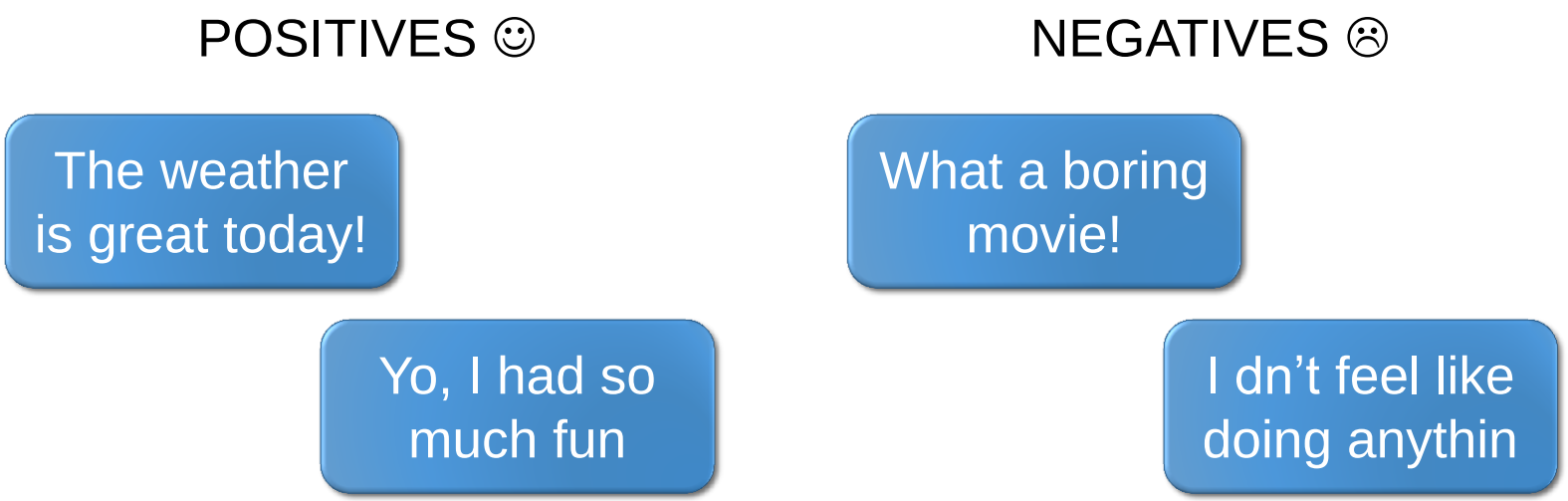
TEXT CLASSIFICATION

Themistoklis Diamantopoulos

Text Classification Problem

- Sentiment Analysis
  - Identify positive and negative emotions in text
  - Dataset from Twitter: http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/



Source: https://codeburst.io/use-tensorflow-dnnclassifier-estimator-to-classify-mnist-dataset-a7222bf9f940

Feature Representation

- Each word has an ID (lower ID ←→ higher frequency)

PHRASES

"Is is a common word“
"So is the“
"the is common“
"disco is not common"

SEQUENCES

[1, 1, 4, 2]
[1, 3]
[3, 1, 2]
[1, 2]

PADDED SEQUENCES

[1, 1, 4, 2]
[0, 0, 1, 3]
[0, 3, 1, 2]
[0, 0, 1, 2]

DICTIONARY

Word	Index	Word	Index
a	**4**	not	8
common	**2**	so	6
disco	7	**the**	**3**
is	**1**	word	5

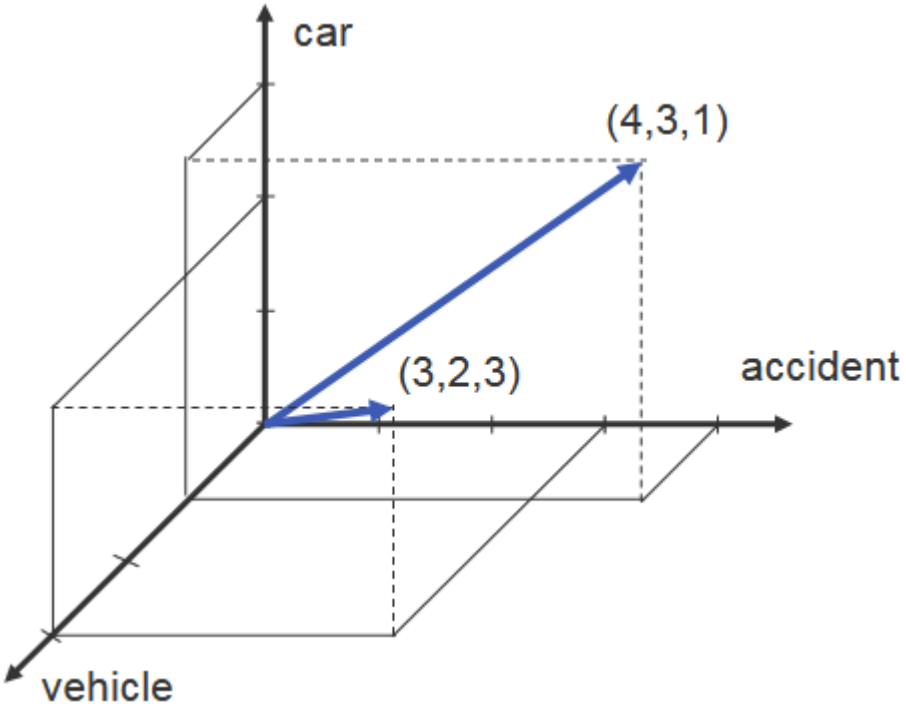
Source: http://www.developintelligence.com/blog/2017/06/practical-neural-networks-keras-classifying-yelp-reviews/

One-hot Encoding

- Document Representation (similar to Vector Space Model)
- Each word is a dimension
- Each document is a vector
- Can be used to find similar documents

WORD FREQUENCIES

	Doc1	Doc2
accident	4	3
car	3	2
vehicle	1	3



Source: http://slideplayer.com/slide/4174408/

Solution using MLP

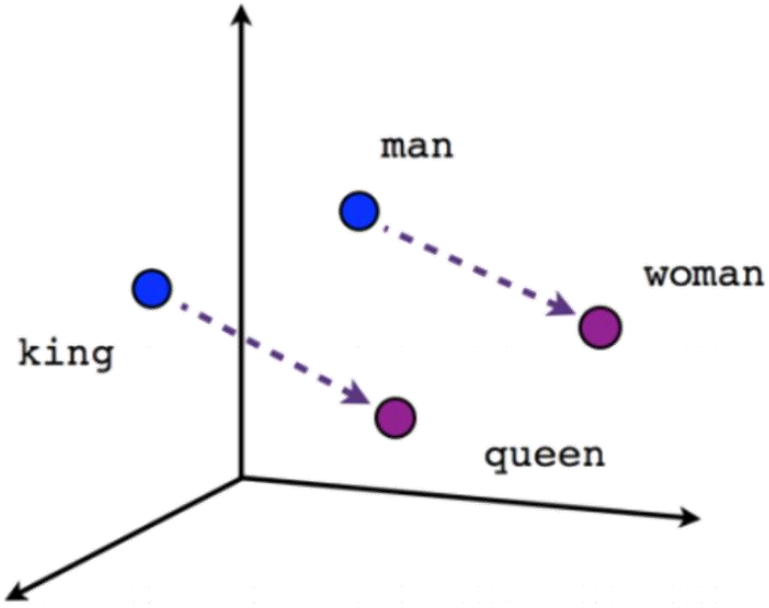
- 3-layer fully connected network
- Input vector size: 3000 (number of words)
- Output layer: 2 nodes
- 2 Intermediate layers

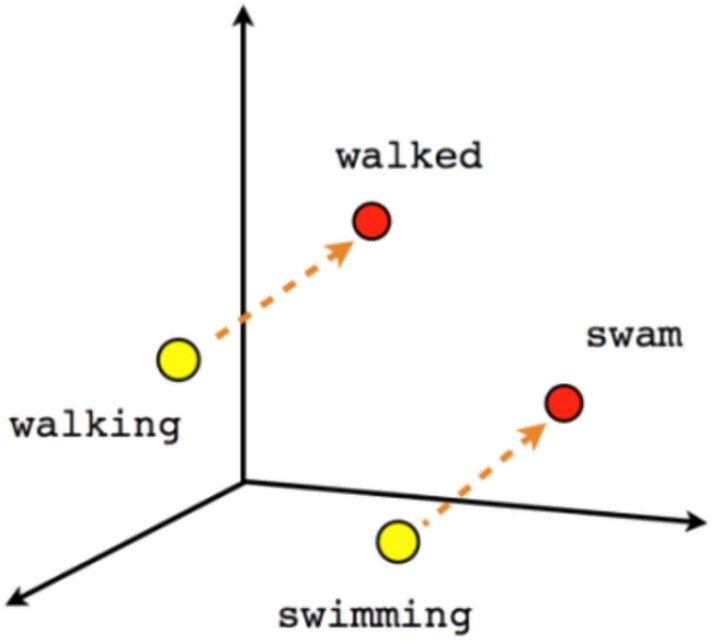
Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 512)	1536512
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 256)	131328
dropout_4 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 2)	514

Source: https://github.com/wxs/keras-mnist-tutorial/blob/master/MNIST%20in%20Keras.ipynb

Word Embedding

- Distributed representation of words
- Each word is a vector
- Similar words have similar vectors
- Can be used to generate analogies



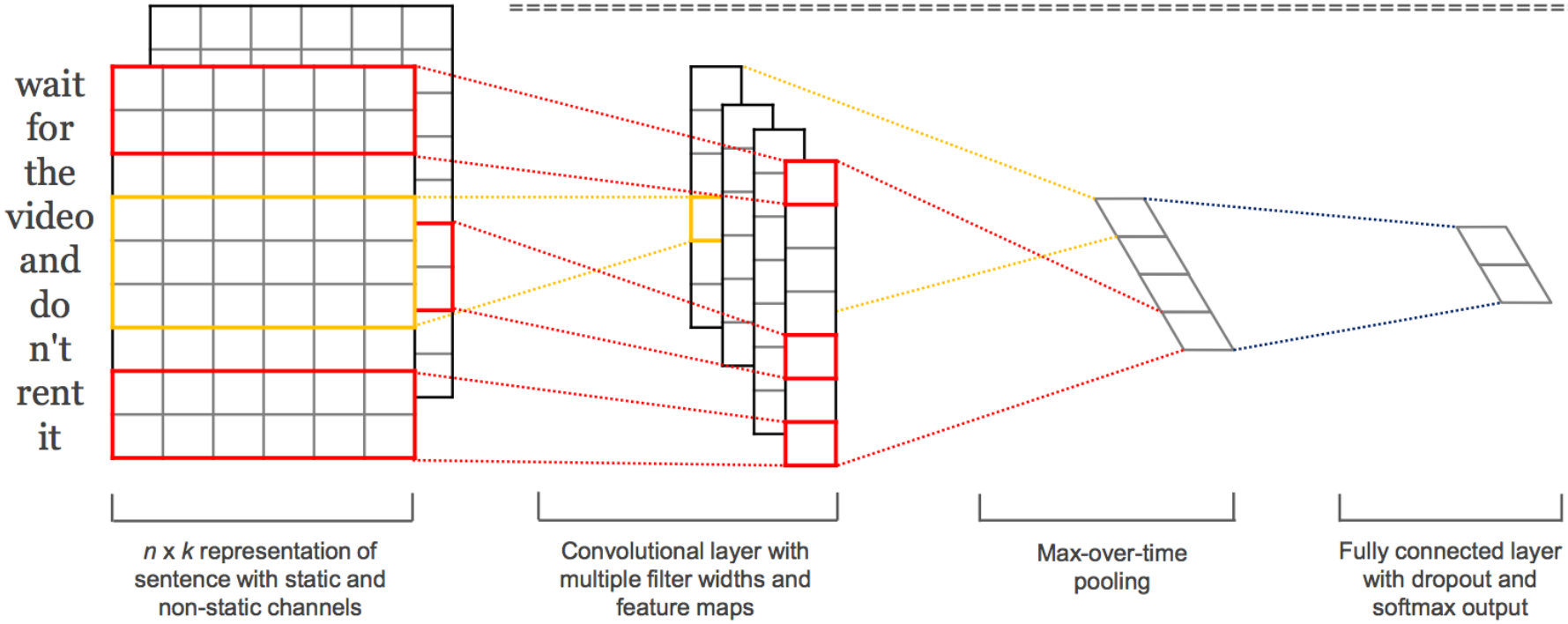


Source: http://www.developintelligence.com/blog/2017/06/practical-neural-networks-keras-classifying-yelp-reviews/

Solution using CNN

- Embedding
- 1 Conv. layer
- Connected MLP

Layer (type)	Output Shape	Param #
embedding_7 (Embedding)	(None, 300, 64)	192000
conv1d_2 (Conv1D)	(None, 299, 100)	12900
global_max_pooling1d_2 (Glob	(None, 100)	0
dense_7 (Dense)	(None, 256)	25856
dense_8 (Dense)	(None, 1)	257

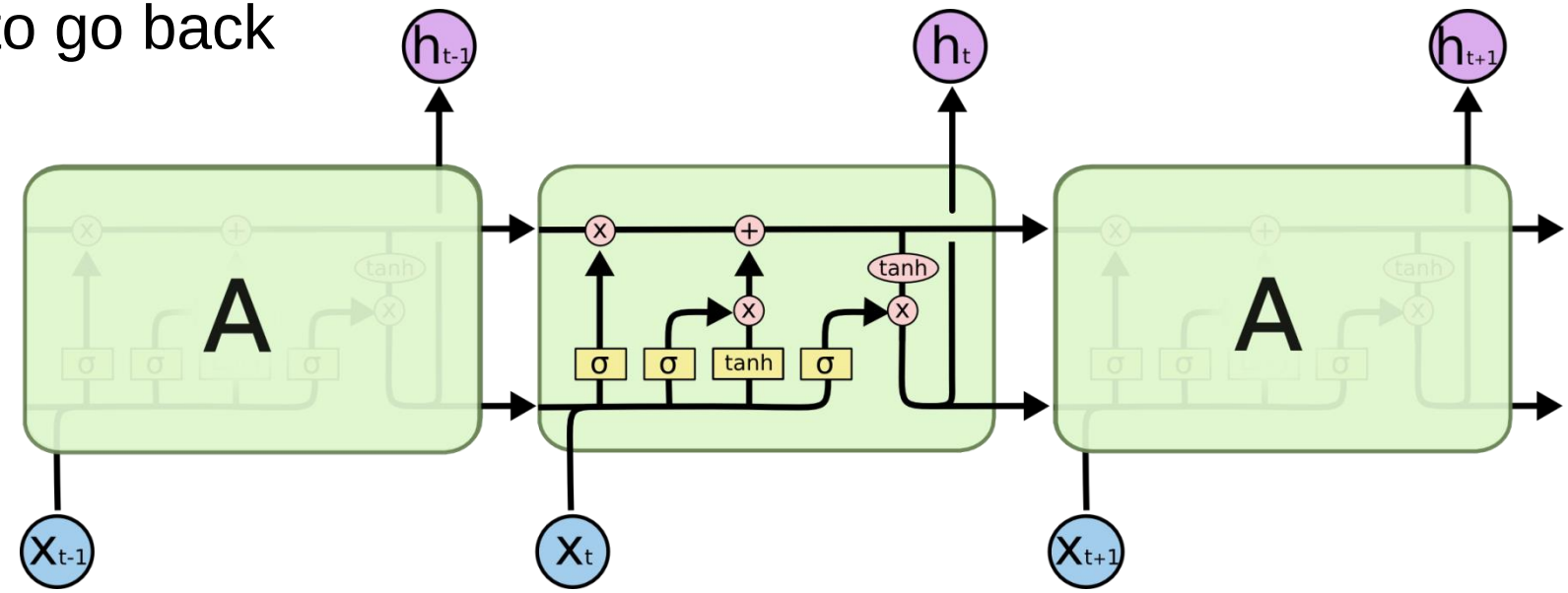


Source: https://machinelearningmastery.com/best-practices-document-classification-deep-learning/

Solution using LSTM

- Embedding
- 1 LSTM
- Like RNN but also determines how much to go back

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 300, 64)	192000
lstm_1 (LSTM)	(None, 64)	33024
dense_1 (Dense)	(None, 1)	65



Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/