

# Najważniejsze architektury

---

w głębokim uczeniu maszynowym

# Pojęcia na dziś

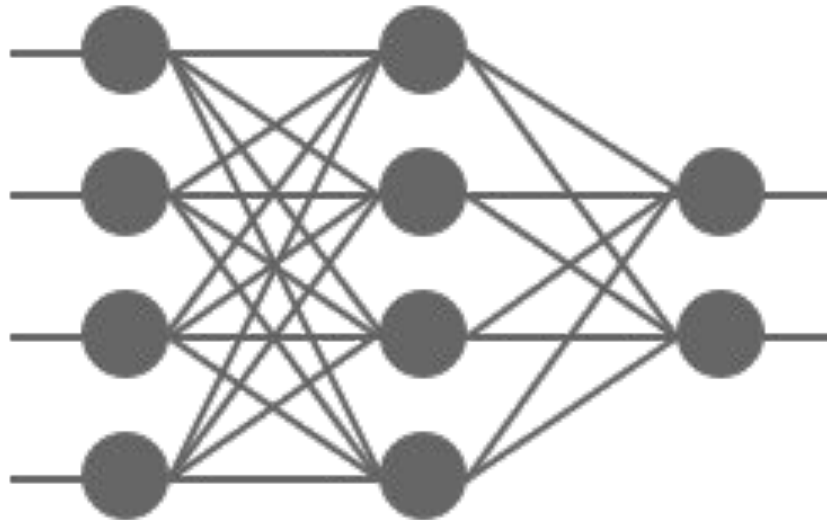
- FCN
- CNN
- RNN
- Transformery



**FCN**

# FCN

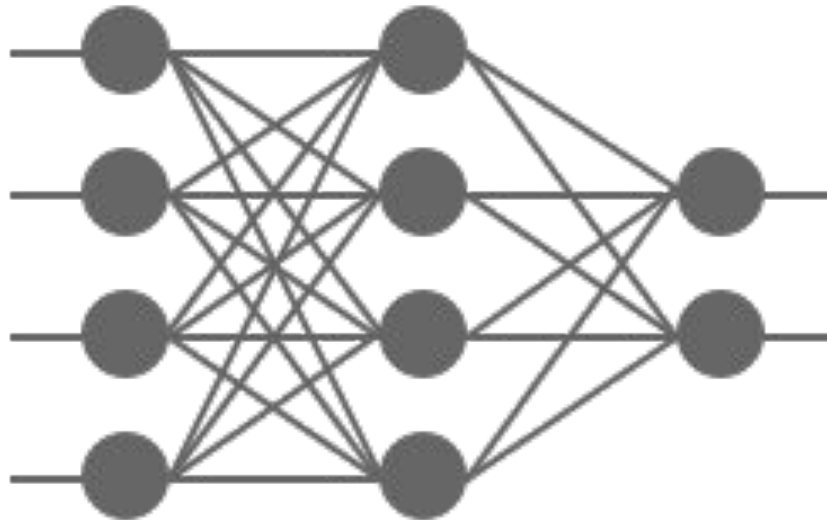
Fully Connected Network



- wejściem do każdego neuronu jest **ważona suma wartości** wszystkich neuronów w poprzedniej warstwie
- wagi dla całej warstwy opisuje **macierz parametrów**, który model się uczy w czasie treningu
- po obliczeniu ważonych sum używa się **funkcji aktywacji**

# FCN

Fully Connected Network



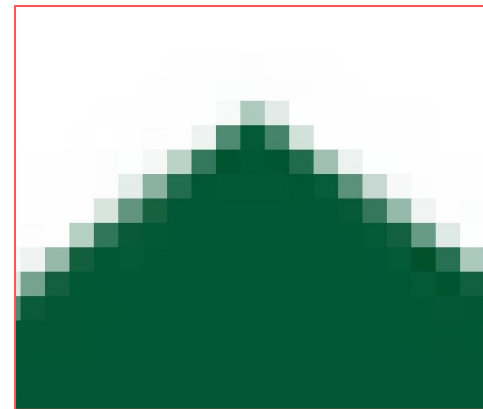
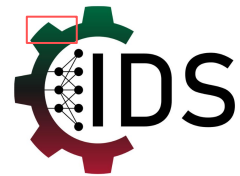
ZBYT PROSTE? PATRZ: MLP-MIXER :)

- wejściem do każdego neuronu jest **ważona suma wartości** wszystkich neuronów w poprzedniej warstwie
- wagi dla całej warstwy opisuje **macierz parametrów**, który model się uczy w czasie treningu
- po obliczeniu ważonych sum używa się **funkcji aktywacji**

**CNN**

# CNN - komputerowa reprezentacja obrazu

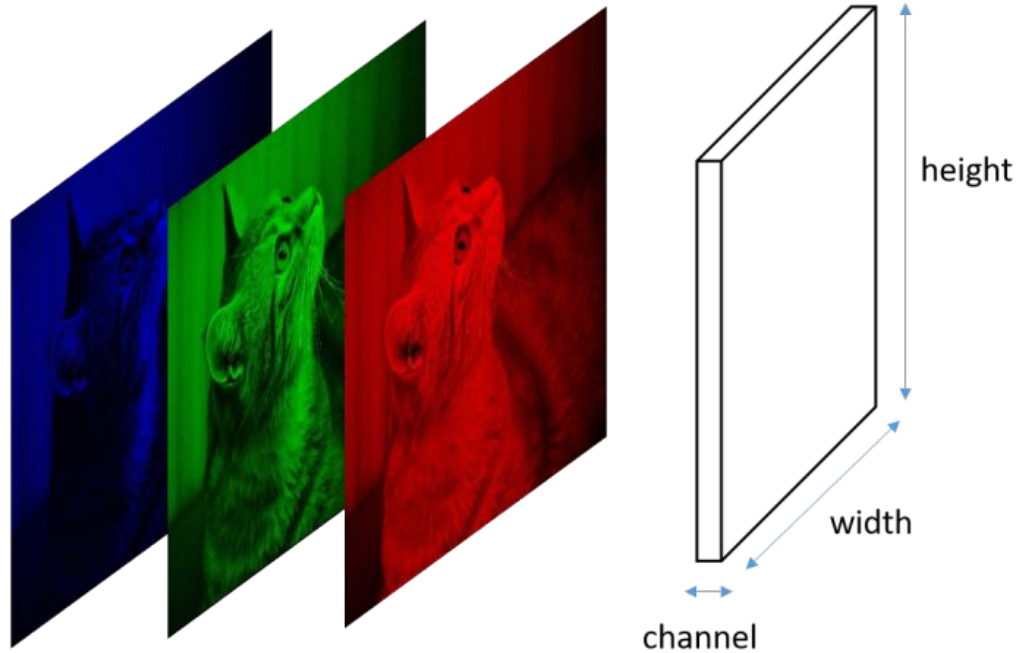
Convolutional Neural Network



Obraz jest **tablicą liczb** (np. 0-255)!

# CNN - komputerowa reprezentacja obrazu

Convolutional Neural Network





# CNN - filtrowanie / konwolucja

Convolutional Neural Network

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

\*

1	0	-1
1	0	-1
1	0	-1

=

6		

$7 \times 1 + 4 \times 1 + 3 \times 1 + 2 \times 0 + 5 \times 0 + 3 \times 0 + 3 \times -1 + 3 \times -1 + 2 \times -1 = 6$

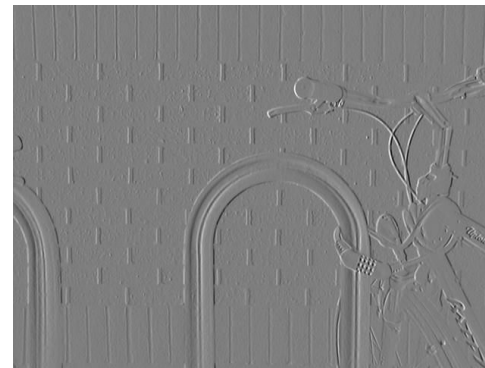
- filtr **przesuwa się po całym obrazie** i oblicza **iloczyn skalarny**
- początki - lata 60., przybliżenie gradientu jasności obrazu
- wykrywa **cechy obrazu**

# CNN - filtrowanie / konwolucja

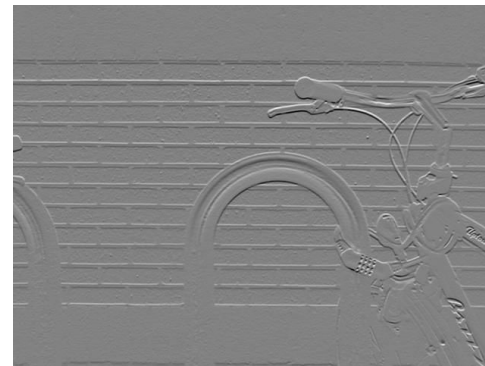
Convolutional Neural Network



$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$



$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



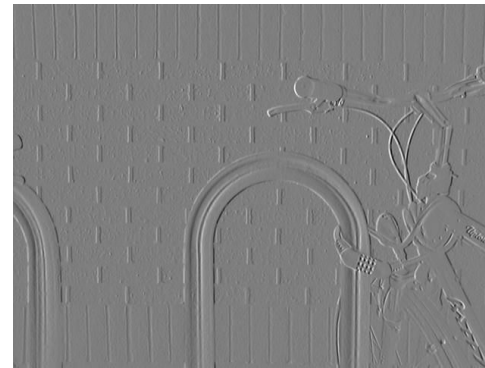
# CNN - filtrowanie / konwolucja

Convolutional Neural Network

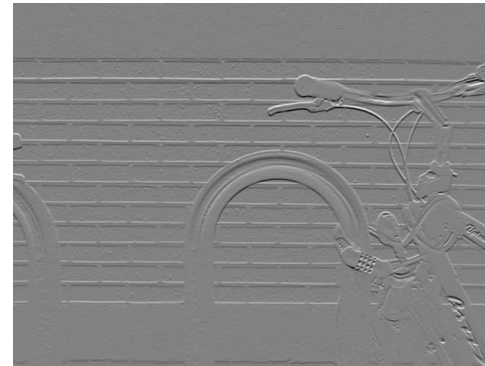
tego uczy się model  
(tzw. feature map / kernel / filter)



$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

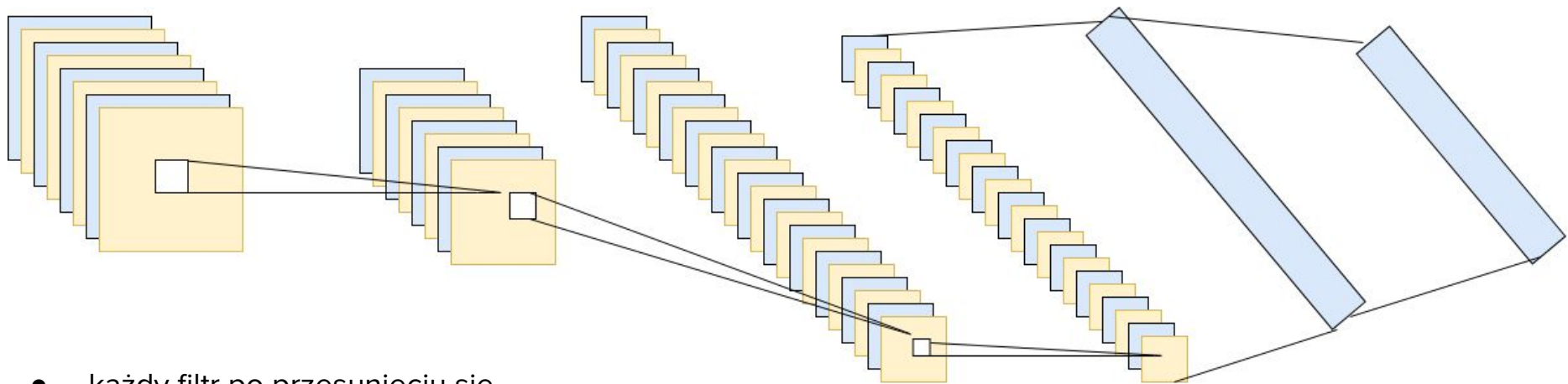


$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



# CNN - architektura

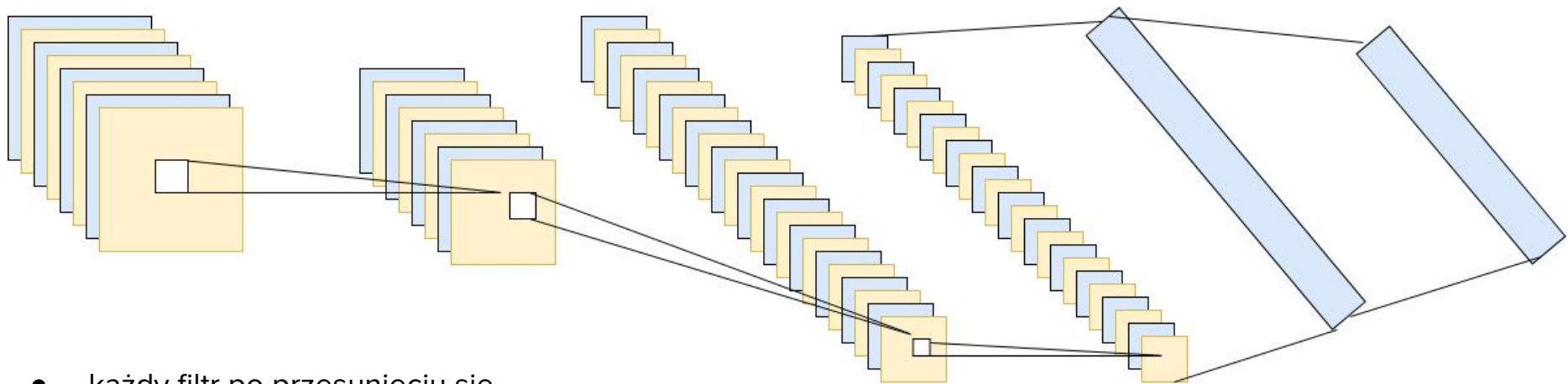
Convolutional Neural Network



- każdy filtr po przesunięciu się przez objętość w warstwie  $l$  **produkuje 1 kanał w warstwie  $l+1$**  (filtr ma tyle kanałów, ile poprzednia warstwa)
- na końcu rozwijamy wszelkie wartości w jeden długi wektor i używamy zwykłych warstw gęsto połączonych

# CNN - architektura

Convolutional Neural Network

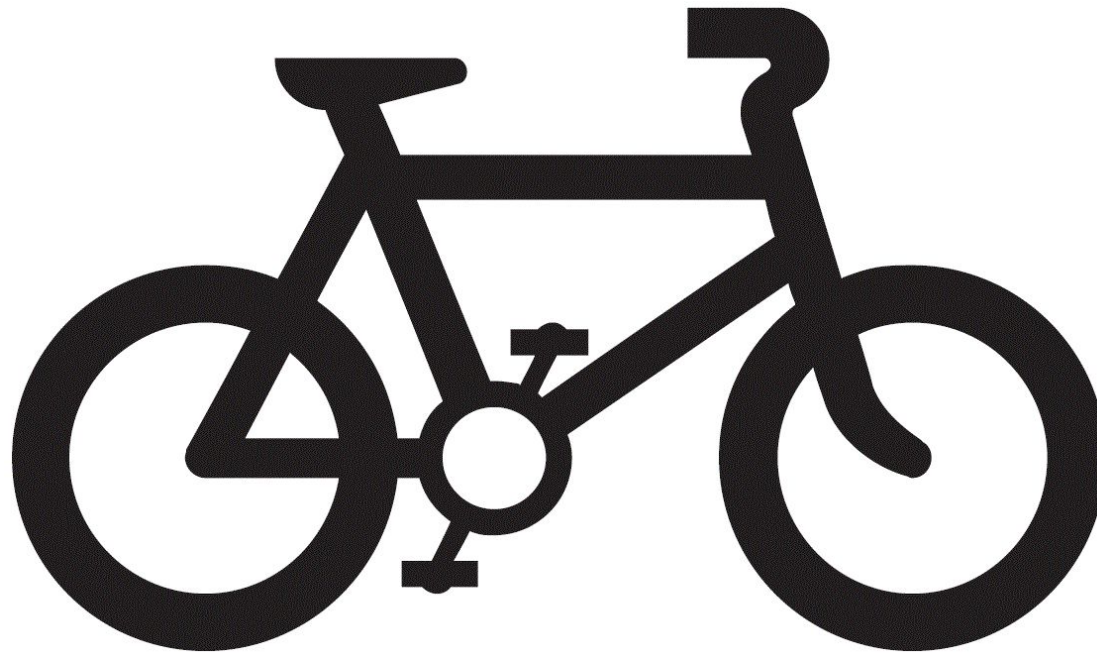


- każdy filtr po przesunięciu się przez objętość w warstwie  $l$  **produkuje 1 kanał w warstwie  $l+1$**  (filtr ma tyle kanałów, ile poprzednia warstwa)
- na końcu rozwijamy wszelkie wartości w jeden długi wektor i używamy zwykłych warstw gęsto połączonych

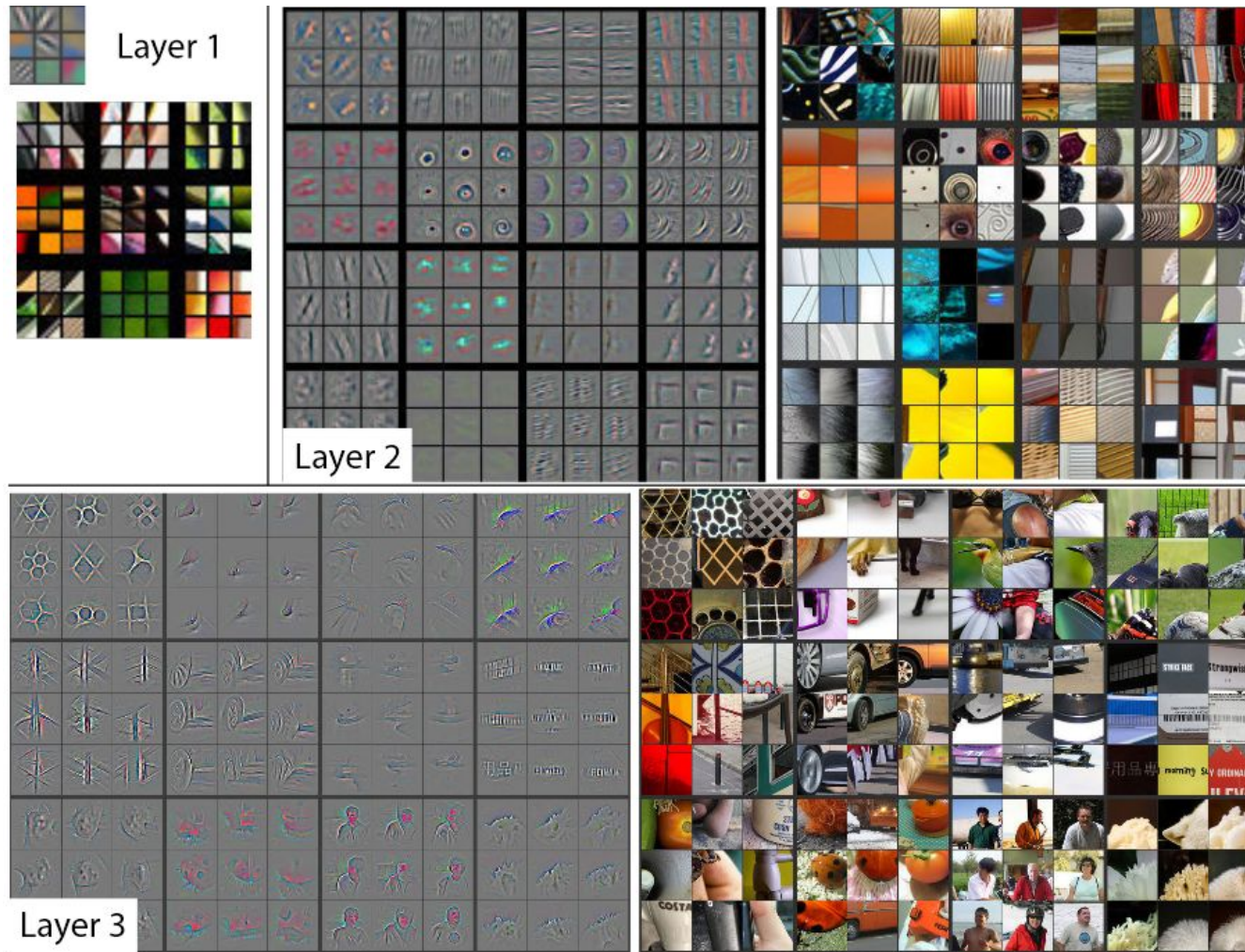
**TRANSFER LEARNING!**

# CNN - intuicja

Convolutional Neural Network





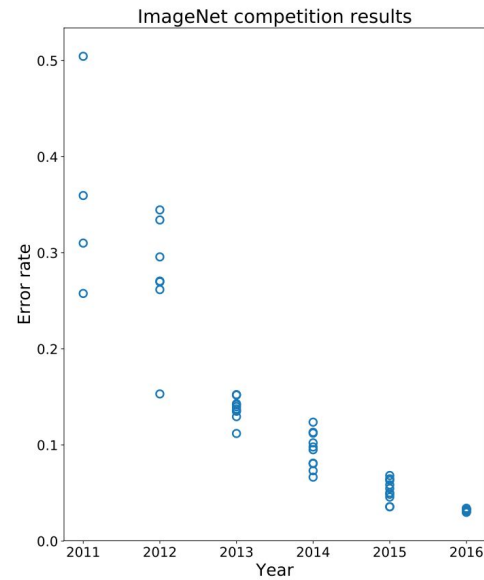


# CNN - historia

Convolutional Neural Network



LeNet (1998)



AlexNet (2012)

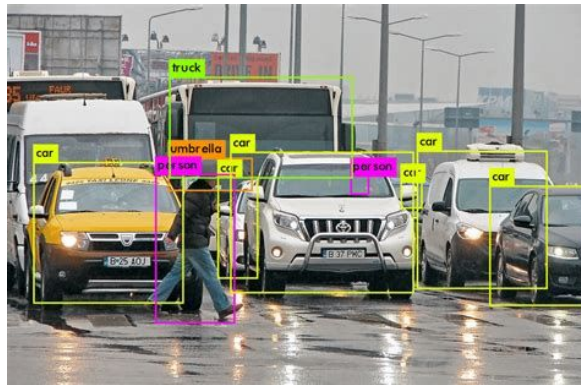
?

2021

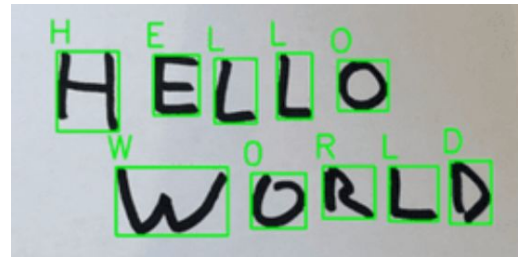


# CNN - zastosowania

Convolutional Neural Network



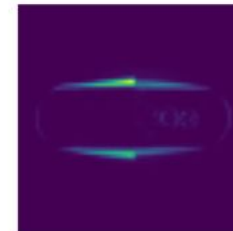
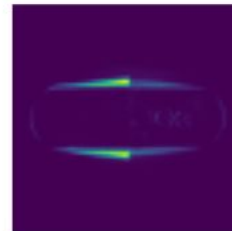
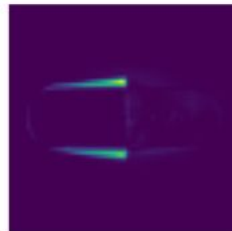
detekcja obiektów



rozpoznawanie tekstu



segmentacja



detekcja anomalii

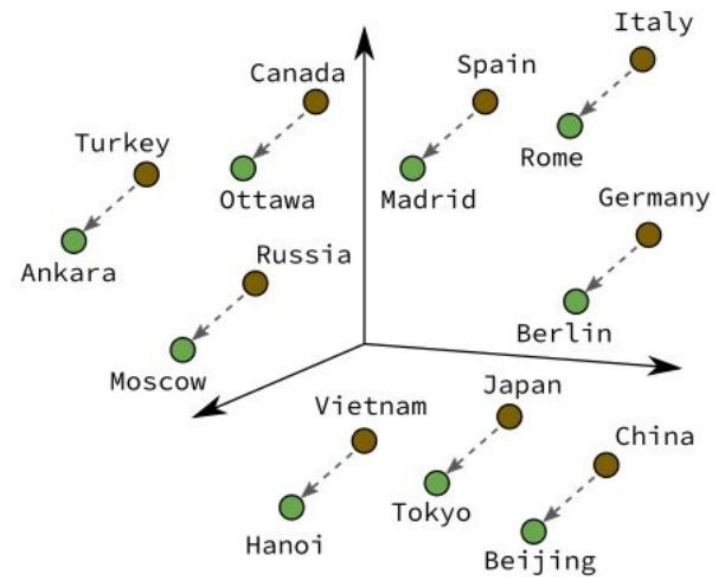
**RNN**

# RNN - reprezentacja słów

Recurrent Neural Network

	cat	mat	on	sat	the
<b>the</b> =>	0	0	0	0	1
<b>cat</b> =>	1	0	0	0	0
<b>sat</b> =>	0	0	0	1	0

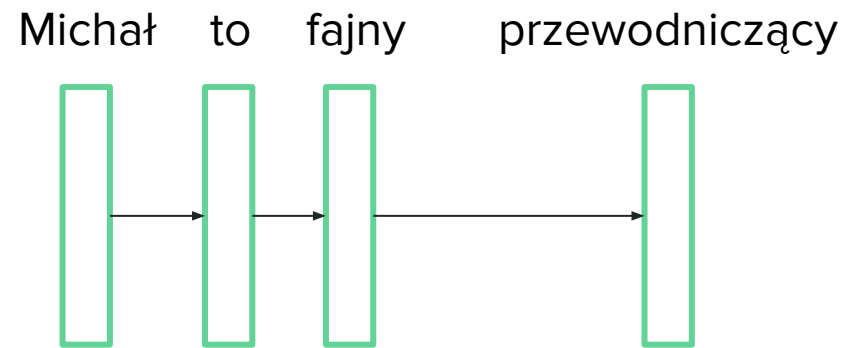
one-hot encoding



word embedding

# RNN - intuicja

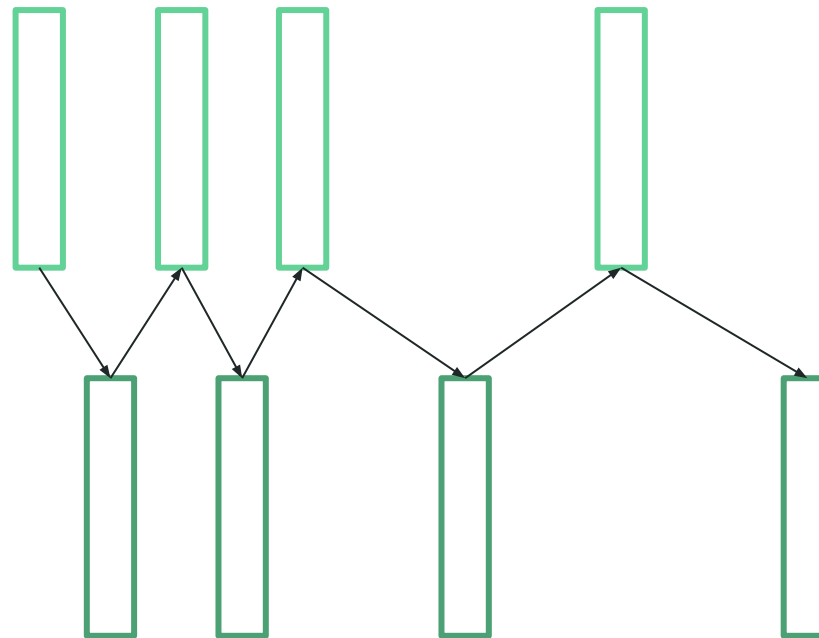
Recurrent Neural Network



# RNN - intuicja

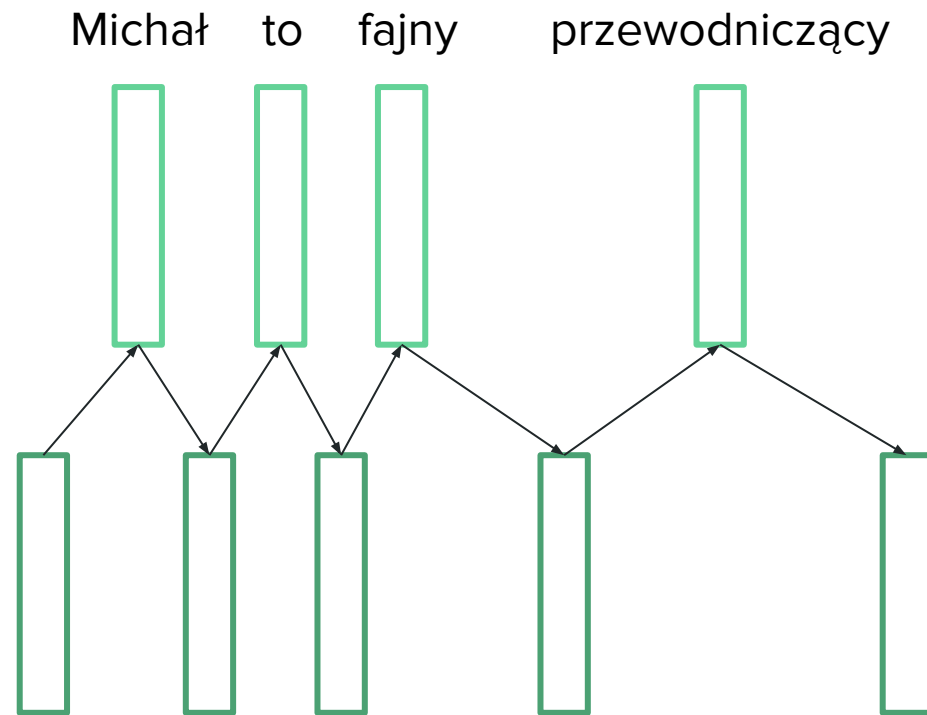
Recurrent Neural Network

Michał to fajny przewodniczący



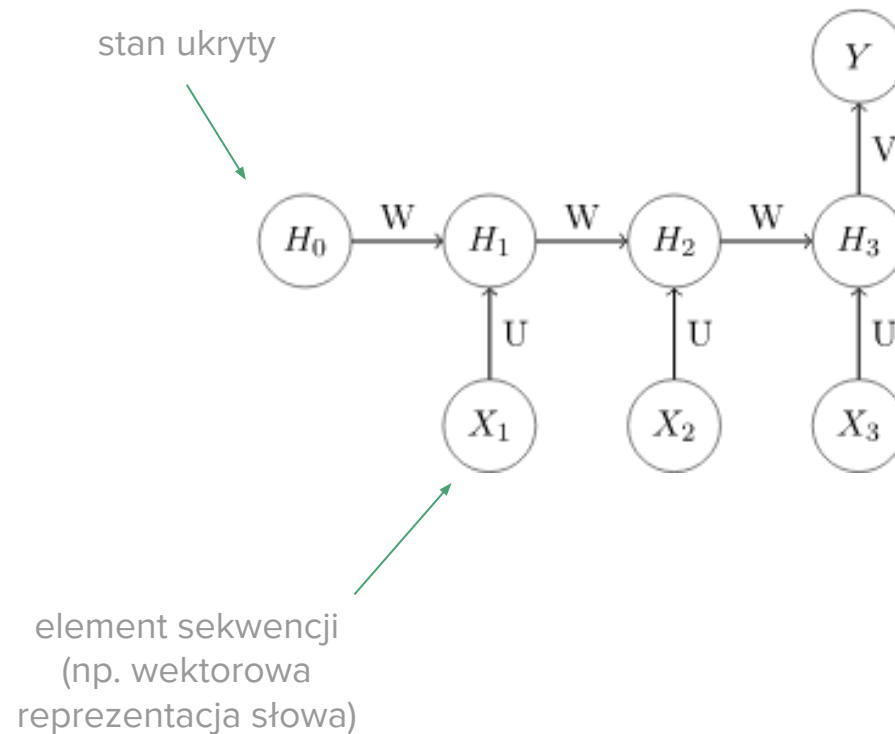
# RNN - intuicja

Recurrent Neural Network



# RNN - architektura

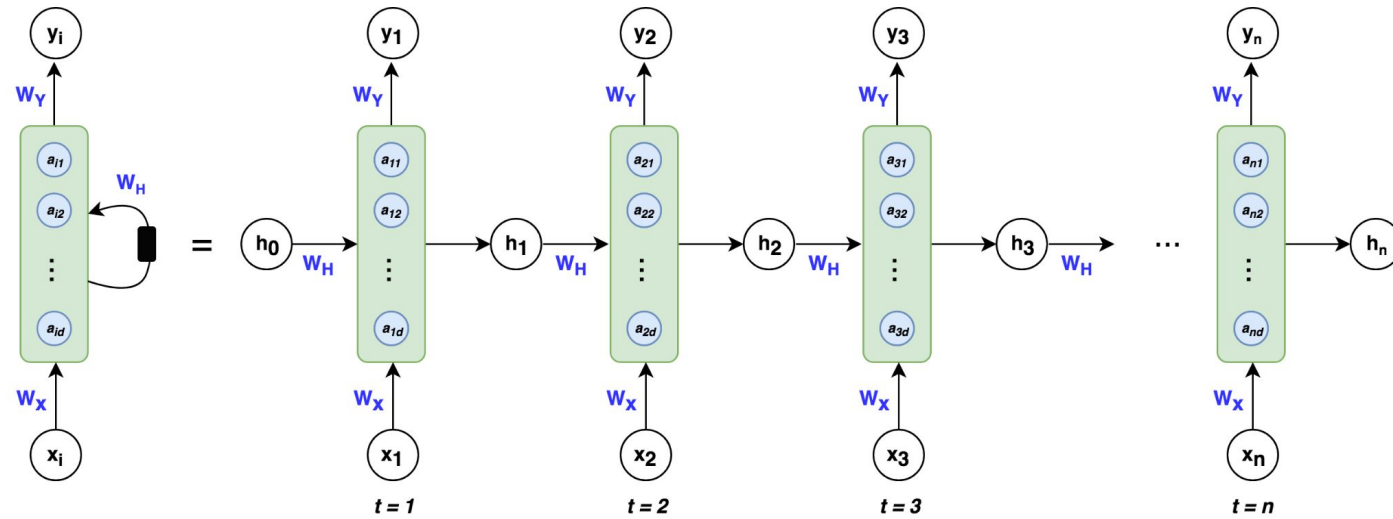
Recurrent Neural Network



- początkowy stan ukryty **nie zawiera żadnej informacji** (np. wektor zer)
- nowy stan ukryty to **suma (dosłownie) poprzedniego stanu ukrytego i aktualnego wejścia** (przemnożonych przez macierze / przetworzonych przez warstwę gęstopołączoną)
- wyjście to przetworzony ostatni stan ukryty
- **są tylko 3 macierze** (te same, kroków może być nieskończenie wiele)

# RNN - inne reprezentacje

Recurrent Neural Network

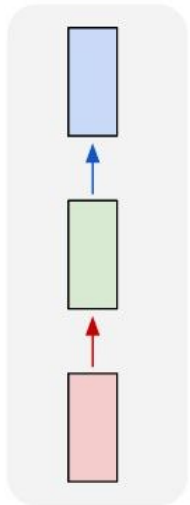




# RNN - różne możliwości

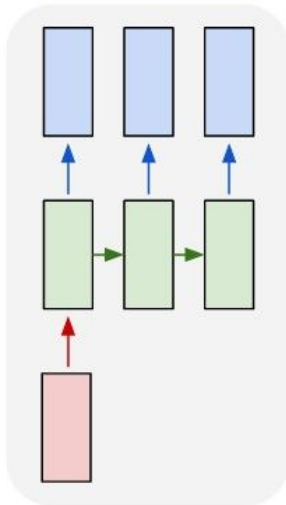
Recurrent Neural Network

one to one



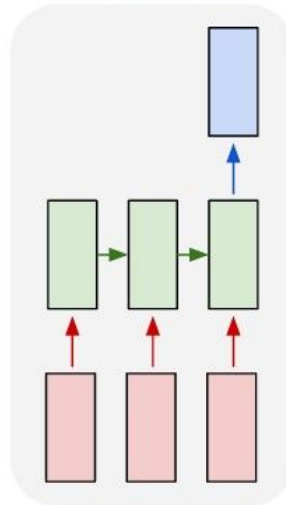
po prostu  
NN

one to many



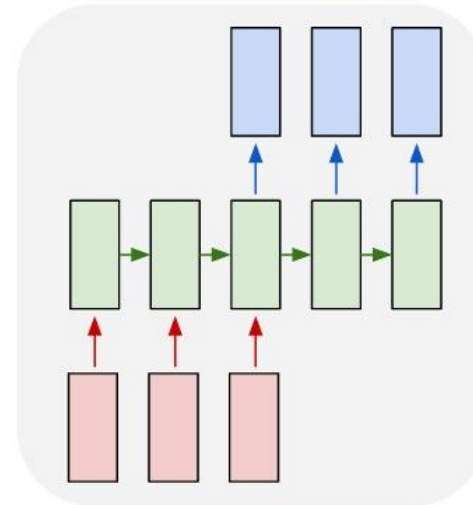
generowanie  
muzyki

many to one



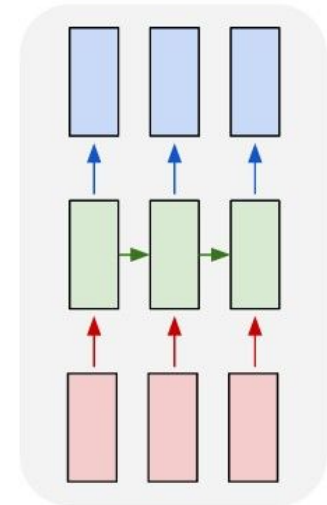
analiza  
sentymentu

many to many



tłumaczenie

many to many

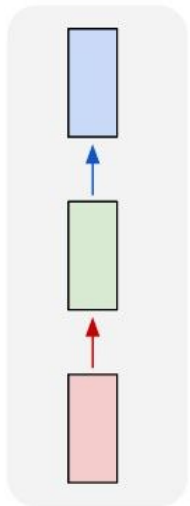


NER

# RNN - różne możliwości

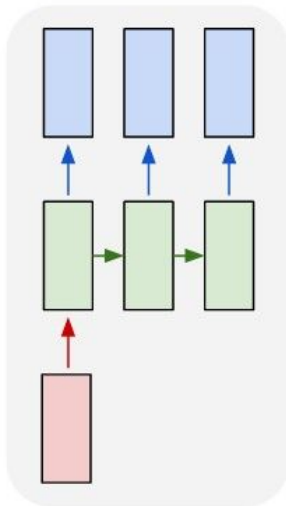
Recurrent Neural Network

one to one



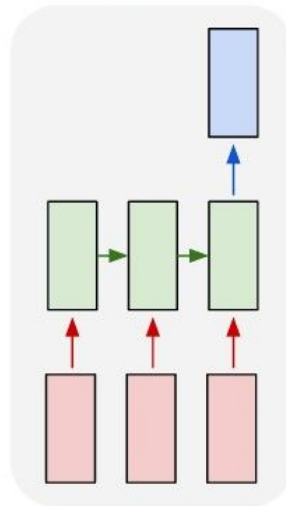
po prostu  
NN

one to many



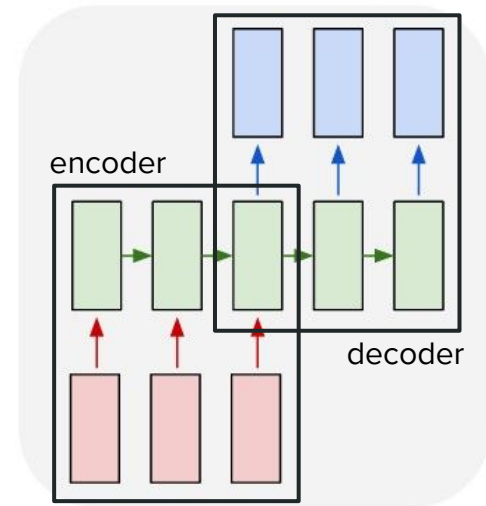
generowanie  
muzyki

many to one



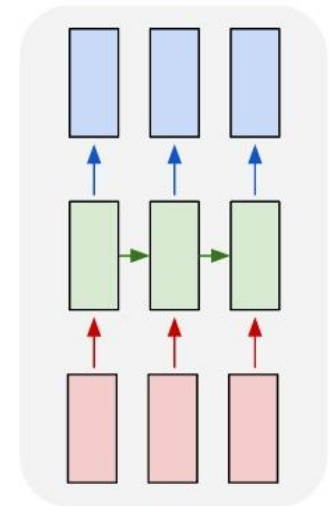
analiza  
sentymentu

many to many



tłumaczenie

many to many



NER

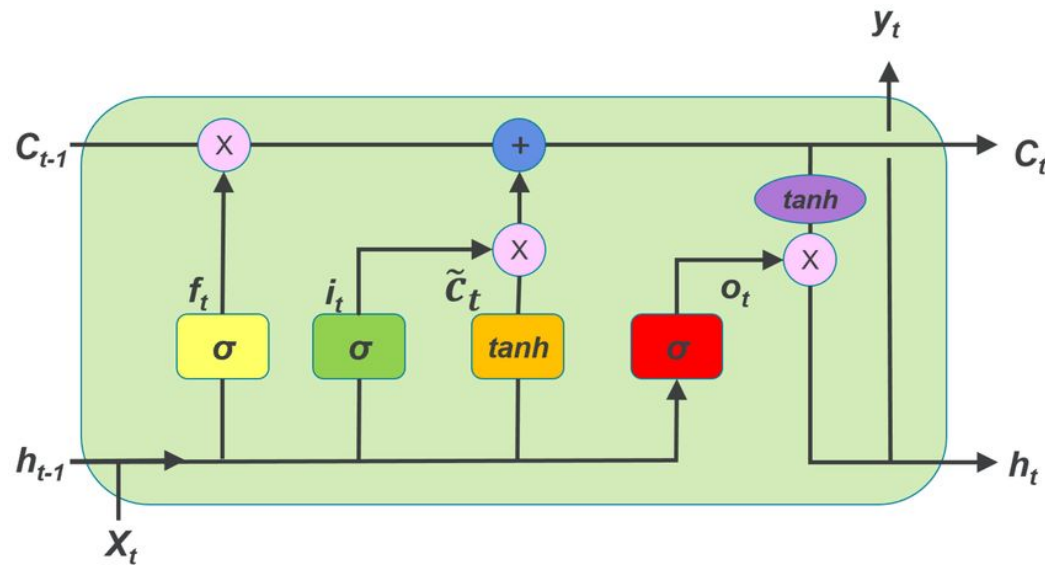
# RNN - ulepszenia

Recurrent Neural Network

**GRU** (Gated Recurrent Unit) - dla krótszych, mniej złożonych sekwencji

i

**LSTM** (Long Short-Term Memory) - dla dłuższych, bardziej złożonych sekwencji

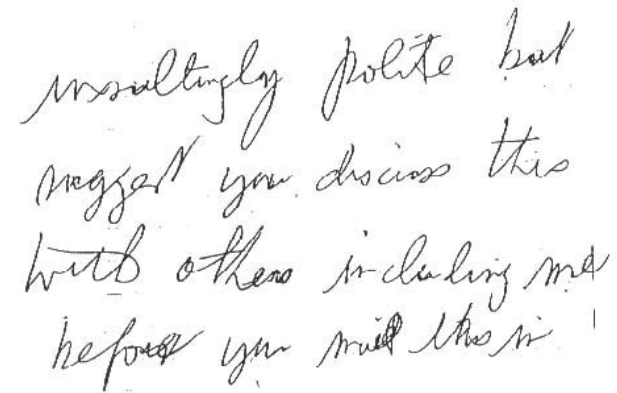


# RNN - zastosowania

Recurrent Neural Network



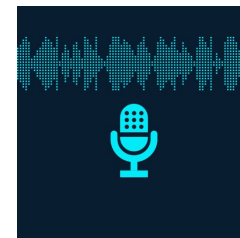
modelowanie języka



rozpoznawanie pisma



tłumaczenie



rozpoznawanie mowy