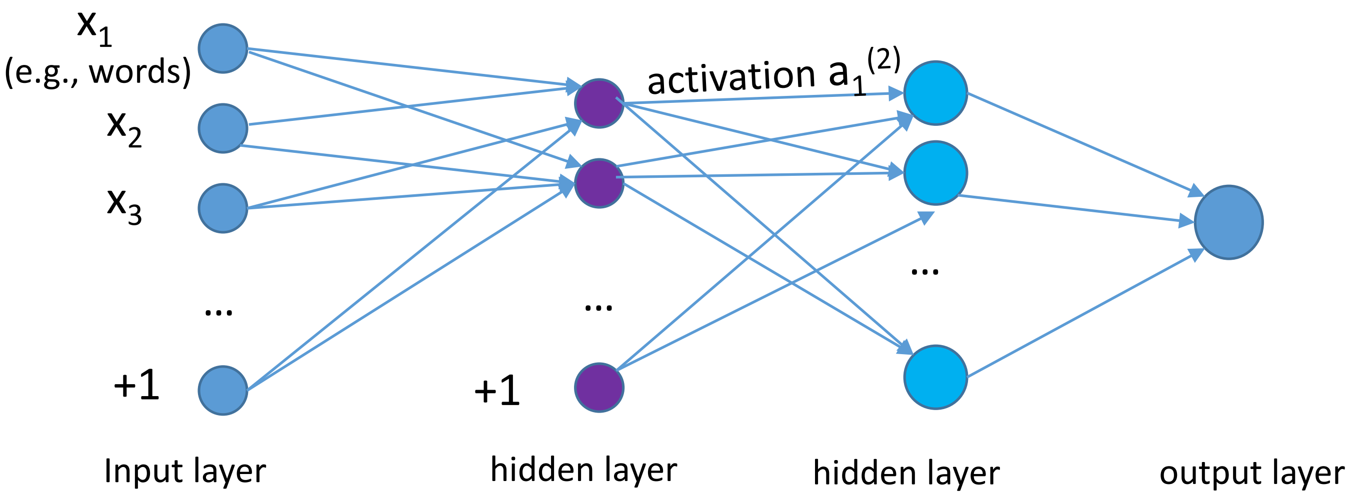
Deep Learning – What is it?

- Representation learning for automatically learning good features or representations
  - <u>Representational learning</u>: "learning representations of the data that make it easier to extract useful information when building classifiers or other predictors" (Bengio, Courville, & Vincent, 2013)



http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.

Deep Learning

- Deep learning models are not new (started ~1960s)
- In 2006 deep learning methods start to outperform other machine learning methods
  - A lot of data
  - Faster machines, GPU
  - New models/algorithms
- A history of deep learning models can be found at:

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

https://arxiv.org/pdf/1404.7828.pdf

Deep Learning in NLP

Language analysis

- Speech
- Morphology
- Syntax
- Semantics

- Machine translation
- Sentiment analysis
- Question answering

Word Representation in Deep NLP

- One-hot representation

In a vocabulary set, each word is represented as a vector. For example, if word *chair* is the 5391th word in that vocabulary, we can represent it as

$$\mathbf{O}_{5391} = \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ 0 \\ 0 \\ \ldots \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

What is a problem with this approach?

Word Representation in Deep NLP

- Featurized representation: word embedding

	Desk	Chair	Mom	Dad	Son	Orange
Gender	0	0	1	-1	-1	0
Age	0.45	0.66	0.85	0.84	0.68	0.02
Food	0.02	0.02	0.01	0.01	0.04	0.96
Size	…					
Cost						
Alive						
Furniture						
…						

E

$\mathbf{e}_{5391}$

Word Representation in Deep NLP

- Featurized representation: word embedding

	Desk	Chair	Mom	Dad	Son	Orange
Gender	0	0	1	-1	-1	0
Age	0.45	0.66	0.85	0.84	0.68	0.02
Food	0.02	0.02	0.01	0.01	0.04	0.96
Size	…					
Cost						
Alive						
Furniture						
…						

E

$\mathbf{e}_{5391}$

$$\mathbf{e}_{5391} = E \cdot O_{5391}$$

Word Representation in Deep NLP

- Featurized representation: word embedding

	Desk	Chair	Mom	Dad	Son	Orange
Gender	0	0	1	-1	-1	0
Age	0.25	0.30	0.85	0.84	0.68	0.02
Food	0.02	0.02	0.01	0.01	0.04	0.96
Size	…					
Cost						
Alive						
Furniture						
…						

300d

E

$\mathbf{e}_{5391}$

Mom told me to do so

______ told me to do so

$\mathbf{e}_{\mathbf{mom}} - e_{\mathbf{dad}}$ $\mathbf{e}_{\mathbf{son}} - e_{\mathbf{dad}}$ $\mathbf{e}_{\mathbf{desk}} - e_{\mathbf{dad}}$

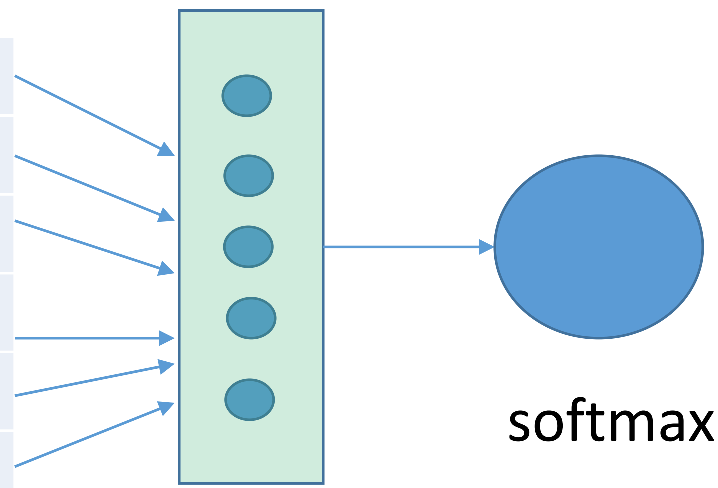
Word Representation in Deep NLP

- Featurized representation: word embedding
- Good when the task has a small labeled training set and a very large unlabeled training set
  - Small labelled training set: dimensions/features of the words are humanly labeled for a small set of texts
  - Large online texts for automatic labelling
- Analogy reasoning: 30 – 75% accuracy
  - King to Queen is as Man to ______

Deep NLP - language model

- I want a glass of orange ___________
- Each word has an embedding e

I	O_{4343}	$E \cdot O$	e_{4343}
want	O_{9665}	$E \cdot O$	e_{9665}
a	O_{1}	$E \cdot O$	e_{1}
glass	O_{3852}	$E \cdot O$	e_{3852}
of	O_{6163}	$E \cdot O$	e_{6163}
orange	O_{6257}	$E \cdot O$	e_{6257}

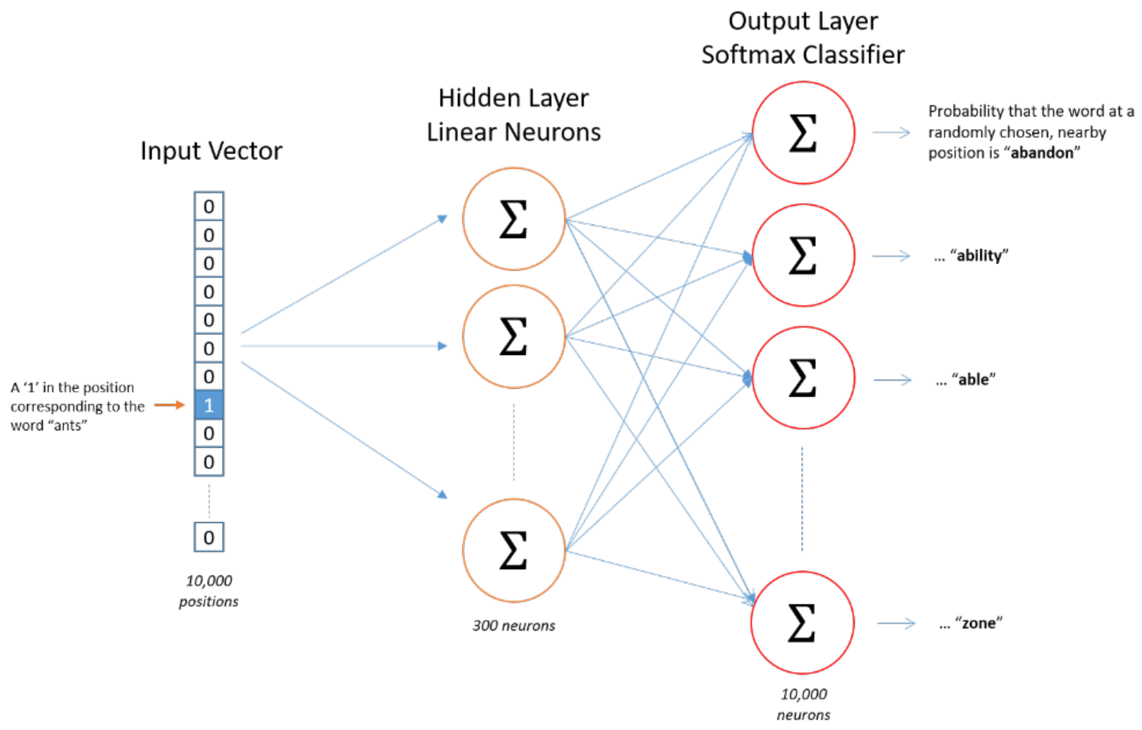


300X 6

Or choose a window instead of using the whole sentence

Deep NLP – How Do You Get E?

- Word2Vec



Goal: Learn the hidden layer weight matrix -> E

Image: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Deep NLP - Word2Vec (Skip-gram)

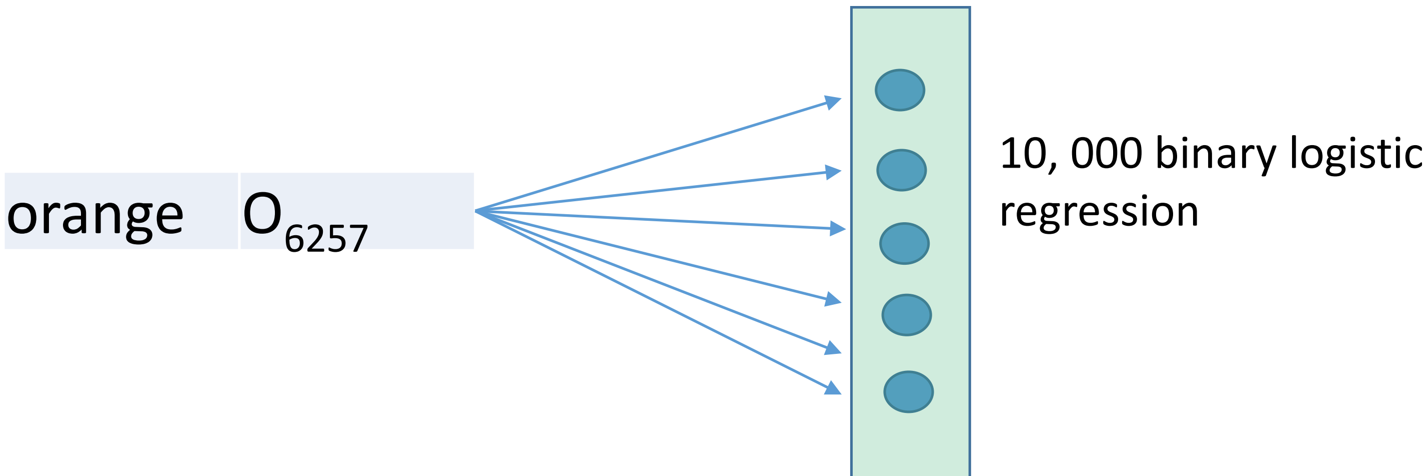
- Skip-gram (one of the two algorithms in word2vec; the other one is CBOW)
  - Learn word embedding using other contexts
  - Use nearby words instead of the whole sentence
  - To learn good word embedding
- Algorithm
  - Pick a context word
  - Input: one-hot vector of the context word
  - Output: probability of a word being the target word (Softmax function)
  - Problem: the sum of vocabulary is necessary when calculating each probability
  - One solution: hierarchical Softmax (Huffman tree)

Deep NLP - Word2Vec (Skip-gram)

- Skip-gram negative sampling (SGNG)
  - Pick a context word and a target word (in the window)
  - For k times, we take random words from the dictionary and label them all 0 (negative)
    - K = 5 – 20 for smaller datasets
    - K = 2 – 5 very large datasets
  - The input of the algorithm: one-hot vector of the context word
  - The output of the algorithm: the probability of a word from the dictionary being the target word near to the context word (supervised learning; probability of y = 1 given the context and the chosen words, logistic regression)

Deep NLP - Word2Vec (Skip-gram)

Skip-gram negative sampling (SGNG)



Sample negative words:

- More frequent words are more likely to be selected as negative samples
- Proportional to the frequency of the words (to the ¾)

Deep NLP – GloVe

- Global vector representation

I want a glass of orange juice to go along with my cereal

X_{ij} = the no. of times i appears in the context of j

Context: close proximity (e.g., within 3 words)

Function:

- the inclusion of X_{ij} ($-\log X_{ij}$)
- The weighting function f
  - To address $X_{ij} = 0$
  - To consider the problem of X_{ij} by stop words and rare words

Deep NLP – Sentiment Analysis

The	dessert	is	excellent
8928	2468	4694	3180

The	O_{8928}	$E \cdot O$	e_{8928}
Dessert	O_{2468}	$E \cdot O$	e_{2468}
Is	O_{4694}	$E \cdot O$	e_{4694}
Excellent	O_{3180}	$E \cdot O$	e_{3180}

300d

Sum or Average them

300d

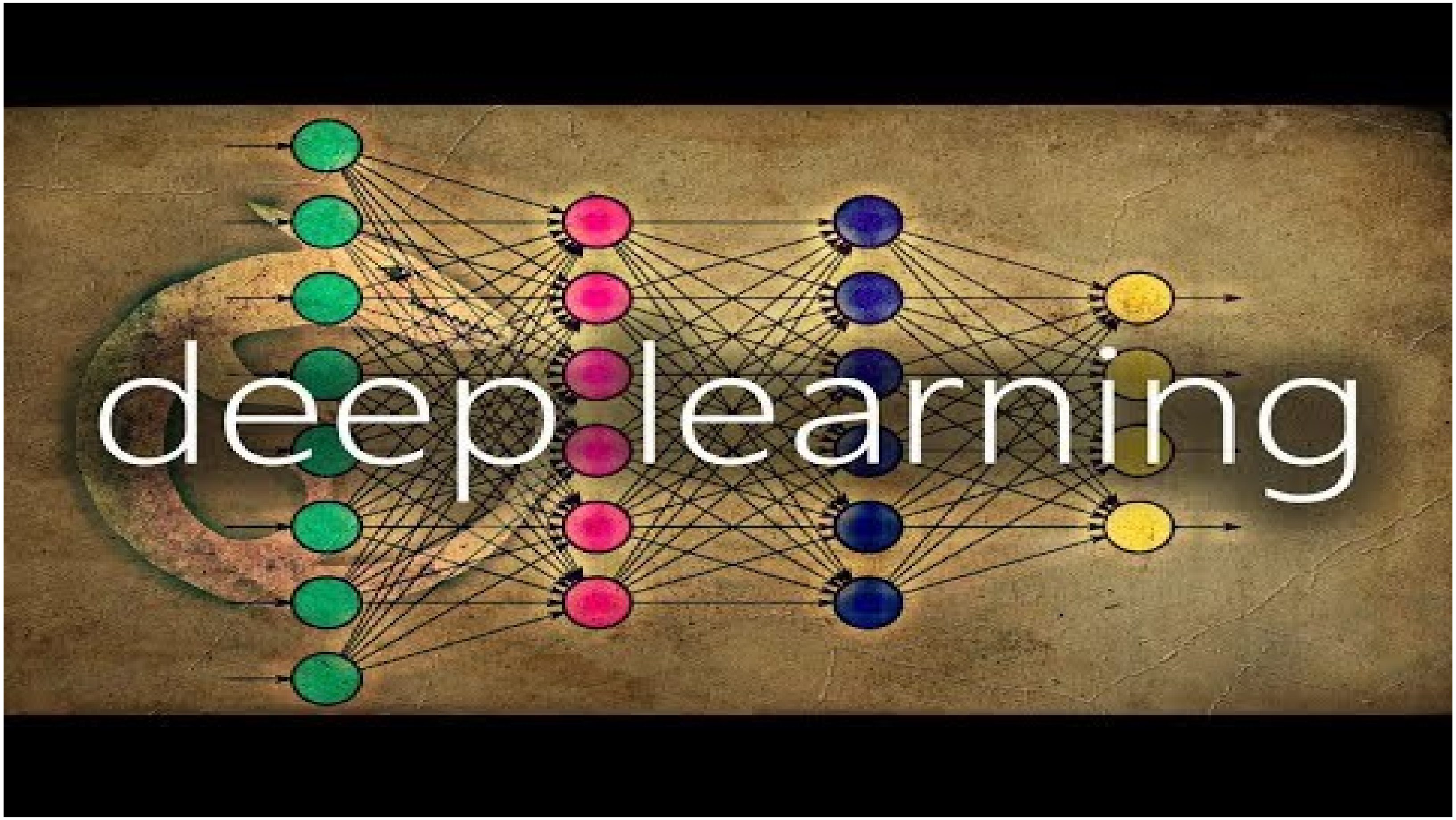
softmax

y

Short or long review -> sum or average, so does not matter

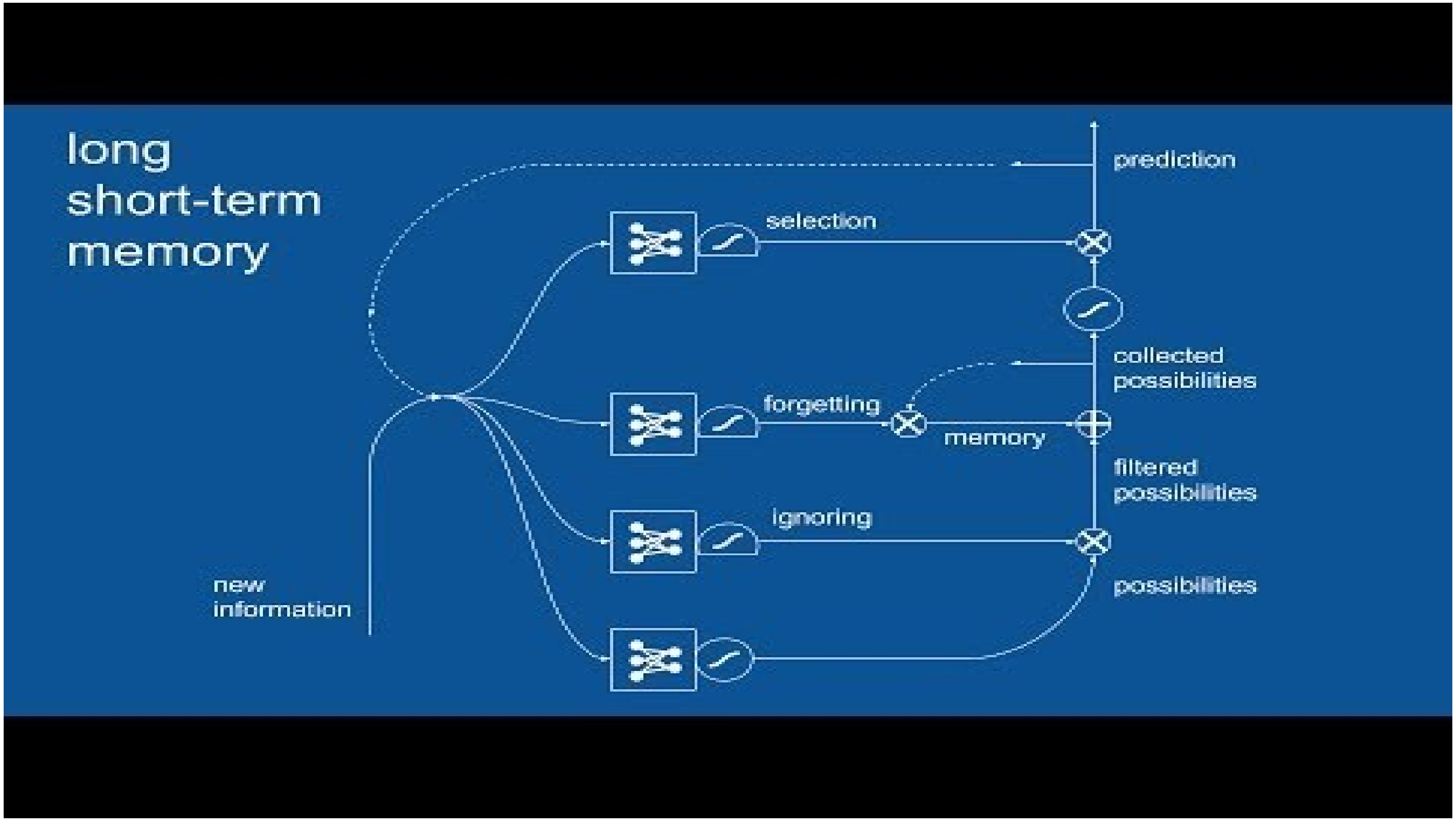
Problem: ignore the word order, the syntactic structure

Deep Learning - CNN



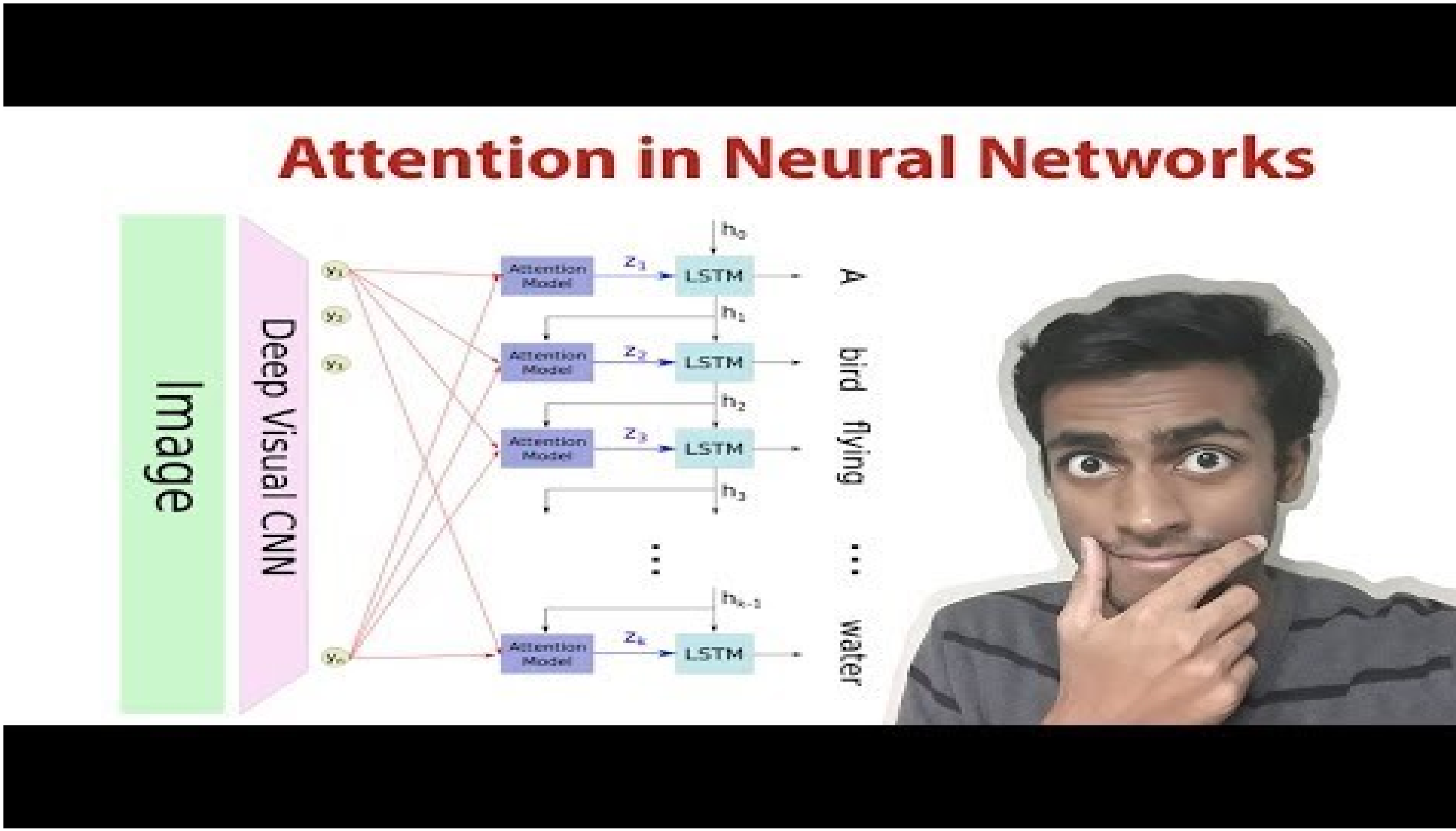
https://www.youtube.com/watch?v=YRhxdVk_sIs

Deep Learning – RNN + LSTMs



https://www.youtube.com/watch?v=WCUNPb-5EYI

Deep Learning – Attention Models



https://www.youtube.com/watch?v=W2rWgXJBZhU

Local interpretable model-agnostic explanation (LIME)

- Interpretable Machine Learning Using LIME Framework

https://www.youtube.com/watch?v=CY3t11vuuOM

Github access: https://github.com/marcotcr/lime

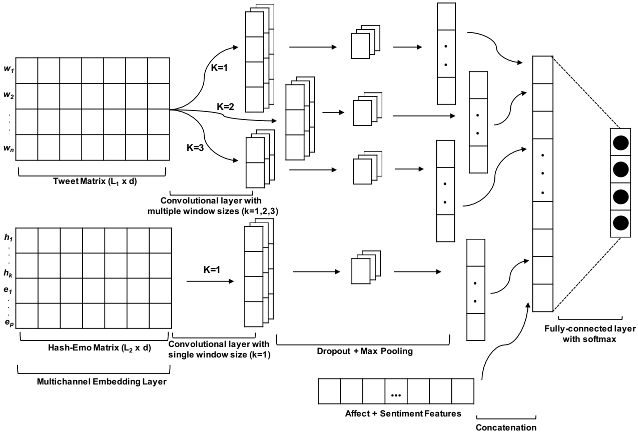
Emotion Recognition

- Emotion recognition in computational linguistics is the process of identifying discrete emotion expressed by humans in text
- Evolution of different social media sites and blogs
- Huge volume of opinionated text with emotional content
- Sentiment analysis deals with polarity of texts (positive, negative or neutral) and the intensity of it, emotion mining deals with identifying human emotion expressed via text

Emotion Recognition

- Four approaches:
  - Keyword based
    - "I passed the test"
    - "Hooray! I passed the test"
  - Learning based
    - can adapt to domain changes
    - depends on keywords as features to certain extent
  - Hybrid based
  - Rule based

Model



w1
w2
wn
Tweet Matrix (L1 x d)
K=1
K=2
K=3
Convolutional layer with multiple window sizes (k=1,2,3)
h1
hk
e1
ep
Hash-Emo Matrix (L2 x d)
K=1
Convolutional layer with single window size (k=1)
Dropout + Max Pooling
Multichannel Embedding Layer
...
Affect + Sentiment Features
Concatenation
Fully-connected layer with softmax

Dataset

Emotion	Dataset			
	BTD	TEC	CBET	SE
joy	409,983	8,240	10,691	3,011
sadness	351,963	3,830	8,623	2,905
anger	311,851	1,555	9,023	3,091
love	175,077	–	9,398	–
thankfulness	80,291	–	8,544	–
fear	76,580	2,816	9,021	3,627
surprise	14,141	3,849	8,552	–
guilt	–	–	8,540	–
disgust	–	761	8,545	–
Total	1,419,886	21,051	80,937	12,634

Table: Basic statistics of the emotion datasets.

Experimental Setup

- Data Cleaning
  - Regular expression based tokenizer - Tweet Tokenizer
    - Preserves hashtags, emoticons, emojis
    - Reduces the length of repeated characters to three ("Haaaaaapy" will become "Haaapy")
  - Remove urls
  - Replace slang words (e.g., "nvm" will become "never mind")
  - Lowercase all the letters
  - Normalize certain negative words (e.g., "won't" will become "will not")
  - Remove stop-words (In this work, we used customized stop word list)
  - Normalize the repetitions of two punctuation marks (! and ?) (e.g., "!!!" will become "! <repeat>")
  - Strip off "#" symbols from all the hashtags within the tweets (e.g., "#depressed" will become "depressed")
  - Keep tokens with more than one character

Experimental Setup

- Input Features
  - Pretrained word embeddings (GloVe)
  - Affect features
    - Valence, arousal and dominance scores (Warriner et al., 2013)
    - NRC Emotion Lexicon (Mohammad and Turney, 2013)
    - NRC Affect Intensity Lexicon (Mohammad and Bravo-Marquez, 2017)
  - Sentiment features
    - MPQA (Wilson et al., 2005)
    - BingLiu (Hu and Liu, 2004)
    - AFINN (Nielsen, 2011)

Result

Emotion	Dataset											
	BTD			TEC			CBET			SemEval		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
joy	68.4	77.4	72.6	67.4	77.1	71.8	58.1	56.1	57.1	78.5	70.1	74.1
sadness	72.7	74.5	73.6	48.8	53.7	50.9	38.0	43.3	40.5	62.6	41.0	49.6
anger	74.7	79.1	76.8	34.5	23.8	27.7	49.3	52.1	50.7	59.7	63.6	61.6
love	57.0	46.4	51.1	—	—	—	65.4	53.3	58.7	—	—	—
thankfulness	63.2	55.3	59.0	—	—	—	66.1	68.0	67.0	—	—	—
fear	57.6	38.3	46.0	61.5	57.2	58.6	70.3	69.6	70.0	51.6	71.9	60.1
surprise	88.1	16.1	27.1	55.9	50.2	52.5	51.0	55.3	53.0	—	—	—
guilt	—	—	—	—	—	—	53.8	49.6	51.6	—	—	—
disgust	—	—	—	67.4	77.1	71.8	59.3	61.0	60.2	—	—	—
Avg.	**68.9**	**55.3**	**58.0**	**55.9**	**56.5**	**55.6**	**56.8**	**56.5**	**56.5**	**63.1**	**61.7**	**61.3**

Table: Results (in %) of our model (MC-CNN) for four emotion-labeled datasets.

Result

Models	Dataset			
	BTD	*TEC*	*CBET*	*SE*
CNN	66.1	54.3	53.8	56.3
MC-CNN†	68.5	57.6	56.1	59.8
MC-CNN†‡	69.2	58.9	56.4	62.0

Models	Dataset		
	STS-Gold	*STS-Test*	*SS-Twitter*
CNN	86.2	75.1	59.1
MC-CNN†	88.5	70.6	63.2
MC-CNN†‡	90.7	81.5	64.6

Table: Comparison of results (accuracy in %) of three variants of our model. † represents the inclusion of Hash-Emo embedding into the network. ‡ represents the inclusion of external features into the network.