

analiza danych

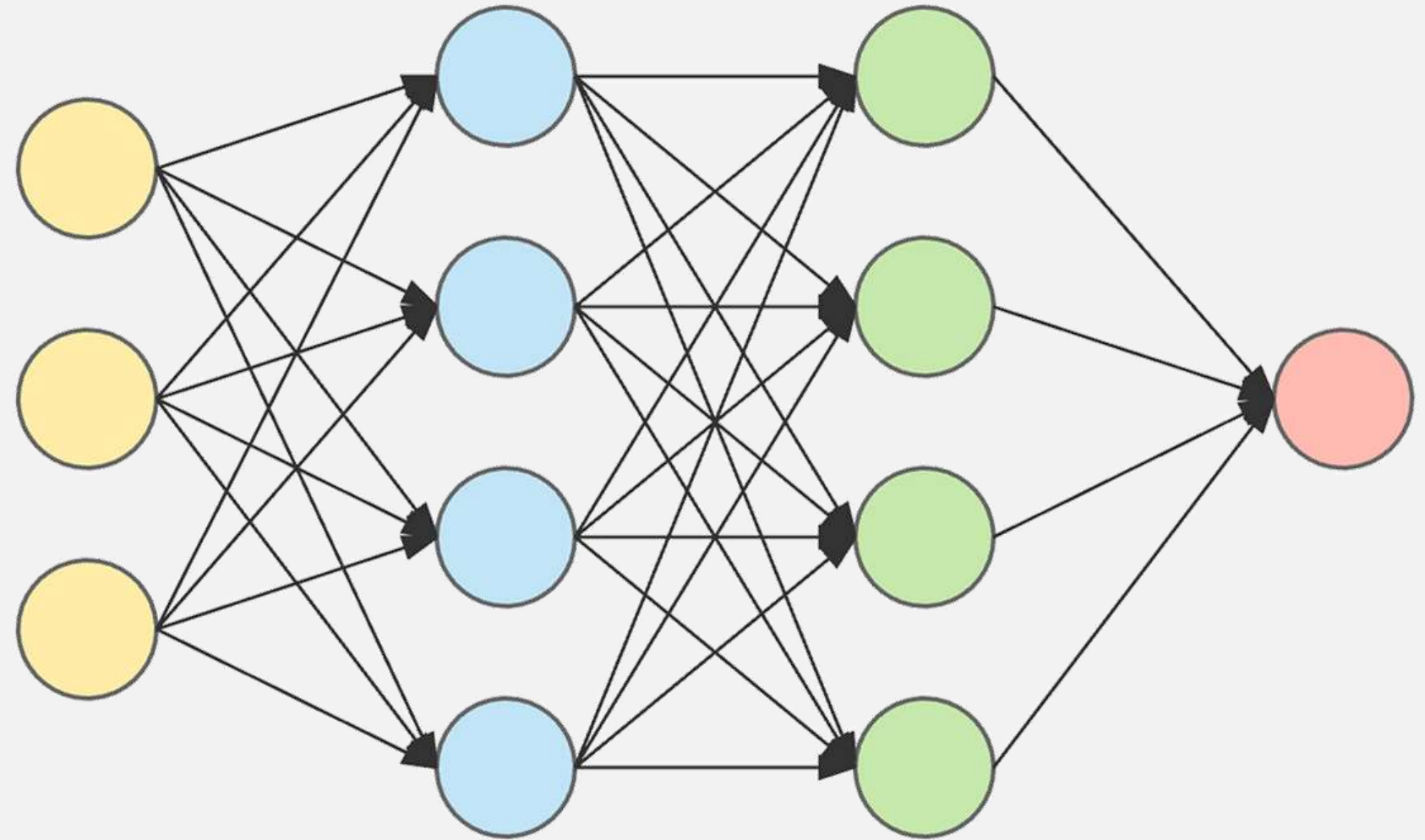
1. Wykład wstępny
2. Elementy algebry liniowej
3. Wybrane aspekty obliczeń komputerowych
4. Architektura pojedynczej sieci neuronowej cz.1
- 5. Architektura pojedynczej sieci neuronowej cz. 2**
6. Architektura modelu uczenia głębokiego - wprowadzenie
7. Architektura modelu uczenia głębokiego – rozszerzenia
8. Walidacja modelu
9. Przykłady innych struktur sieci
10. Przykłady własnych badań cz. 1
11. Przykłady własnych badań cz. 2
12. -15. Prezentacje zaliczeniowe

-
1. DNN
 2. CNN
 3. RNN
 4. LSTM

DNN

dense neural network

DNN



1. $\mathbf{z}_1 = \mathbf{x}W_1$
2. $\mathbf{z}_2 = \mathbf{a}_1W_2$
3. $\mathbf{z}_3 = \mathbf{a}_2W_3$

$$\begin{aligned} &\rightarrow \mathbf{a}_1 = f(\mathbf{z}_1) \\ &\rightarrow \mathbf{a}_2 = f(\mathbf{z}_2) \\ &\rightarrow \hat{\mathbf{y}} = f(\mathbf{z}_3) \end{aligned}$$

CNN

convolutional neural network

- Oryginalnie stosowane do rozpoznawania obrazów → problem klasyfikacyjny
- Cechy charakterystyczne obrazów
 - wielowymiarowa struktura (2D) → 2 źródła informacji:
 - wartość zmiennej = feature = wartość pixela
 - położenie zmiennych względem siebie
 - Istnieją grupy pixeli o podobnych wartościach = korelacja między pixelami
 - Bardzo wiele pixeli = features = bardzo duże rozmiary modelu:
 $\mathbf{x}, \mathbf{W}_i, \mathbf{z}_i, \mathbf{a}_i$

- Bardzo wiele pixeli = bardzo duże rozmiary modelu
- Kumulowanie wartości sąsiadujących pixeli
- Grupa kumulowanych pixeli = kernel
- Operacja kumulacji
 - Konwolucja / Convolution
 - Wyrażenie wartości kilku pixeli jedną liczbą
 - $\sum_{i=1}^P w_i p_i$
 - P liczba pixeli
 - w_i waga i-tego pixela
 - p_i wartość i-tego pixela

CNN

3 x 3 kernel → przykład

Wartości pixeli

1	2	1
3	1	3
2	1	1

CNN

3 x 3 kernel → przykład

Wartości pixeli

1	2	1
3	1	3
2	1	1

Kernel → wagi

-1	0	1
-2	0	2
-1	0	1

CNN

3 x 3 kernel → przykład

Wartości pixeli

1	2	1
3	1	3
2	1	1

Kernel → wagi

-1	0	1
-2	0	2
-1	0	1

Konwolucja

	$\sum_{i=1}^P w_i p_i$	

CNN

3 x 3 kernel → przykład

Wartości pixeli

Kernel → wagi

Konwolucja

$$1 \cdot (-1) + 2 \cdot 0 + 1 \cdot 1 + 3 \cdot (-2) + 1 \cdot 0 + 3 \cdot 2 + 2 \cdot (-1) + 1 \cdot 0 + 1 \cdot 1$$

1	2	1
3	1	3
2	1	1

-1	0	1
-2	0	2
-1	0	1

	-1	

CNN

3 x 3 kernel → przykłady struktur kerneli wykrywanie:

linii pionowych

linii poziomych

narożników

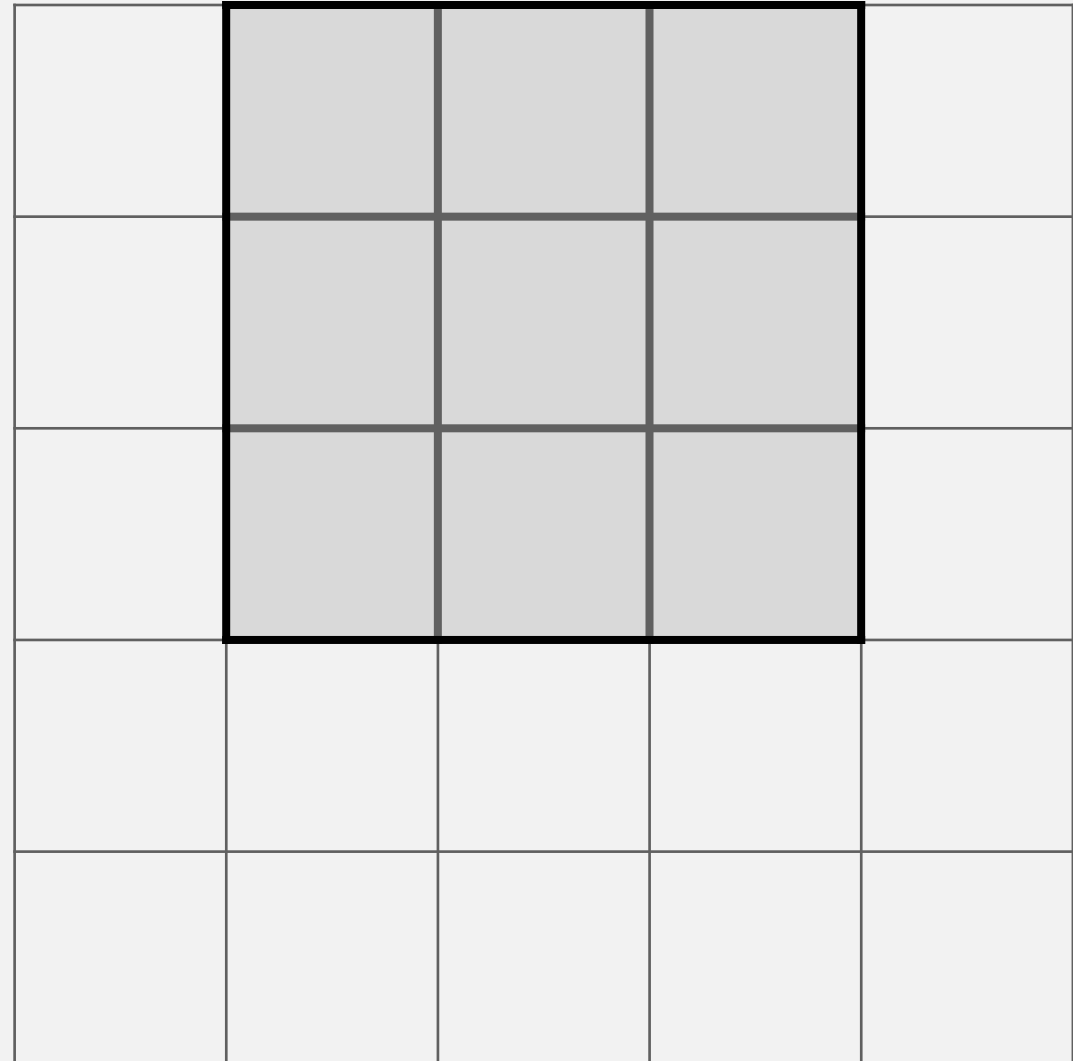
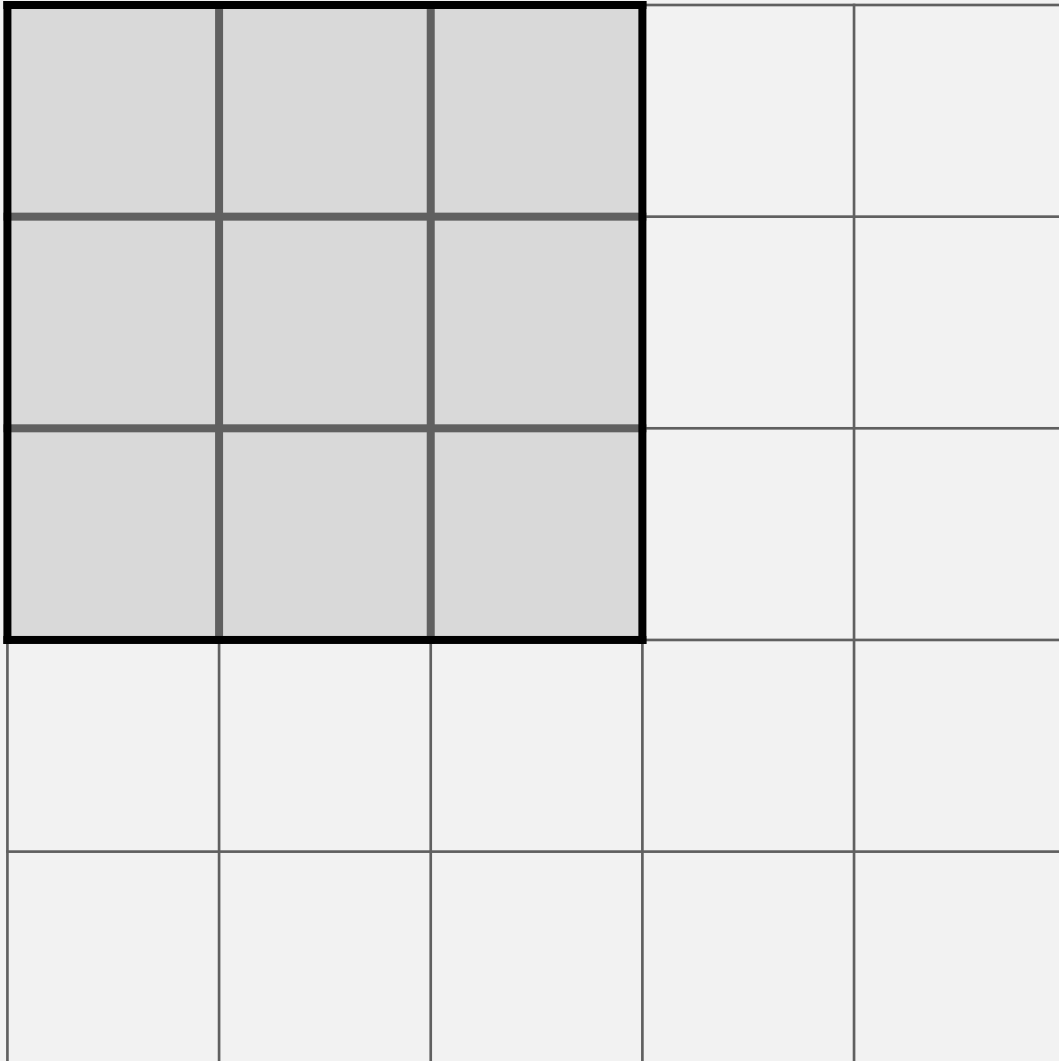
- W rzeczywistości wagi są estymowane w trakcie procesu uczenia

-1	1	-1
-1	1	-1
-1	1	-1

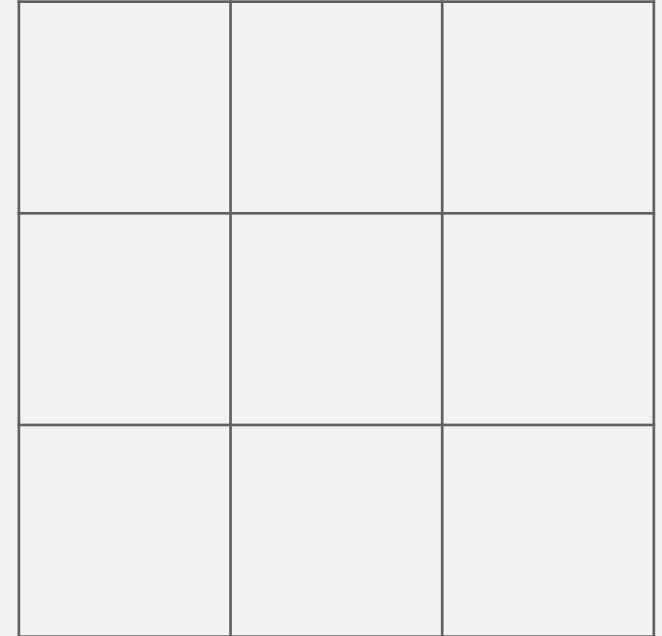
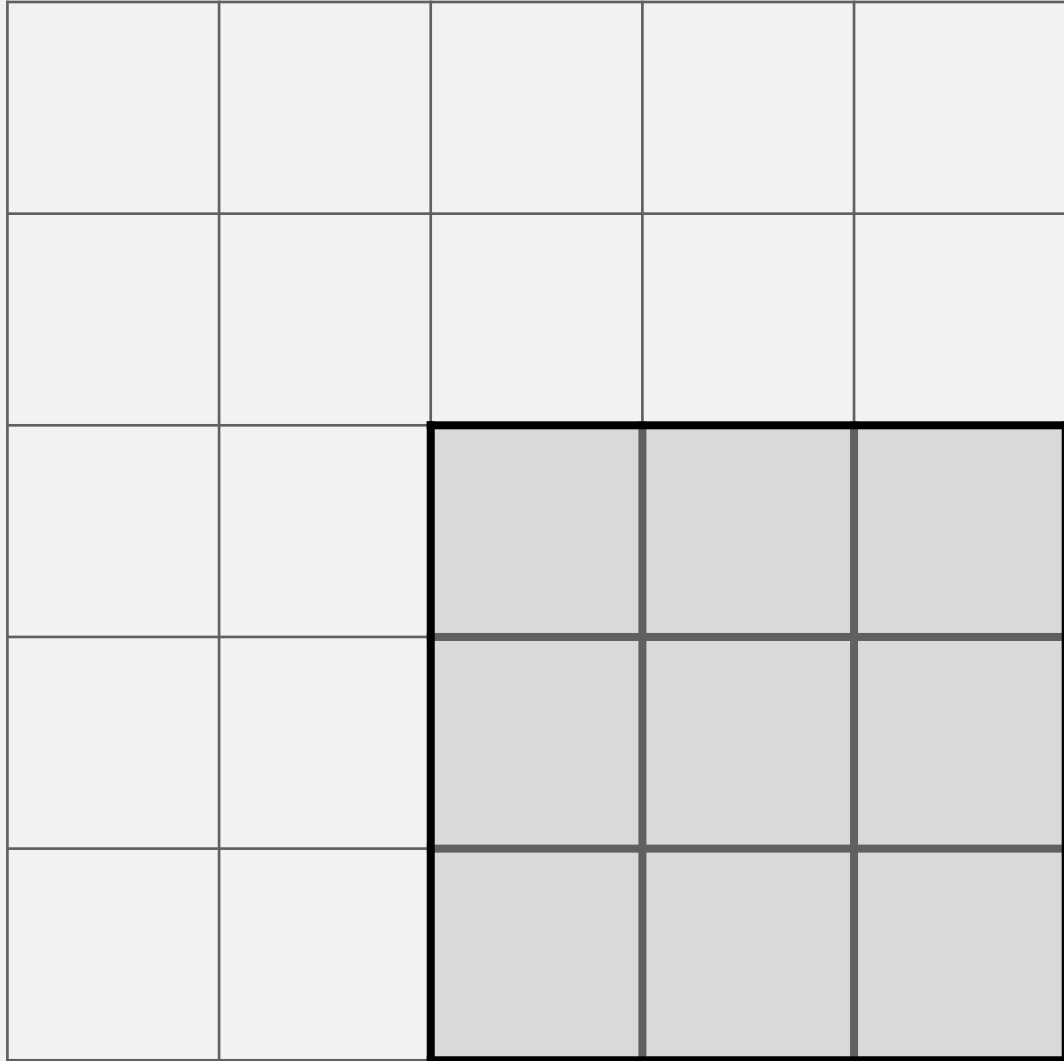
-1	-1	-1
1	1	1
-1	-1	-1

-1	-1	-1
-1	1	1
-1	1	1

Konwolucja całego obrazu



Konwolucja całego obrazu



Dodatkowe aspekty

- Obrazy wielowymiarowe np. 3 kolory
- Padding
- Stride

CNN

```
from keras.models import Sequential
from keras.layers import Dense, MaxPooling2D, Conv2D

model = Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(150, 150, 3)))      #32 numer of kernels, (3,3) kernel size
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())      # flatten 2D outputs to 1D for Dense
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

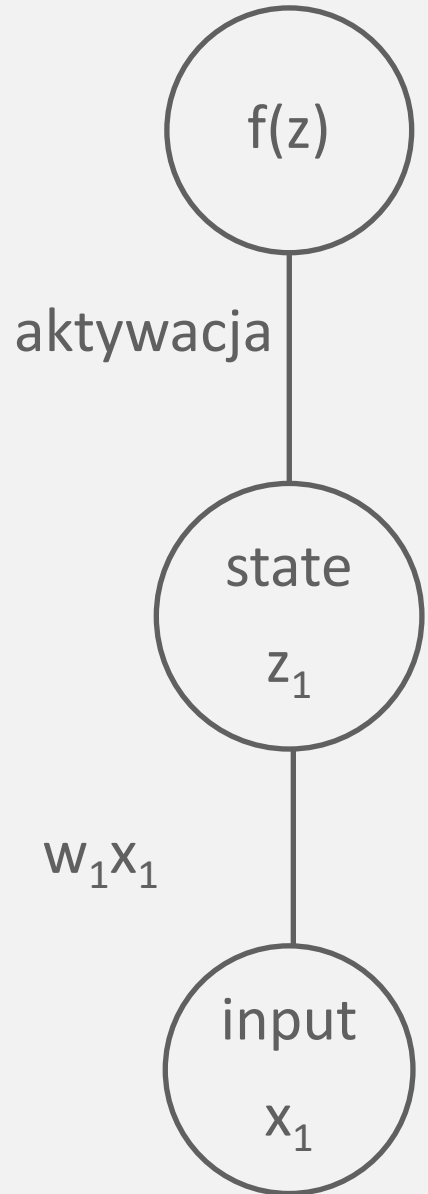
model.compile(loss='binary_crossentropy', optimizer='Adam',
metrics=['acc'])
```


RNN

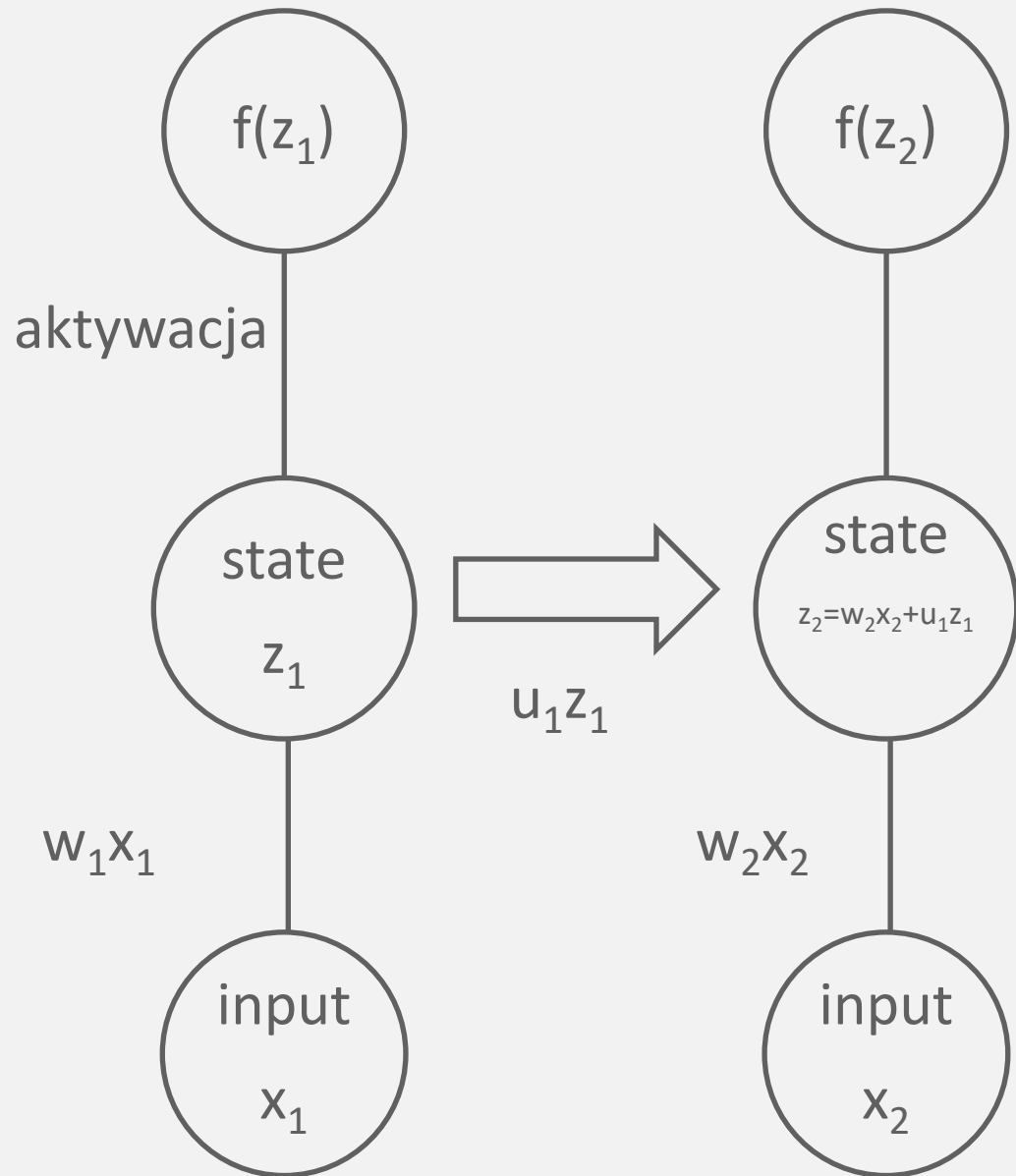
recurrent neural network

- Analiza szeregów czasowych
- Analiza tekstów → NIE NLP
- Uwzględnienie kontekstu każdej obserwacji = feature
 - Wykorzystanie informacji o poprzednich wartościach
- Predykcja pogody
 - Uwzględnienie poprzednich pomiarów temp., wilgotności, itp
- Klasyfikacja tekstów → pozytywne / krytyczne
 - Uwzględnienie kontekstu zdania, a nie zbioru wyrazów

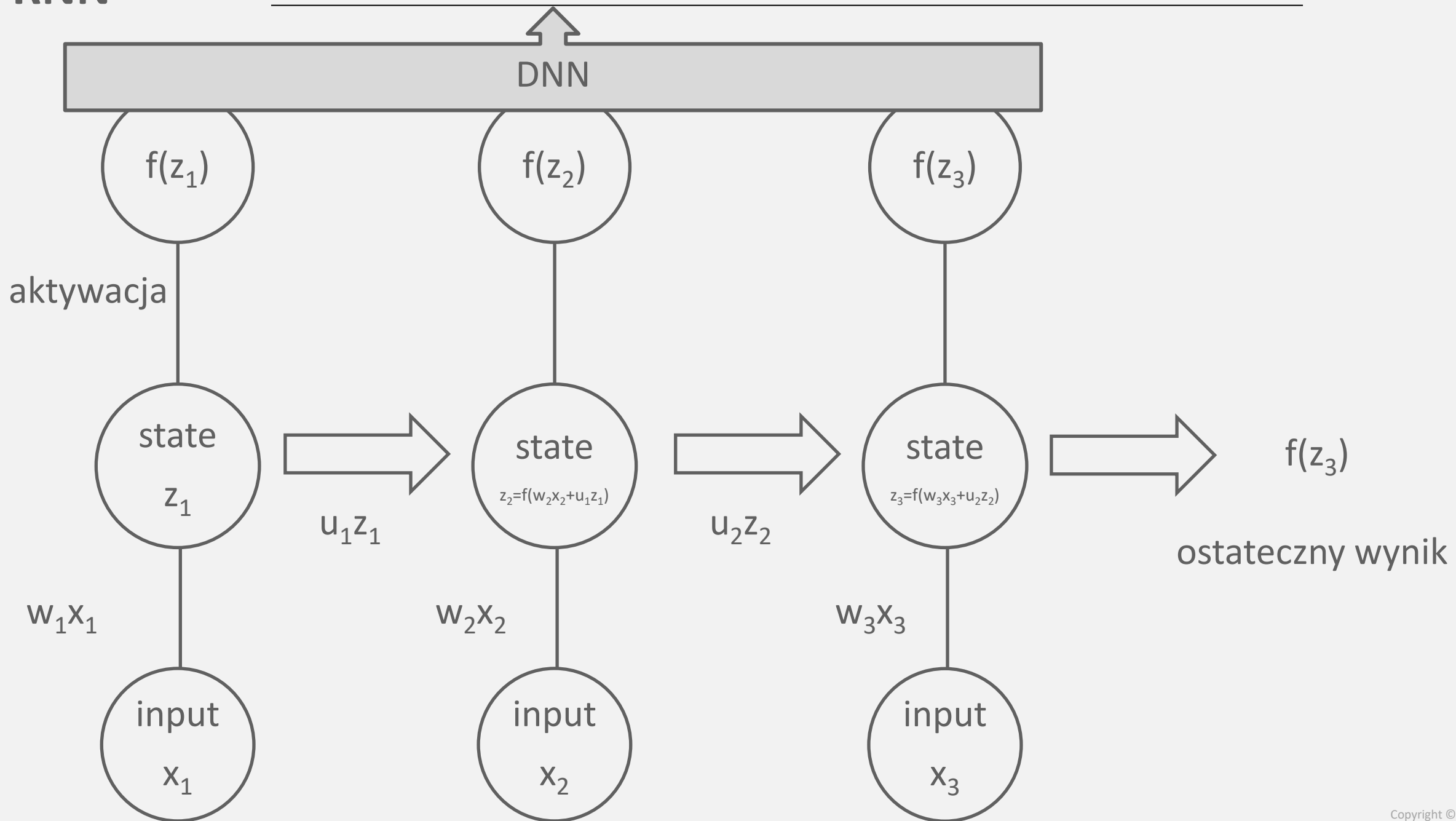
RNN



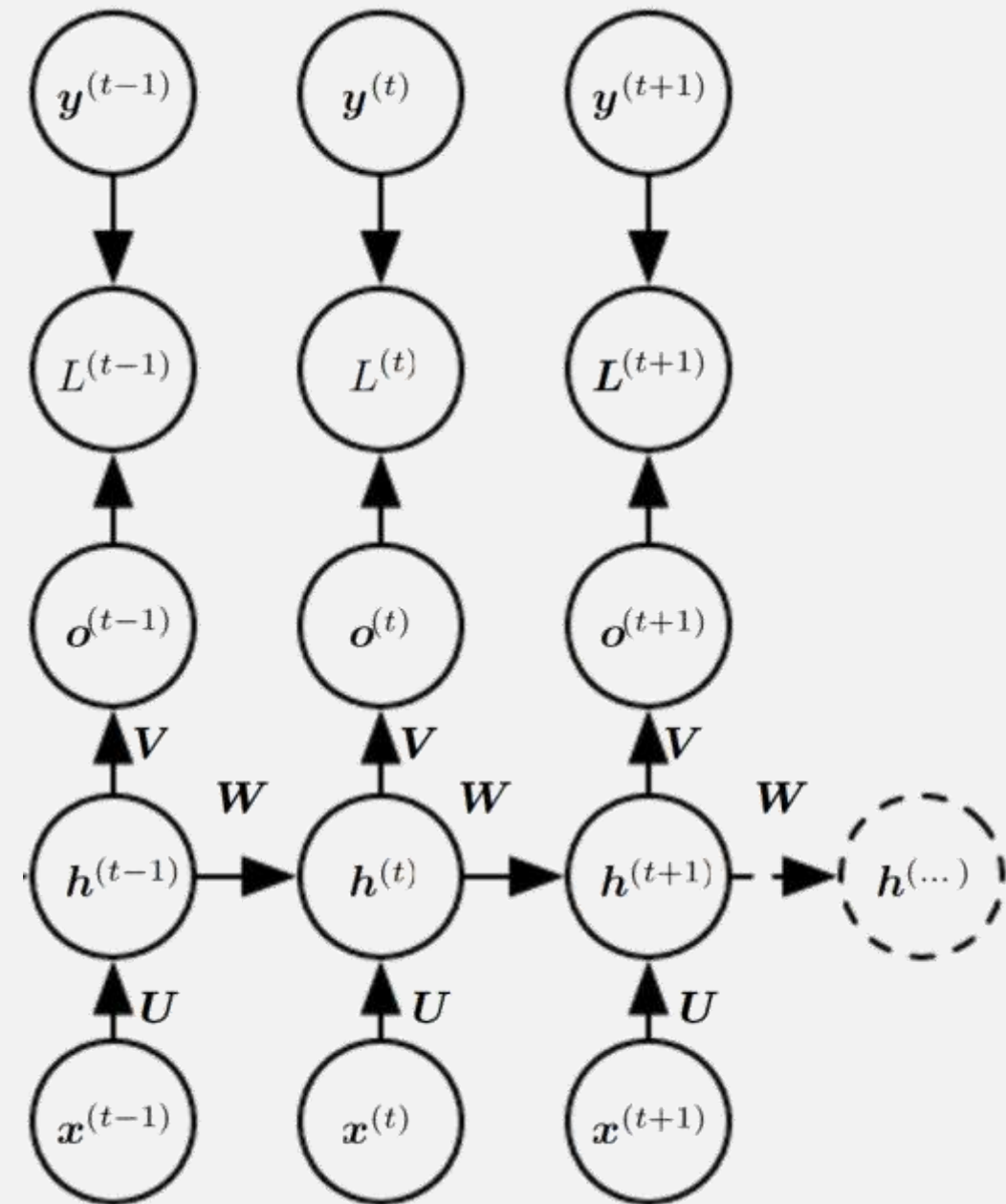
RNN



RNN

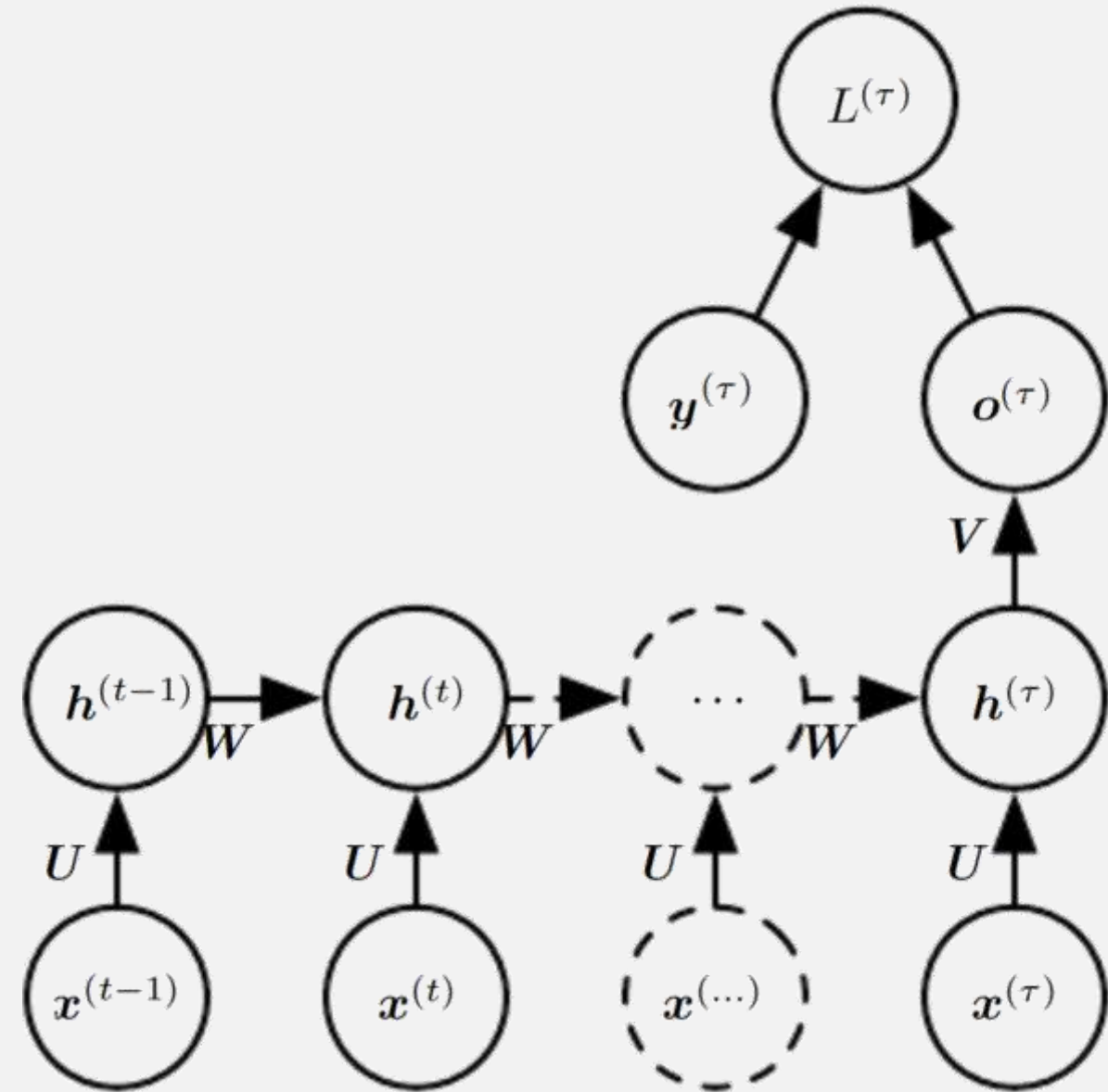


RNN



Uogólniona architektura RNN przykład 1

- y wartość wynikowa
- L *loss – funkcja straty = różnica między y i \hat{y}*
- $o = \hat{y}$ output – wartość f aktywacji
- h hidden – warstwa ukryta
- x input
- t indeks sekwencji, słowo, czas



RNN

```
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense

model = Sequential()
model.add(SimpleRNN(32))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='Adam', loss='binary_crossentropy',
metrics=['acc'])

model.fit(input_train, y_train, epochs=10, batch_size=128,
validation_split=0.2)
```

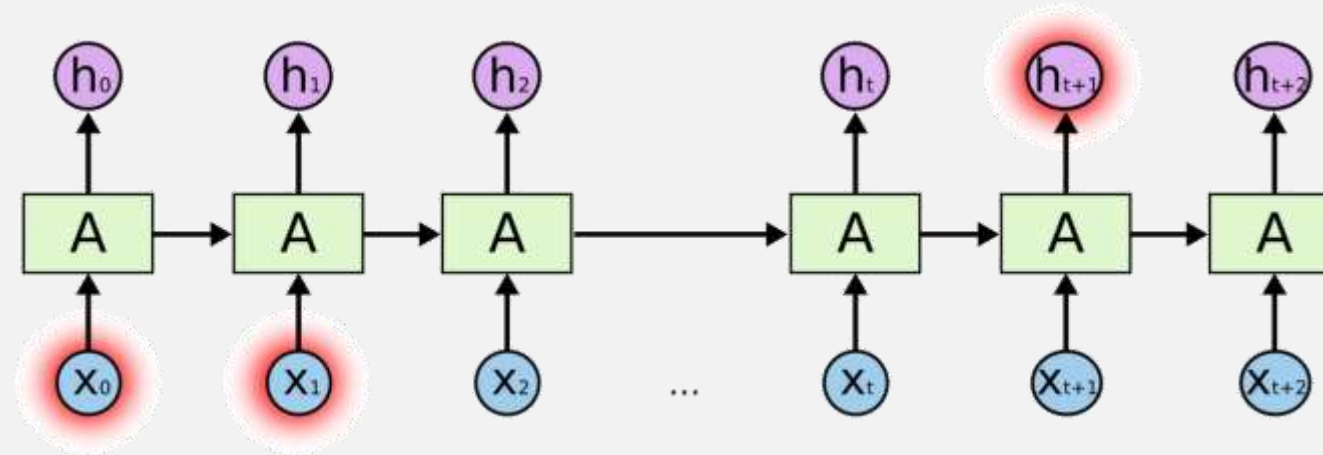

LSTM

long short term memory

neural networks

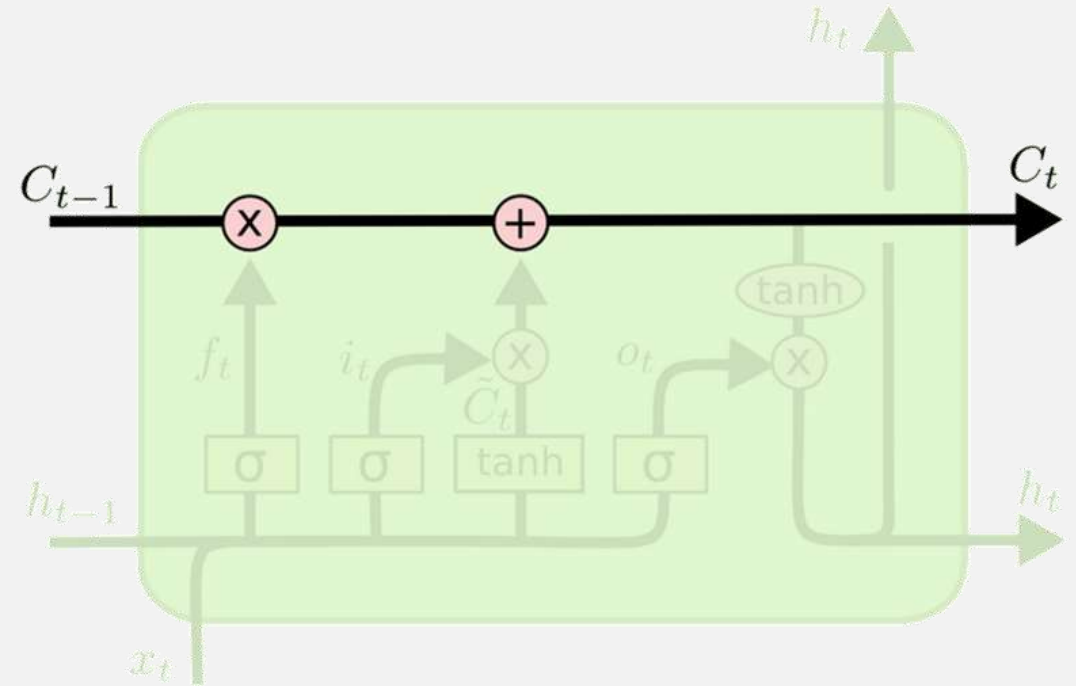
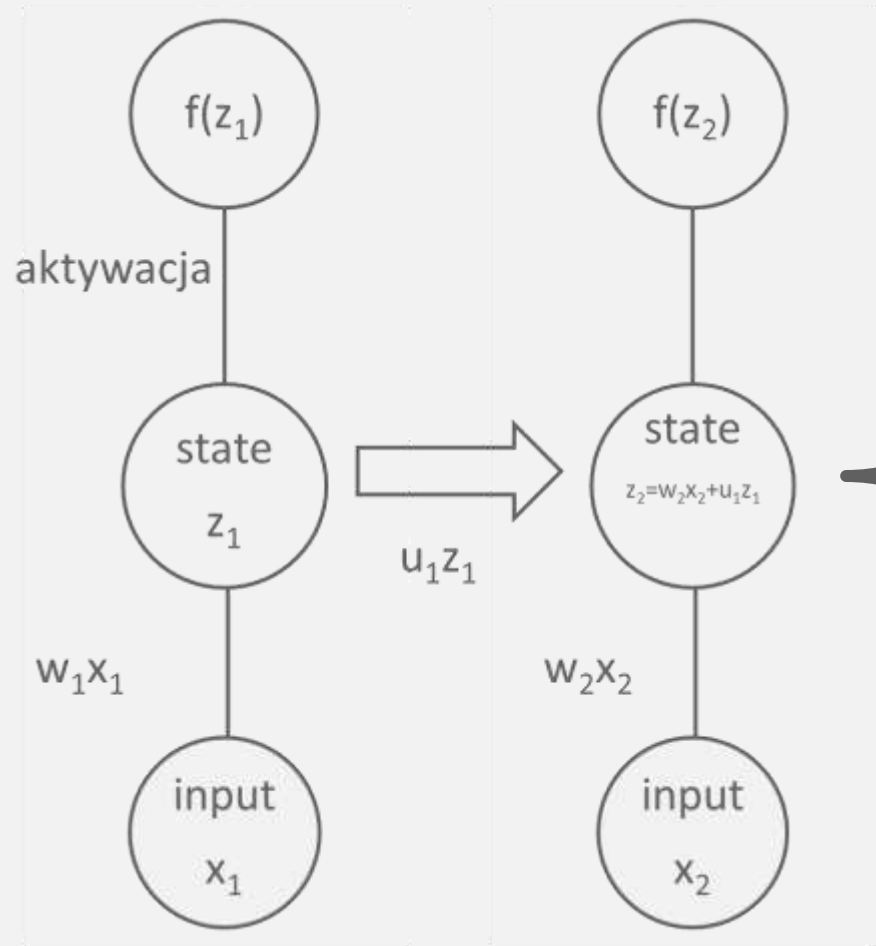
LSTM

- RNN problem \rightarrow zanik informacji o poprzednich wartościach oddalonych w czasie



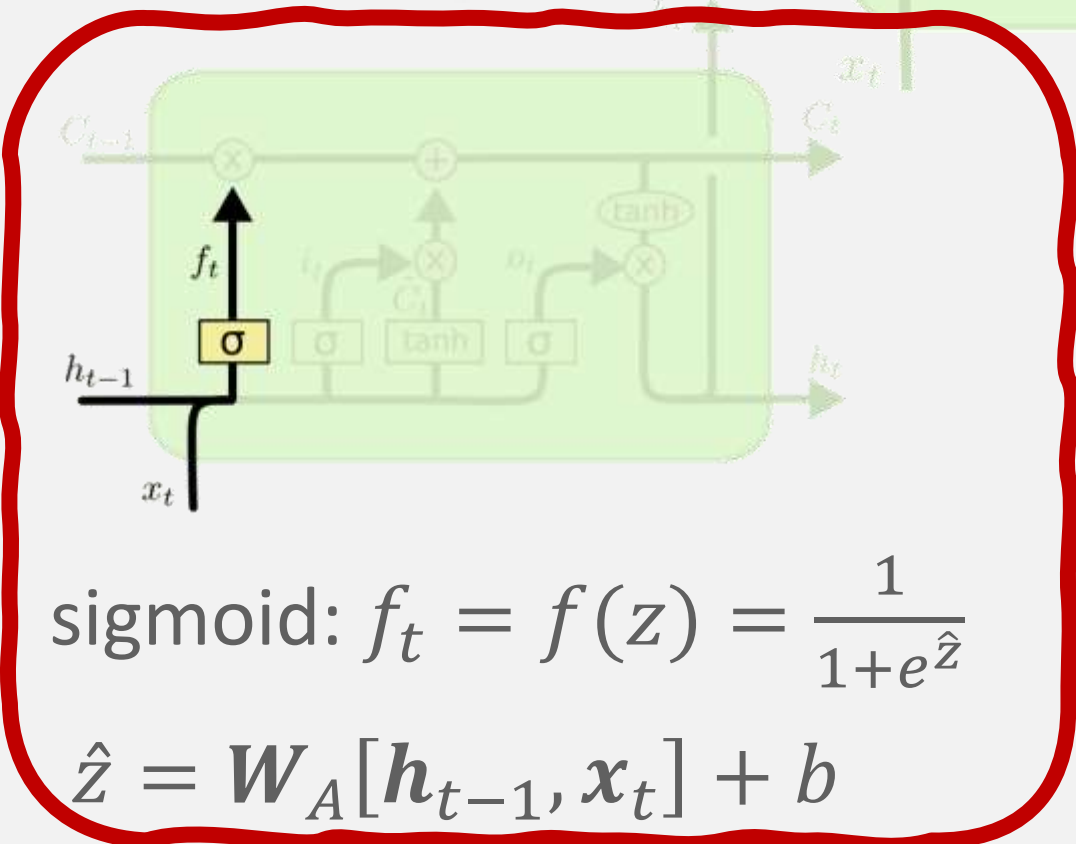
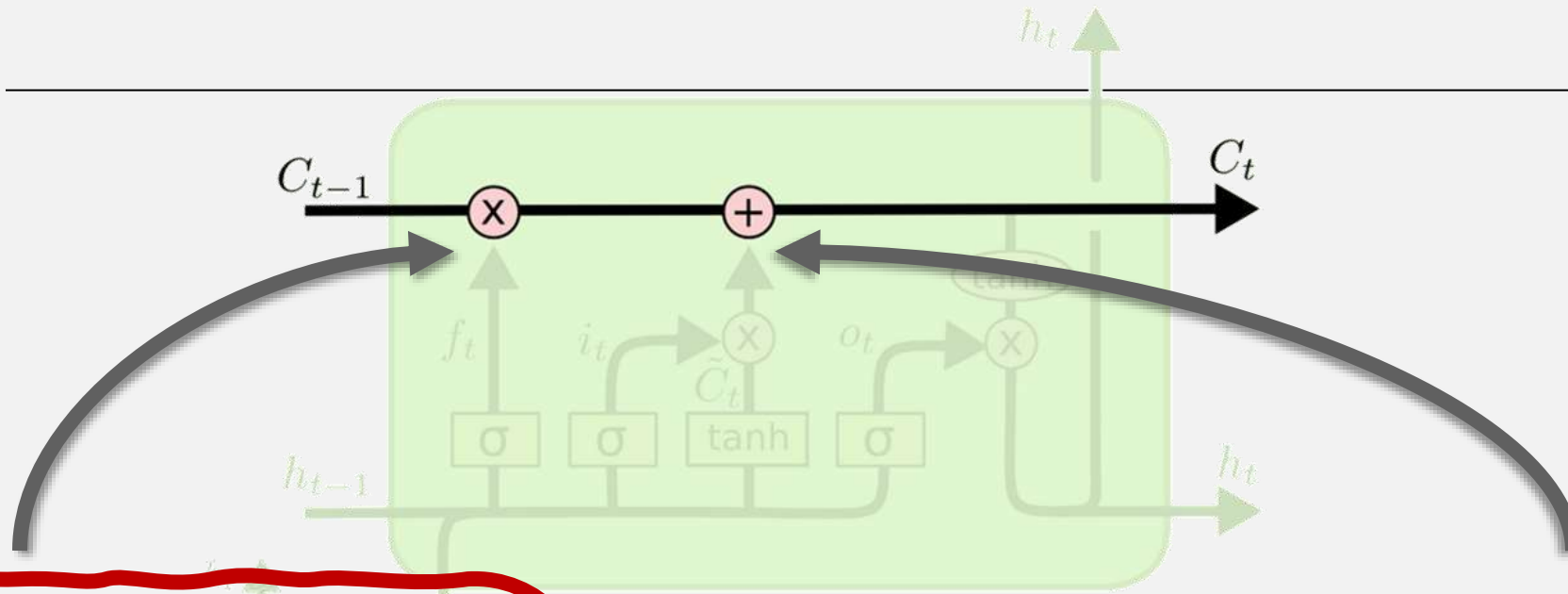
- Modyfikacja RNN \rightarrow LSTM \rightarrow ważenie poprzedniej $h_{(t-1)}$ i aktualnej x_t informacji

LSTM



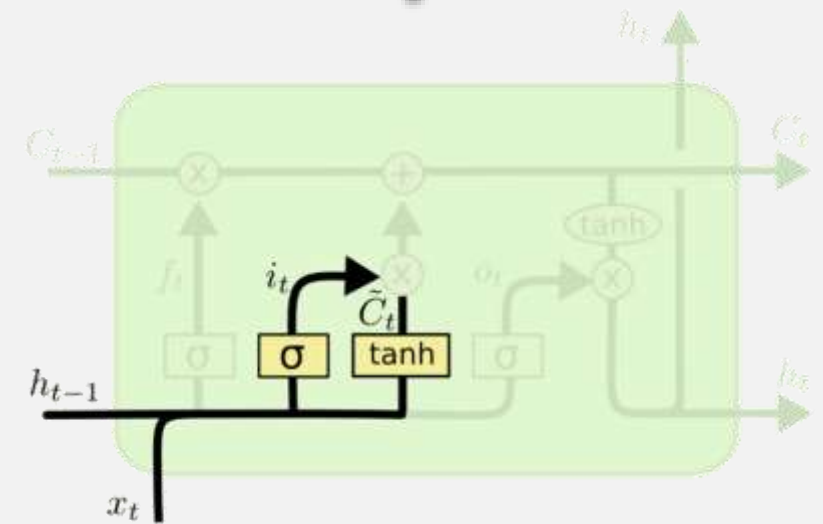
Stan $C_t \rightarrow$ ważona informacja z poprzedniego stanu C_{t-1} i aktualnej wartości $x_t \rightarrow$ wagi = wartości funkcji aktywacji

LSTM

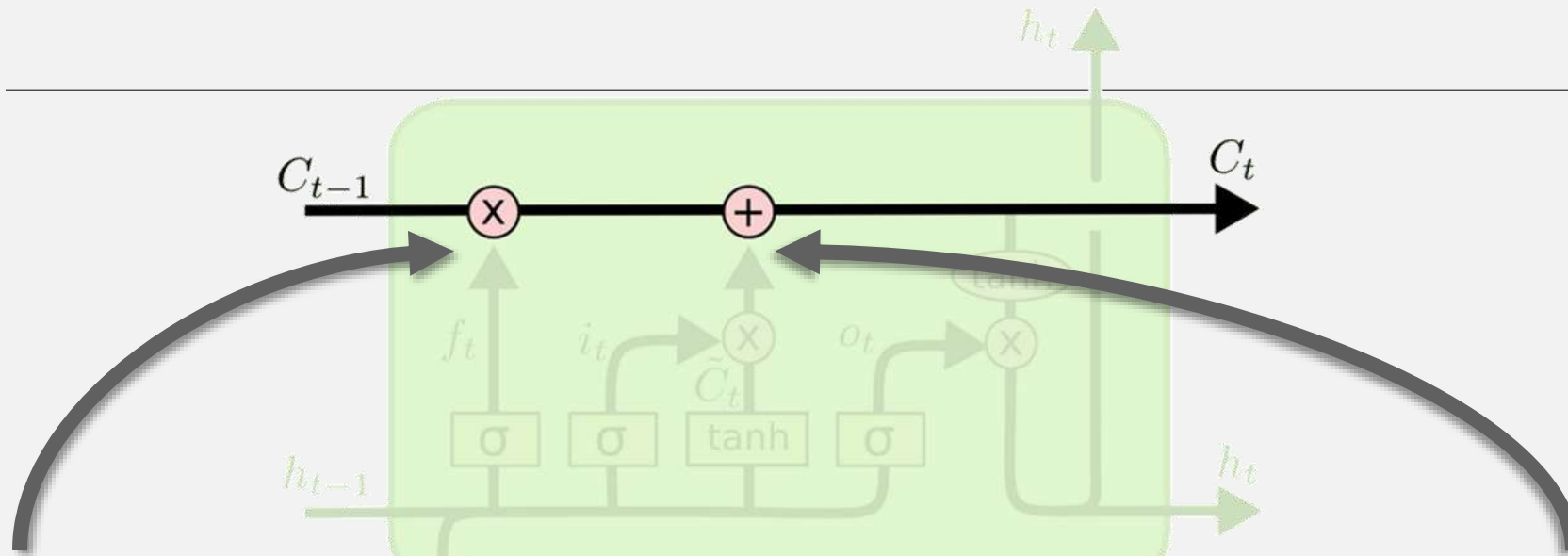


sigmoid: $f_t = f(z) = \frac{1}{1+e^{\hat{z}}}$

$\hat{z} = \mathbf{W}_A[\mathbf{h}_{t-1}, \mathbf{x}_t] + b$



LSTM

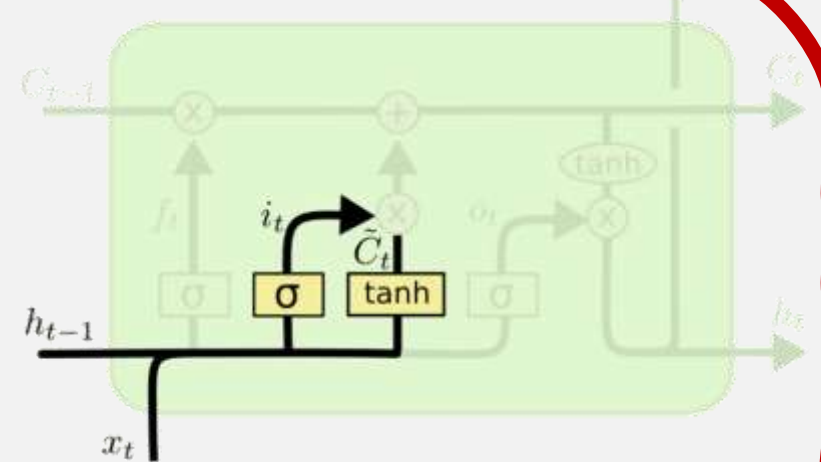
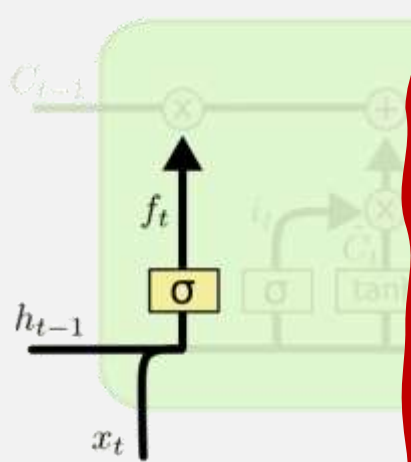


sigmoid: $i_t = f(z) = \frac{1}{1+e^{-\hat{z}}}$

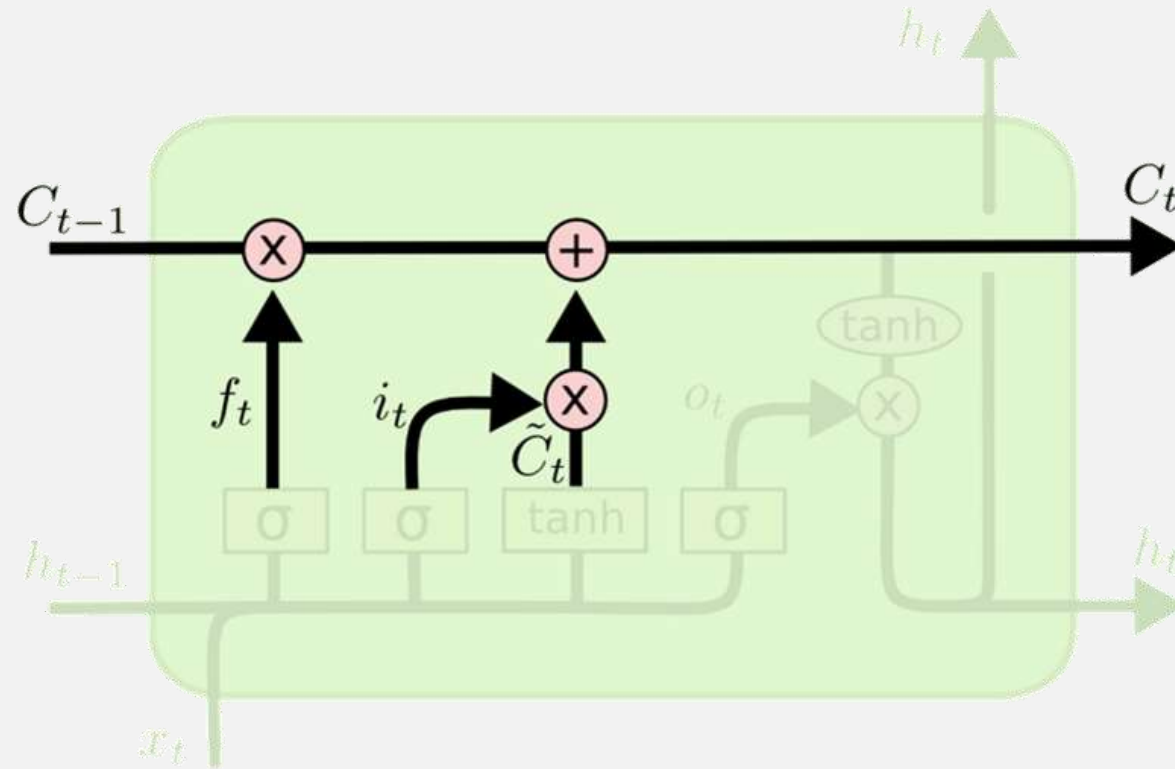
$$\hat{z} = \mathbf{W}_B[\mathbf{h}_{t-1}, \mathbf{x}_t] + b$$

tanh: $j_t = f(z) = f(z) = \frac{e^{\hat{z}} - e^{-\hat{z}}}{e^{\hat{z}} + e^{-\hat{z}}}$

$$\hat{z} = \mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + b$$

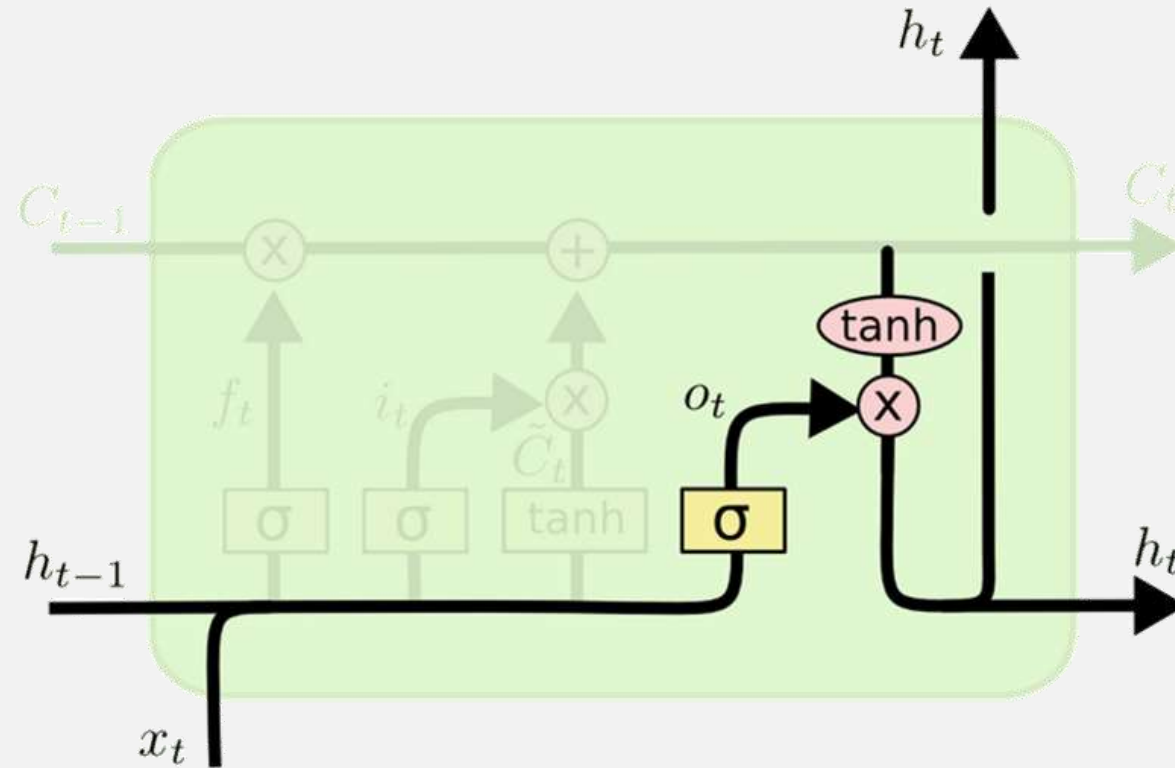


LSTM



Final state w czasie t: $C_t = f_t(\mathbf{W}_A) C_{t-1} + i_t (\mathbf{W}_B) j_t (\mathbf{W}_C)$

LSTM



$$\text{sigmoid: } k_t = f(z) = \frac{1}{1+e^{\hat{z}}} \quad \hat{z} = \mathbf{W}_D [\mathbf{h}_{t-1}, \mathbf{x}_t] + b$$

Final **output** w czasie t:

$$h_t = k_t(\mathbf{W}_D) \tanh(\mathbf{W}_E, C_t) = k_t(\mathbf{W}_D) \frac{e^{\hat{C}_t} - e^{-\hat{C}_t}}{e^{\hat{C}_t} + e^{-\hat{C}_t}}$$

LSTM

```
from keras.models import Sequential
from keras.layers import LSTM, Dense

model = Sequential()
model.add(LSTM(32))
model.add(Dense(1, activation='sigmoid'))

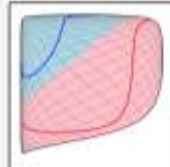
model.compile(optimizer='Adam', loss='binary_crossentropy',
metrics=['acc'])

model.fit(input_train, y_train, epochs=10, batch_size=128,
validation_split=0.2)
```

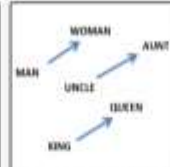

1. DNN
2. CNN
3. RNN
4. LSTM

1. Co oznacza skrót CNN?
2. Jakie dodatkowe źródło informacji wykorzystuje CNN?
3. Na czym polega konwolucja?
4. Co oznacza skrót RNN
5. Jakie dodatkowe źródło informacji wykorzystuje RNN?

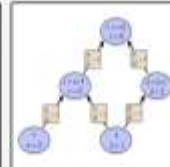
Neural Networks (General)



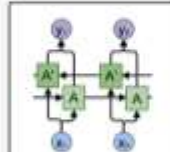
Neural Networks, Manifolds, and Topology



Deep Learning, NLP, and Representations

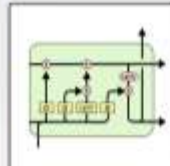


Calculus on Computational Graphs: Backpropagation

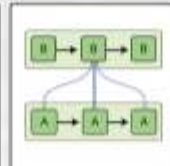


Neural Networks, Types, and Functional Programming

Recurrent Neural Networks



Understanding LSTM Networks

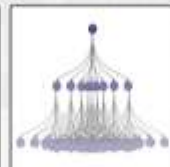


Attention and Augmented Recurrent Neural Networks On Data

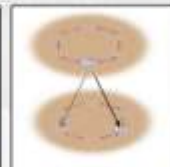
Convolutional Neural Networks



Conv Nets: A Multiple Perspective



Understanding Convolution



Groups & Group Convolution



Deconvolution and Checkerboard Artifacts On Data