Sieci Neuronowe – laboratorium  
  
Ćwiczenie nr. 4  
  
Kajetan Pynka, 254495

Spis treści

[Wstęp teoretyczny 3](#_Toc121590615)

[Porównanie komórek RNN i LSTM 4](#_Toc121590616)

[Rozmiar wymiaru wyjściowego warstwy embeddingu 4](#_Toc121590617)

[Rozmiar wymiaru wyjściowego warstwy embeddingu – 30 4](#_Toc121590618)

[Rozmiar wymiaru wyjściowego warstwy embeddingu – 50 7](#_Toc121590619)

[Rozmiar wymiaru wyjściowego warstwy embeddingu – 80 8](#_Toc121590620)

[Rozmiar wymiaru wyjściowego warstwy embeddingu – 120 9](#_Toc121590621)

[Rozmiar wymiaru wyjściowego warstwy embeddingu – 150 10](#_Toc121590622)

[Liczba jednostek warstwy rekurencyjnej 11](#_Toc121590623)

[Liczba jednostek warstwy rekurencyjnej – 30 11](#_Toc121590624)

[Liczba jednostek warstwy rekurencyjnej – 50 13](#_Toc121590625)

[Liczba jednostek warstwy rekurencyjnej – 80 14](#_Toc121590626)

[Liczba jednostek warstwy rekurencyjnej – 120 15](#_Toc121590627)

[Liczba jednostek warstwy rekurencyjnej – 150 16](#_Toc121590628)

[Funkcja aktywacji 17](#_Toc121590629)

[Funkcja aktywacji – tangens hiperboliczny 17](#_Toc121590630)

[Funkcja aktywacji – sigmoid 19](#_Toc121590631)

[Funkcja aktywacji – ReLU 20](#_Toc121590632)

[Efektywność paddingu 21](#_Toc121590633)

[Rozmiar paddingu / długość maski 21](#_Toc121590634)

[Rozmiar paddingu / długość maski – 50 21](#_Toc121590635)

[Rozmiar paddingu / długość maski – 80 23](#_Toc121590636)

[Rozmiar paddingu / długość maski – 100 24](#_Toc121590637)

[Rozmiar paddingu / długość maski – 130 25](#_Toc121590638)

[Rozmiar paddingu / długość maski - 150 26](#_Toc121590639)

[Wnioski 27](#_Toc121590640)

# Wstęp teoretyczny

W ramach ćwiczenia zaimplementowałem modele sieci rekurencyjnej, raz wykorzystujące zwykłe komórki RNN, a w przypadku drugiego komórki LSTM (long short-term memory). Struktura sieci i cały proces nauki został zrealizowany przy użyciu języka Python 3.10 oraz biblioteki ‘keras’ będącej składową większej biblioteki ‘tensorflow’. Oba modele jako funkcję straty zakładały binarną entropię krzyżową oraz używały optymalizatora RMSprop. W ramach ćwiczeń wykonałem następujące eksperymenty: porównanie komórek RNN i LSTM (na podstawie wartości: rozmiaru embeddingu, liczby wymiarów oraz użytej funkcji aktywacji), ocena użyteczności i efeketywności paddingu (ze względu na rozmiar maski). Z danych wyciągnięte zostało najpopularniejsze 15000 słów (wdg wewnętrznej częstotliwości występowania danego słowa).

Dla każdego eksperymentu przyjęte zostały następujące wartości hiperparametrów (tam, gdzie dany parametr nie był oczywiście przedmiotem badania):

* Rozmiar paczki = 128
* Liczba epok = 10
* Funkcja aktywacji = ReLU
* Rozmiar wymiaru wejściowego warstwy embeddingu = 15000
* Rozmiar wymiaru wyjściowego warstwy embeddingu = 64
* Rozmiar paddingu / długość maski = 100
* Liczba jednostek warstwy rekurencyjnej = 10

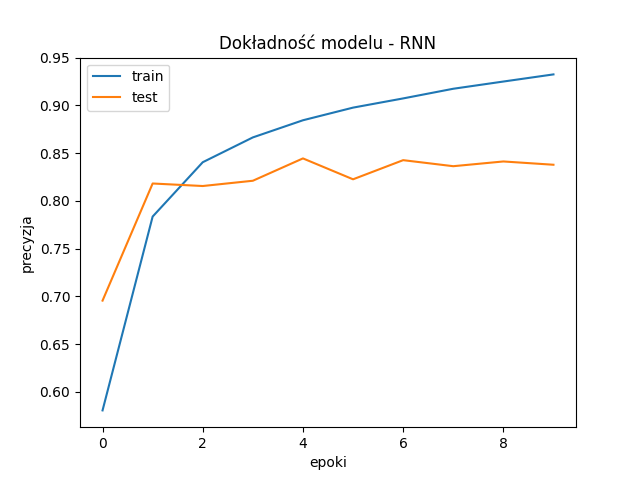
Dla każdego eksperymentu sporządzone zostały wykresy:

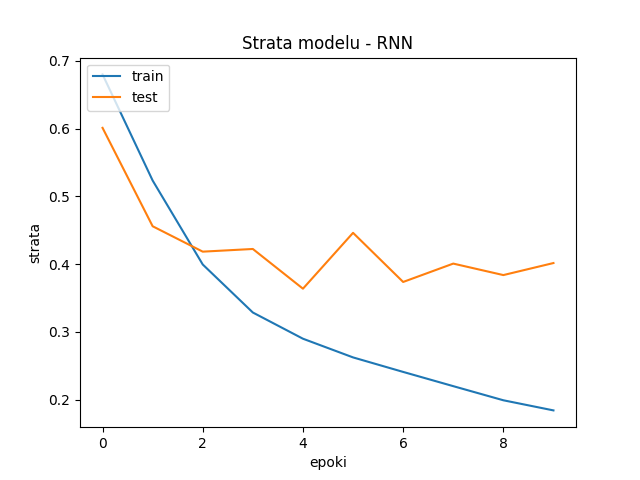
* precyzji/skuteczności modelu (w skali 0-1) na zbiorze treningowym oraz testowym w zależności od liczby epok dla komórek RNN i LSTM
* wartość funkcji straty (binarnej entropii krzyżowej) na zbiorze treningowym oraz testowym w zależności od liczby epok dla komórek RNN i LSTM
* tablicy pomyłek – zawsze po lewej stronie tablica pomyłek dla komórek RNN a po lewej stronie dla komórek LSTM

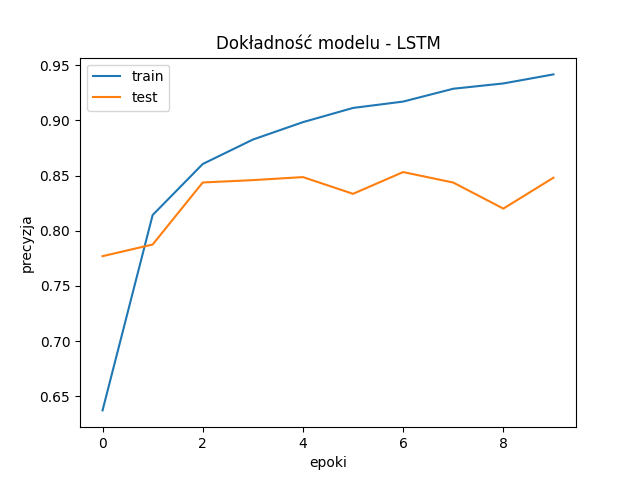
# Porównanie komórek RNN i LSTM

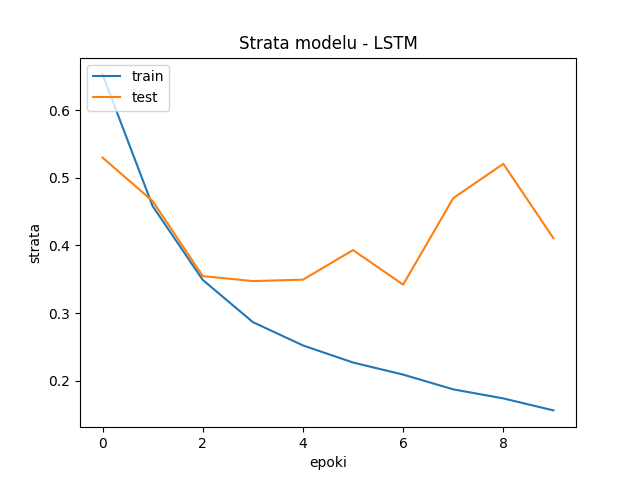
## Rozmiar wymiaru wyjściowego warstwy embeddingu

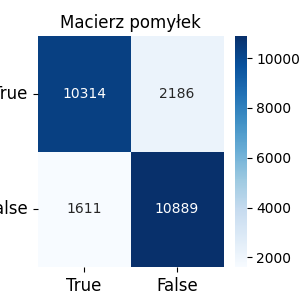
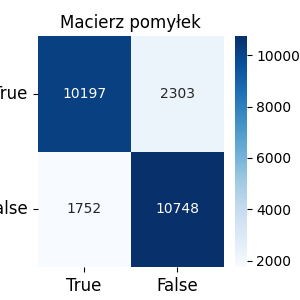
### Rozmiar wymiaru wyjściowego warstwy embeddingu – 30



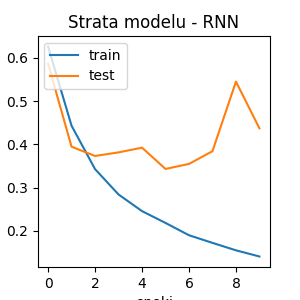
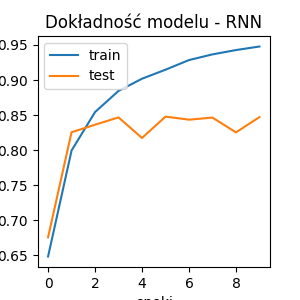


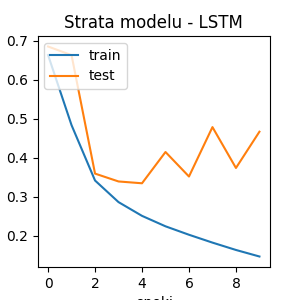
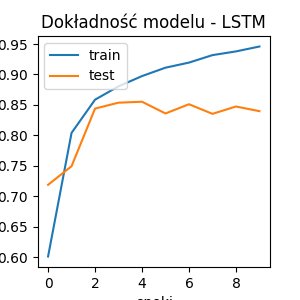


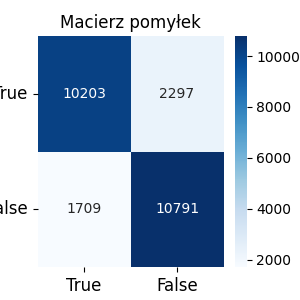
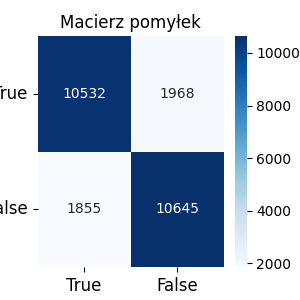




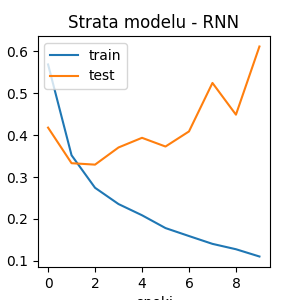
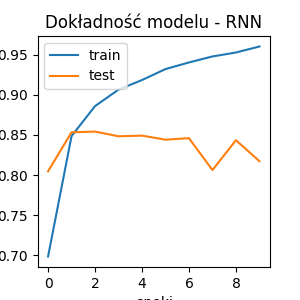
### Rozmiar wymiaru wyjściowego warstwy embeddingu – 50

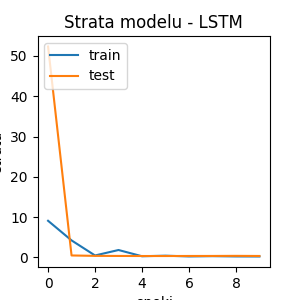
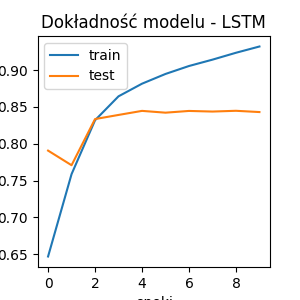


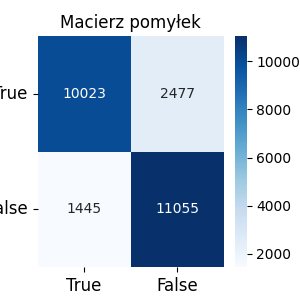
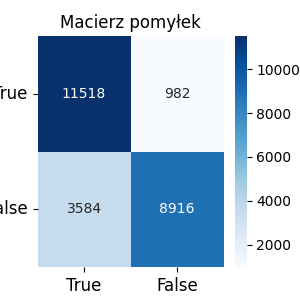




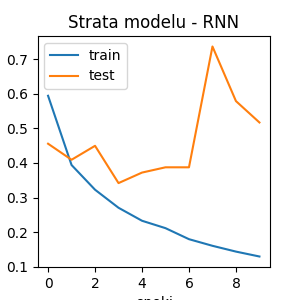
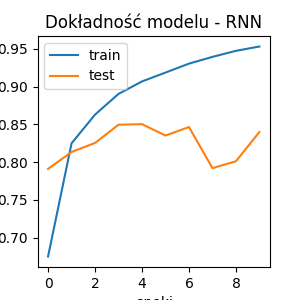
### Rozmiar wymiaru wyjściowego warstwy embeddingu – 80

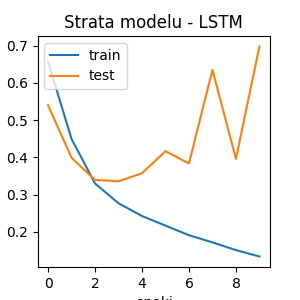
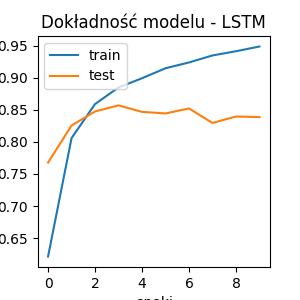


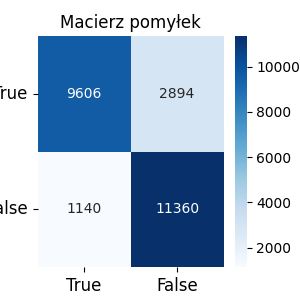
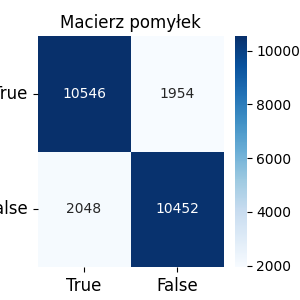




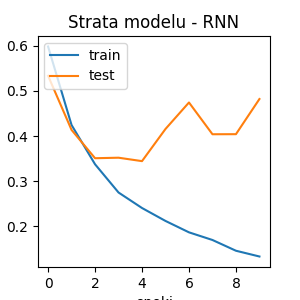
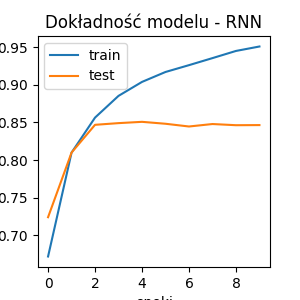
### Rozmiar wymiaru wyjściowego warstwy embeddingu – 120

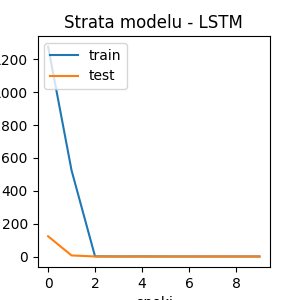
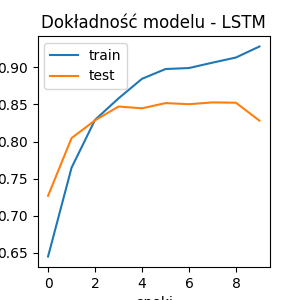


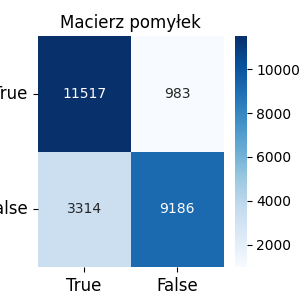
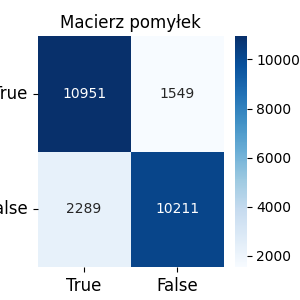




### Rozmiar wymiaru wyjściowego warstwy embeddingu – 150

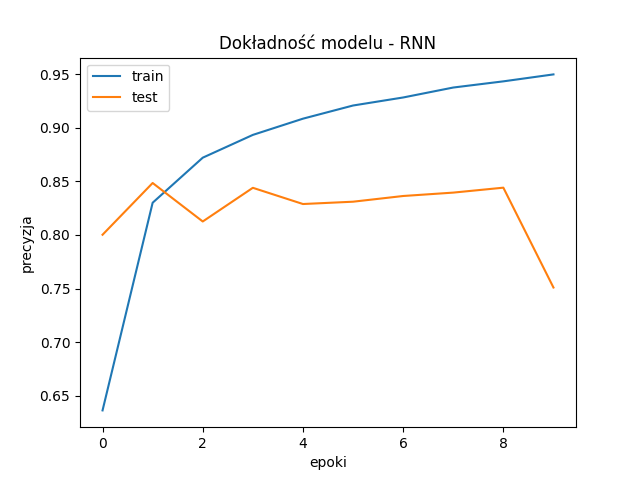


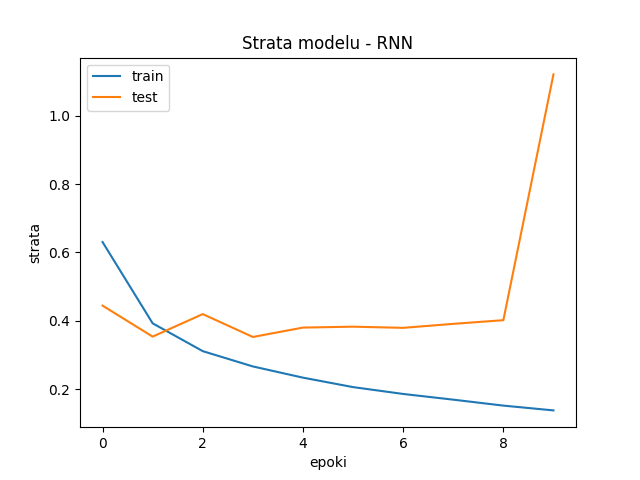


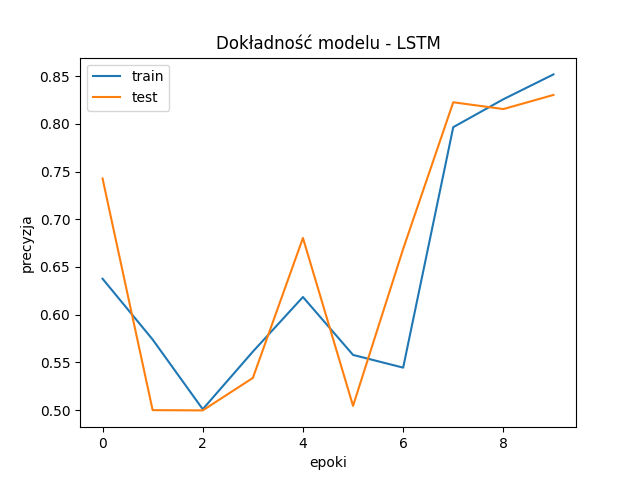


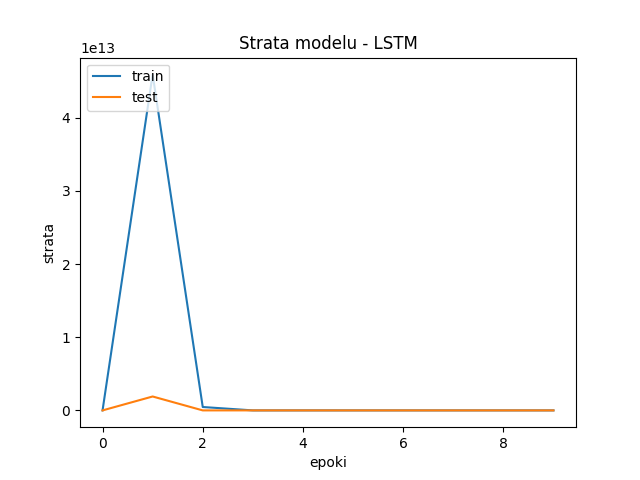
## Liczba jednostek warstwy rekurencyjnej

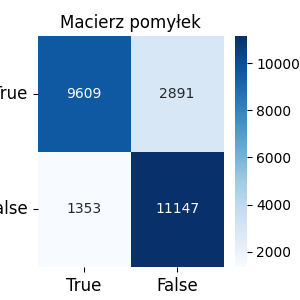
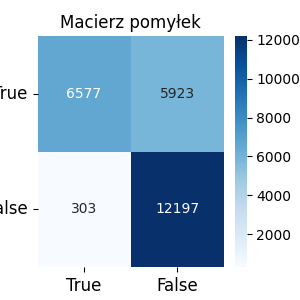
### Liczba jednostek warstwy rekurencyjnej – 30



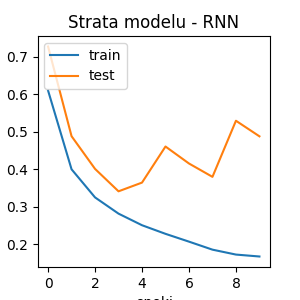
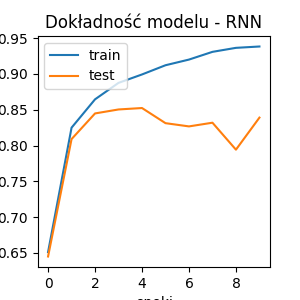


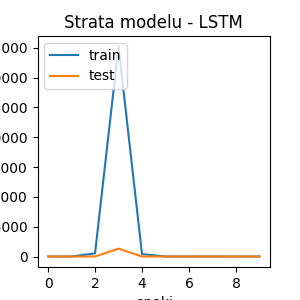
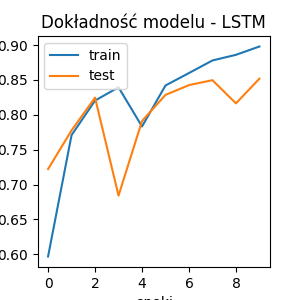


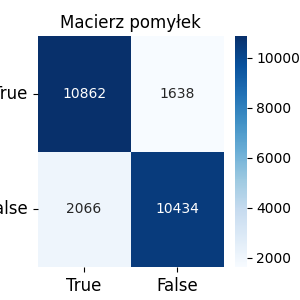
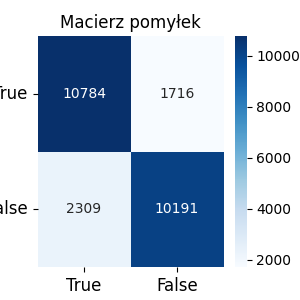




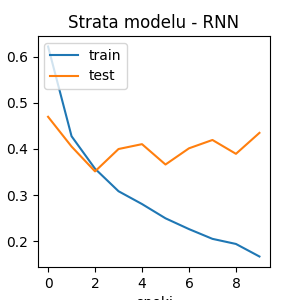
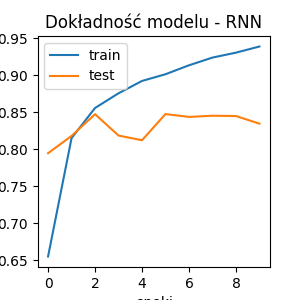
### Liczba jednostek warstwy rekurencyjnej – 50

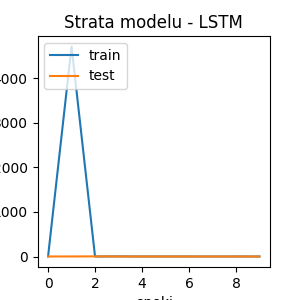
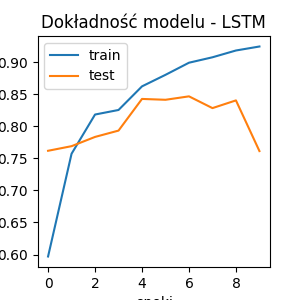


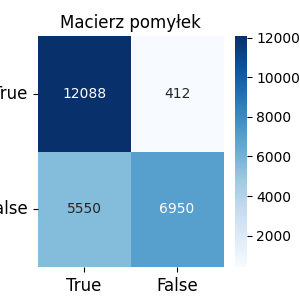
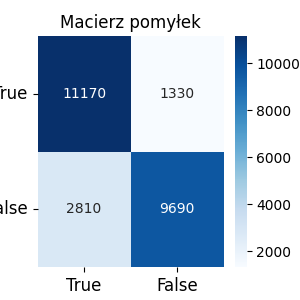




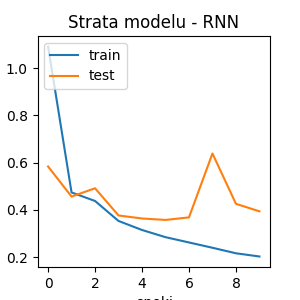
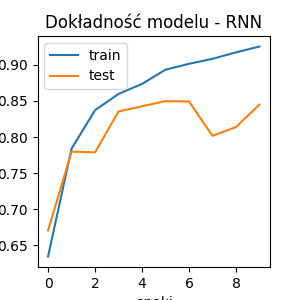
### Liczba jednostek warstwy rekurencyjnej – 80

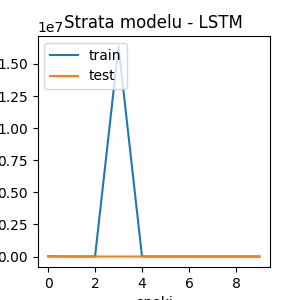
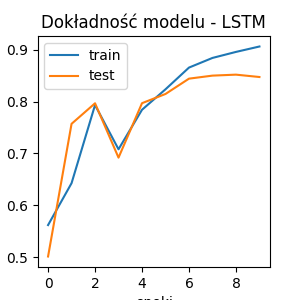


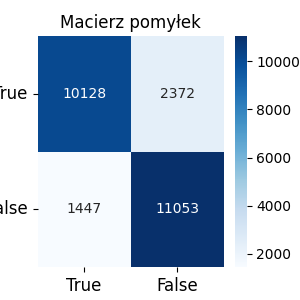
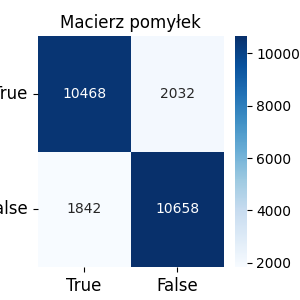




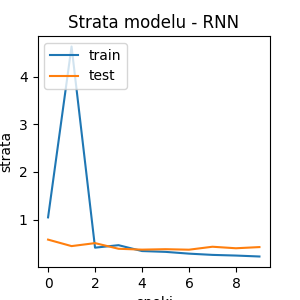
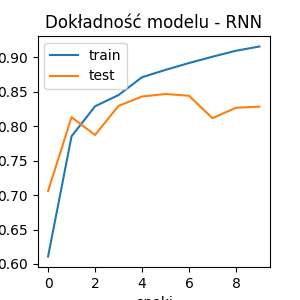
### Liczba jednostek warstwy rekurencyjnej – 120

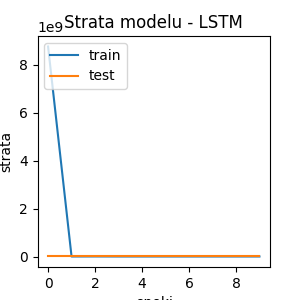
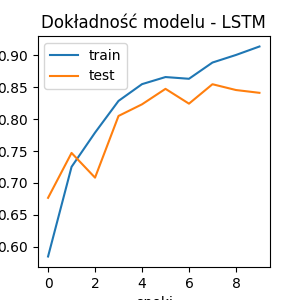


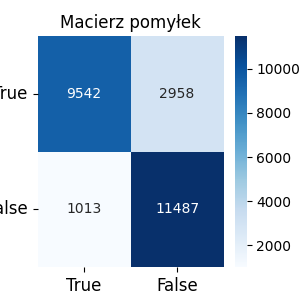
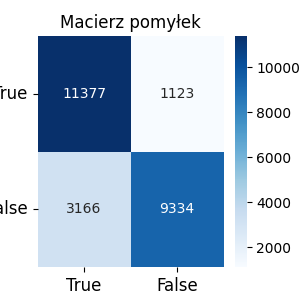




### Liczba jednostek warstwy rekurencyjnej – 150

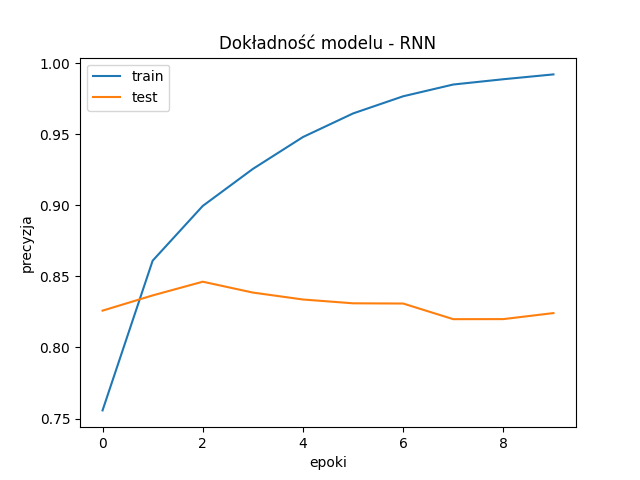


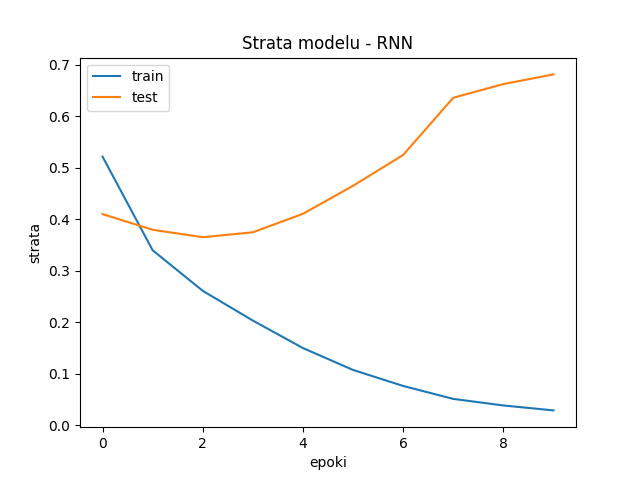


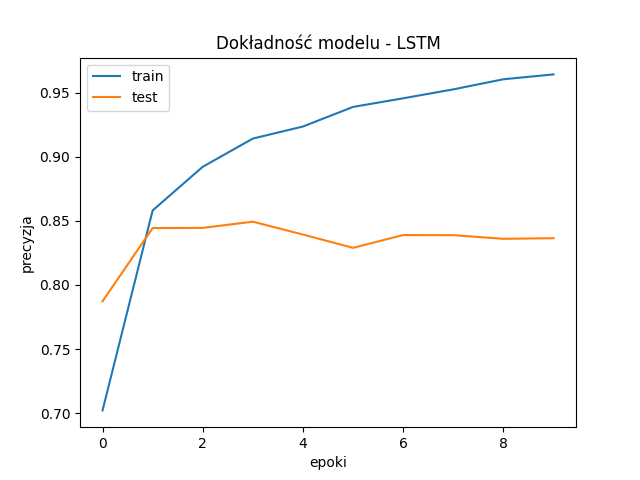


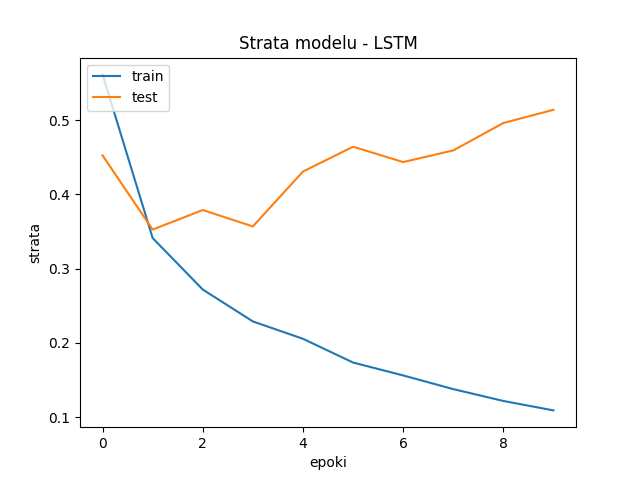
## Funkcja aktywacji

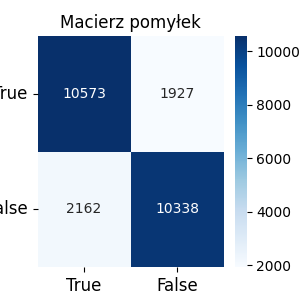
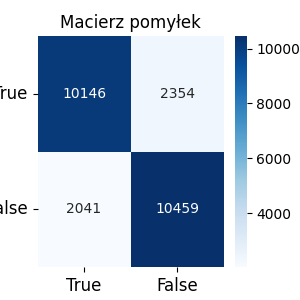
### Funkcja aktywacji – tangens hiperboliczny



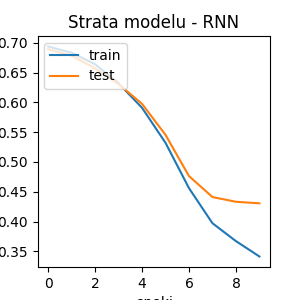
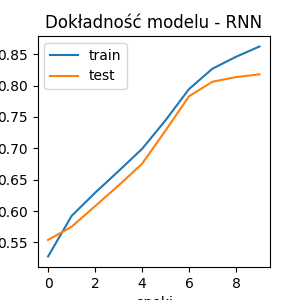


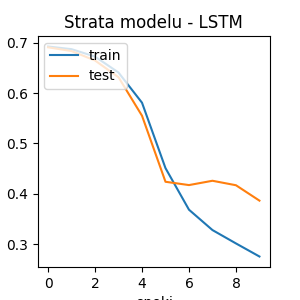
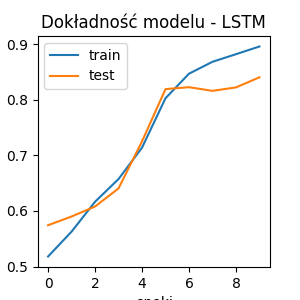


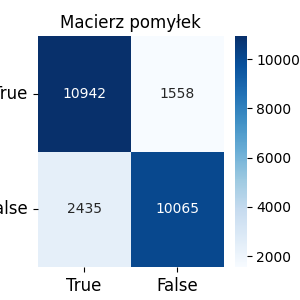
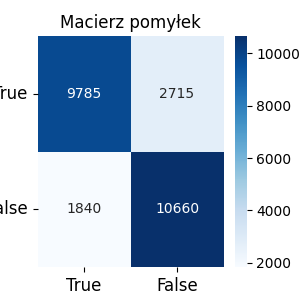




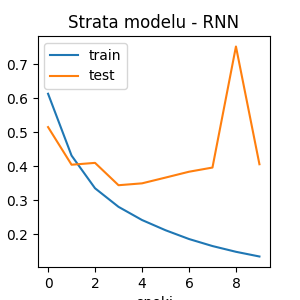
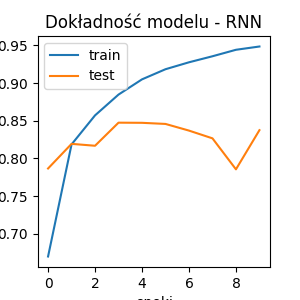
### Funkcja aktywacji – sigmoid

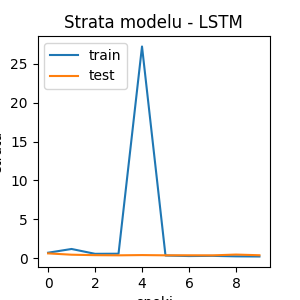
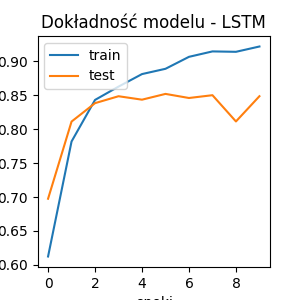


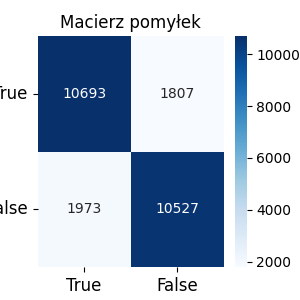
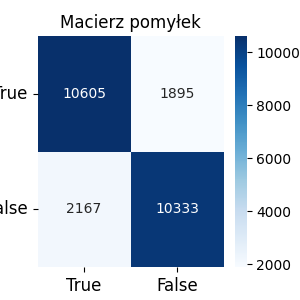




### Funkcja aktywacji – ReLU



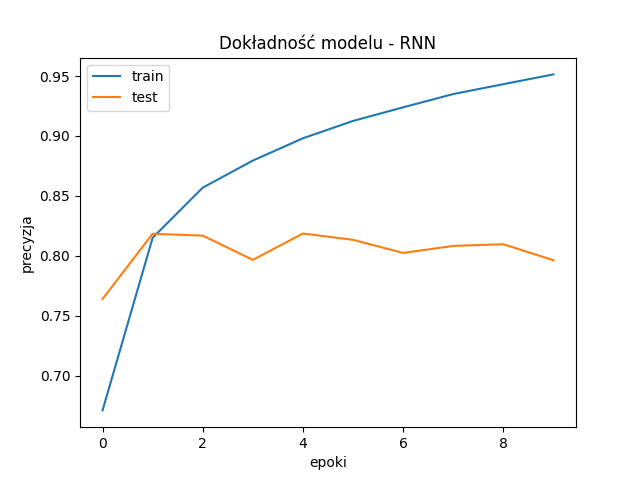
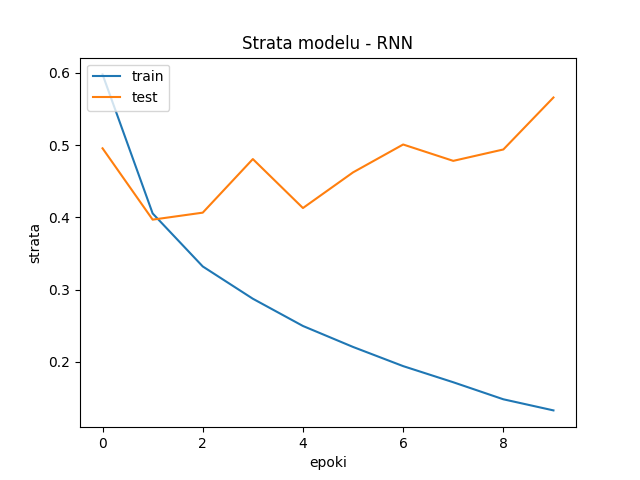


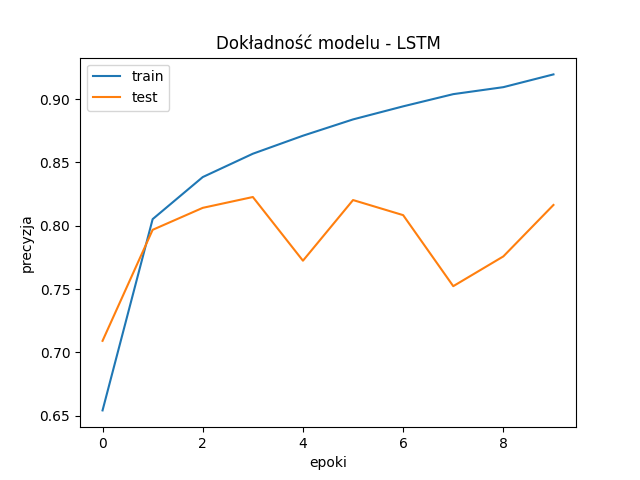


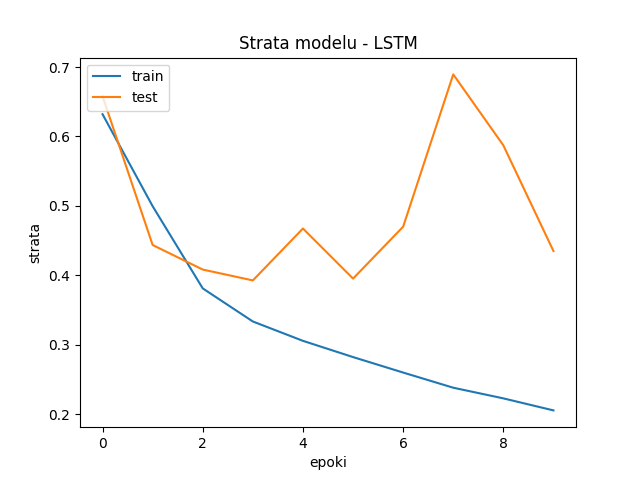
# Efektywność paddingu

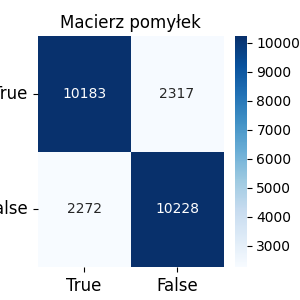
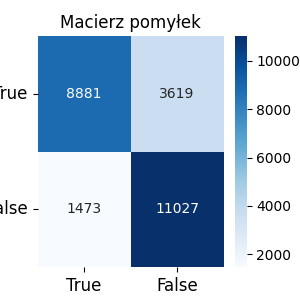
## Rozmiar paddingu / długość maski

### Rozmiar paddingu / długość maski – 50

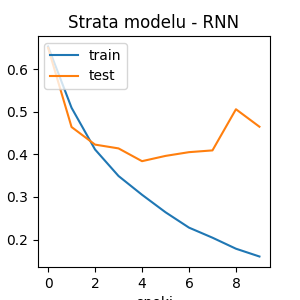
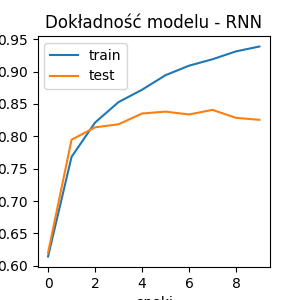


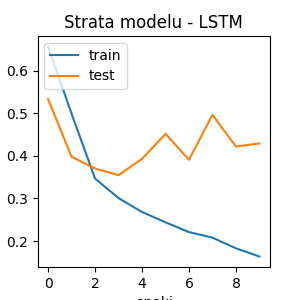
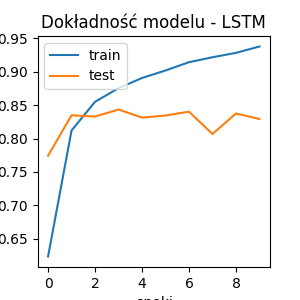


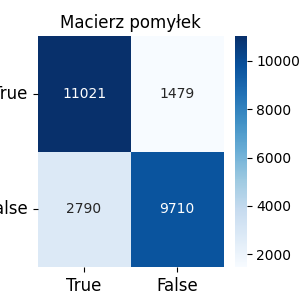
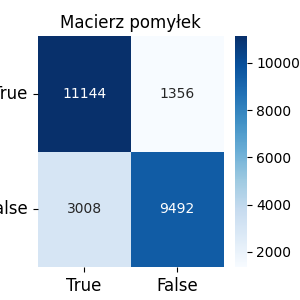




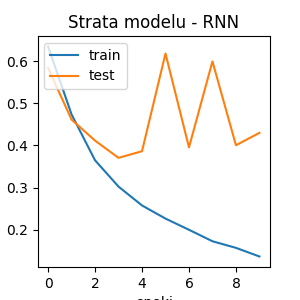
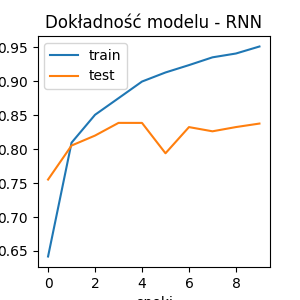
### Rozmiar paddingu / długość maski – 80

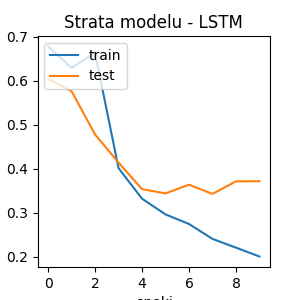
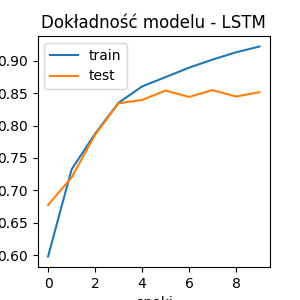


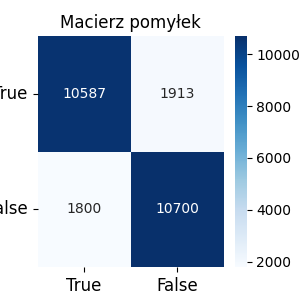
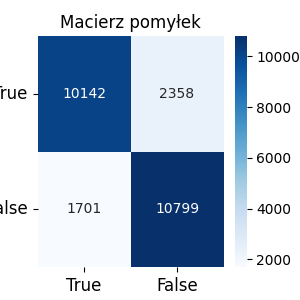




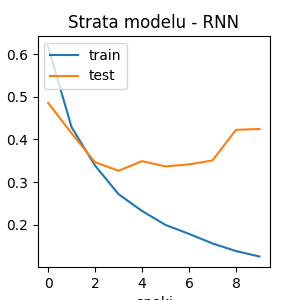
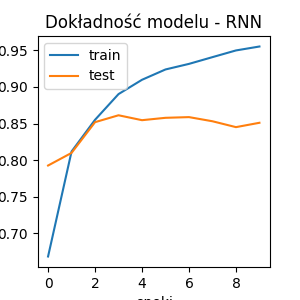
### Rozmiar paddingu / długość maski – 100

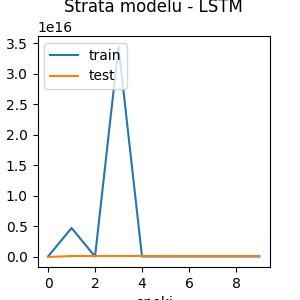
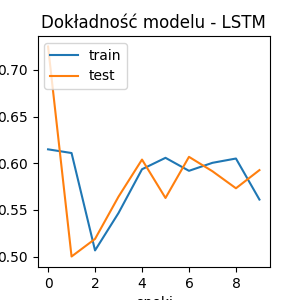


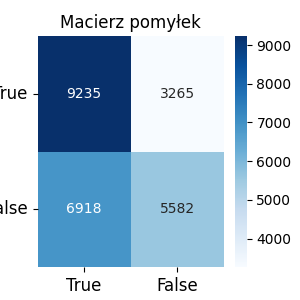
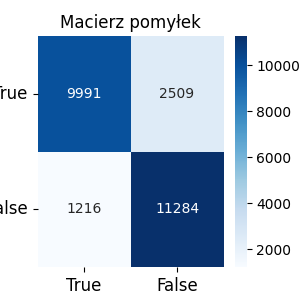




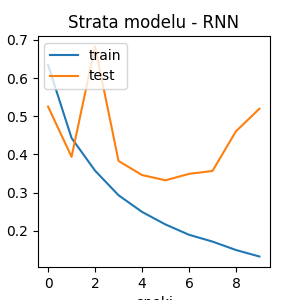
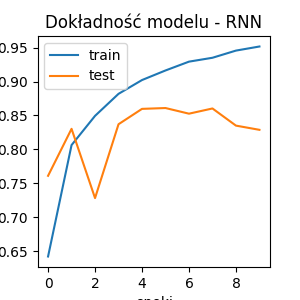
### Rozmiar paddingu / długość maski – 130

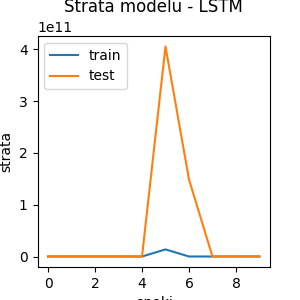
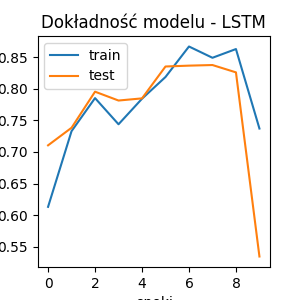


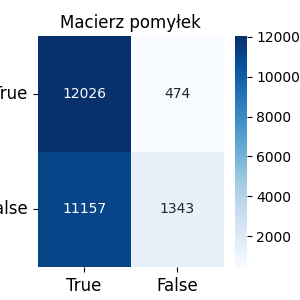
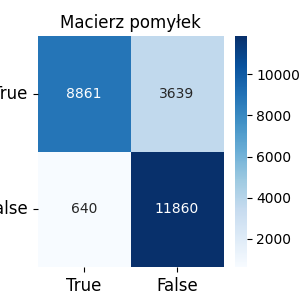




### Rozmiar paddingu / długość maski - 150







# Wnioski

Po przeprowadzeniu eksperymentów mogę stwierdzić, że sieć konwolucyjna idealnie nadaje się do zadania klasyfikacji obrazów i zapewnia dużo lepszą skuteczność oraz wydajność od sieci MLP. Warstwa poolingu zależy w dużej mierze od rozmiarów obrazów wejściowych oraz szczegółowości tych obrazów. Dla zbioru MNIST nie zauważyłem by zmiana wielkości okna w znaczący sposób wpływała na skuteczność modelu. Do usuwaniu ‘szumu’ z bardziej złożonych obrazów jest na pewno bardzo istotną techniką. Technika ‘dropout’ w efektywny sposób pozwala pozbyć się zjawiska overfittingu, natomiast zbyt mocne usuwanie neuronów może doprowadzić do bardzo wolnego procesu nauczania.