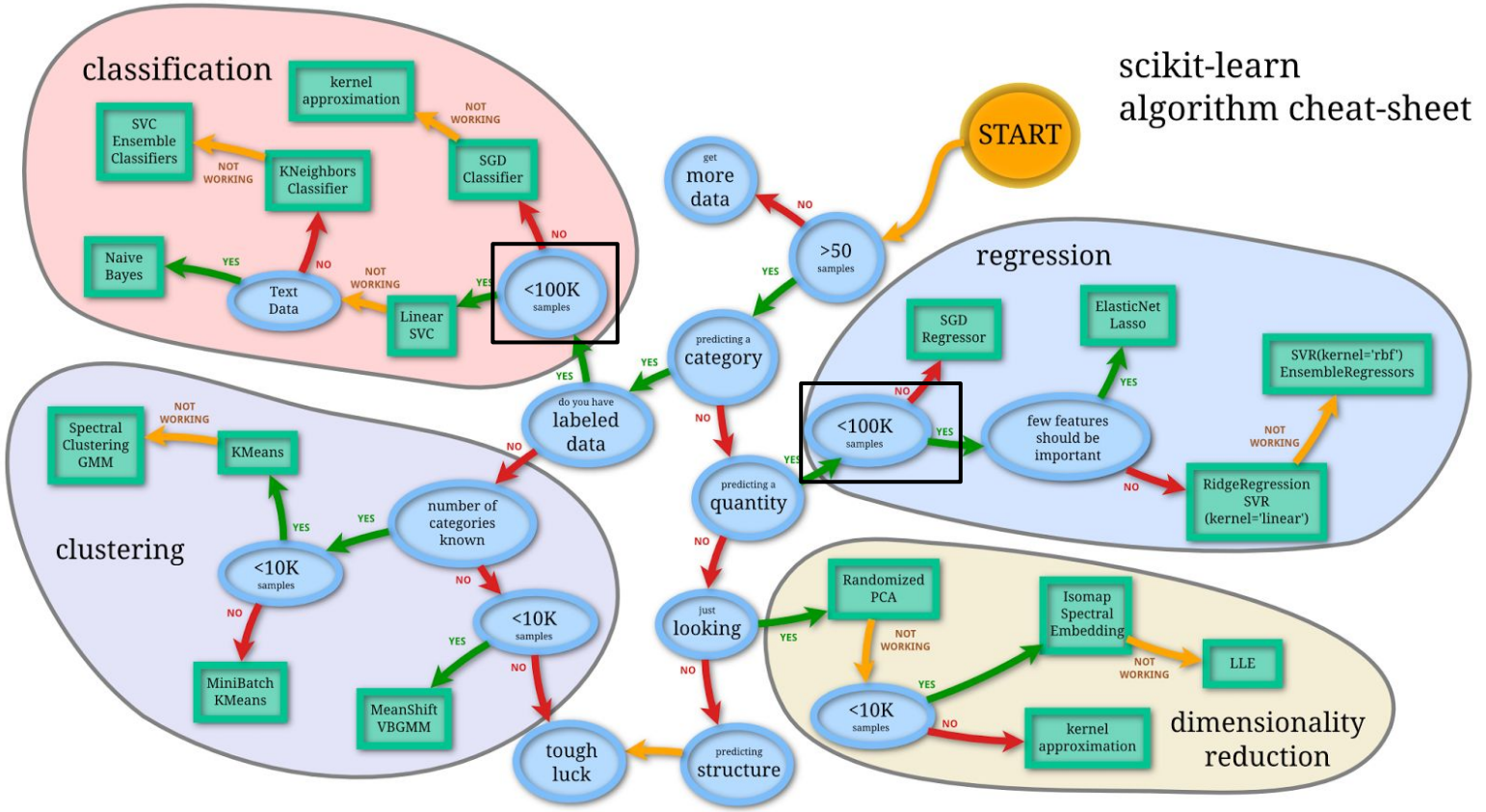
CNN

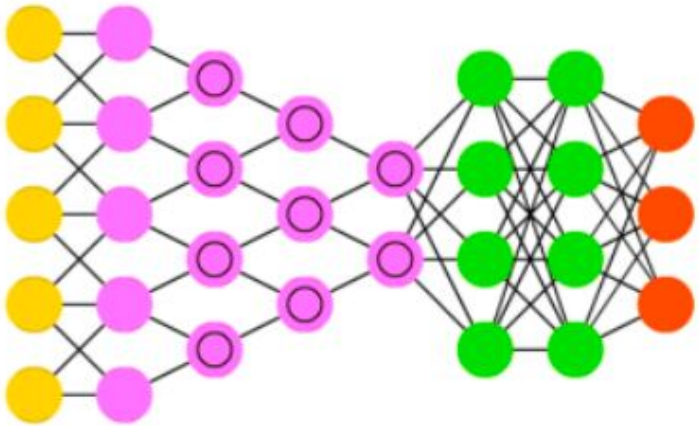
Juan Luis García Mendoza, Mario Ezra Aragón, Adrián Pastor López Monroy, Luis Villaseñor Pineda, Manuel Montes y Gómez

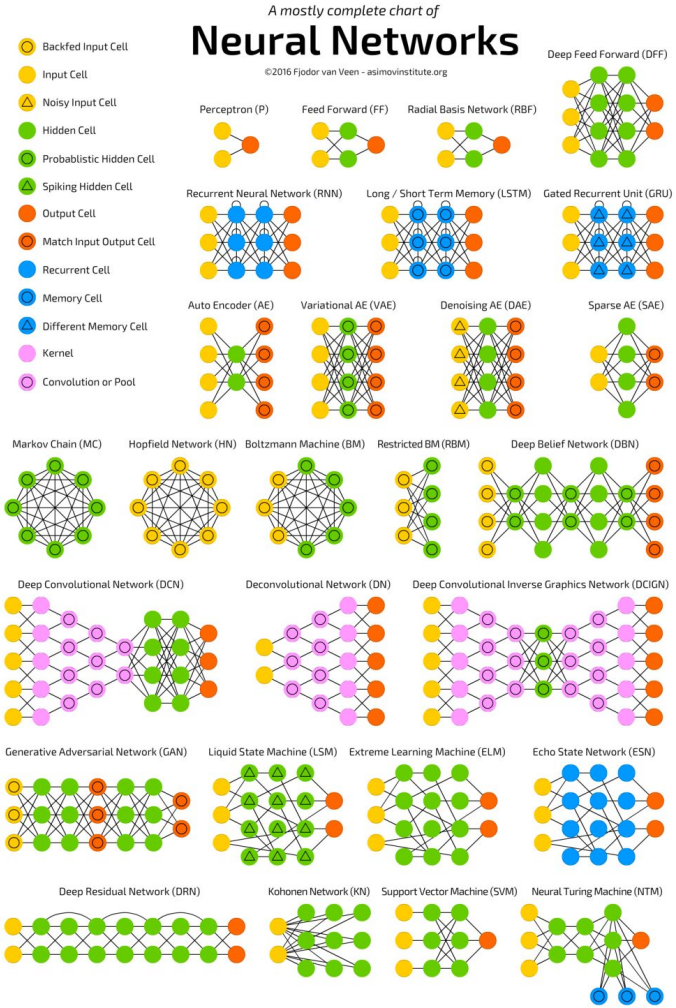
Overview

- Explaining How Convolutions Work
- Pooling
- CNN in NLP


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
ElasticNet Lasso
SVR(kernel='rbf') EnsembleRegressors
predicting a category
YES
NO
<100K samples
NO
YES
few features should be important
YES
NOT WORKING
RidgeRegression SVR (kernel='linear')
NO
YES
do you have labeled data
NO
YES
predicting a quantity
NO
YES
clustering
Spectral Clustering GMM
NOT WORKING
KMeans
YES
number of categories known
YES
NO
<10K samples
YES
NO
<10K samples
NO
MiniBatch KMeans
MeanShift VBGMM
just looking
NO
YES
Randomized PCA
NOT WORKING
Isomap Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
tough luck
predicting structure

Deep Convolutional Network (DCN)



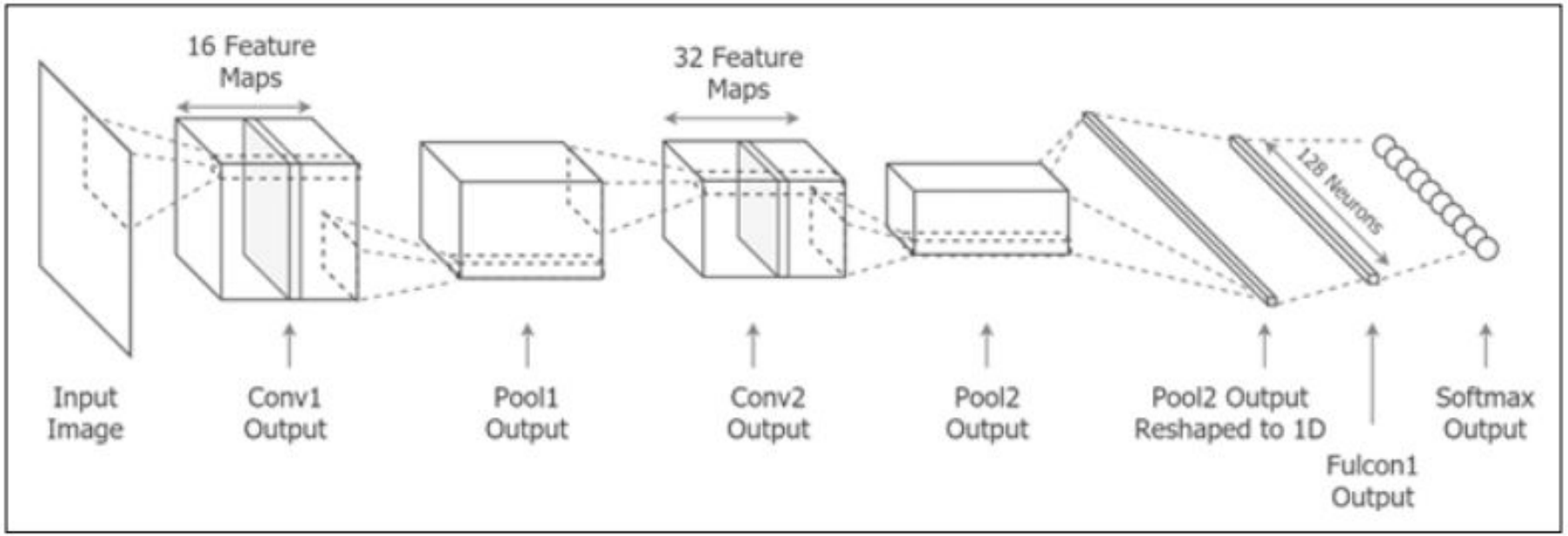


Stefan Kojouharov, Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Big Data

Terminology

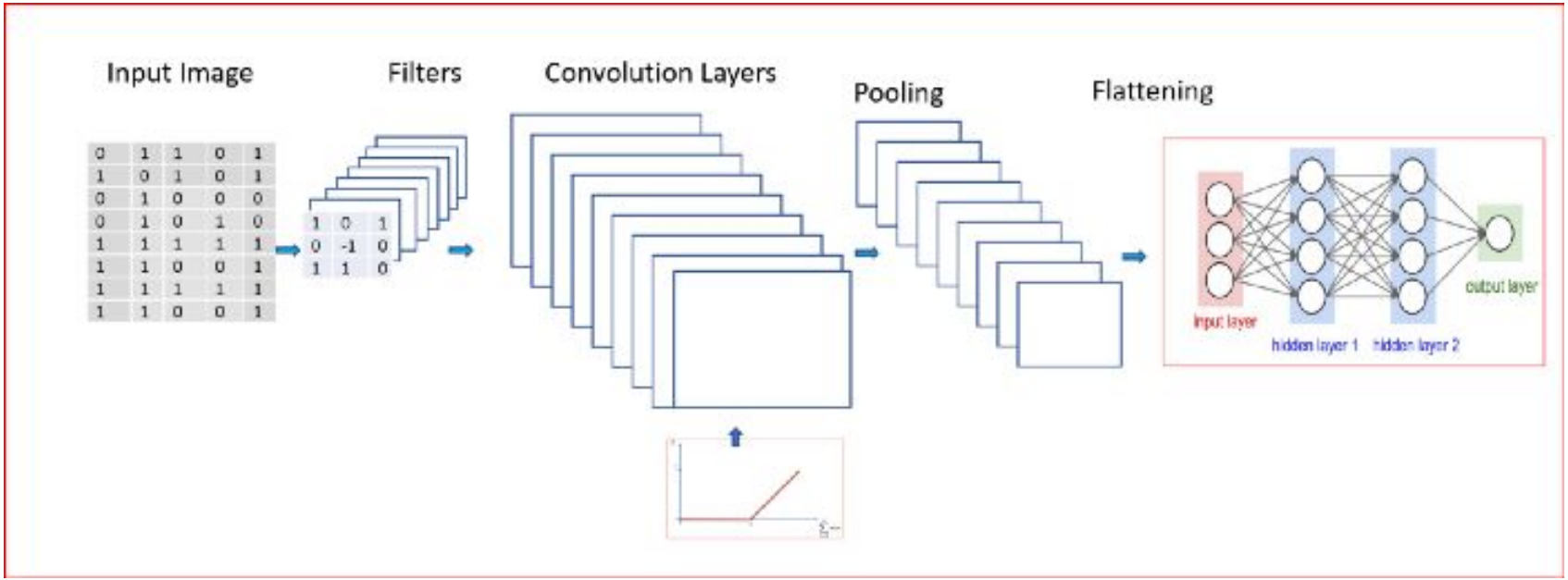
- **filter size:** refers to the window size of the convolution operation
- **stride:** refers to the distance between two movements of the convolution window
- **padding:** refers to the way you handle boundaries of the input
- **pooling operation:** was introduced to CNNs mainly for reducing the size of the intermediate outputs.

Architecture



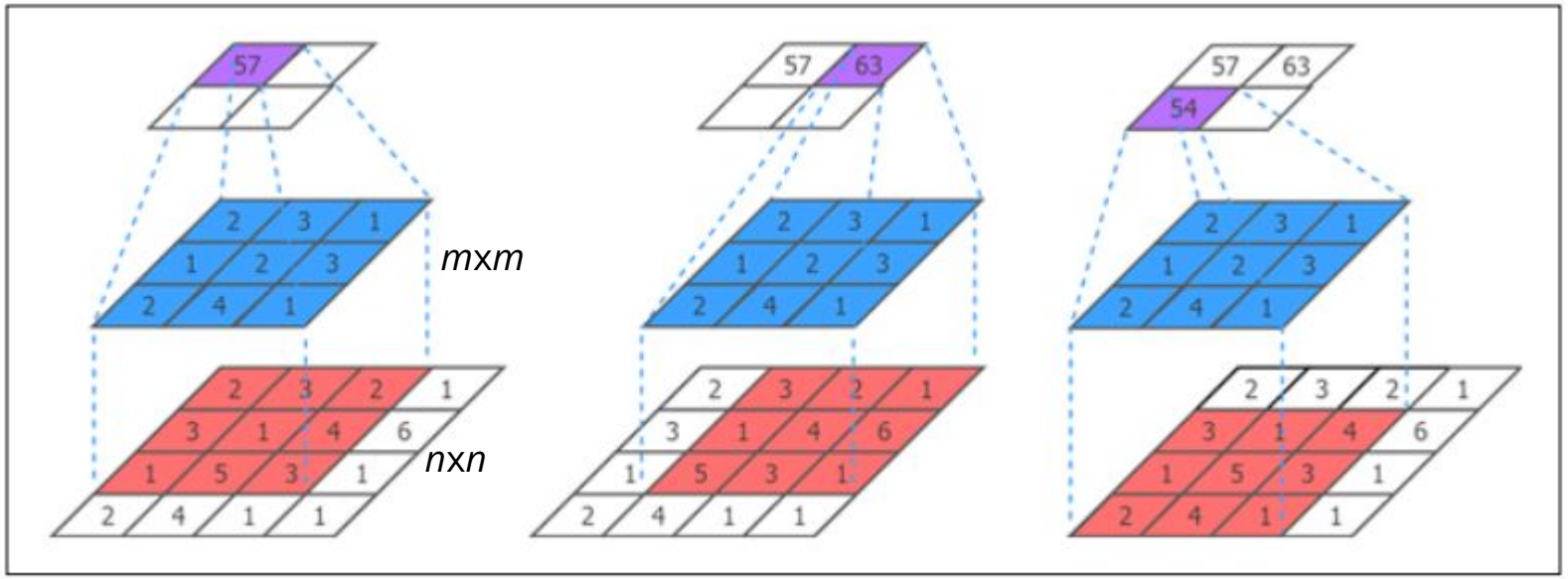
Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Architecture



Renu Khandelwal, Convolutional Neural Network: Feature Map and Filter Visualization

Standard Convolution

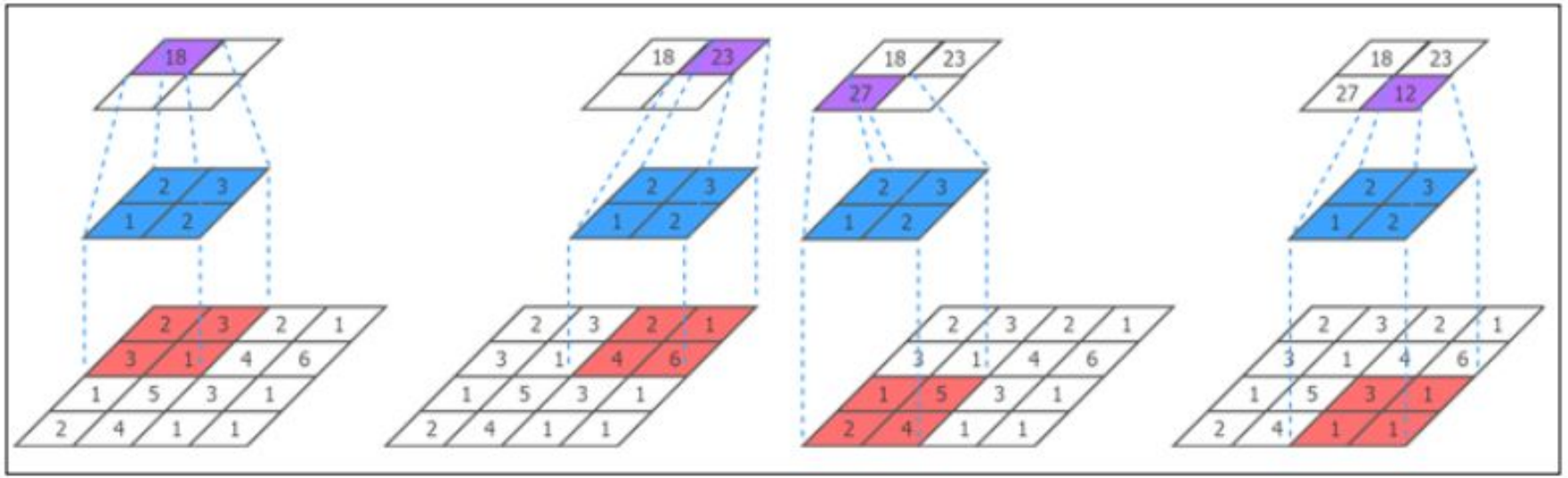


$$h_{i,j} = \sum_{k=1}^{m}\sum_{l=1}^{m} w_{k,l} x_{i+k-1,j+l-1} \; where 1 \le i, j \le n-m+1$$

filter size (m) = 3, stride = 1, padding = None

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Convolution with stride

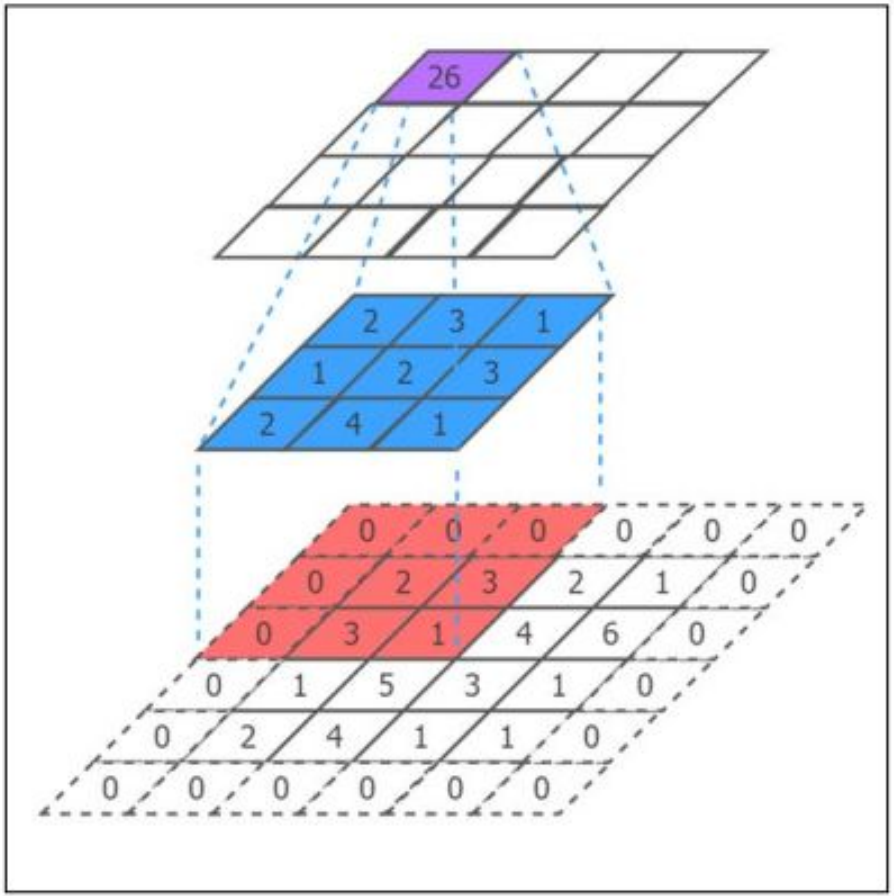


$$h_{i,j} = \sum_{k=1}^{m}\sum_{l=1}^{m} w_{k,l}\, x_{(i-1)\times s_i + k,(j-1)\times s_j + l} \; where \; 1 \le i \le floor\left[(n-m)/s_i\right] + 1 \; and \; floor\left[(n-m)/s_j\right] + 1$$

filter size (m) = 2, stride = 2, padding = None

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Convolution with padding

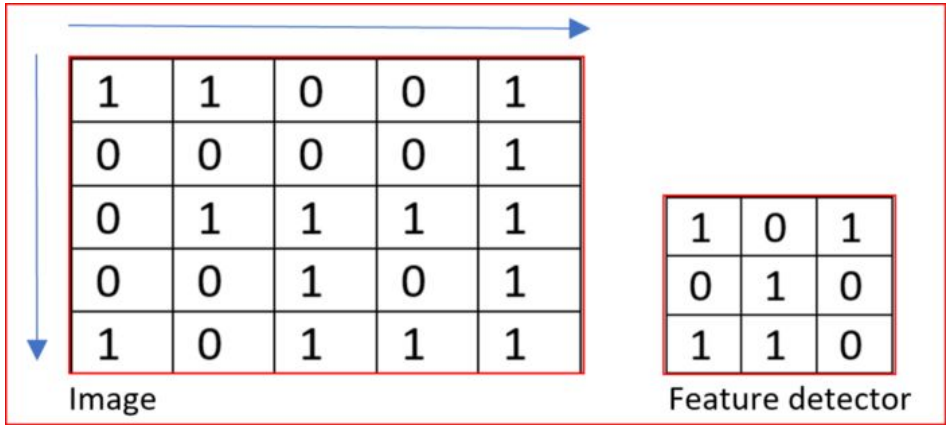


$$h_{i,j} = \sum_{k=1}^{m}\sum_{l=1}^{m} w_{k,l} x_{i+k-(m-1),j+l-(m-1)} \; where 1 \le i,j \le n$$

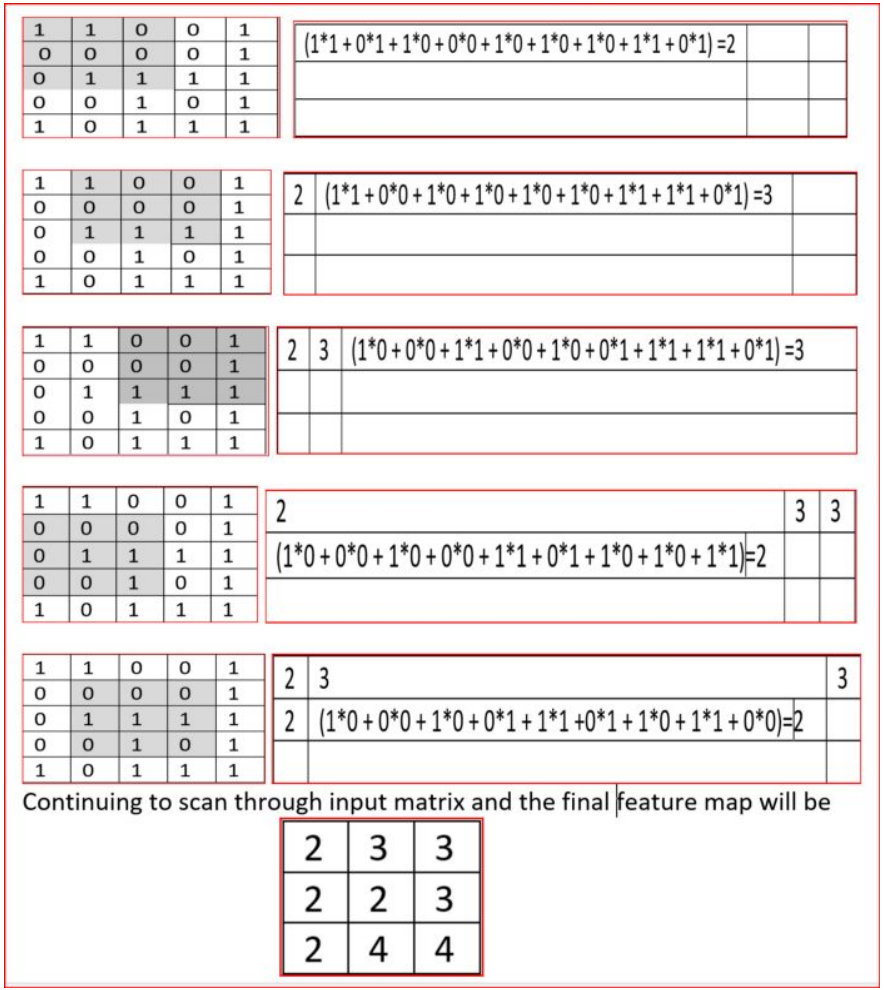
filter size (m) = 3, stride = 1, padding = zero

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Convolution (example)

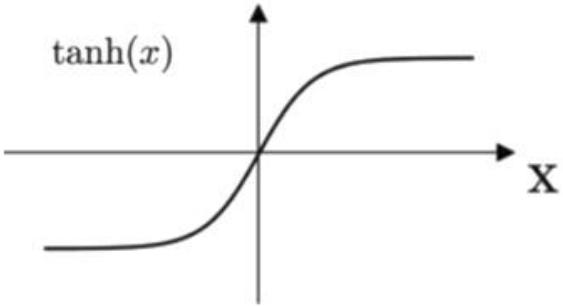


filter size (m) =**?**, stride = **?**, padding = **?**

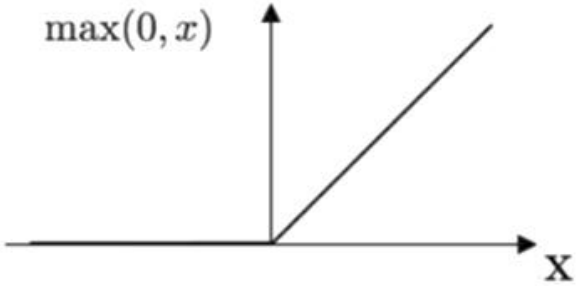


Activation functions

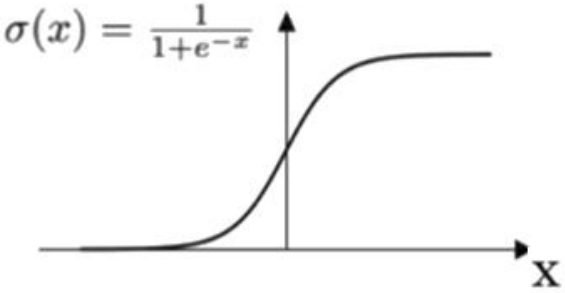
Tanh



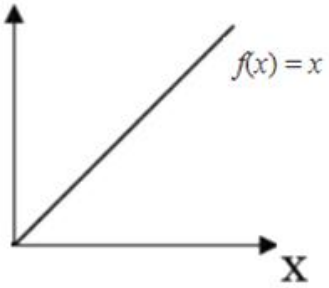
ReLU



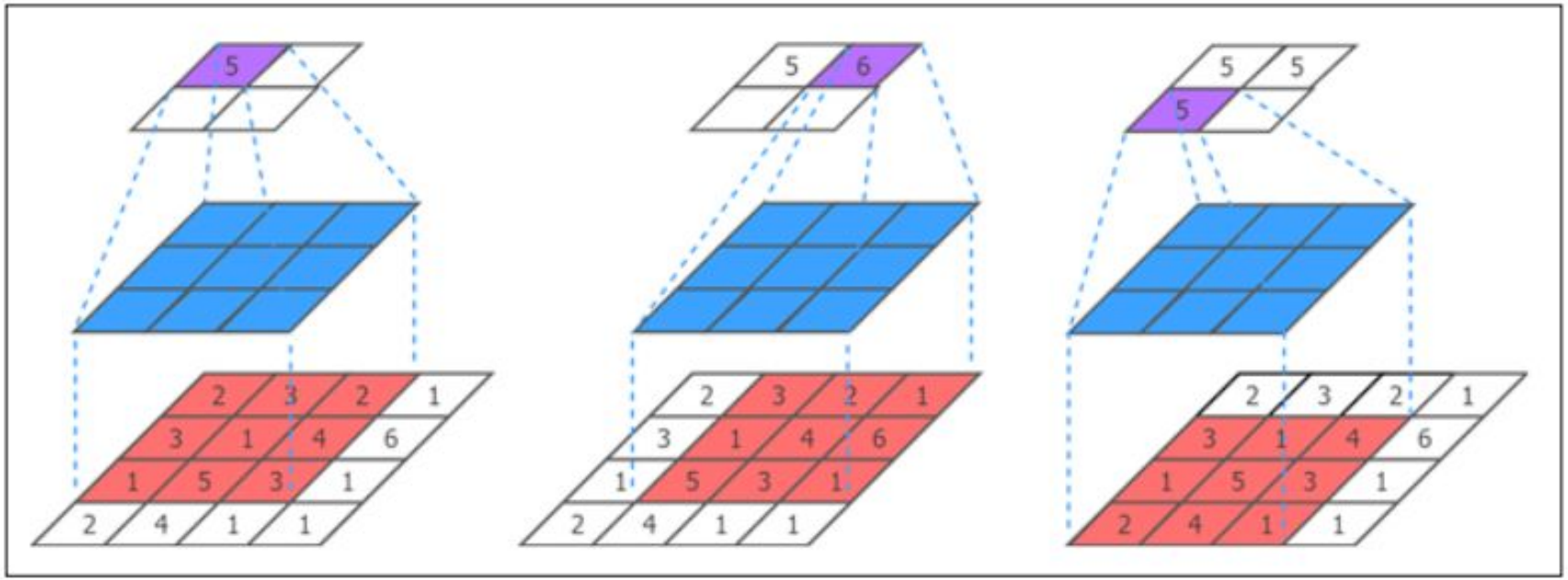
Sigmoid



Linear



Pooling operation

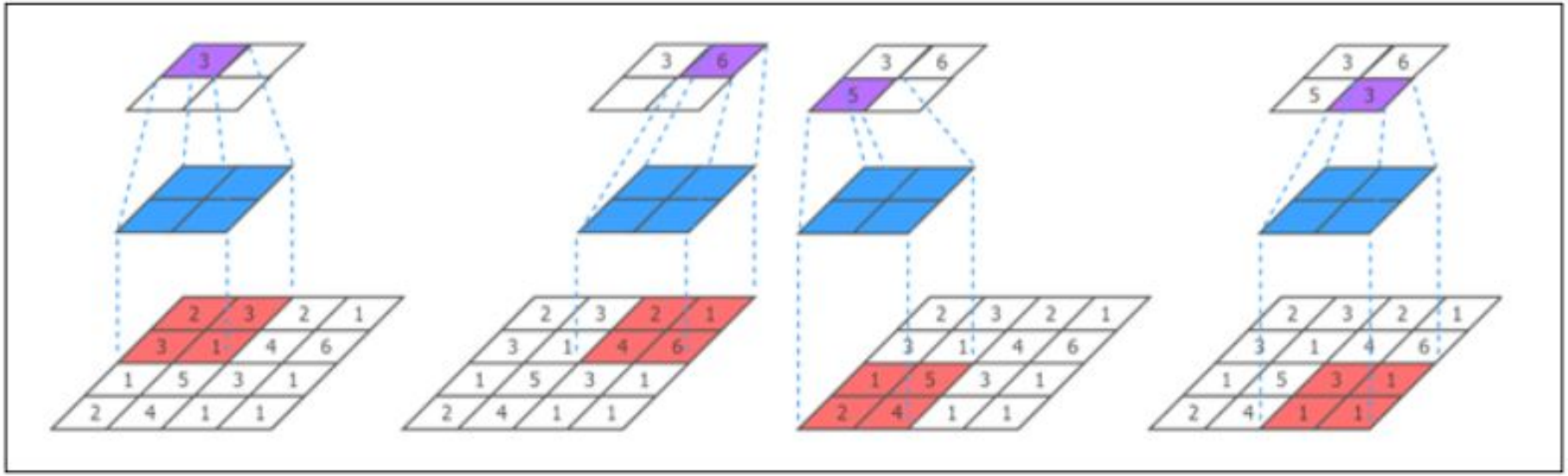


$$h_{i,j} = \max\left(\left\{x_{i,j}, x_{i,j+1}, \ldots, x_{i,j+m-1}, x_{i+1,j}, \ldots, x_{i+1,j+m-1}, \ldots, x_{i+m-1,j}, \ldots, x_{i+m-1,j+m-1}\right\}\right) where 1 \le i, j \le n-m+1$$

filter size (m) = 3, stride = 1, padding = zero

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Max pooling with stride



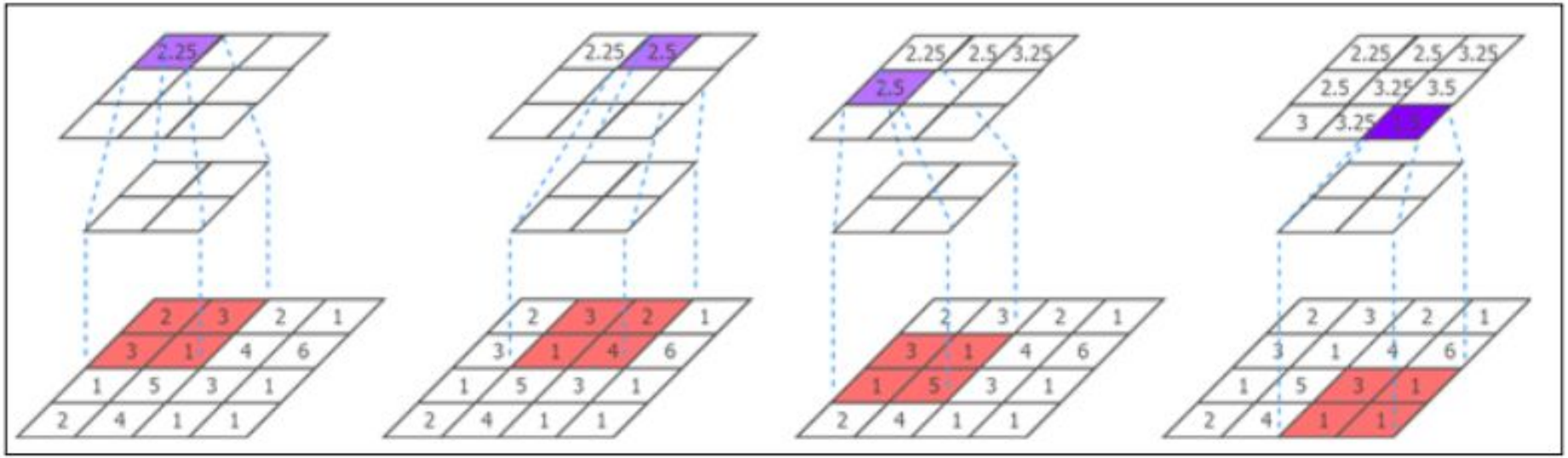
$$h_{i,j} = max\left(\left\{x_{(i-1)\times s_i+1,(j-1)\times s_j+1}, x_{(i-1)\times s_i+1,(j-1)\times s_j+2}, \ldots, x_{(i-1)\times s_i+1,(j-1)\times s_j+m}, x_{(i-1)\times s_i+2,(j-1)\times s_j+1}, \ldots, x_{(i-1)\times s_i+2,(j-1)\times s_j+m}, \ldots, x_{(i-1)\times s_i+m,(j-1)\times s_j+1}, \ldots, x_{(i-1)\times s_i+m,(j-1)\times s_j+m}\right\}\right)$$

$$where 1 \le i \le floor\left[(n-m)/s_i\right]+1 and 1 \le j \le floor\left[(n-m)/s_i\right]+1$$

filter size (m) = 2, stride = 2, padding = zero

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Average pooling

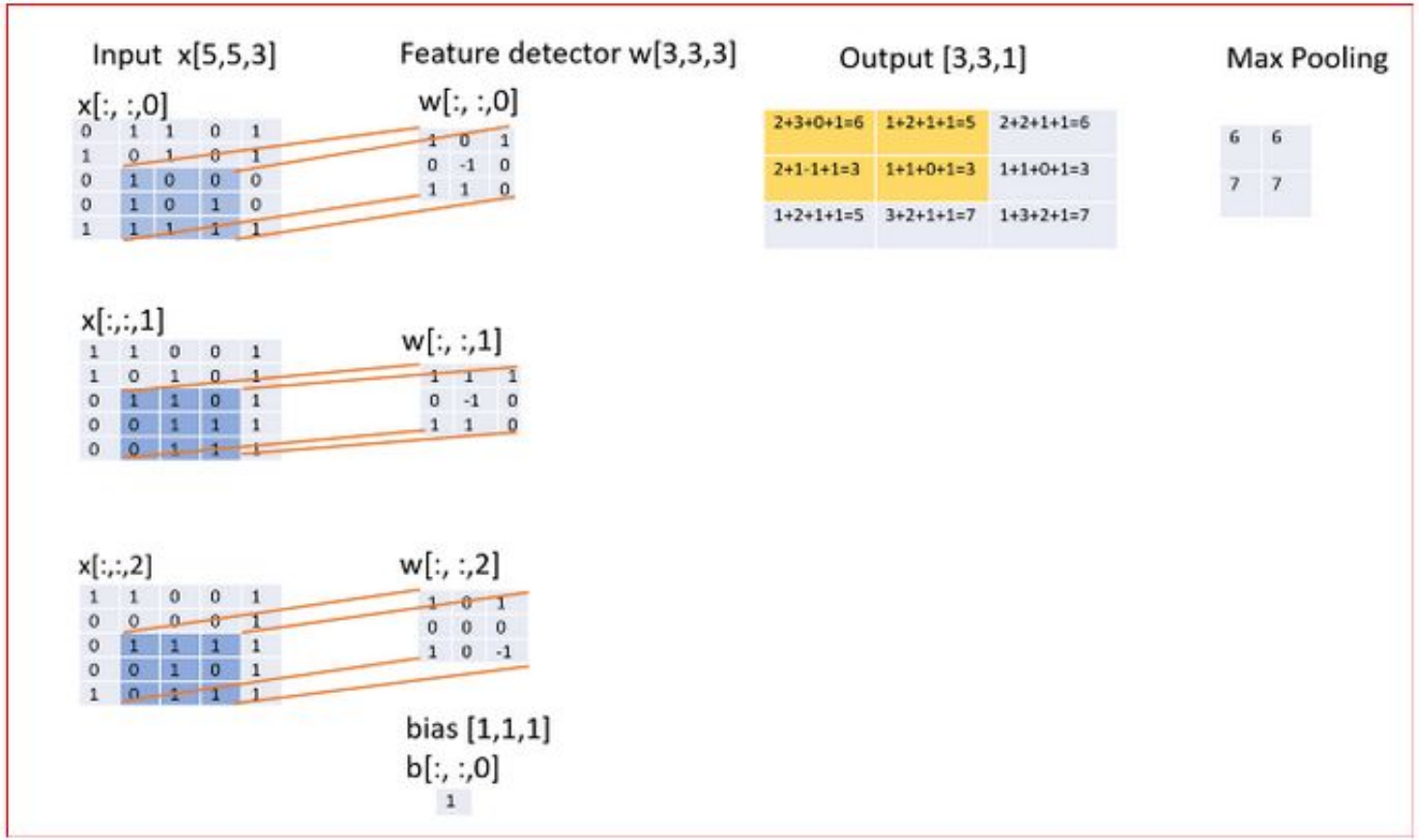


$$h_{i,j} = \frac{x_{i,j}, x_{i,j+1}, \ldots, x_{i,j+m-1}, x_{i+1,j}, \ldots, x_{i+1,j+m-1}, \ldots, x_{i+m-1,j}, \ldots, x_{i+m-1,j+m-1}}{m \times m} \forall i \geq 1, j \leq n - m + 1$$

filter size (m) = 2, stride = 1, padding = zero

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

Example



Renu Khandelwal, Convolutional Neural Network(CNN) Simplified

Fully connected layers

- The initial **fully connected layer** found immediately after the **last convolution or pooling layer.**
- The weight matrix will be $w^{(m,\ \boldsymbol{h}x\boldsymbol{w}x\boldsymbol{d})}$.
- During inference (or prediction), we reshape the output of the last convolution/pooling layer to be of size (***h***x***w***x***d***, 1)
- The **output layer** can be a **softmax classification** layer for a classification problem or a **linear layer** for a regression problem.

Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

CNN in NLP (input)

Let a sentence of **p** words and the length of the input sentences **n** words:

- We will pad the sentence with some special word (if **p< n**).
- We will represent each word in the sentence by a vector of size **k** (one-hot-encoded representation, word2vec, glove).
- A batch of sentences of size b can be represented by a ***b***x***n***x***k*** matrix.

CNN in NLP (input)

k=**13**

Bob and Mary are friends. p=5

Bob plays soccer. p=3

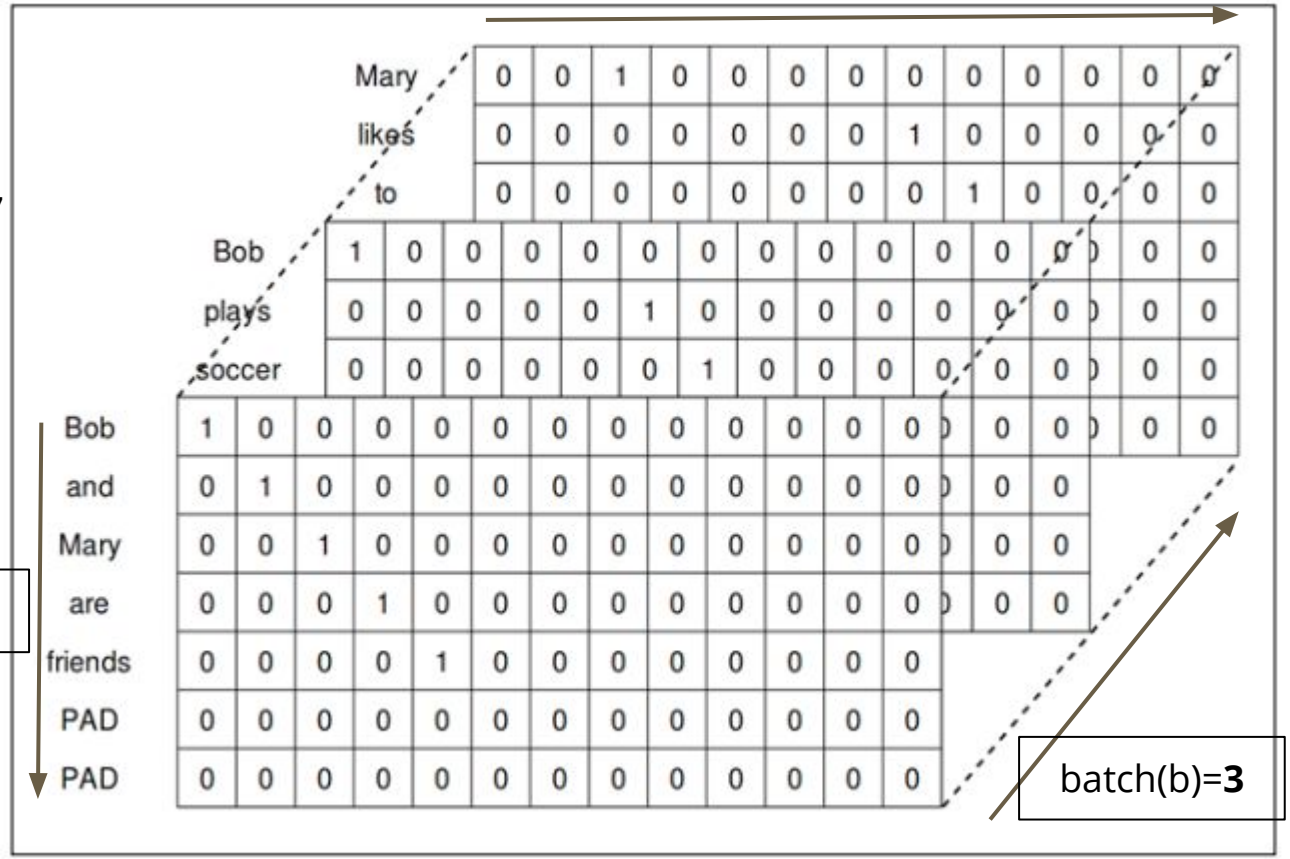
Mary likes to sing in the choir. p=7

special word=**PAD**

word representation: **one-hot**

n=**7**

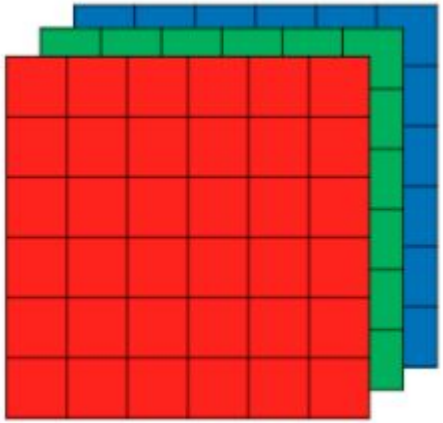
batch(b)=**3**



Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

CNN in NLP (convolution operation)

Image



6 x 6 x 3

hx**w**x**d**

Text (one sentence)

Bob	1	0	0	0	0	0	0	0	0	0	0	0	0
and	0	1	0	0	0	0	0	0	0	0	0	0	0
Mary	0	0	1	0	0	0	0	0	0	0	0	0	0
are	0	0	0	1	0	0	0	0	0	0	0	0	0
friends	0	0	0	0	1	0	0	0	0	0	0	0	0
PAD	0	0	0	0	0	0	0	0	0	0	0	0	0
PAD	0	0	0	0	0	0	0	0	0	0	0	0	0

7 x 13

hx**w** -> **n**x**k**

Prabhu, Understanding of Convolutional Neural Network (CNN) — Deep Learning

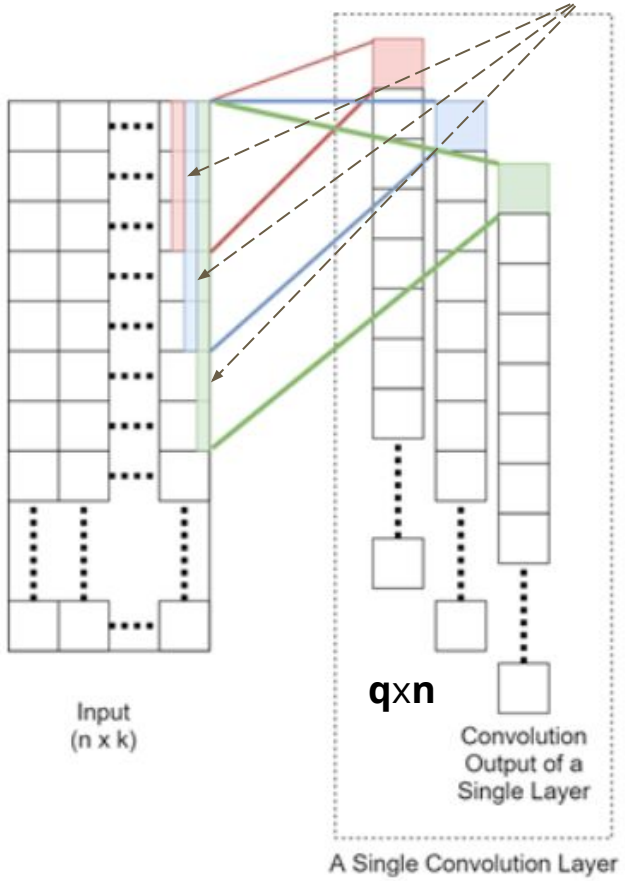
Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

CNN in NLP (convolution operation)

By convolving the input **x** (**n**x**k**) with a weight matrix **W** (**m**x**k**), we will produce an output of **h** (**1**x**n**)

$$h_{i,1} = \sum_{j=1}^{m} \sum_{l=1}^{k} w_{j,l} x_{i+j-1,l} \qquad h = W * x + b$$

- We have parallel layers with different convolution filter sizes.
- Each convolution layer outputs a hidden vector of size **1**×**n**.
- We will concatenate these outputs to form the input to the next layer of size **q**x**n.**
- **q** is the number of parallel layers we will use.



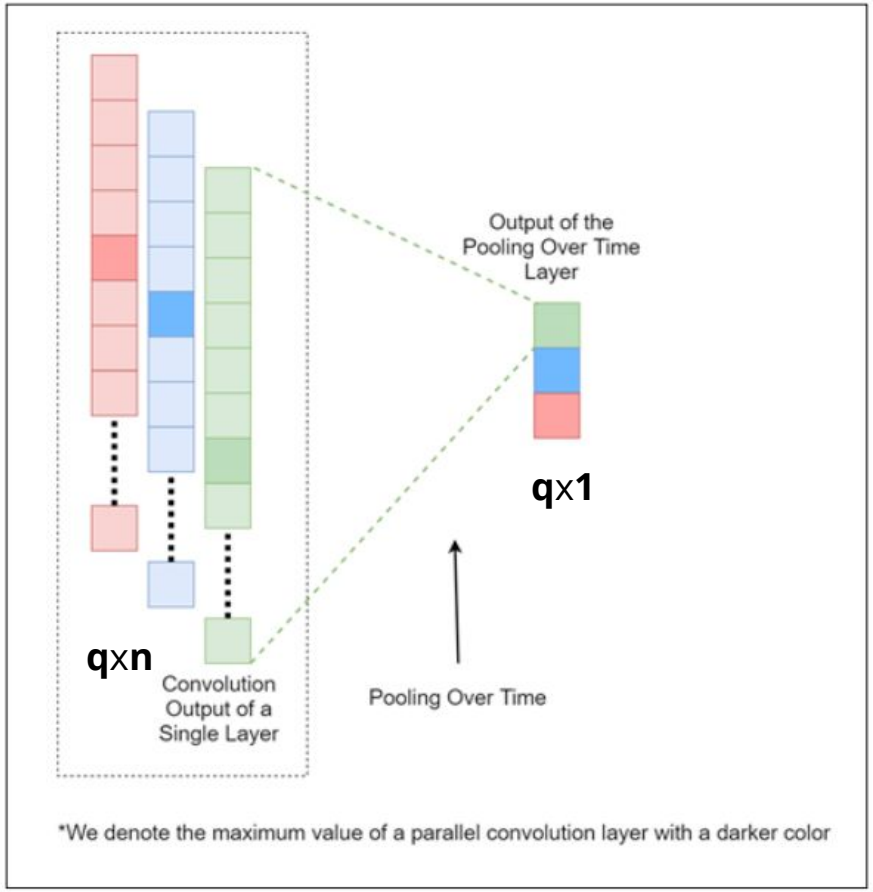
Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

CNN in NLP (pooling operation)

- Let's assume the output of the last layer **h** is of size **q**x**n.**

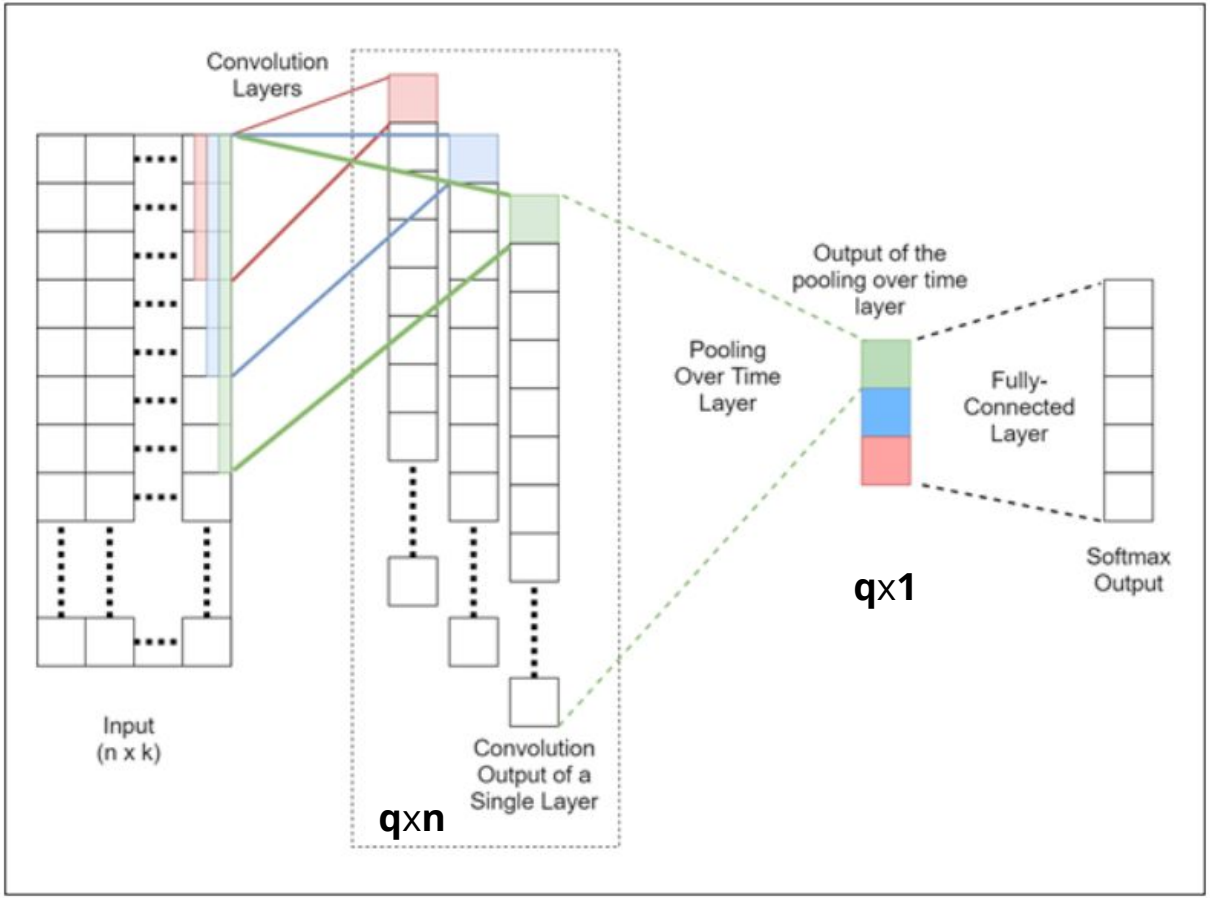
- The pooling layer would produce an output **h'** of size **q**×**1** output.

$$h_{i,1}' = \left\{ max\left(h^{(i)} \right) where 1 \le i \le q \right\}$$



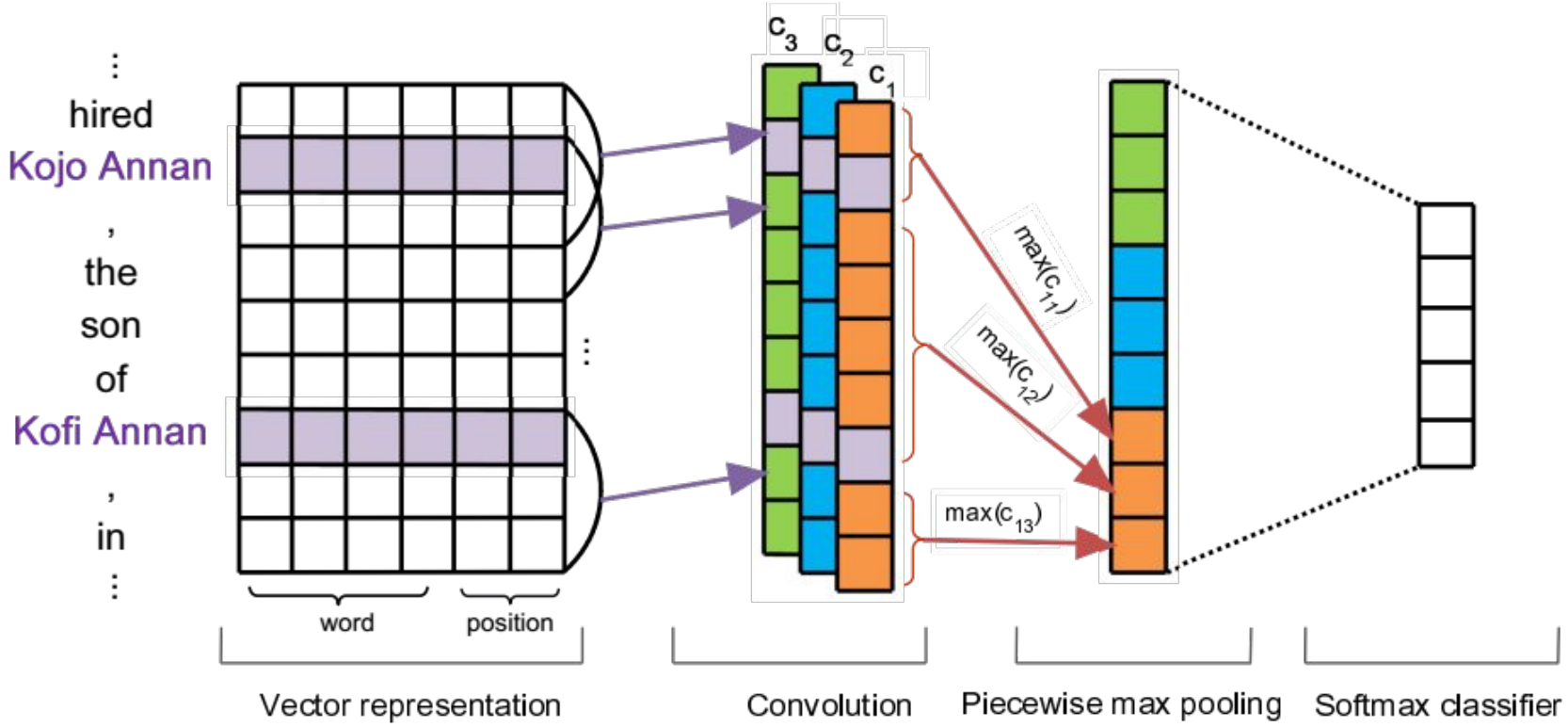
Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

CNN in NLP (architecture)



Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

CNN in NLP (relation extraction)



Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1753–1762).

References

- Ganegedara, T. (2018). Natural Language Processing with TensorFlow. Packt Publishing Ltd. https://github.com/PacktPublishing/Natural-Language-Processing-with-TensorFlow
- Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1753–1762).

Auxiliar Bibliography

- Stefan Kojouharov, Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Big Data
- Renu Khandelwal, Convolutional Neural Network: Feature Map and Filter Visualization
- Renu Khandelwal, Convolutional Neural Network(CNN) Simplified
- Prabhu, Understanding of Convolutional Neural Network (CNN) — Deep Learning
- Activation Function

Questions?

Exercises

- Sentence Classification with CNN (Simple Architecture)
- Sentence Classification with CNN (Complex Architecture)

CNN

Juan Luis García Mendoza, Mario Ezra Aragón, Adrián Pastor López Monroy, Luis Villaseñor Pineda, Manuel Montes y Gómez