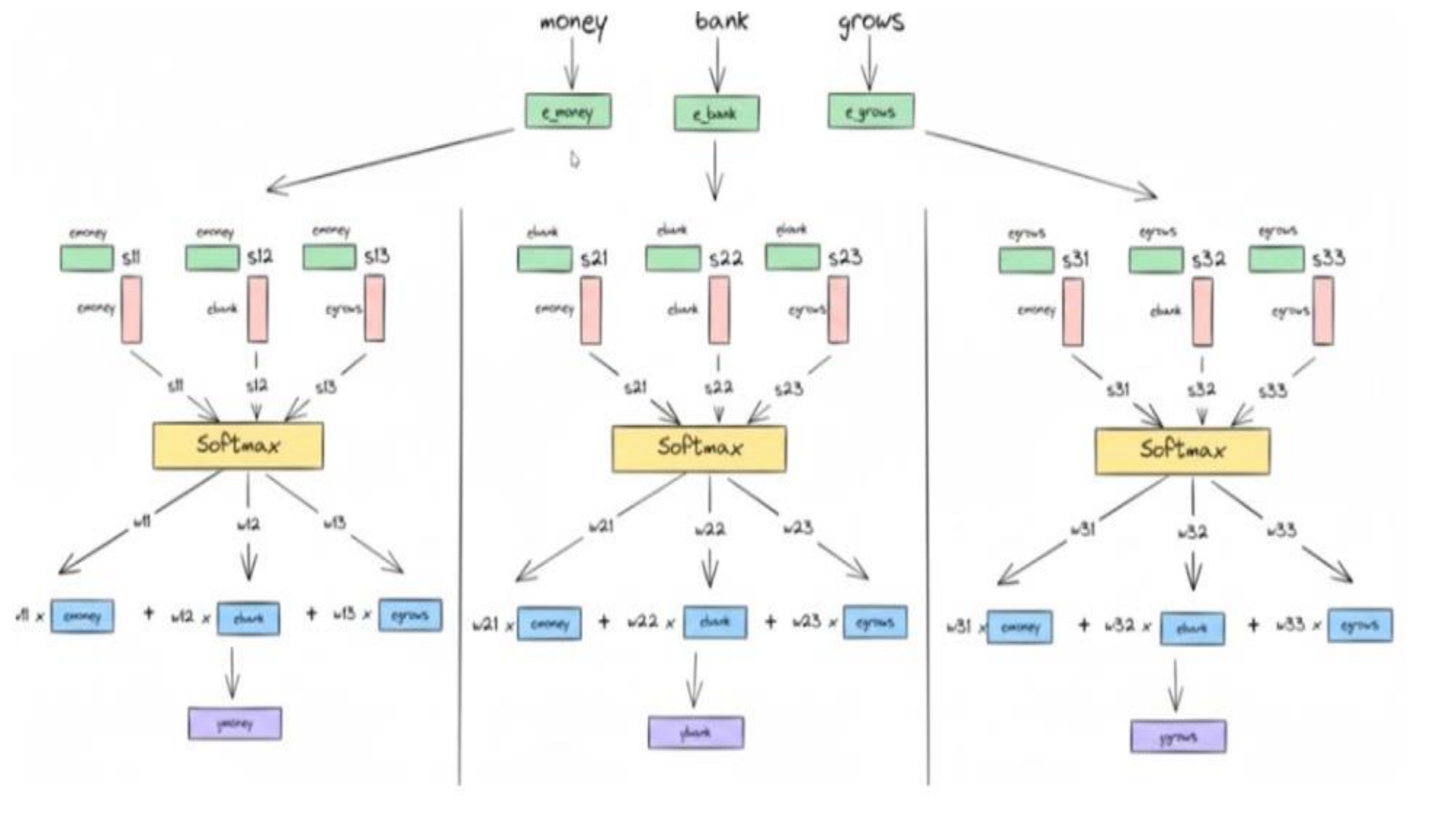


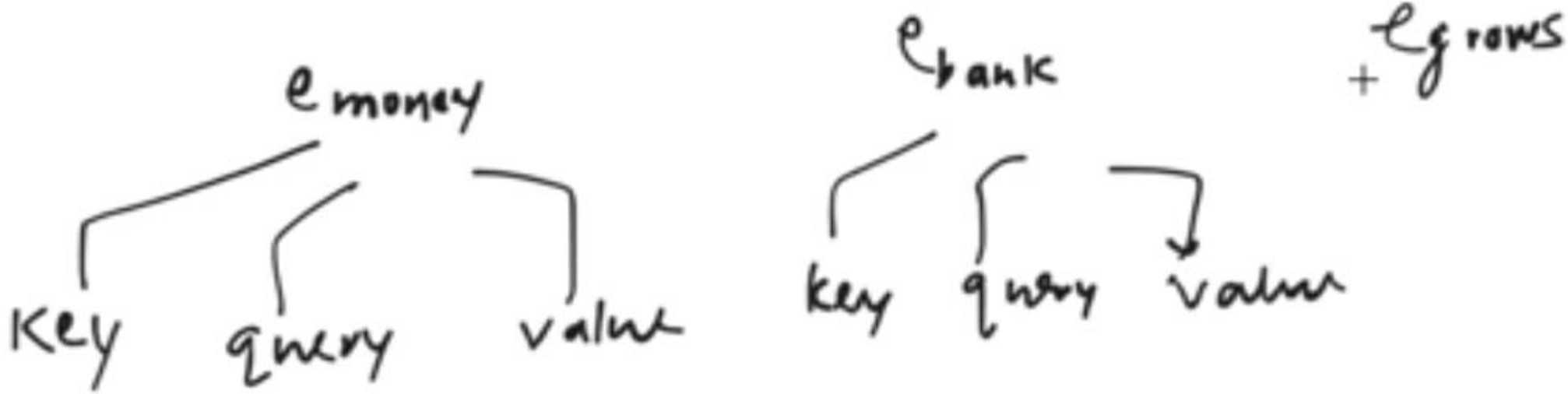
- Where will you add parameter

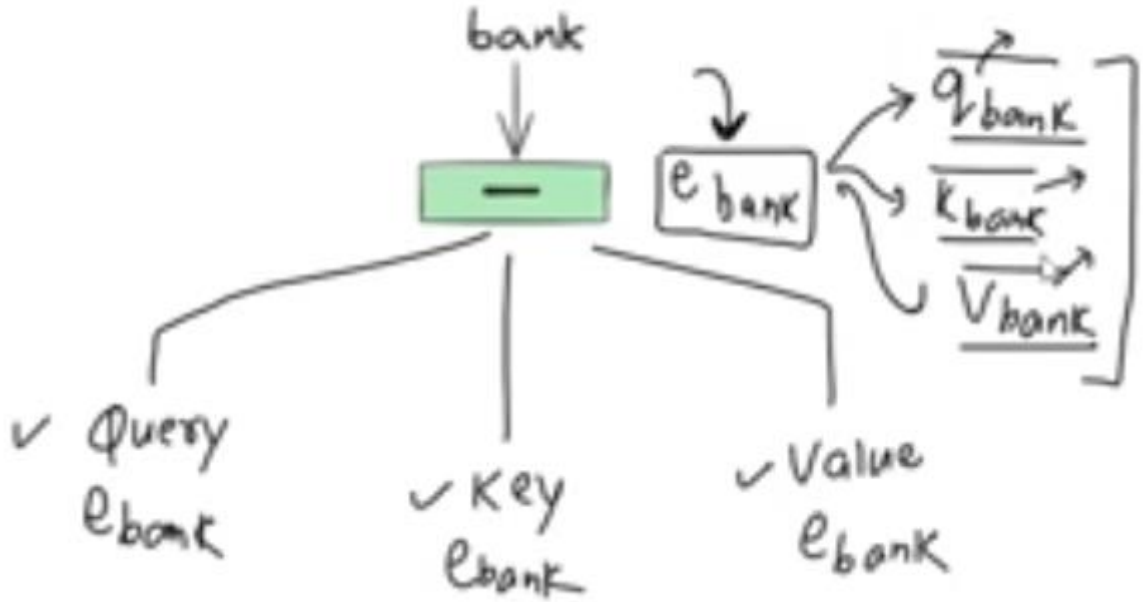


Query, Key , value

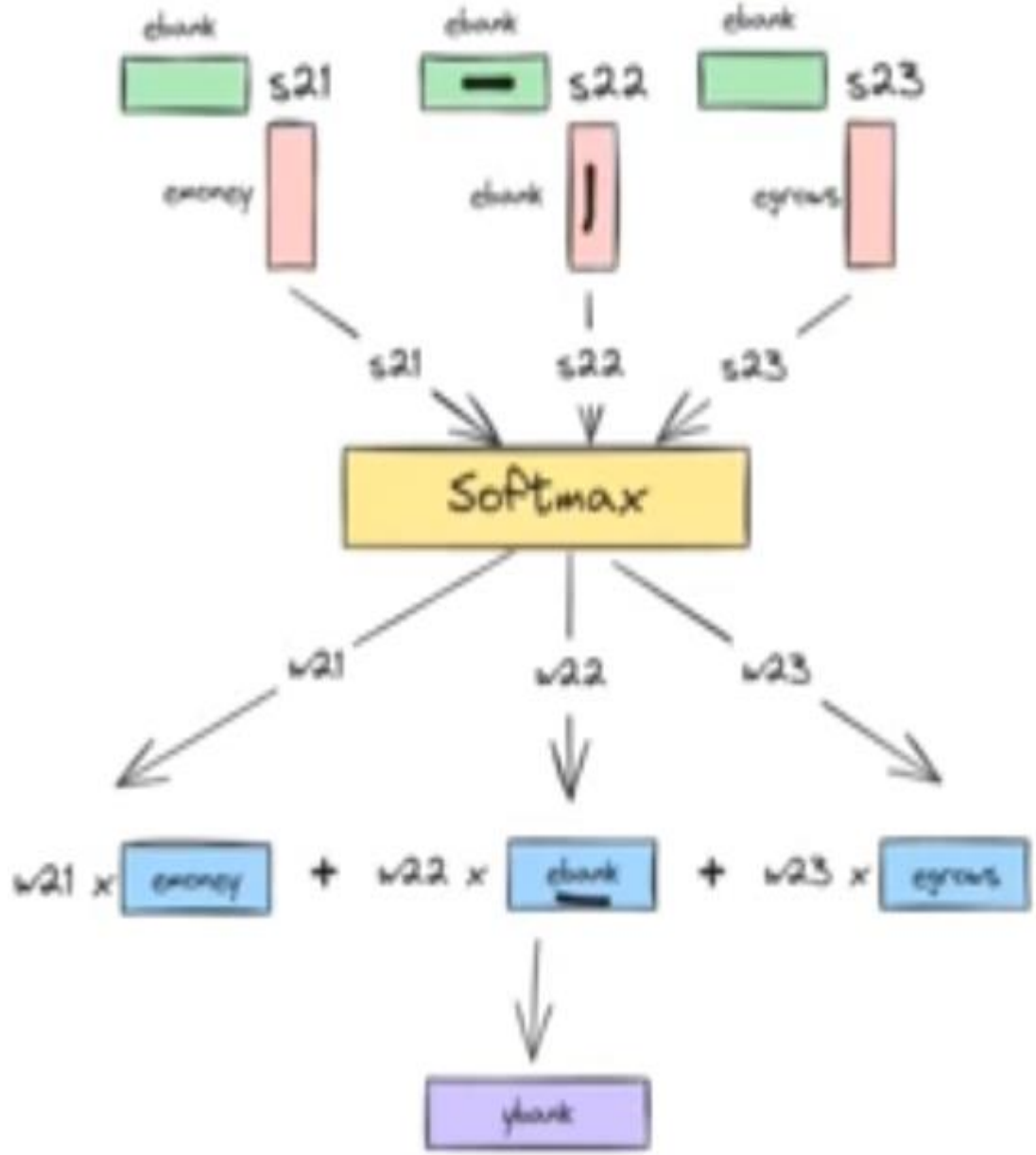
D=[a:2, b: 3, c:3]


money
bank
grows
e_money
e_bank
e_grows
Softmax
Softmax
Softmax


e money
key
query
value
e bank
key
query
value
+ e grows



Separation of Concerns



Person: Author/writer of books → autobiography
Age=35
Shadi.com

profile
specification/requirements
search(query)
match
meeting/call

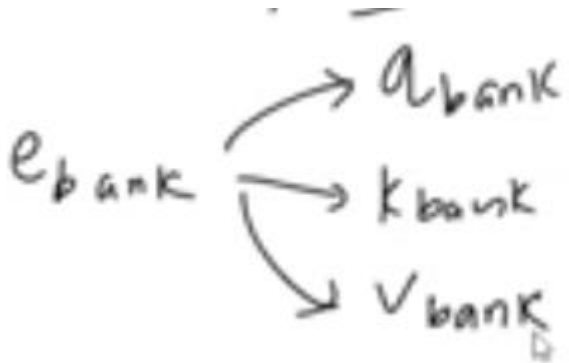
Ebank

key

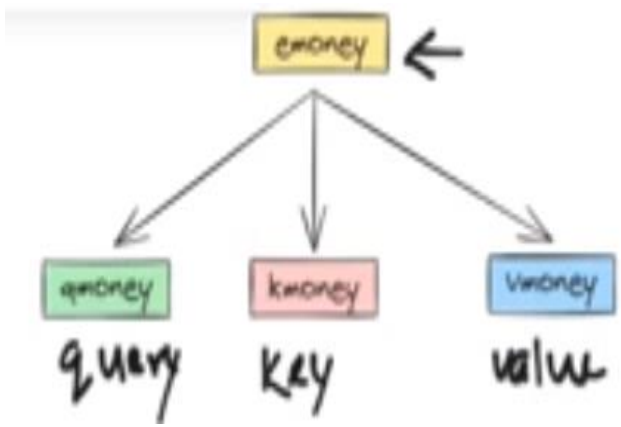
Query

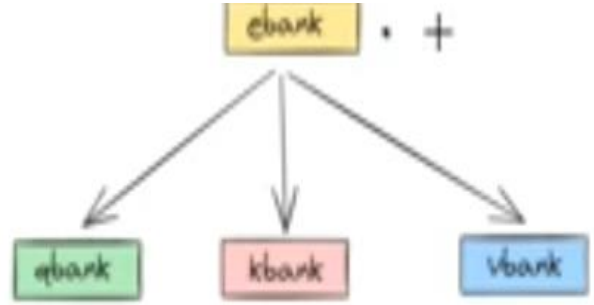
value

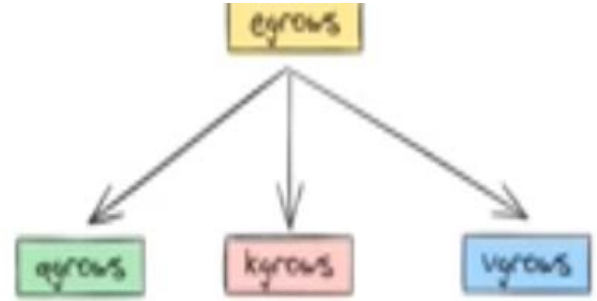
Autobiography
profile
search
match

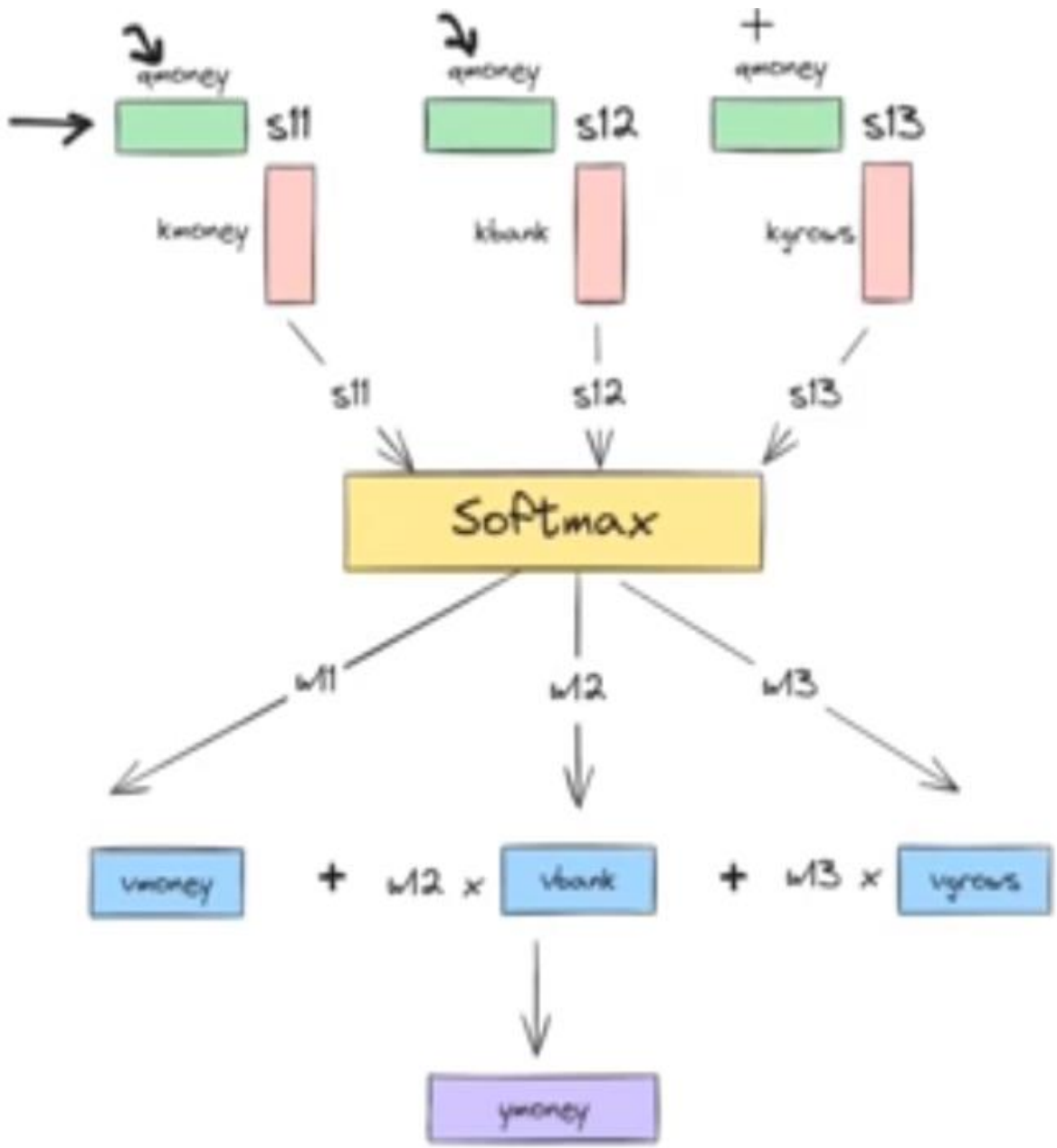


Data

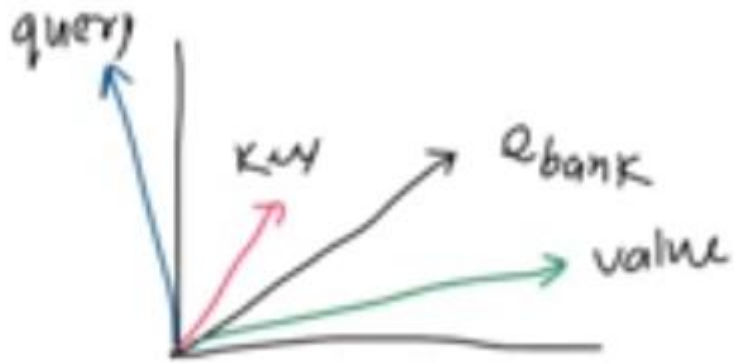

emoney
qmoney
kmoney
vmoney
query
key
value

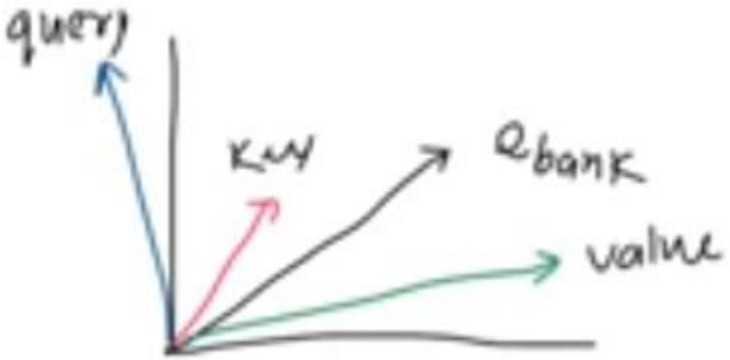

ebank
qbank
kbank
vbank


eyrows
qyrows
kyrows
vyrows

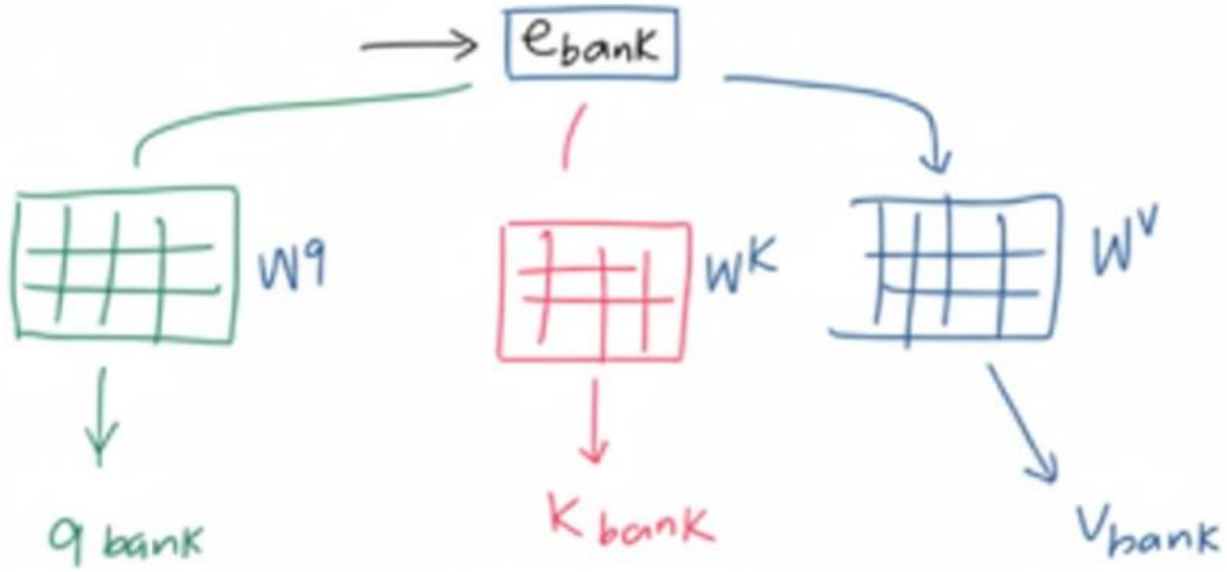

qmoney
kmoney
s11
qmoney
kbank
s12
qmoney
kgrass
s13
s11
s12
s13
Softmax
w11
w12
w13
vmoney
+
w12 x
vbank
+
w13 x
vgrass
ymoney

money →


query
Kbank
Qbank
value

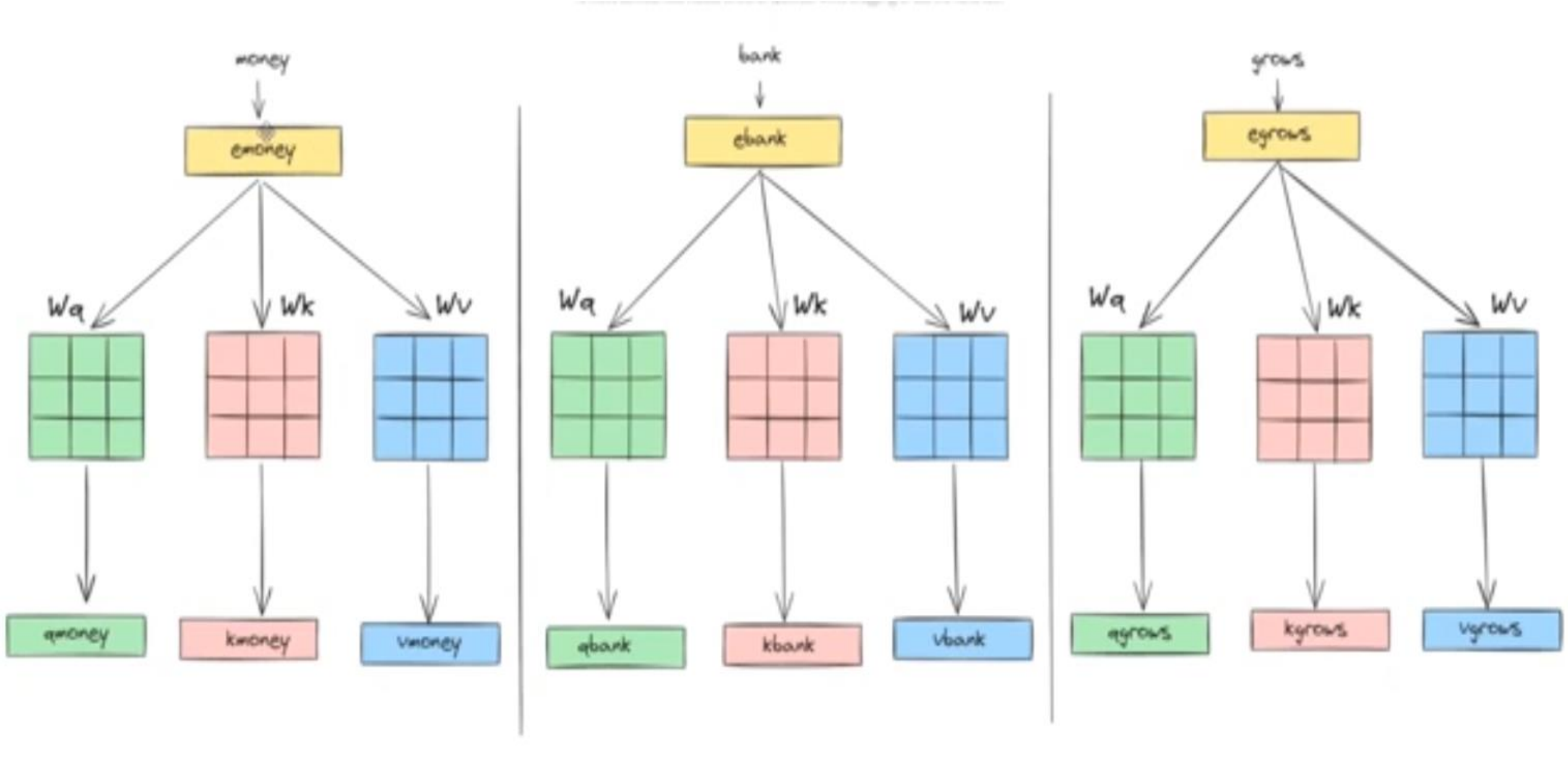


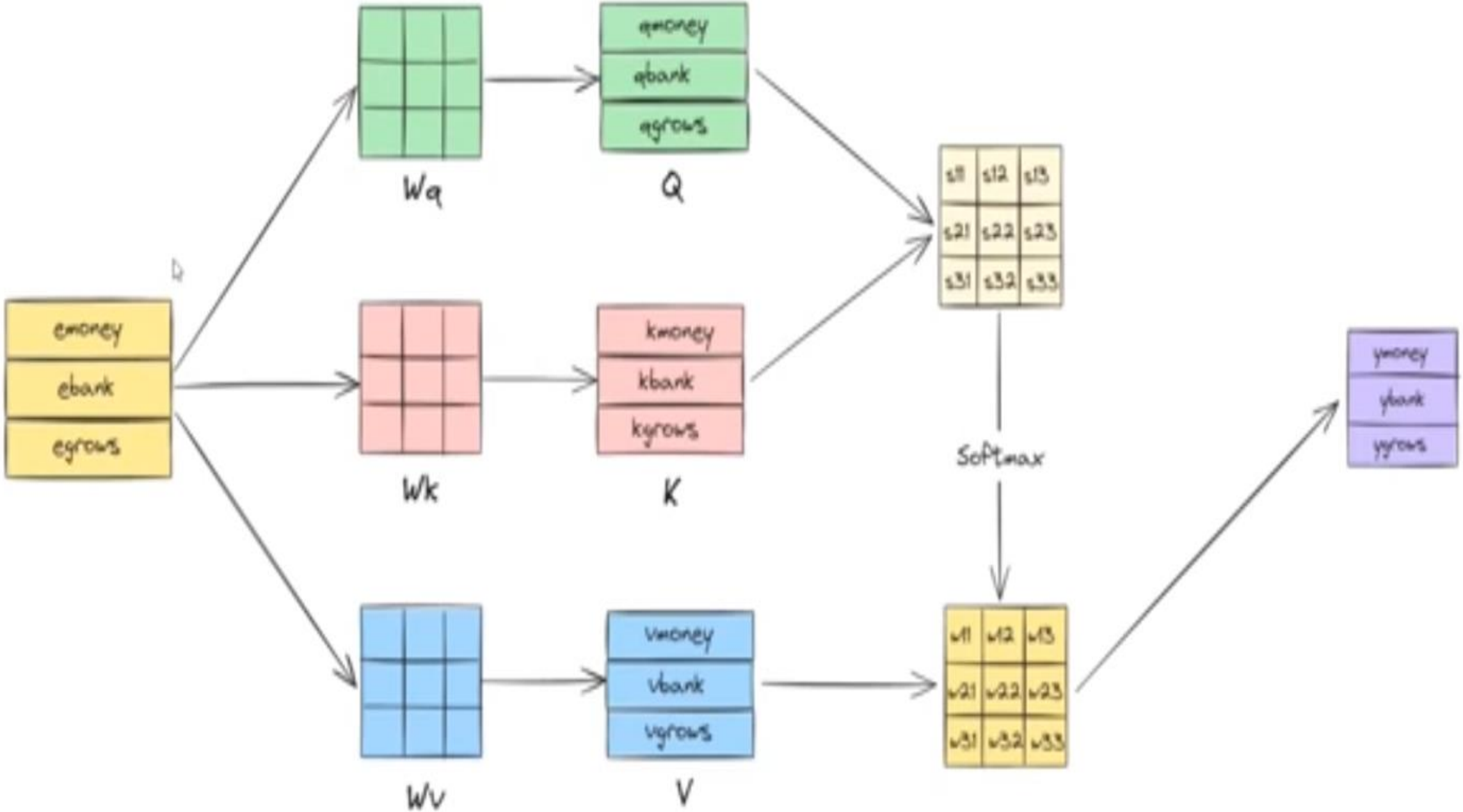
Change magnitude (increase or decrease)
Linear Transformation (change direction)



Training

Forward propagation
backward propagation


money
emoney
Wq
Wk
Wv
qmoney
kmoney
vmoney
bank
ebank
Wq
Wk
Wv
qbank
kbank
vbank
grows
egrows
Wq
Wk
Wv
qgrows
kgrows
vgrows


emoney
ebank
egrows
Wq
qmoney
qbank
qgrows
Q
Wk
kmoney
kbank
kgrows
K
Wv
vmoney
vbank
vgrows
V
Softmax
ymoney
ybank
ygrows