Word embedding techniques

CBOW, Skip gram

- Word2vec
- GloVe
- fastText

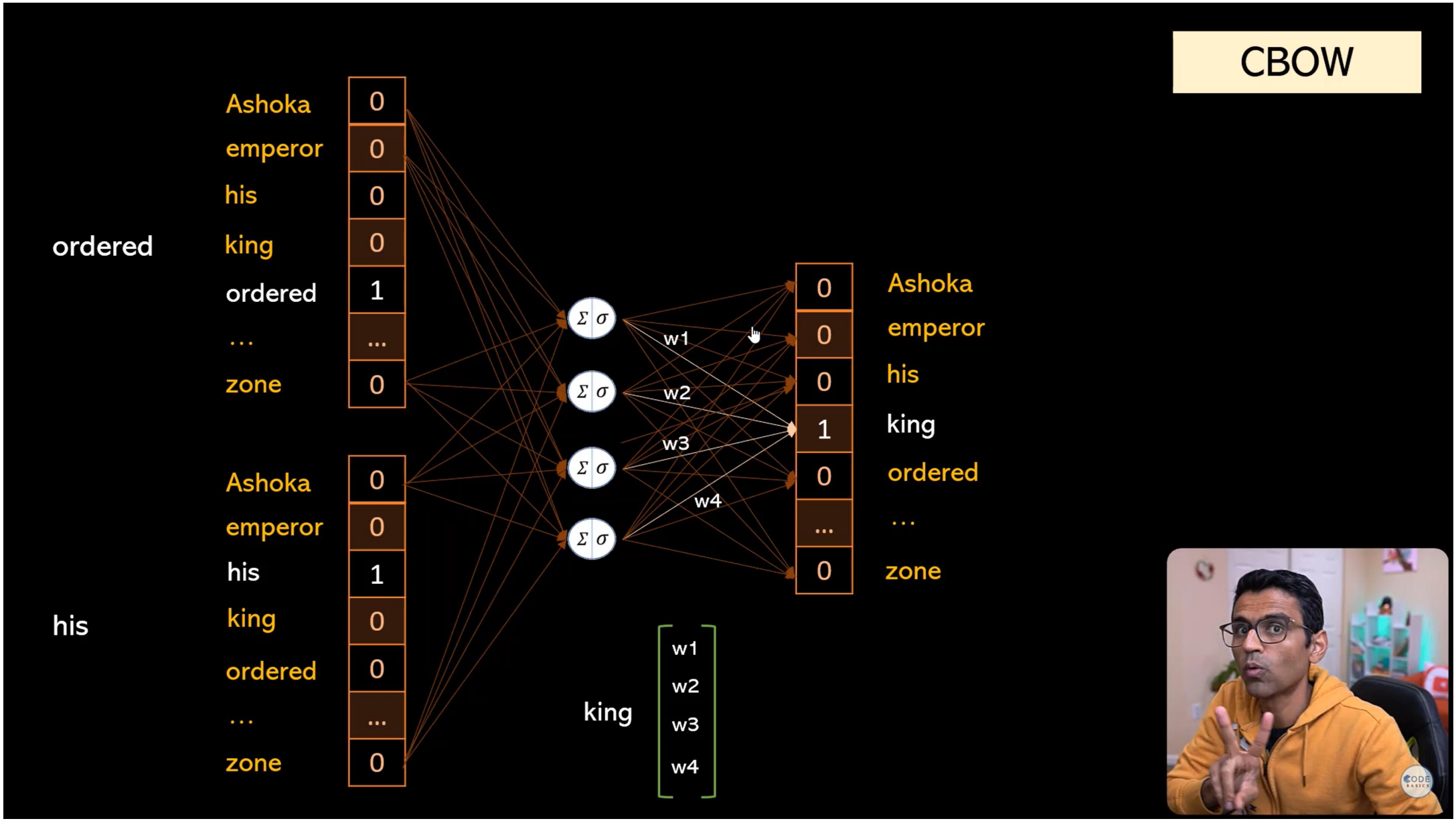
Based on transformer architecture

- BERT
- GPT

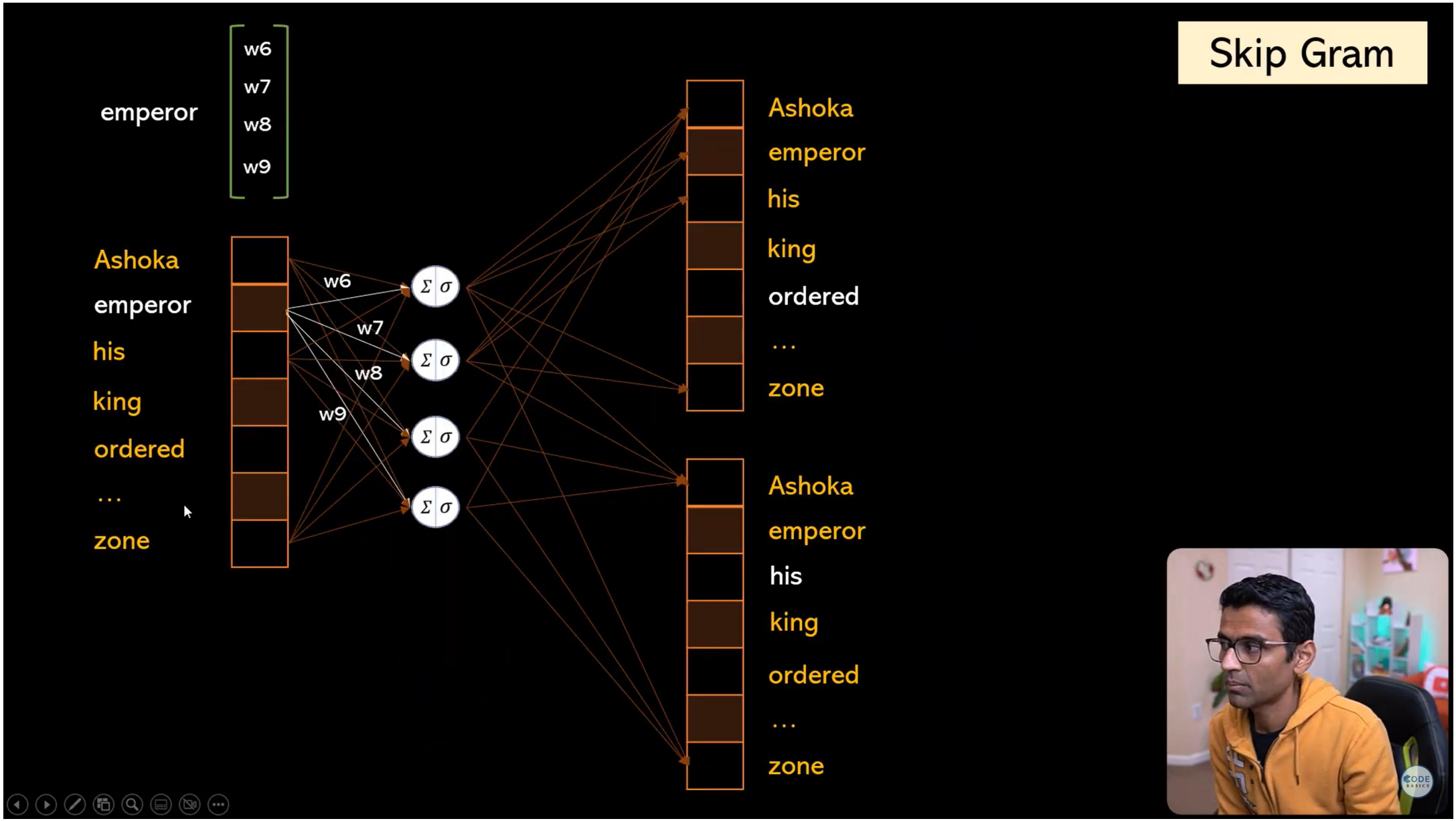
Based on LSTM

- ELMo

CBOW


ordered
his
Ashoka
emperor
his
king
ordered
…
zone
0
0
0
0
1
...
0
Ashoka
emperor
his
king
ordered
…
zone
0
0
1
0
0
...
0
Σ σ
w1
w2
w3
w4
king
w1
w2
w3
w4
0
0
0
1
0
...
0
Ashoka
emperor
his
king
ordered
…
zone

Skip Gram


emperor
w6
w7
w8
w9
Ashoka
emperor
his
king
ordered
…
zone
w6
w7
w8
w9
Σ σ
Σ σ
Σ σ
Σ σ
Ashoka
emperor
his
king
ordered
…
zone
Ashoka
emperor
his
king
ordered
…
zone

Word2vec

fastText

Word2vec

Unit on which neural network is trained is WORD

fastText

Word2vec

Unit on which neural network is trained is WORD

capable

fastText

Unit on which neural network is trained is CHARACTER n GRAM

Word2vec

Unit on which neural network is trained is WORD

capable

fastText

Unit on which neural network is trained is CHARACTER n GRAM

capable n = 3

Word2vec

Unit on which neural network is trained is WORD

capable

fastText

Unit on which neural network is trained is CHARACTER n GRAM

capable n = 3

cap

apa

pab

abl

ble

Word2vec

Unit on which neural network is trained is WORD

capable

capability OOV

fastText

Unit on which neural network is trained is CHARACTER n GRAM

capable n = 3

cap

apa

pab

abl

ble

fastText Fun Facts

fastText can handle OOV better than word2vec

fastText is often a first choice when you want to train custom word embeddings for your domain

fastText is a technique (similar to word2vec) as well as a library