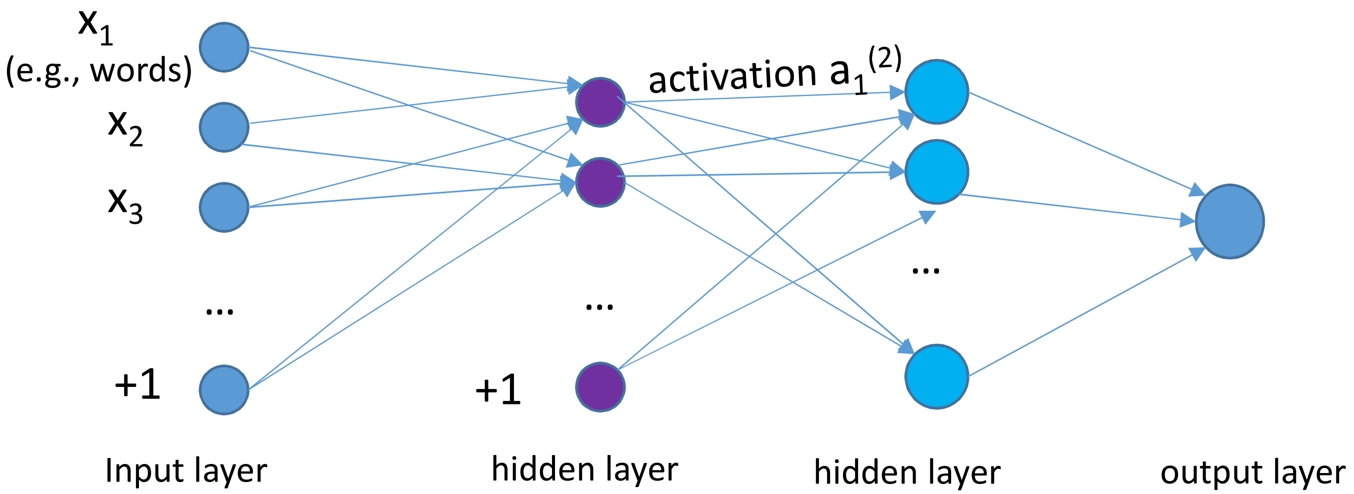
Introduction to Deep NLP

Word Embedding
Convolutional Neural Network (CNN)
CNN in NLP (Example)

Deep Learning – What is it?

- Representation learning for automatically learning good features or representations
  - Representational learning: "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

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

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						
…						

$\mathbf{E}$

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

$\mathbf{e}_{5391}$

$$\mathbf{e}_{5391} = \boldsymbol{E} \cdot \boldsymbol{O}_{5391}$$

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

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

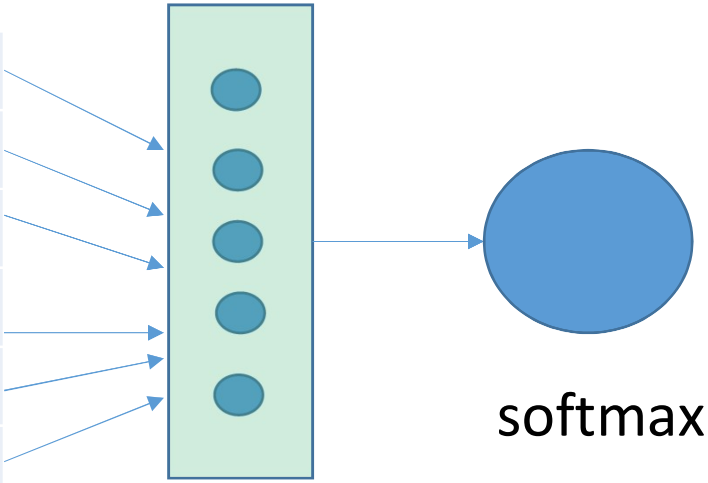
Word Representation in Deep NLP

- 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

Word Embedding – How Do You Get E?

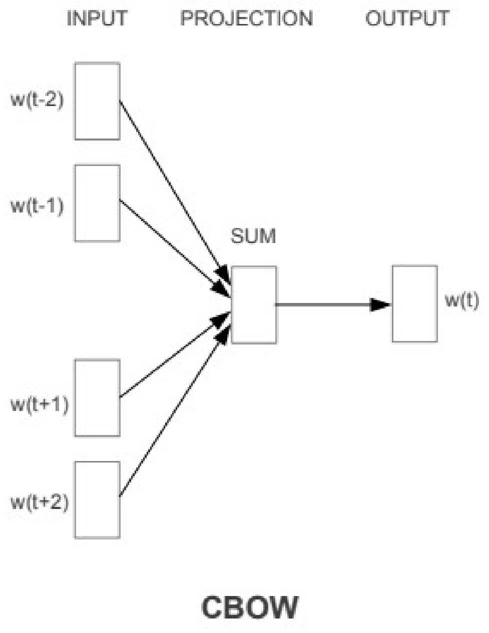
Word2Vec
GloVe
fastText

Word Embedding – Word2Vec

- Continuous Bag of Words (CBOW): predict the target word given the context (i.e., surrounding words)

  Example: “the quick brown fox jumps over the lazy dog”

  If window size is 2, we have examples like ([quick, fox], brown), ([the, brown], quick), etc.



Goal: Learn the hidden layer weight matrix -> E

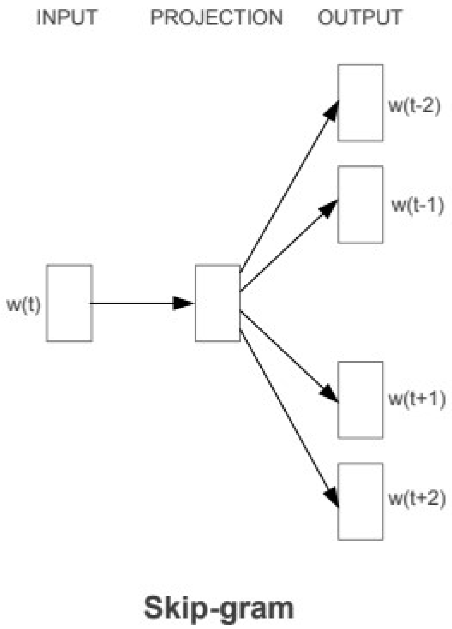
The CBOW model architecture
(Source: https://arxiv.org/pdf/1301.3781.pdf Mikolov el al.)

Word Embedding – Word2Vec

- Skip-gram: predict the context word given the target (i.e., surrounding words)

  Example: "the quick brown fox jumps over the lazy dog"

  If window size is 2, we have examples like (brown, [quick, fox]), (quick, [the, brown]), etc.

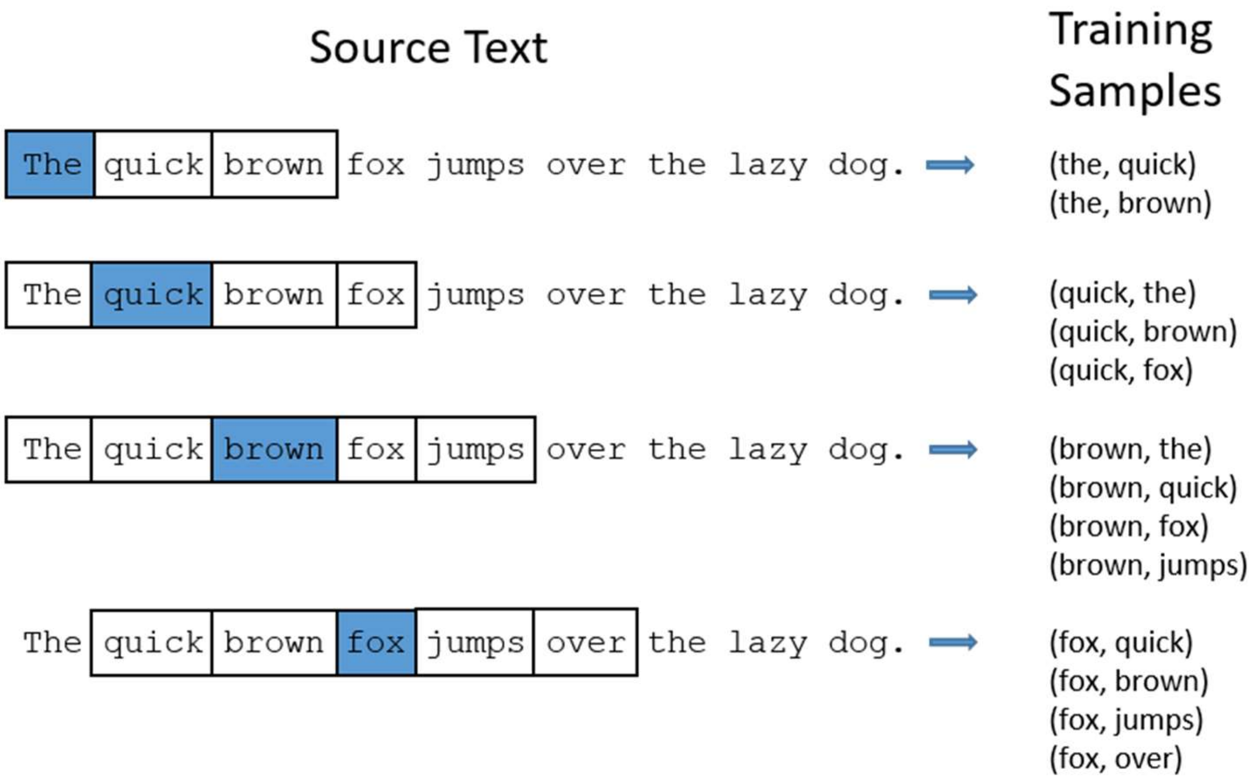


Goal: Learn the hidden layer weight matrix -> E

The Skip-gram model architecture
(Source: https://arxiv.org/pdf/1301.3781.pdf Mikolov el al.)

Word Embedding – Word2Vec

- Skip-gram model

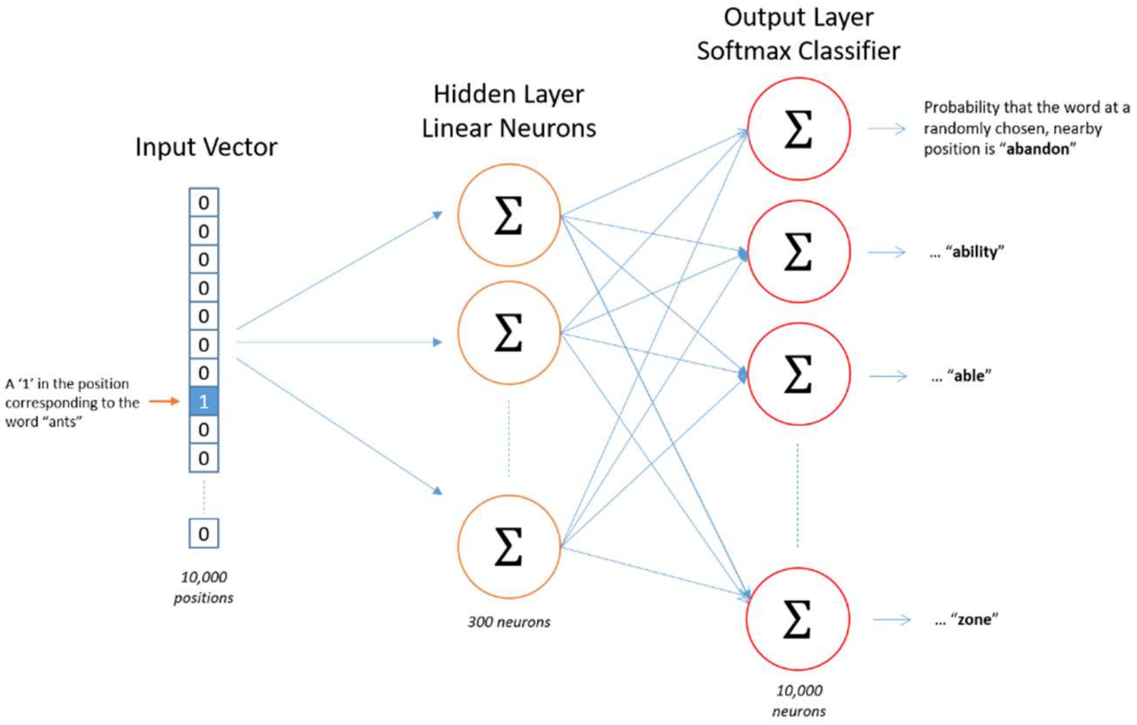


Source Text	Training Samples
The quick brown fox jumps over the lazy dog. ⟹	(the, quick) (the, brown)
The quick brown fox jumps over the lazy dog. ⟹	(quick, the) (quick, brown) (quick, fox)
The quick brown fox jumps over the lazy dog. ⟹	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick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

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

Word Embedding – Word2Vec

- Skip-gram model



Problem: the sum of vocabulary is necessary when calculating each probability

One solution: hierarchical Softmax (Huffman tree)

Goal: Learn the hidden layer weight matrix -> E

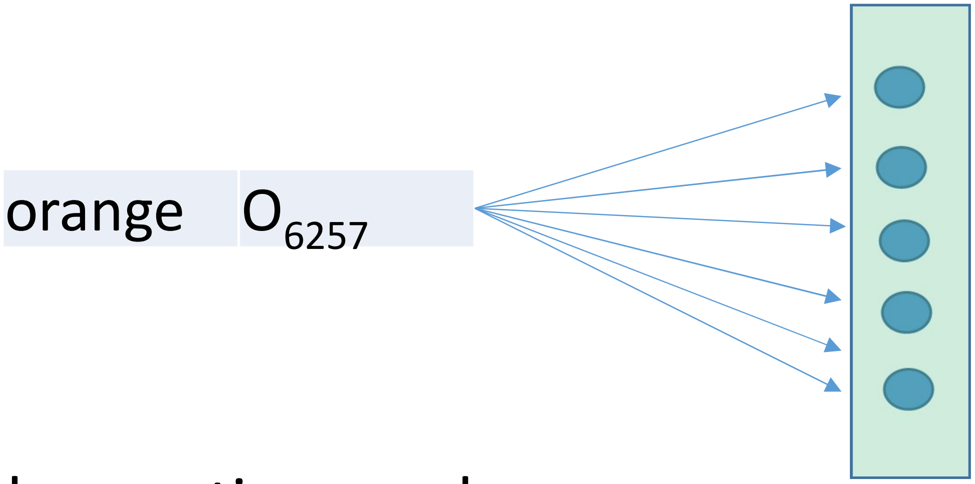
Image: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Word Embedding – Word2Vec

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

Word Embedding – Word2Vec

Skip-gram negative sampling (SGNG)



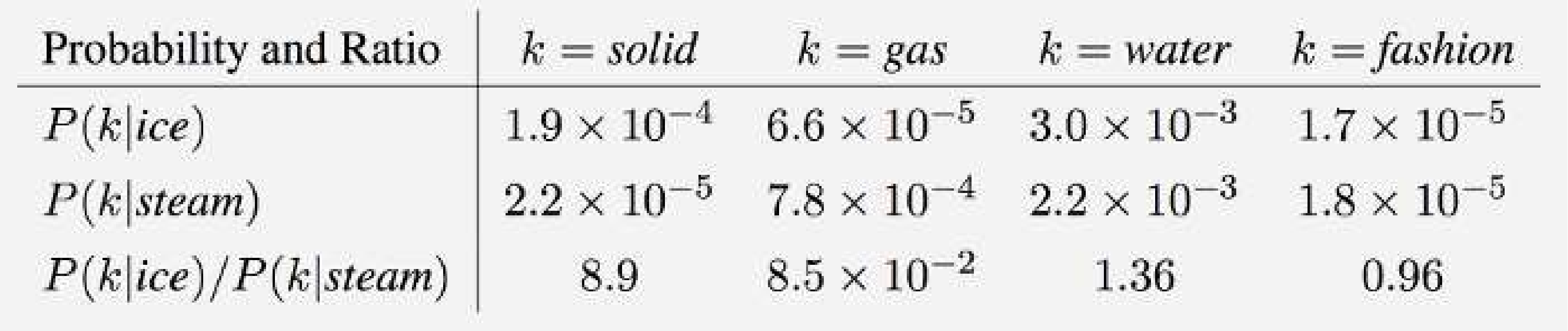
10, 000 binary logistic regression

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)

Word Embedding – GloVe

- Ratios of word-word co-occurrence probabilities can be related to the semantic relationships between the words.



Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Source: https://nlp.stanford.edu/projects/glove/

The relatedness of *ice* and *water* vs. that of *steam* and *water* – similar (ratio = 1.36)
The relatedness of *ice* and *fashion* vs. that of *steam* and *fashion* – similar (ratio = 0.96)

Ice and *solid* are much more related than *steam* and solid – ratio = 8.9
Steam and *gas* are much more related than *ice* and *gas* – ratio << 1

Word Embedding – GloVe

Training data: non-zero entries of a global word-word co-occurrence matrix

Training objective of GloVe: to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence

Example: 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

Word Embedding – fastText

Character-level n-gram representation

Matter, with n = 3, the fastText representation:

<ma, att, the, ter, er>

Context

matter

Target

< and > are boundary symbols distinguishing the ngram of a word (e.g., *the* in this example) from a word itself (e.g., the word *the*)

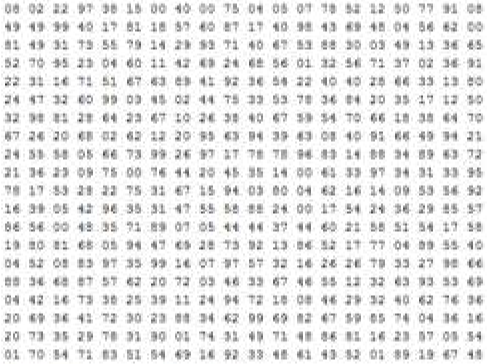
Deep NLP – Some common deep neural network models

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN)

- 2012 ImageNet competition (Alex Krizhevsky)

What We See



```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What Computers See

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Input: the matrix (e.g., 32 X 32 X 3, each element is a number in the range of [0, 255])
Output: a vector (each number is a probability of the image being a certain class, e.g., cat, dog, etc.)

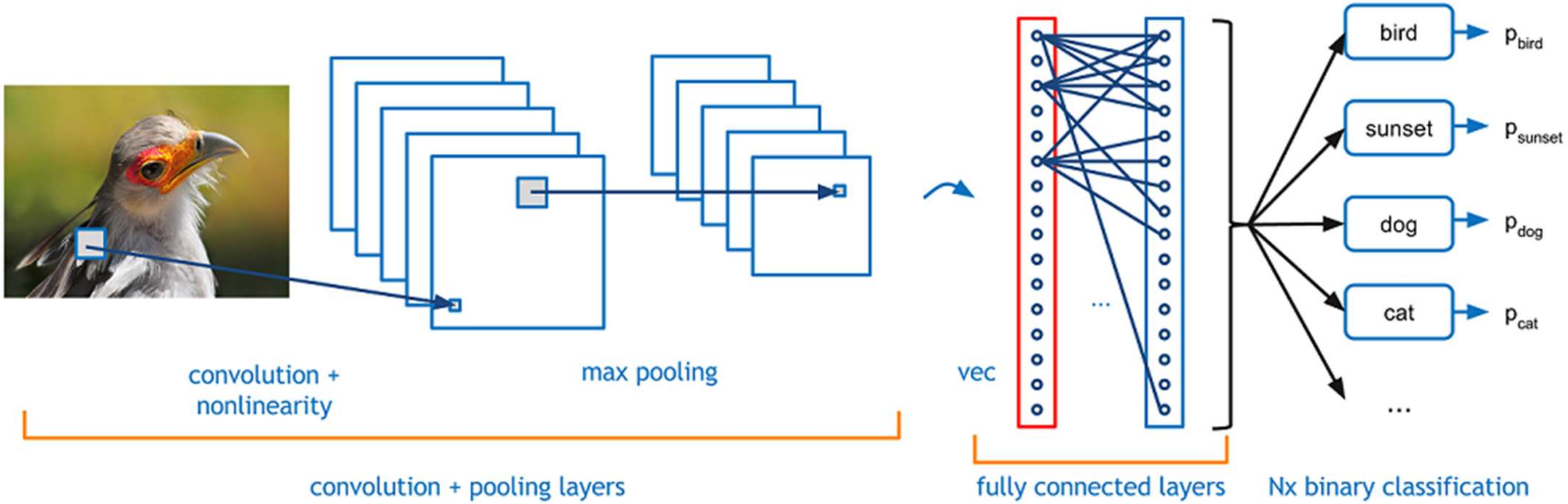
Convolutional Neural Network (CNN)

- How do we process and recognize images? (Hubel and Wiesel, 1962)

For visual perception, our neuronal cells are in charge of different orientation. For example, some will respond to vertical edges, some horizontal, some diagonal, etc. These neuronal cells are organized in columnar architecture and function together to full the visual perception tasks.

Development of orientation specificity in visual cortex
https://www.youtube.com/watch?v=p6Tjm9VCkcI

Convolutional Neural Network (CNN)

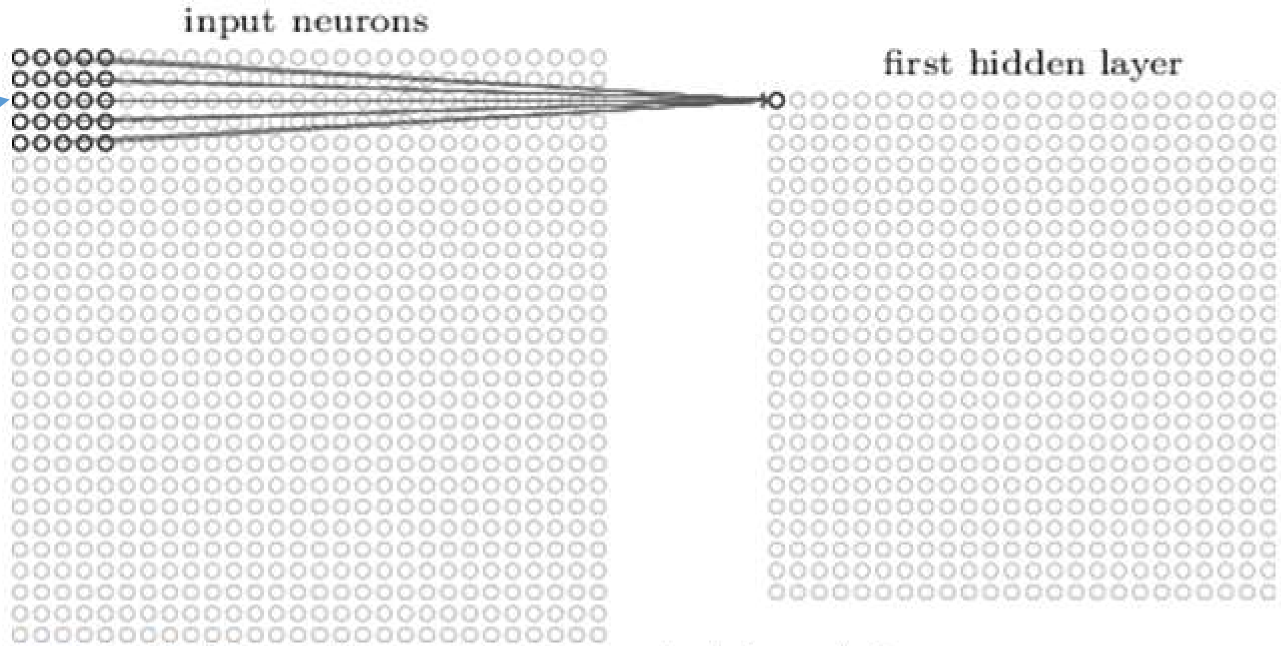


Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN)

Filter (kernel, neuron):e.g.,5X5X1

Convolutional Layer



receptive field

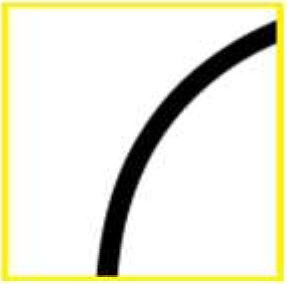
Visualization of 5 x 5 filter convolving around an input volume and producing an activation map

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

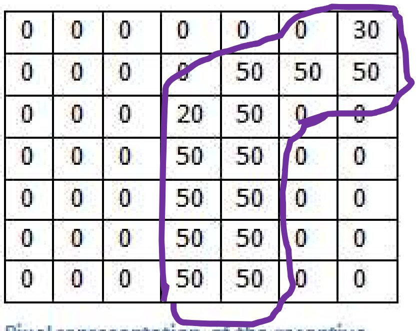
Convolutional Neural Network (CNN)

Convolutional Layer

(curve) Filter: 7 X 7 X 1



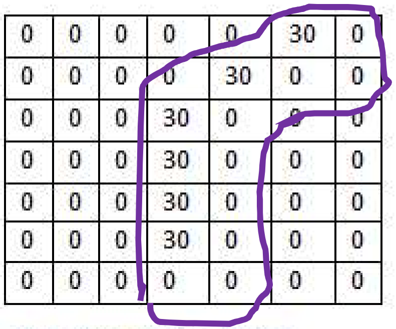
Visualization of the receptive field



0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*



0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

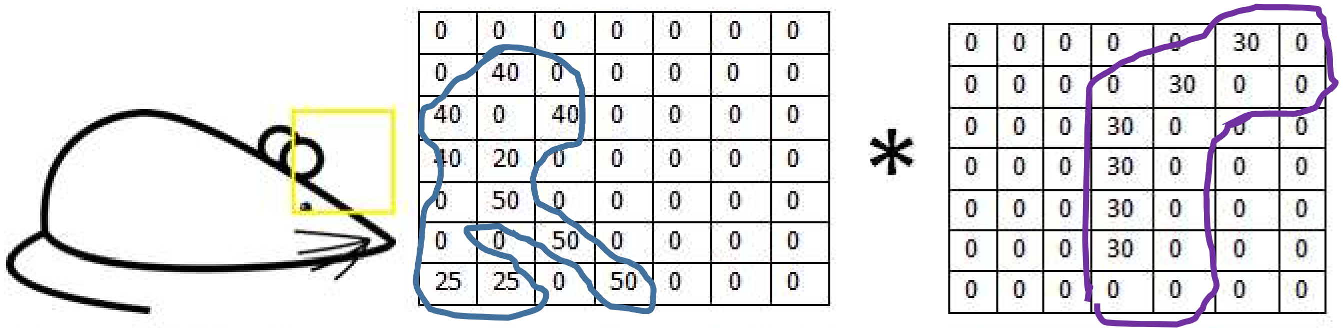
Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN)

Convolutional Layer

Filter: 7 X 7 X 1



Visualization of the filter on the image

Pixel representation of receptive field

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

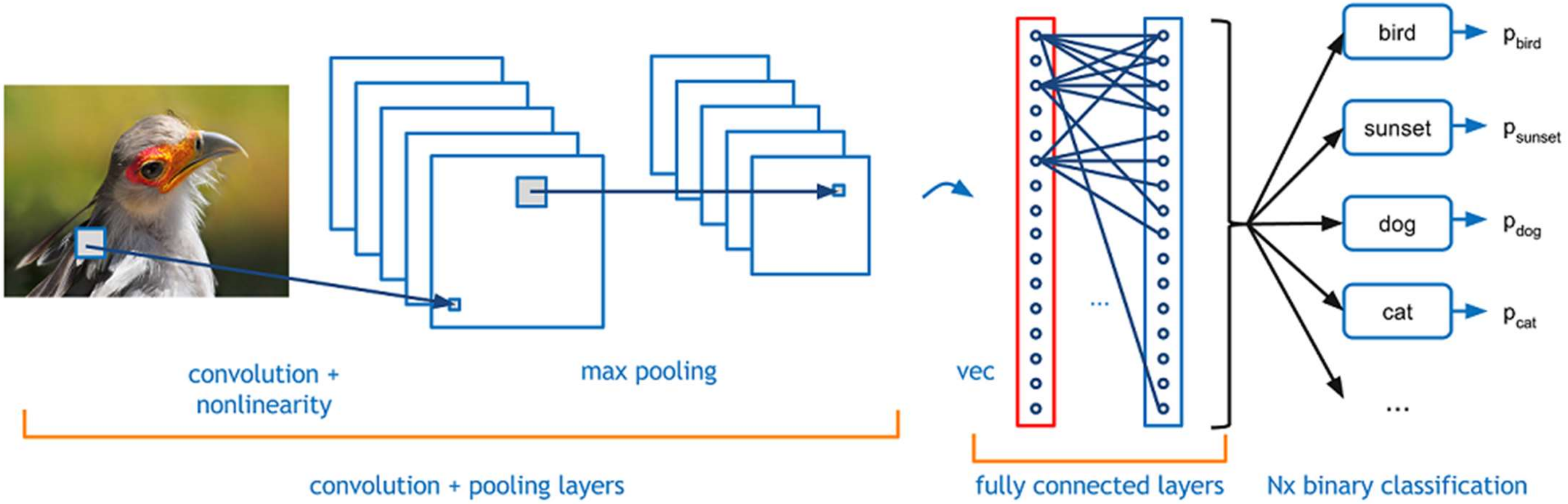
Pixel representation of filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Multiplication and Summation = 0

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN)



Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN) - Training

- The forward pass

Initially, filter value is randomly assigned -> performance is expected to be (very) bad

- The loss function

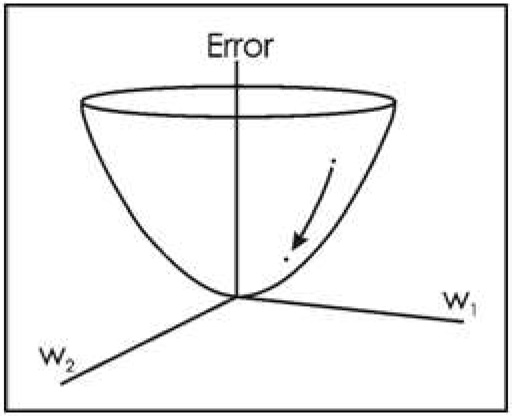
$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

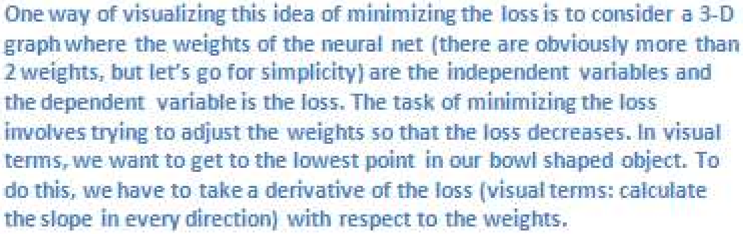
Cost/Error function (mean squared error)

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN) - Training

- The backward pass




One way of visualizing this idea of minimizing the loss is to consider a 3-D graph where the weights of the neural net (there are obviously more than 2 weights, but let's go for simplicity) are the independent variables and the dependent variable is the loss. The task of minimizing the loss involves trying to adjust the weights so that the loss decreases. In visual terms, we want to get to the lowest point in our bowl shaped object. To do this, we have to take a derivative of the loss (visual terms: calculate the slope in every direction) with respect to the weights.

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN) - Training

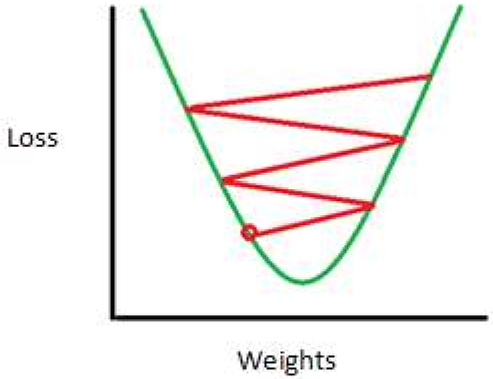
- The weight update

$$w = w_i - \eta \frac{dL}{dW}$$

w = Weight

w_i = Initial Weight

η = Learning Rate



Consequence of a high learning rate where the jumps are too large and we are not able to minimize the loss.

Backpropagation includes: the forward pass, loss function, backward pass, and parameter update

Source: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolutional Neural Network (CNN) - Training

➢The weight update

Epoch: one Epoch is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE.

Batch: one batch contains the training examples present in one weight update (recommended: no more than 32, https://arxiv.org/pdf/1804.07612.pdf)

Iteration: one iteration = total training data/batch size

Convolutional Neural Network in Natural Language Processing

Example study by Islam, Mercer, and Xiao (2019)

Islam, J., Mercer, R., Xiao, L. (2019). Multi-Channel Convolutional Neural Network for Twitter Emotion and Sentiment Recognition, *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (Jun 3 – 5, Minneapolis, USA), Volume 1, 1355-1365.*

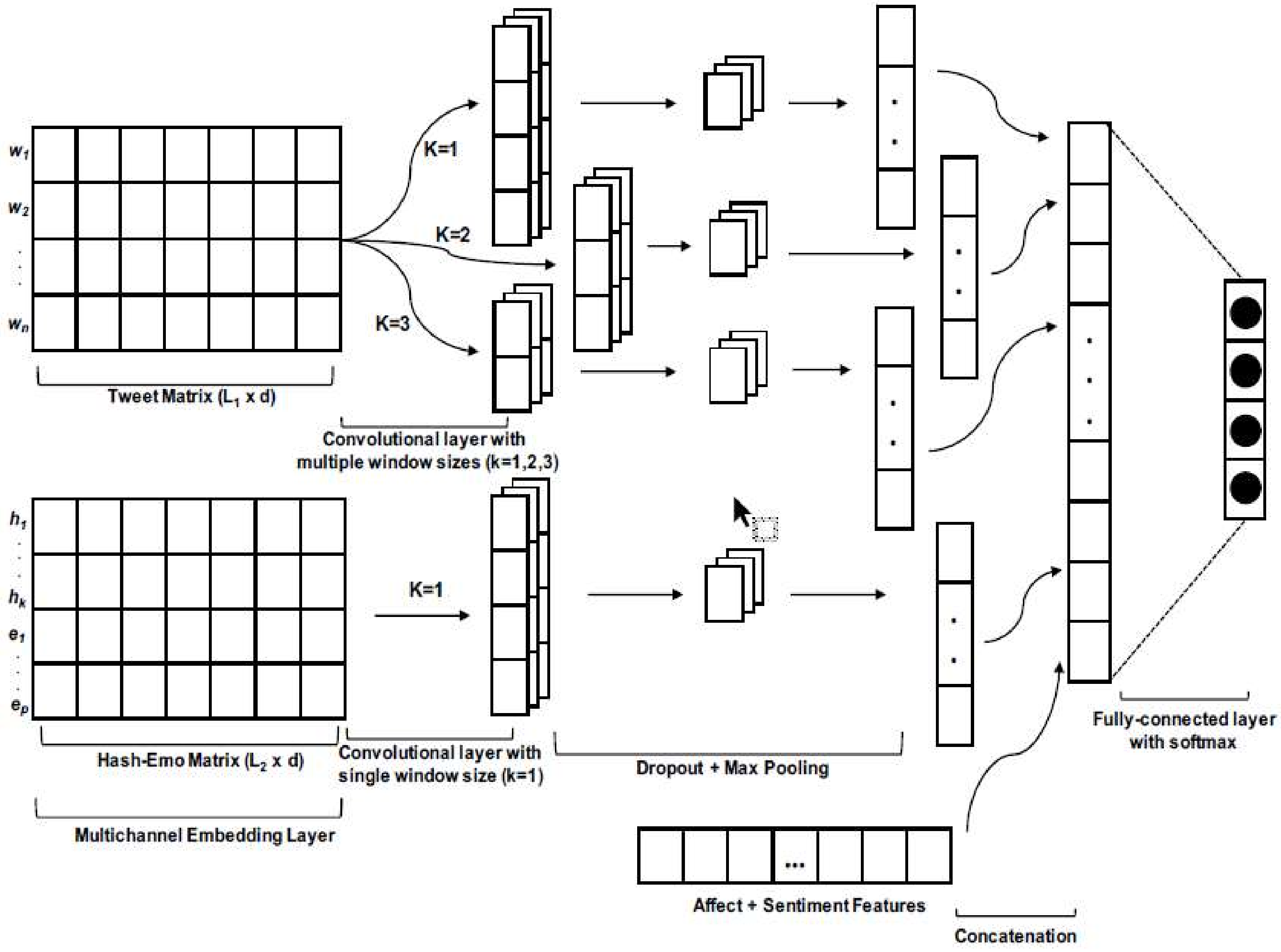
Tweet: “The most memorable shot of Elijah Hughes’s time at **Syracuse**. Pretty darn good one.”

Which emotion is revealed from the tweet above?

Anger Joy Fear Disgust Sadness Thankfulness
….

Islam, J., Mercer, R., Xiao, L. (2019). Multi-Channel Convolutional Neural Network for Twitter Emotion and Sentiment Recognition, *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (Jun 3 – 5, Minneapolis, USA), Volume 1, 1355-1365.*

Multi-Channel CNN Model (Islam, Mercer, & Xiao, 2019)



w_1
w_2
w_n
Tweet Matrix (L_1 x d)
K=1
K=2
K=3
Convolutional layer with multiple window sizes (k=1,2,3)
h_1
h_k
e_1
e_p
Hash-Emo Matrix (L_2 x d)
K=1
Convolutional layer with single window size (k=1)
Multichannel Embedding Layer
Dropout + Max Pooling
...
Affect + Sentiment Features
Concatenation
Fully-connected layer with softmax

Figure 1: Overview of the MC-CNN model

Embedding Layer

Tweet Matrix

- Each tweet ti consists of a sequence of tokens $w_{1,}$ w_2,…w_{ni} . L_1 is the maximum tweet length. Short tweets are padded using zero padding.
- Every word is represented as a d-dimensional word vector
- The publicly available pre-trained GloVe word vectors for Twitter by (Pennington et al., 2014).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543. Association for Computational Linguistics.

Embedding Layer

Hash-Emo Matrix

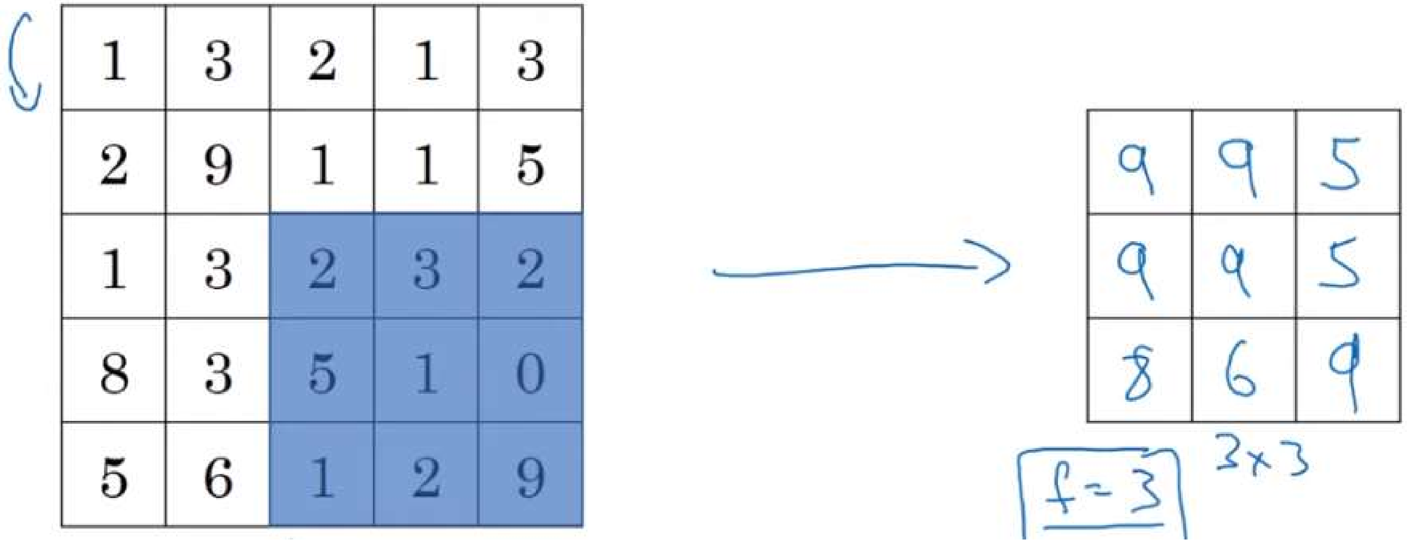
- Hashtags, emoticons and emojis
- for each tweet t_i, we extract hashtags h_1, h_2, … and emoticons/emojis e_1, e_2, e... and concatenate the hashtags and emoticon/emoji vectors
- L_2: the height of the Hash-Emo Matrix. Tweets with the number of hash-emo features less than L_2 are padded with zero while tweets with more hash-emo features than L_2 are truncated.
- d-dimension Word vectors from GloVe
- Random initiation
  - no word vector is found for a particular word
  - emoticons.
  - for emojis, we first map it to something descriptive; and then generate random word vectors

Convolutional Layer

- Apply *m* filters of varying window sizes over the Tweet Matrix from the embedding layer
- window size (k) refers to the number of adjacent word vectors in the Tweet Matrix that are filtered together (when k > 1)

Dropout and Max Pooling Layer

- ReLU is applied before dropout layer
- Dropout is used as a regularization strategy to avoid overfitting
- Max-pooling: the maximum value for each filter



1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

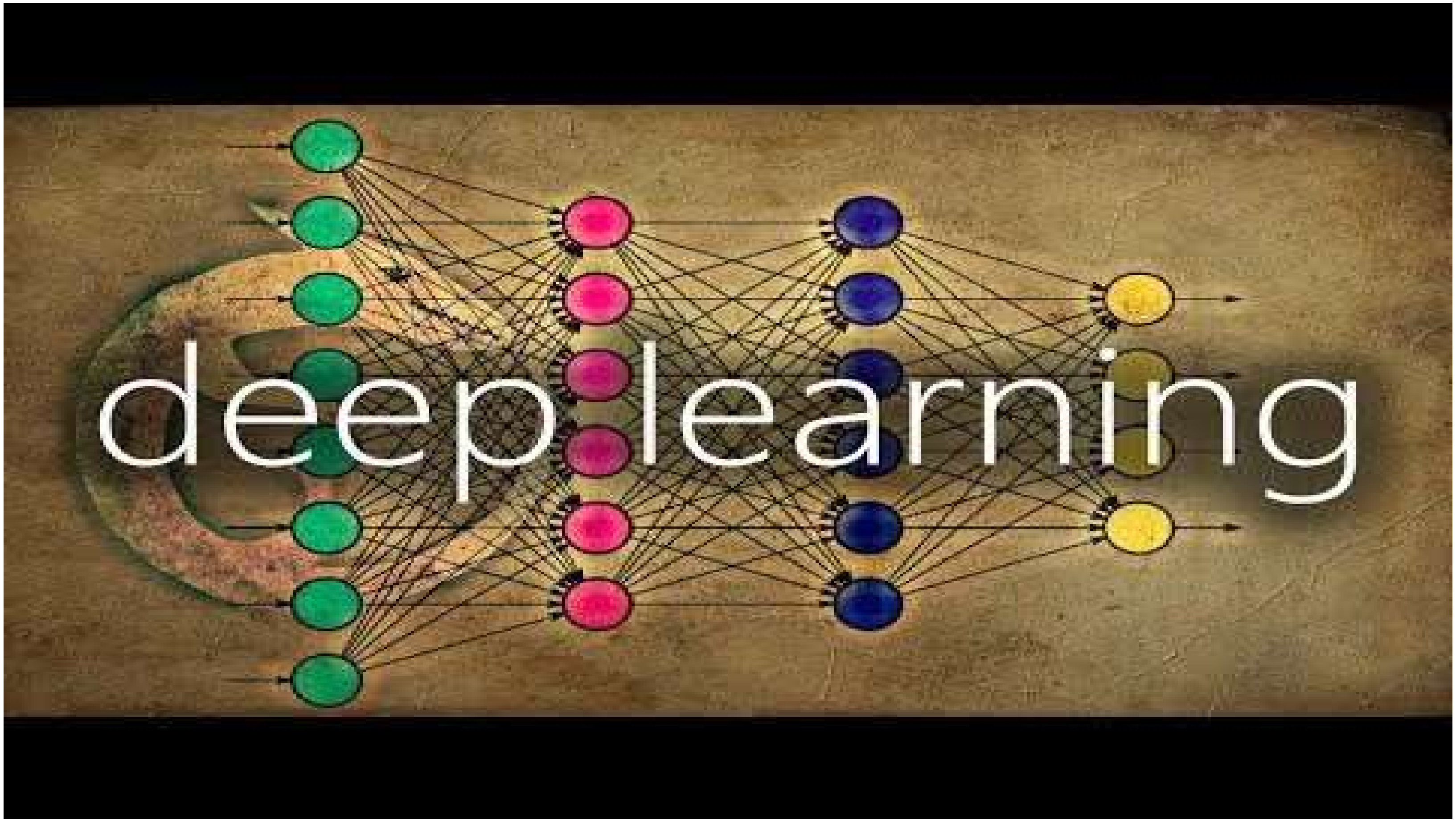
9	9	5
9	9	5
8	6	9

Source:
https://www.youtube.com/watch?v=8oOgPUO-TBY&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=9

Fully Connected Layer

- Maps the inputs to a number of outputs corresponding to the number of classes we have.
- Emotion recognition: a multi-class classification task
  - Softmax as the activation function and categorical cross-entropy as the loss function
  - The output of the softmax function is equivalent to a categorical probability distribution which generally indicates the probability that any of the classes are true

Deep Learning - CNN



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