

Sieci Neuronowe – laboratorium

Ćwiczenie nr. 3

Kajetan Pynka, 254495

Spis treści

Wstęp teoretyczny.....	4
Porównanie sieci konwolucyjnej i MLP.....	5
Rozmiar paczki.....	5
Rozmiar paczki – 5.....	5
Rozmiar paczki – 20.....	7
Rozmiar paczki – 100.....	9
Rozmiar paczki – 500.....	11
Rozmiar paczki – 1000.....	13
Liczba epok.....	15
Liczba epok – 3.....	15
Liczba epok – 5.....	17
Liczba epok – 10.....	19
Liczba epok – 15.....	21
Liczba epok – 20.....	23
Liczba neuronów / filtrów.....	25
Liczba neuronów / filtrów – 10/10.....	25
Liczba neuronów / filtrów – 50/20.....	27
Liczba neuronów / filtrów – 100/30.....	29
Liczba neuronów / filtrów – 200/40.....	31
Liczba neuronów / filtrów – 250/50.....	33
Wpływ rozmiaru okna poolingu oraz rodzaju poolingu.....	35
Kwadratowe okno poolingu.....	35
Okno poolingu – 2x2.....	35
Okno poolingu – 3x3.....	36
Okno poolingu – 4x4.....	37
Okno poolingu – 5x5.....	38
Okno poolingu – 6x6.....	39
Prostokątne okno poolingu.....	40
Okno poolingu – 2x1.....	40
Okno poolingu – 1x2.....	41
Okno poolingu – 2x3.....	42
Okno poolingu – 3x2.....	43
Okno poolingu – 1x5.....	44
Okno poolingu – 5x1.....	45

Typ pooling.....	46
Average pooling.....	46
Max pooling.....	47
Global average pooling.....	48
Global max pooling.....	49
Ocena efektywności techniki 'dropout'.....	50
Prawdopodobieństwo dropout'u - 0.01.....	50
Prawdopodobieństwo dropout'u - 0.05.....	52
Prawdopodobieństwo dropout'u - 0.1.....	54
Prawdopodobieństwo dropout'u - 0.2.....	56
Prawdopodobieństwo dropout'u - 0.4.....	58
Wnioski.....	60

Wstęp teoretyczny

W ramach ćwiczenia zaimplementowałem model sieci konwolucyjnej (dalej CNN) oraz wielowarstwowej sieci perceptronowej (dalej MLP). 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 entropię krzyżową oraz używały optymalizatora Adam. W ramach ćwiczeń wykonałem następujące eksperymenty: porównanie sieci CNN i MLP (na podstawie wartości: rozmiaru paczki, liczby epok, liczby neuronów/filtrów), badania poolingu (ze względu na rozmiar okna oraz rodzaj poolingu), ocena użyteczności techniki 'dropout' (dla różnego prawdopodobieństwa wystąpienia dropout'u).

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

- Liczba neuronów warstwy ukrytej MLP = 128
- Liczba filtrów warstwy konwolucyjnej CNN = 32
- Szerokość okna warstwy konwolucyjnej = 3
- Wysokość okna warstwy konwolucyjnej = 3
- Krok okna warstwy konwolucyjnej = 1
- Szerokość okna warstwy poolingu = 2
- Wysokość okna warstwy poolingu = 2
- Krok okna warstwy poolingu = 2
- Rozmiar paczki = 128
- Liczba epok = 10
- Procent danych przeznaczony na zbiór walidacyjny = 10%

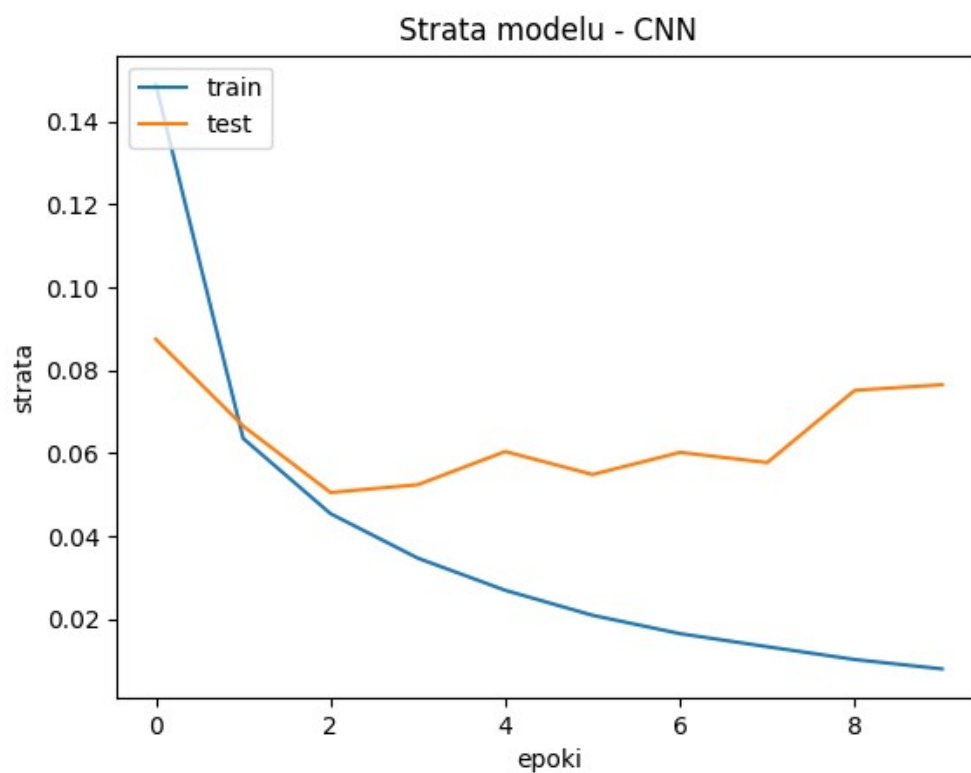
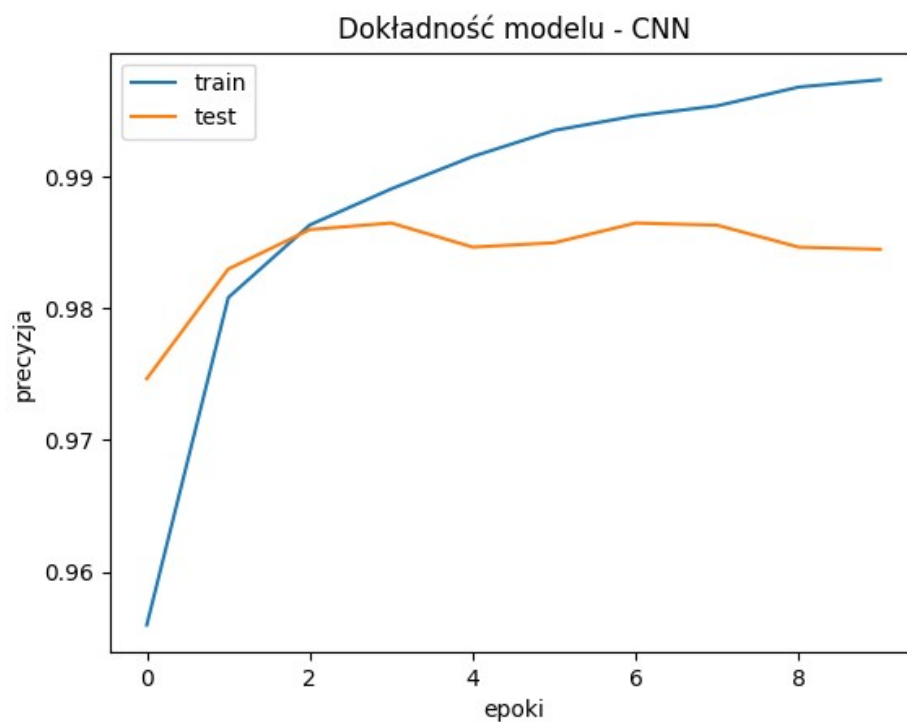
Dla każdego eksperymentu sporządzone zostały wykresy:

- precyzji/skuteczności modelu (w skali 0-1) na zbiorze treningowym oraz testowym w zależności od liczby epok
- wartość funkcji straty (entropii krzyżowej) na zbiorze treningowym oraz testowym w zależności od liczby epok

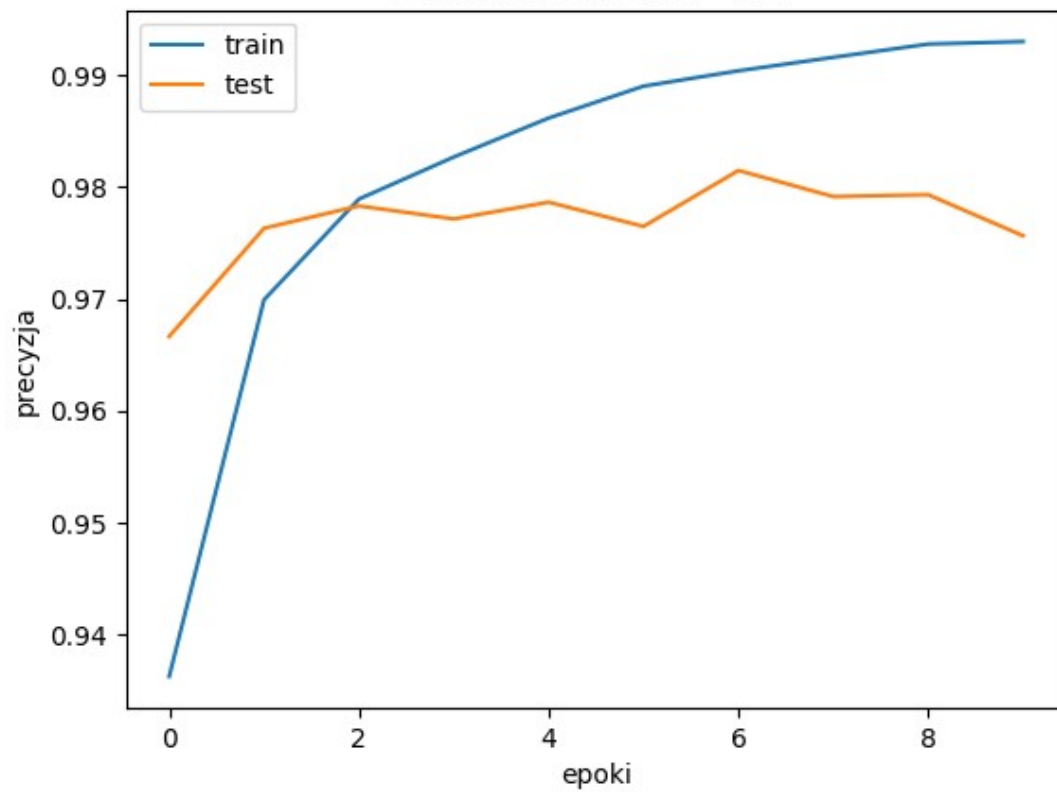
Porównanie sieci konwolucyjnej i MLP

Rozmiar paczki

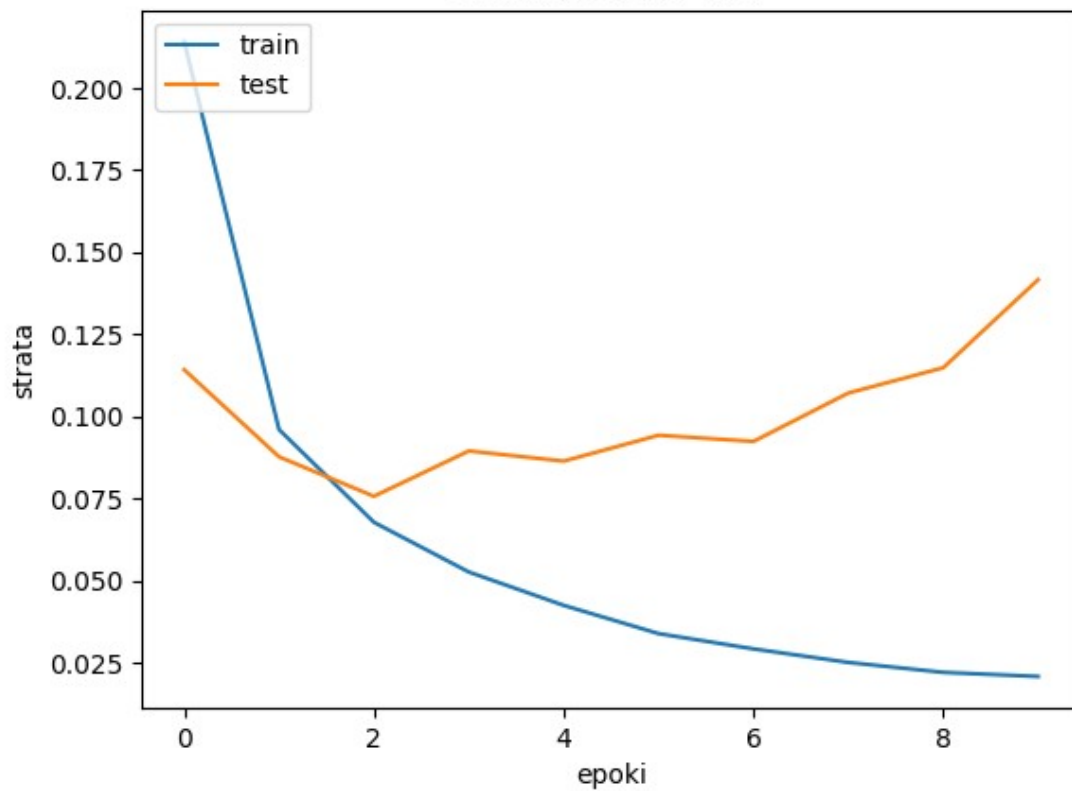
Rozmiar paczki - 5



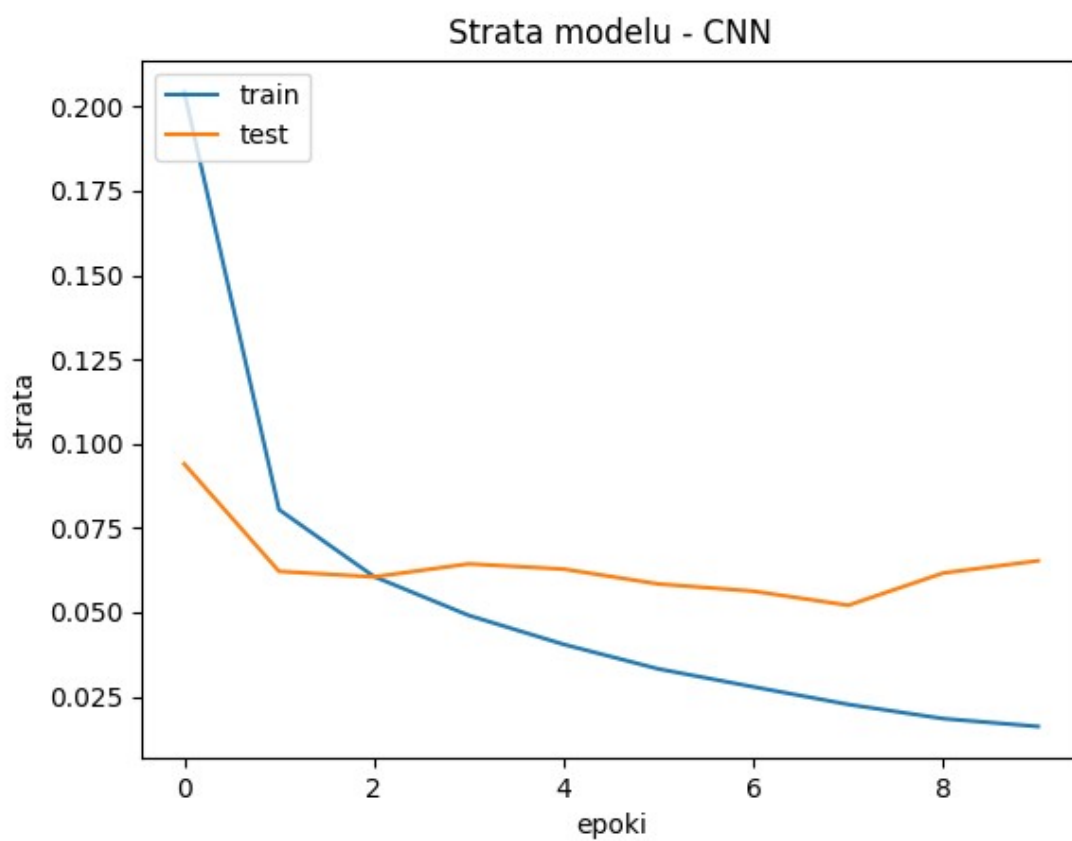
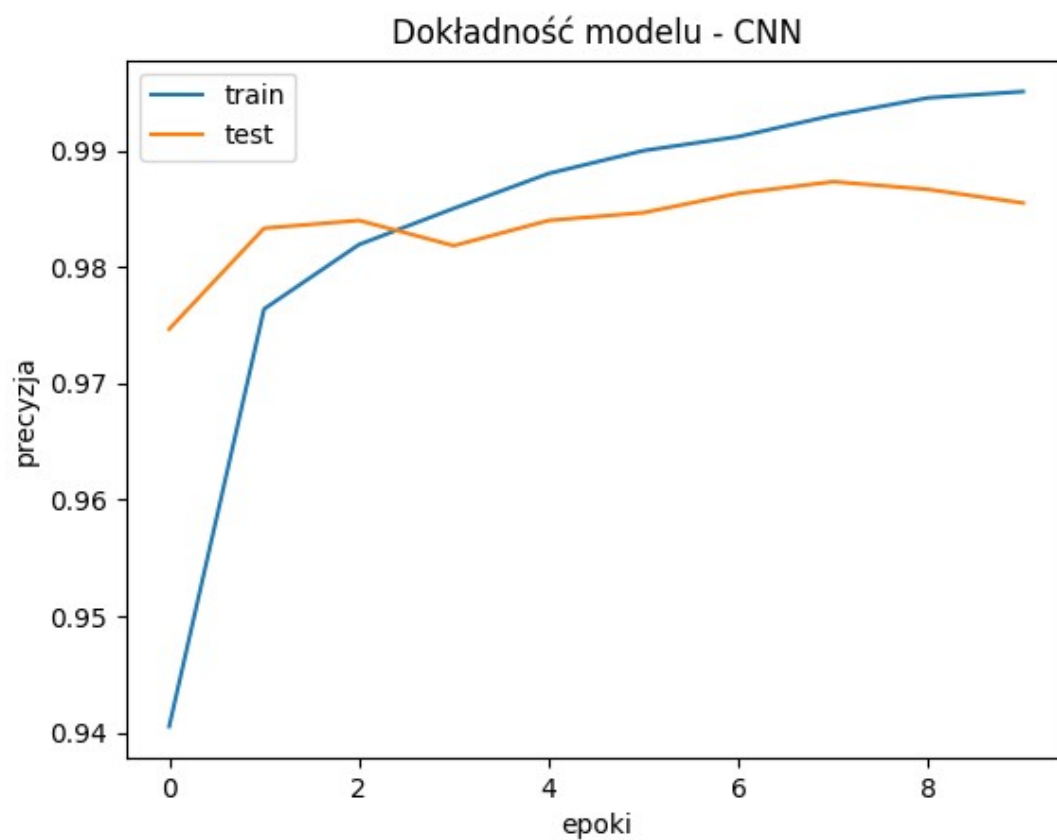
Dokładność modelu - MLP

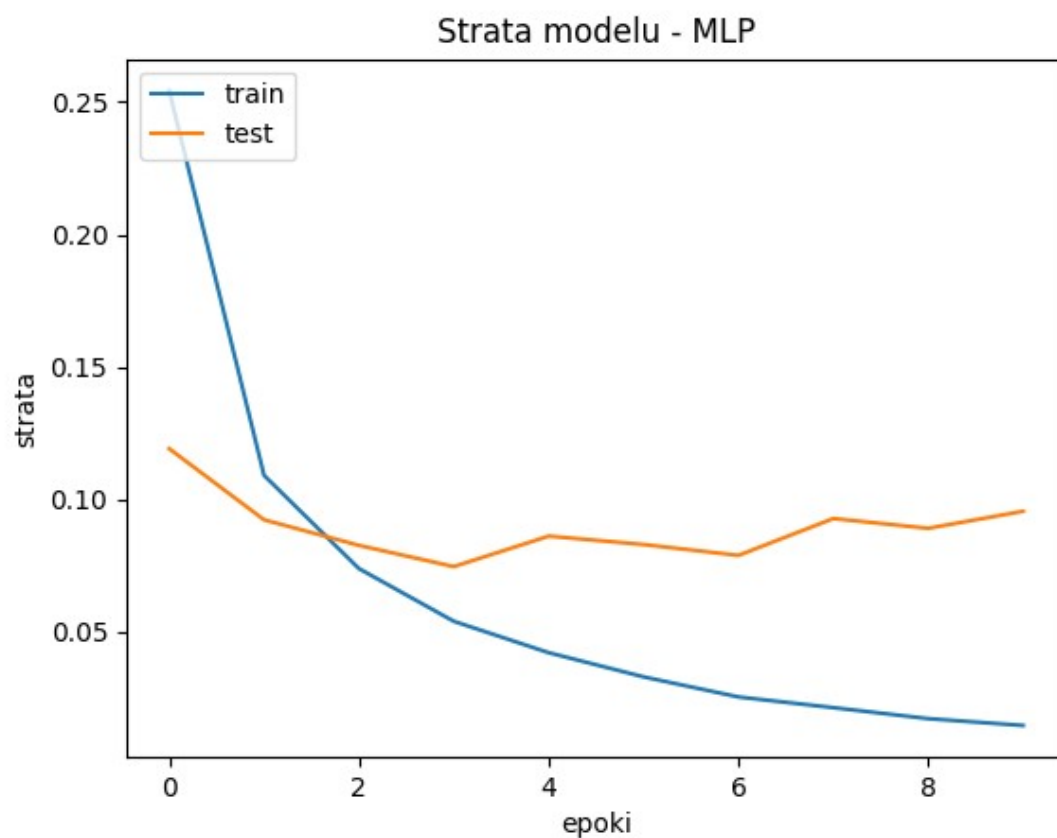
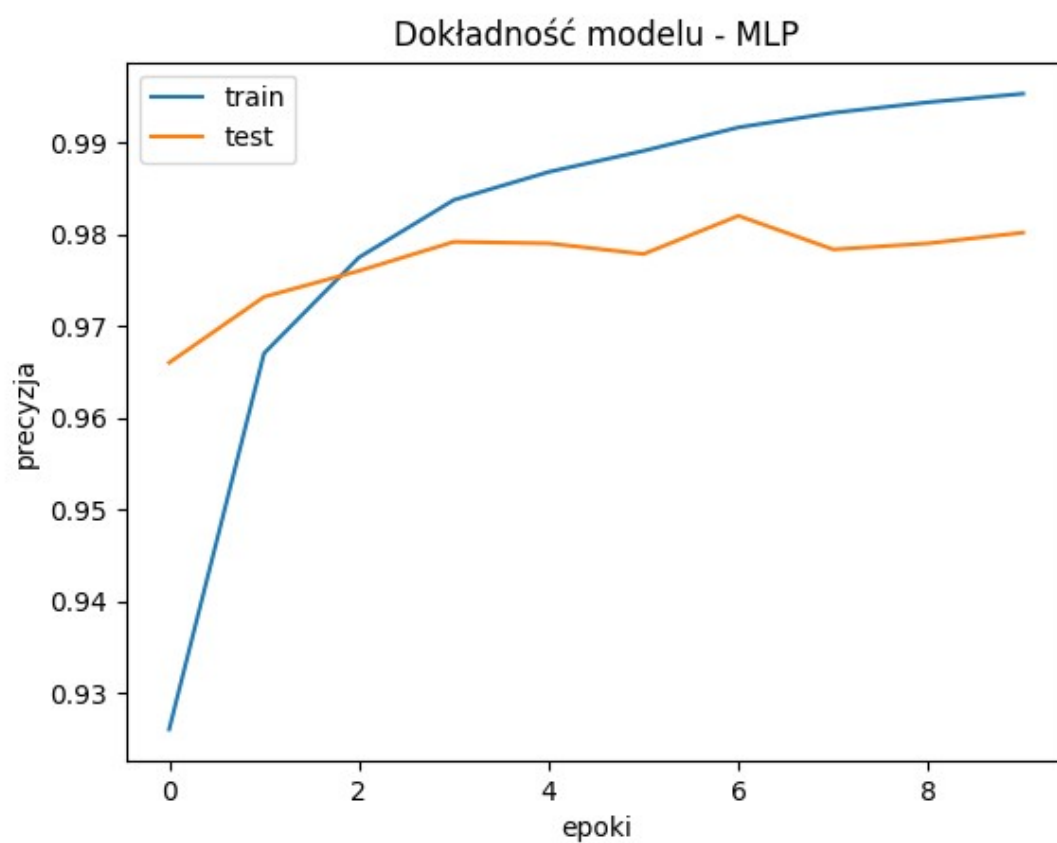


Strata modelu - MLP

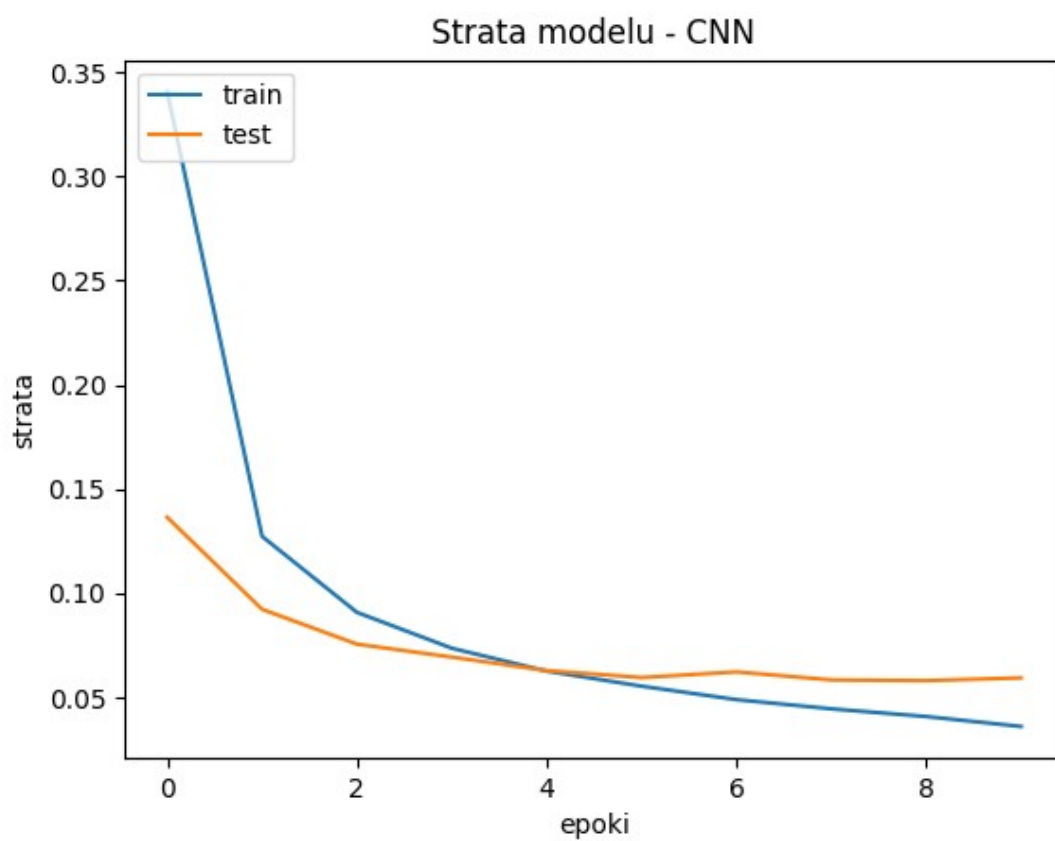
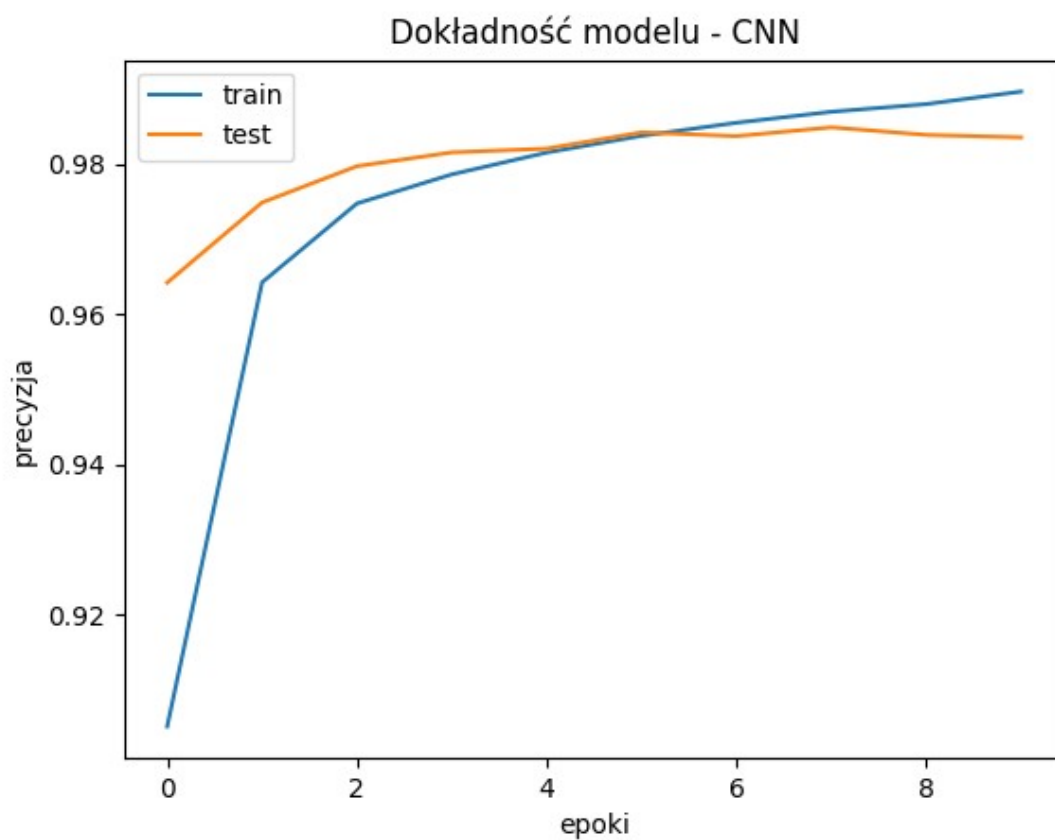


Rozmiar paczki - 20

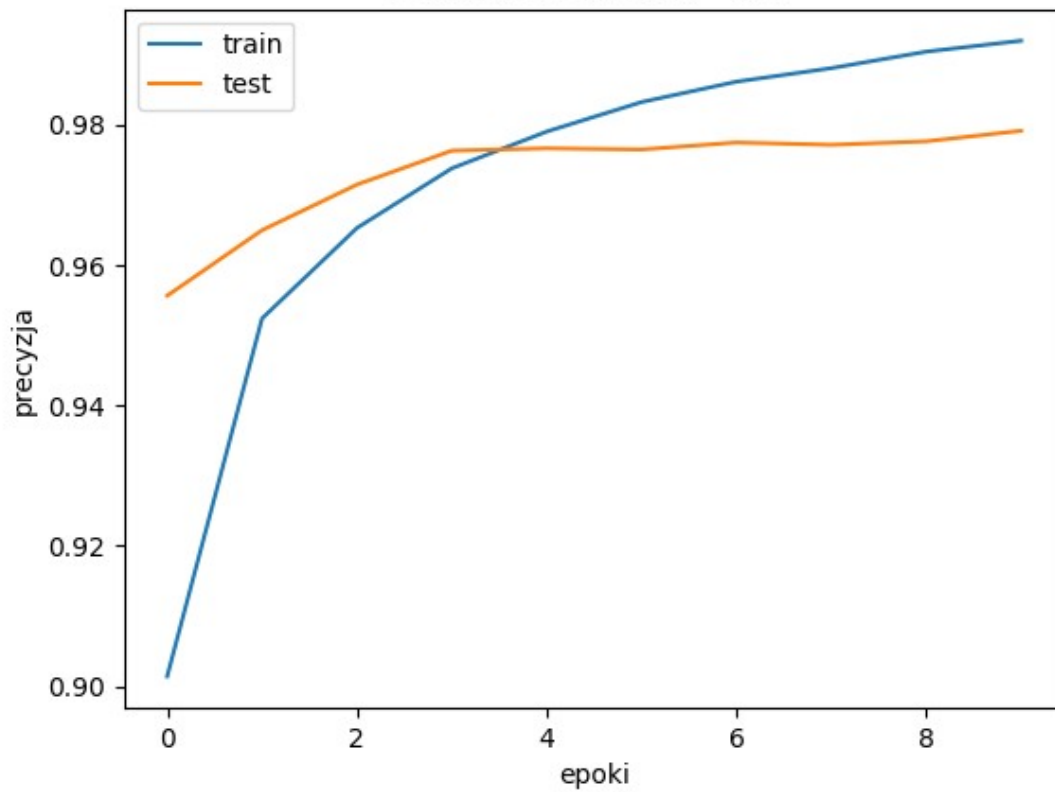




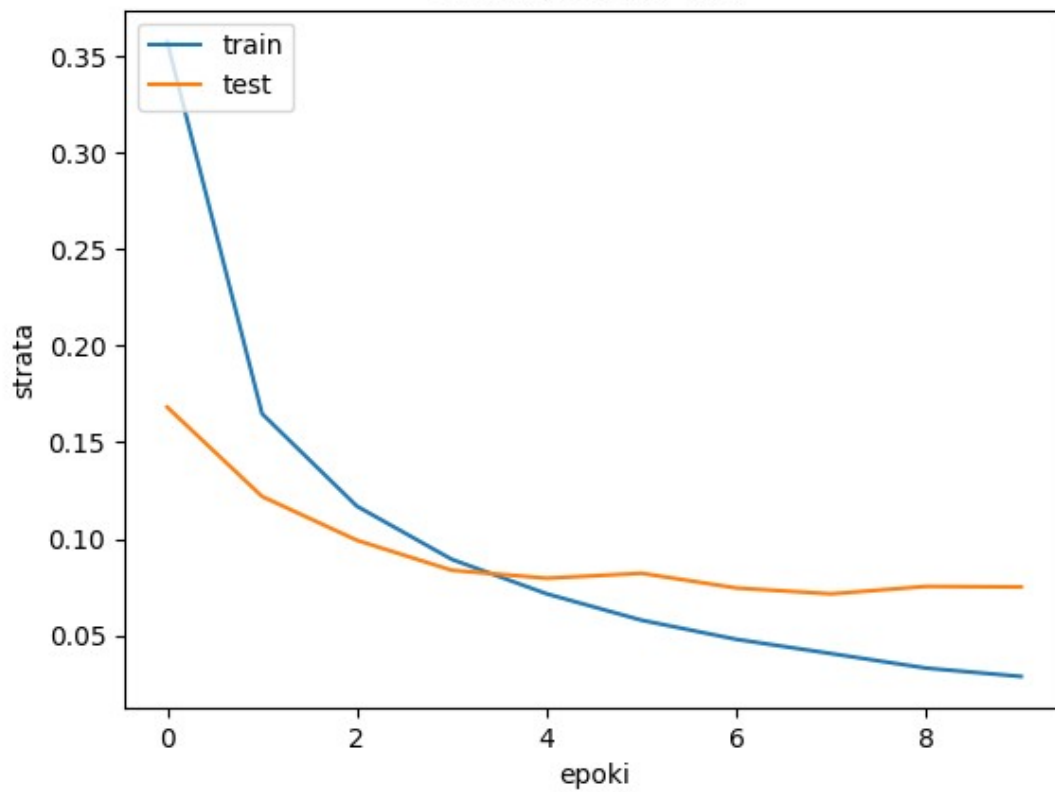
Rozmiar paczki - 100



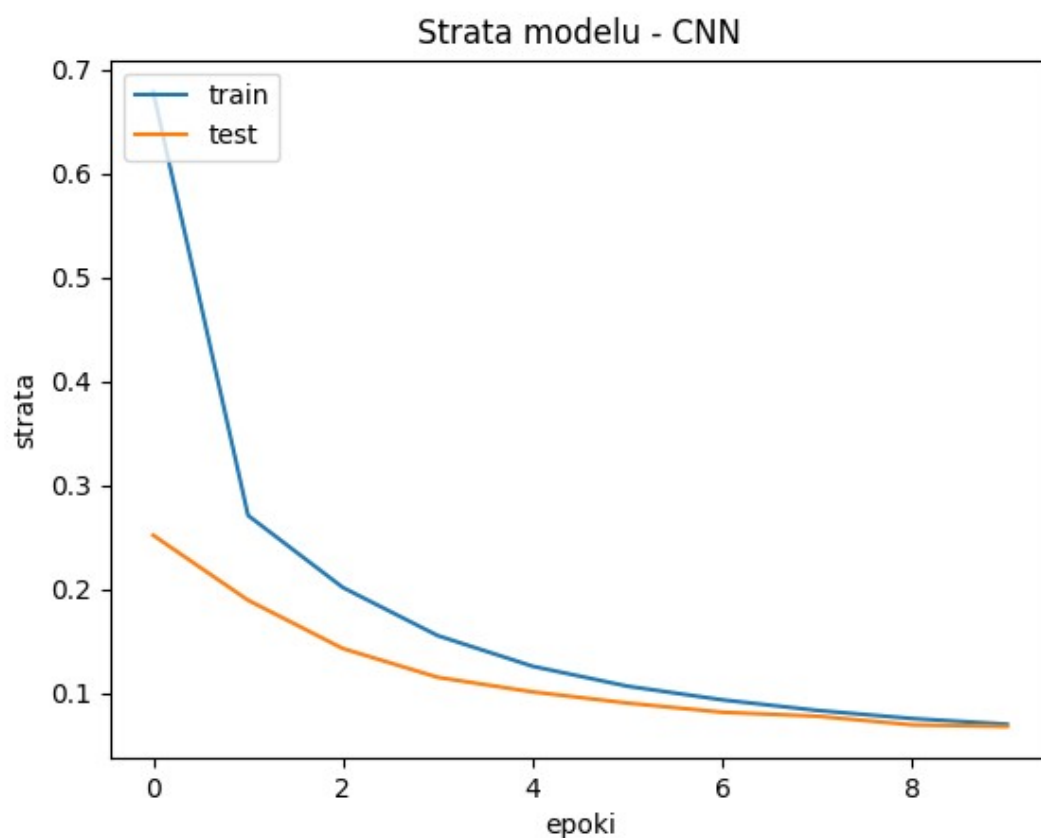
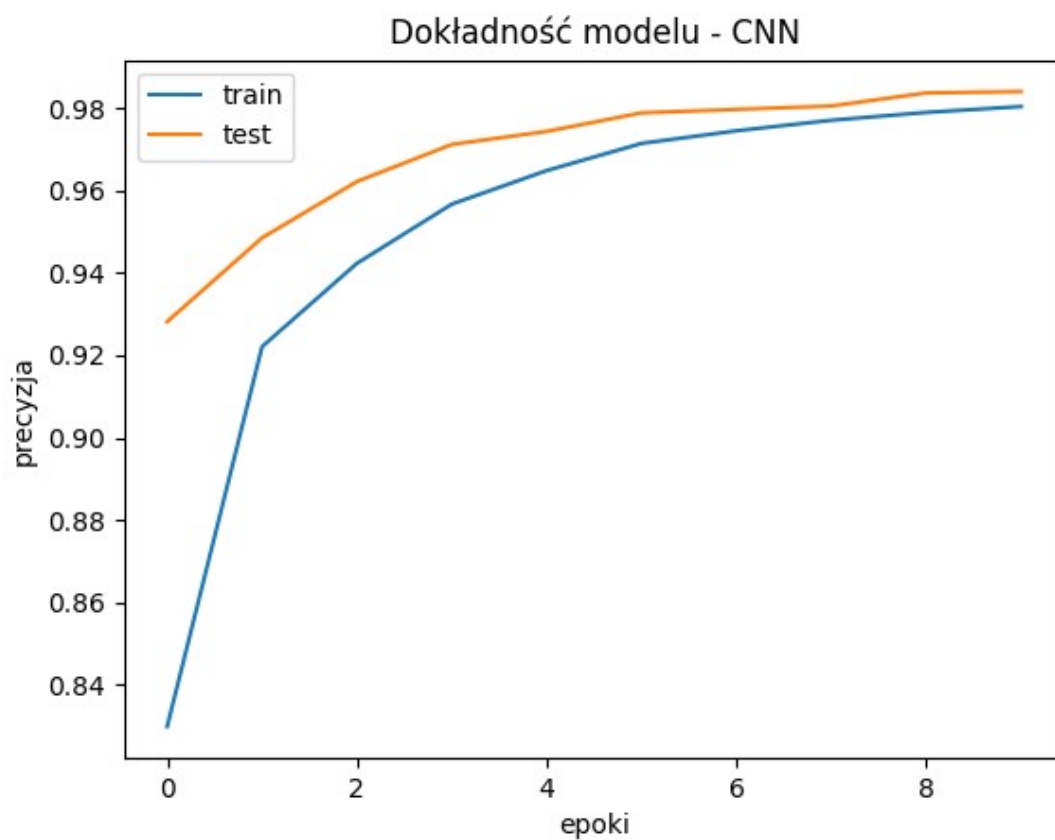
Dokładność modelu - MLP

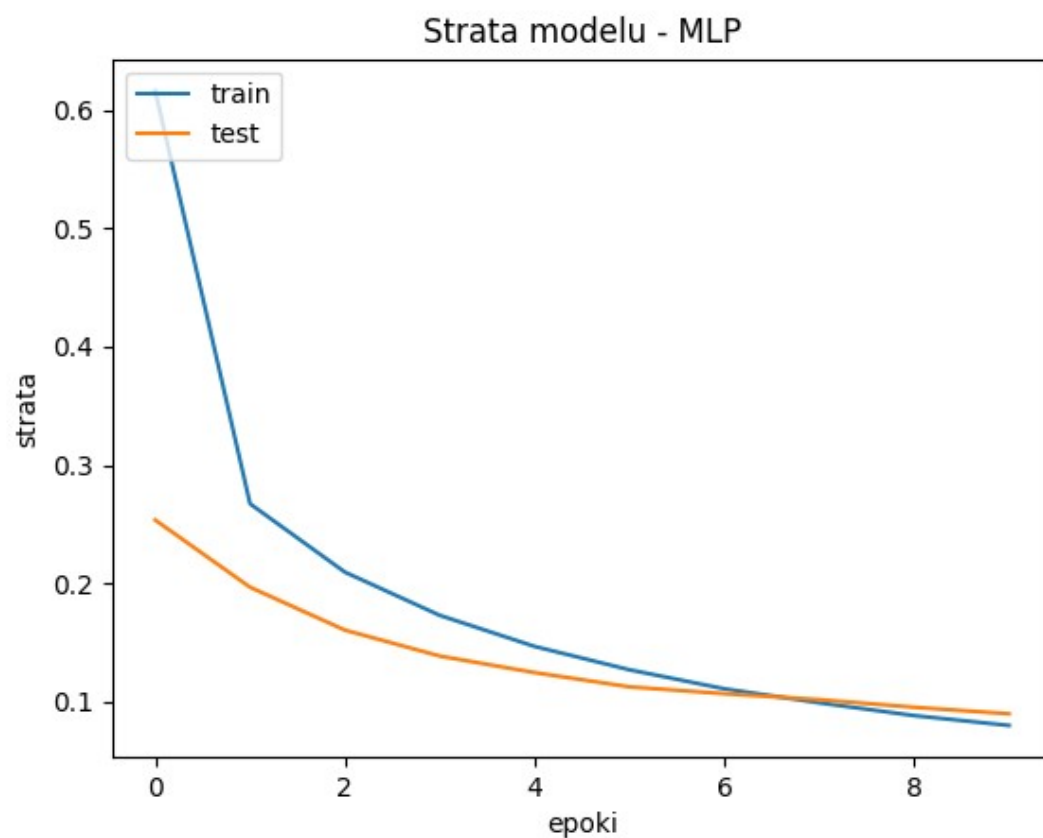
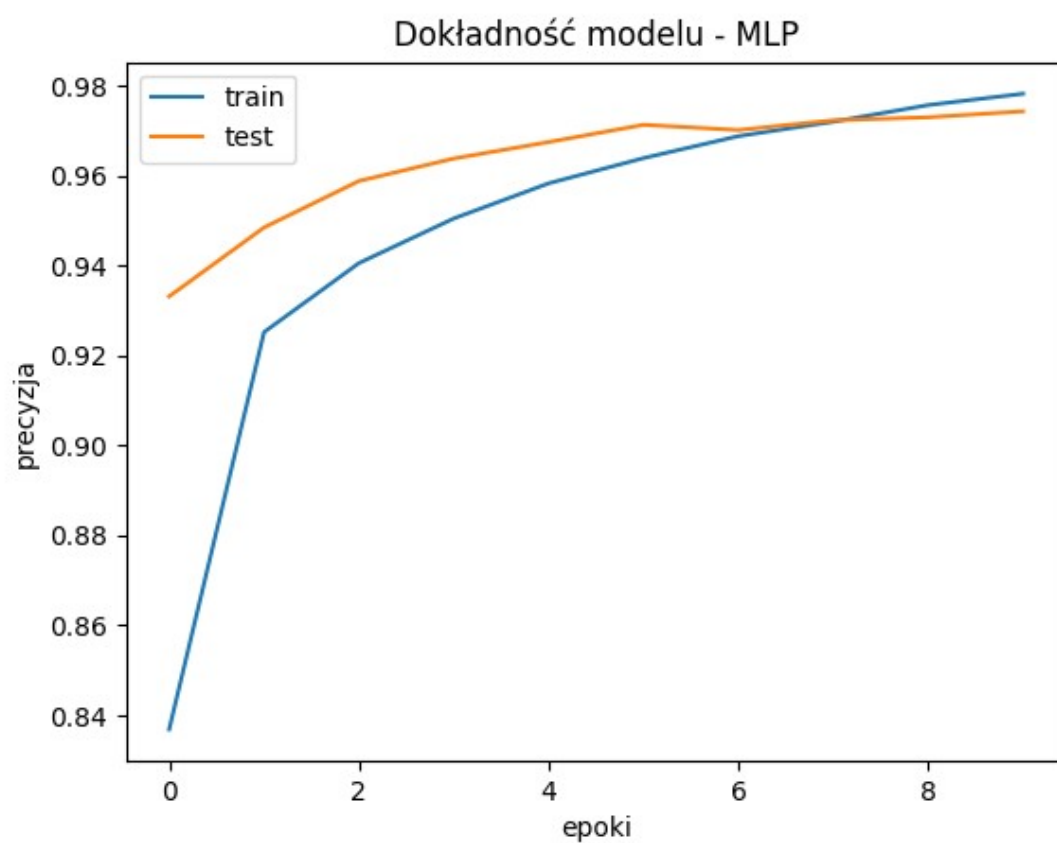


Strata modelu - MLP

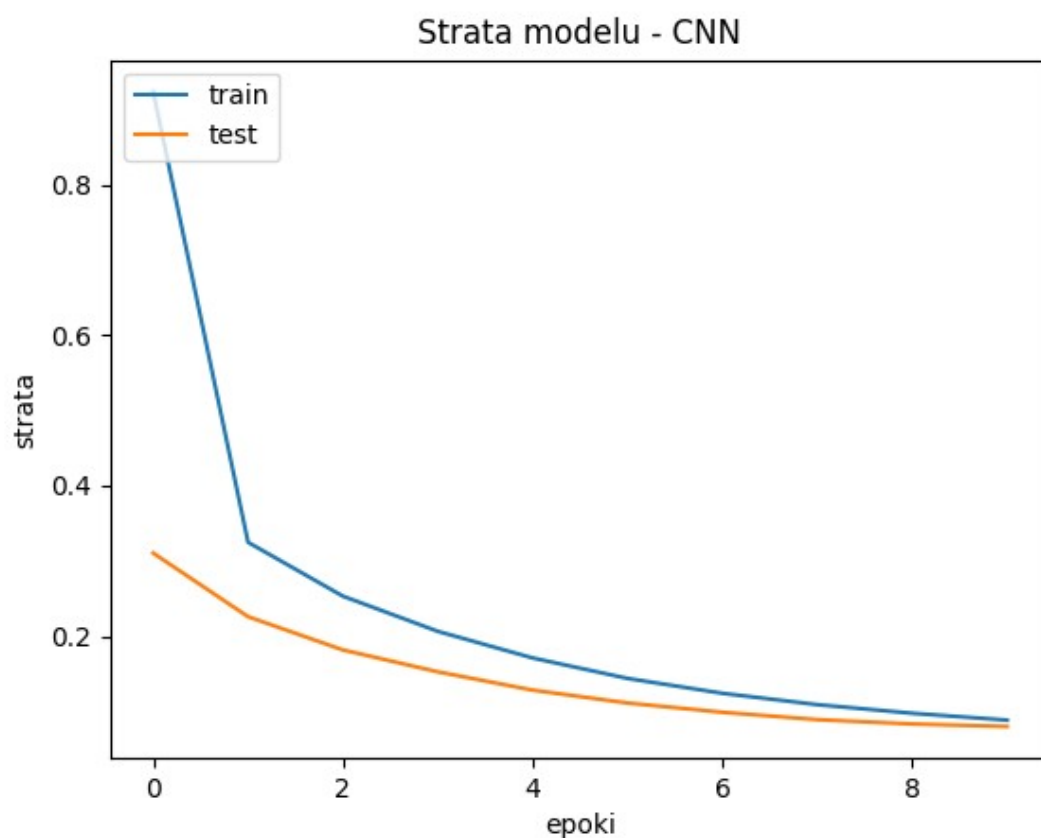
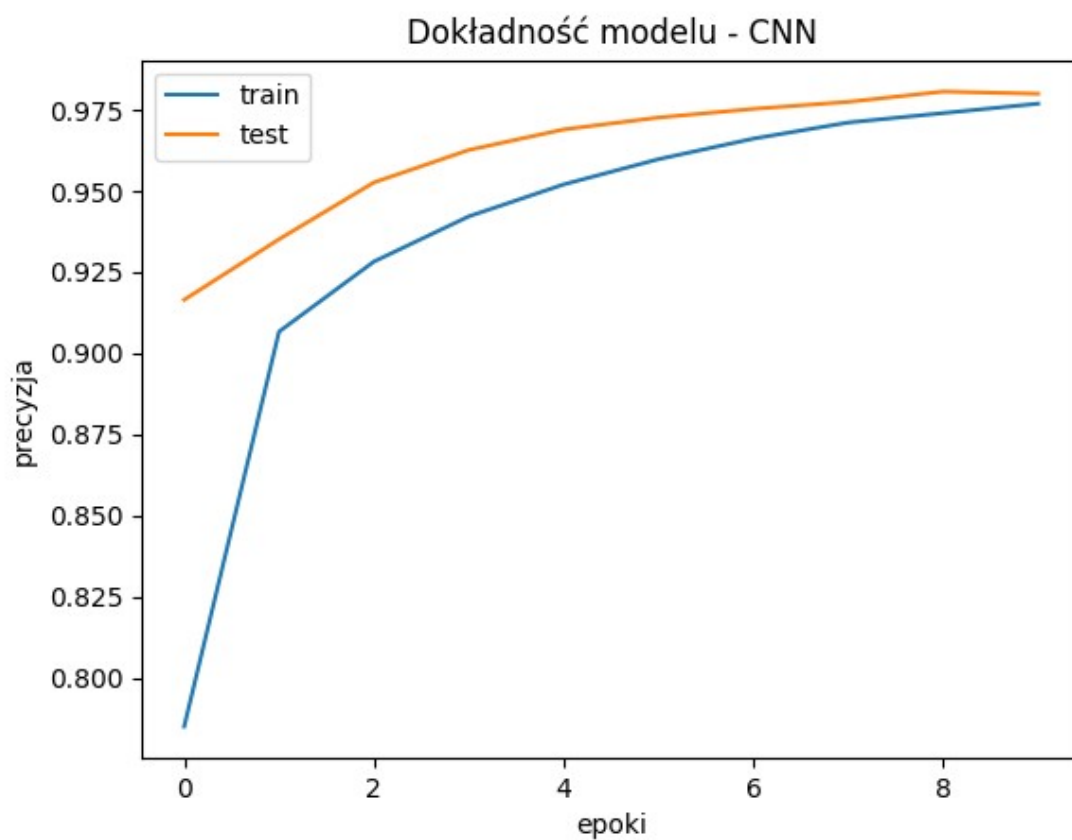


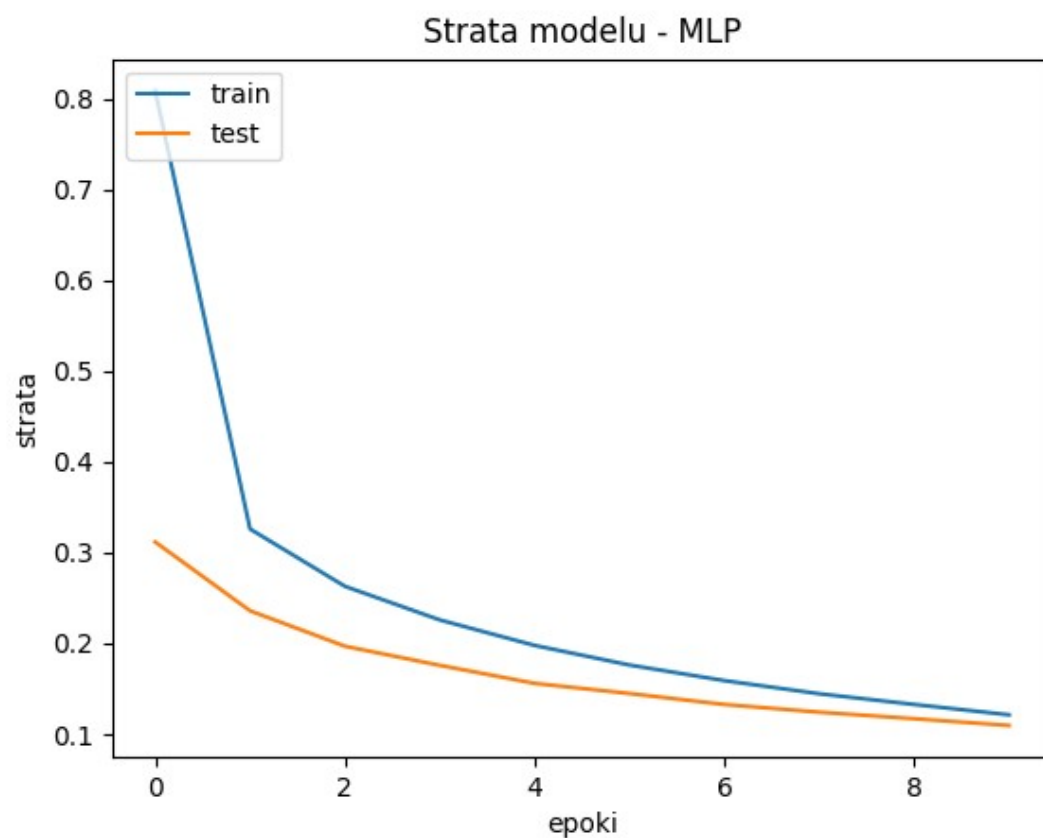
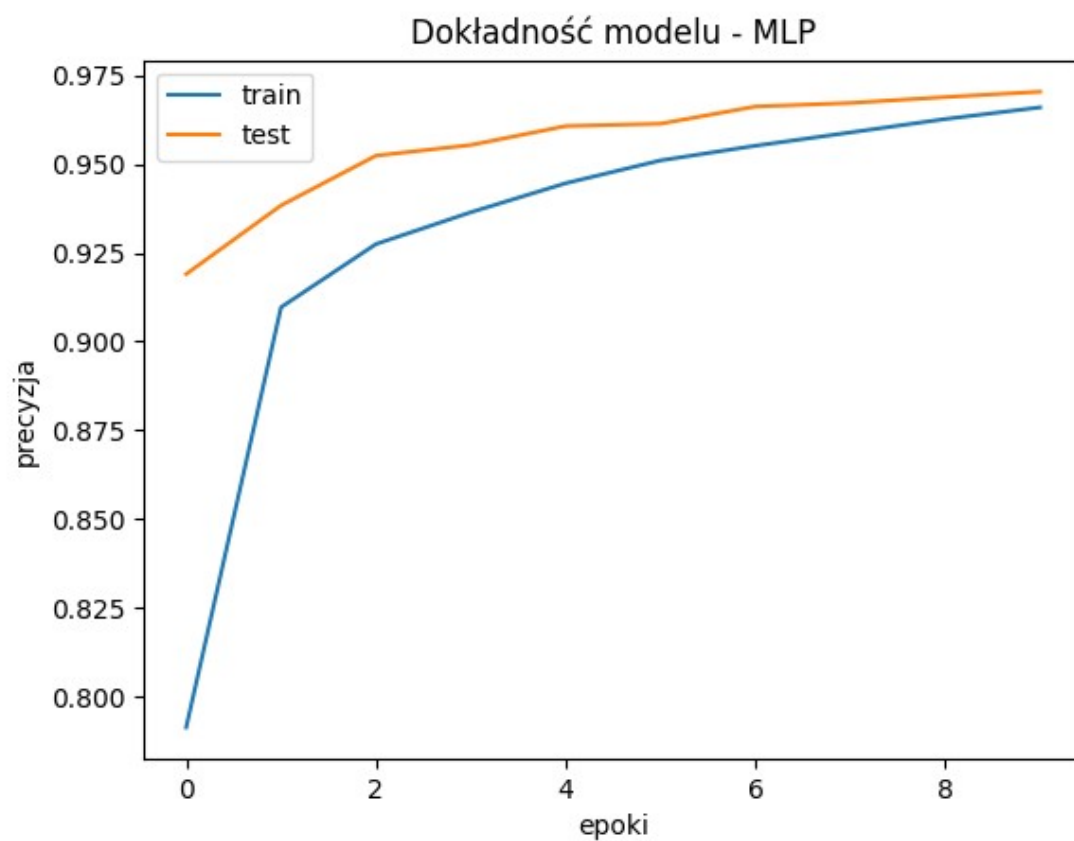
Rozmiar paczki - 500





Rozmiar paczki - 1000



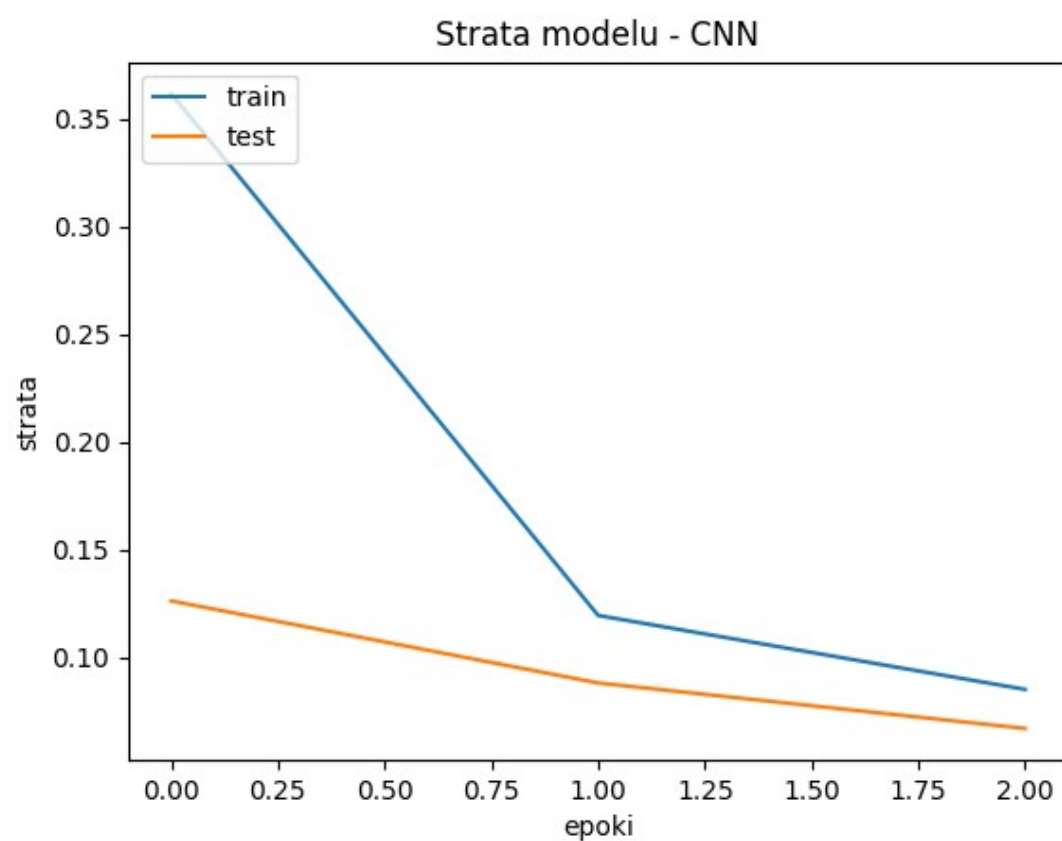
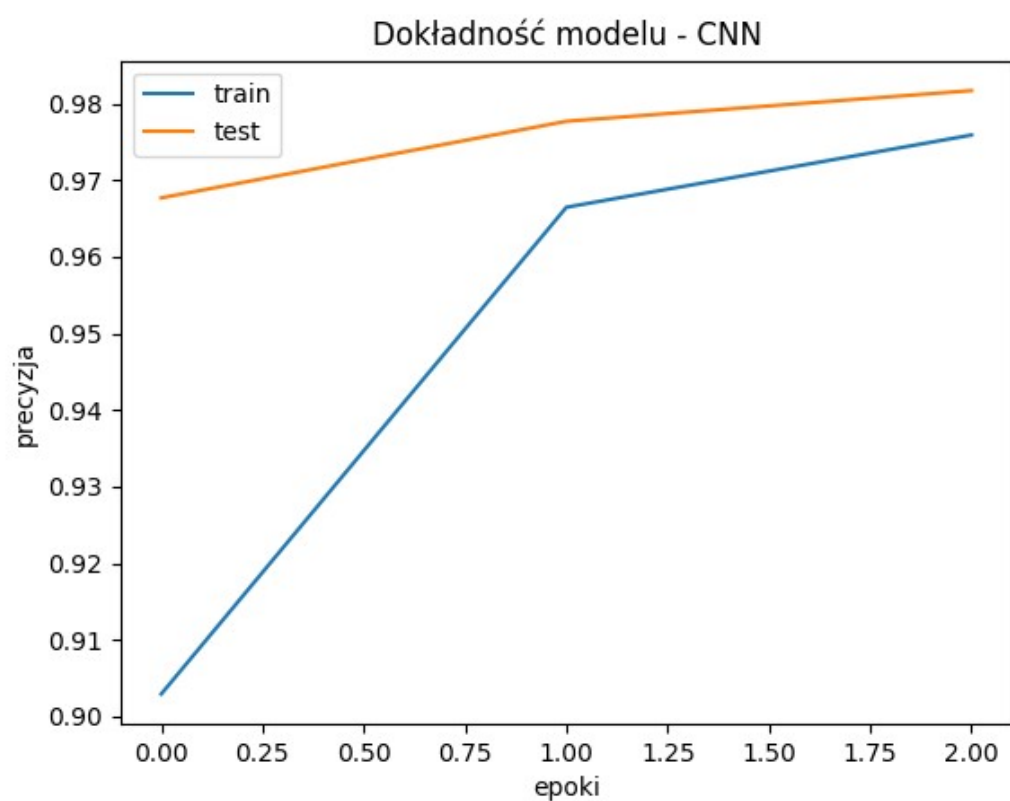


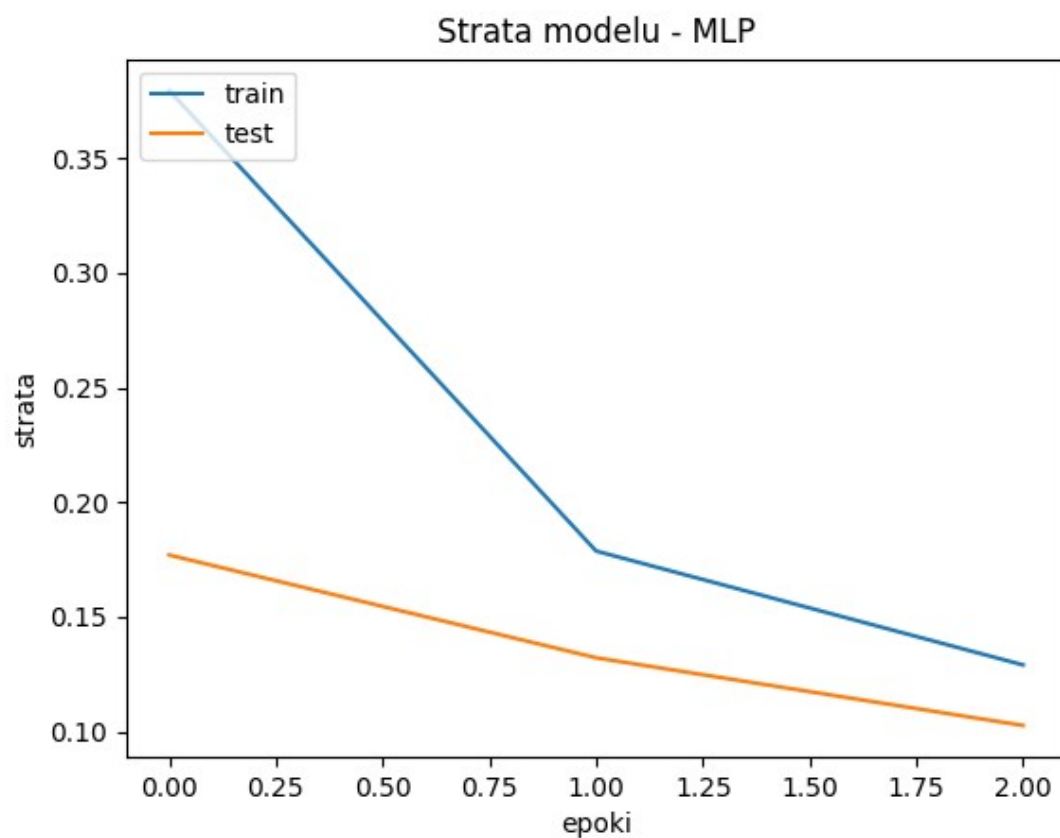
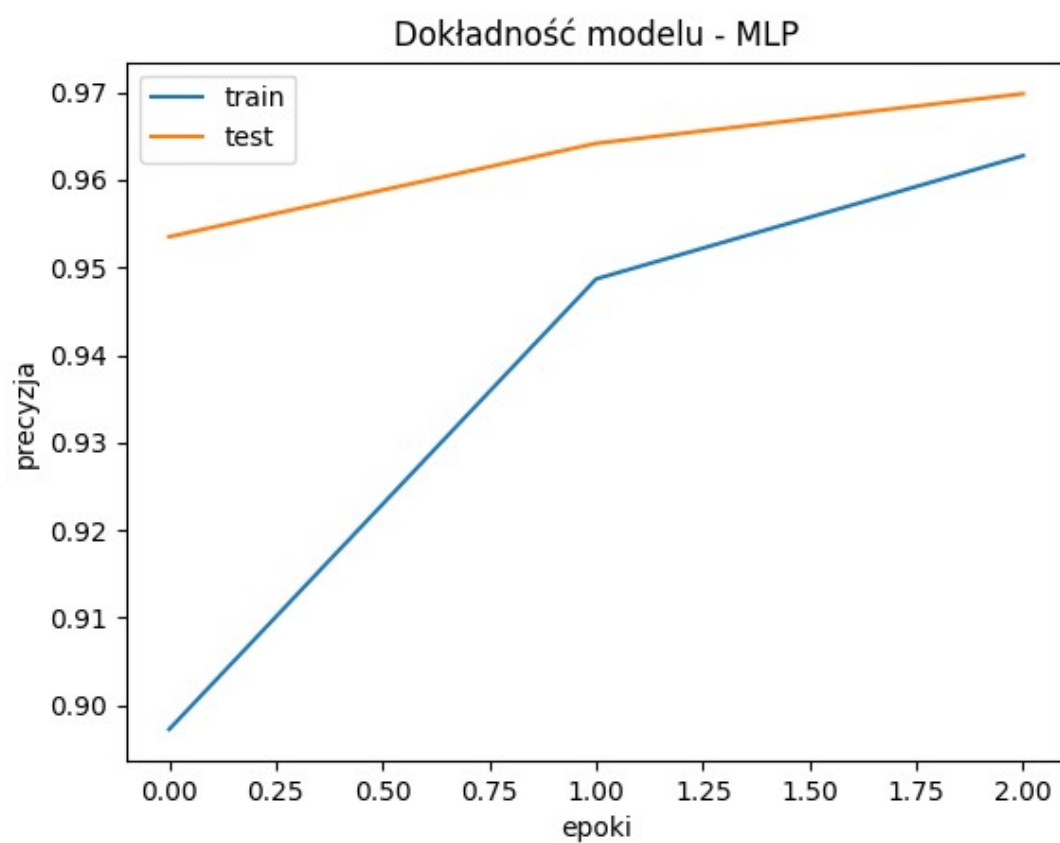
Wnioski

Dla bardzo małych rozmiarów paczki (5 i 20) możemy wyraźnie zaobserwować zjawisko overfittingu, zarówno dla sieci CNN jak i MLP. Powiedziałbym nawet, że wygląda ono gorzej z perspektywy MLP. Dla większych rozmiarów obie sieci wydają się uczyć w podobny sposób i w podobnej liczbie epok.

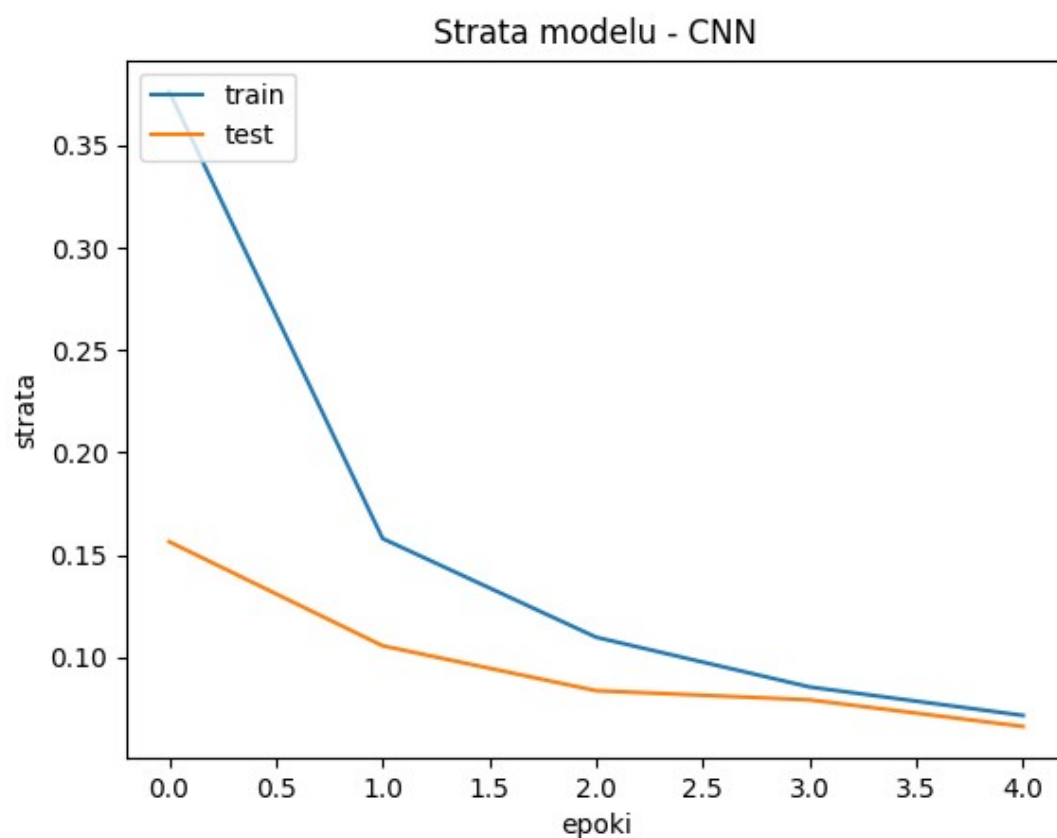
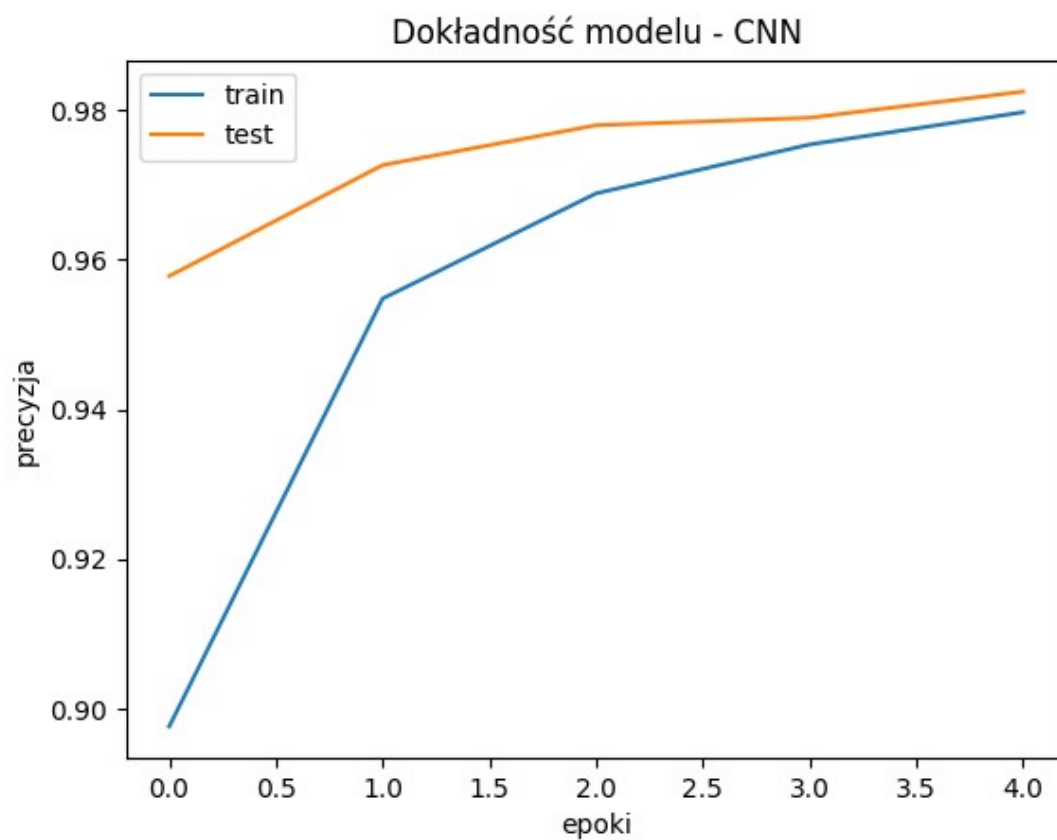
Liczba epok

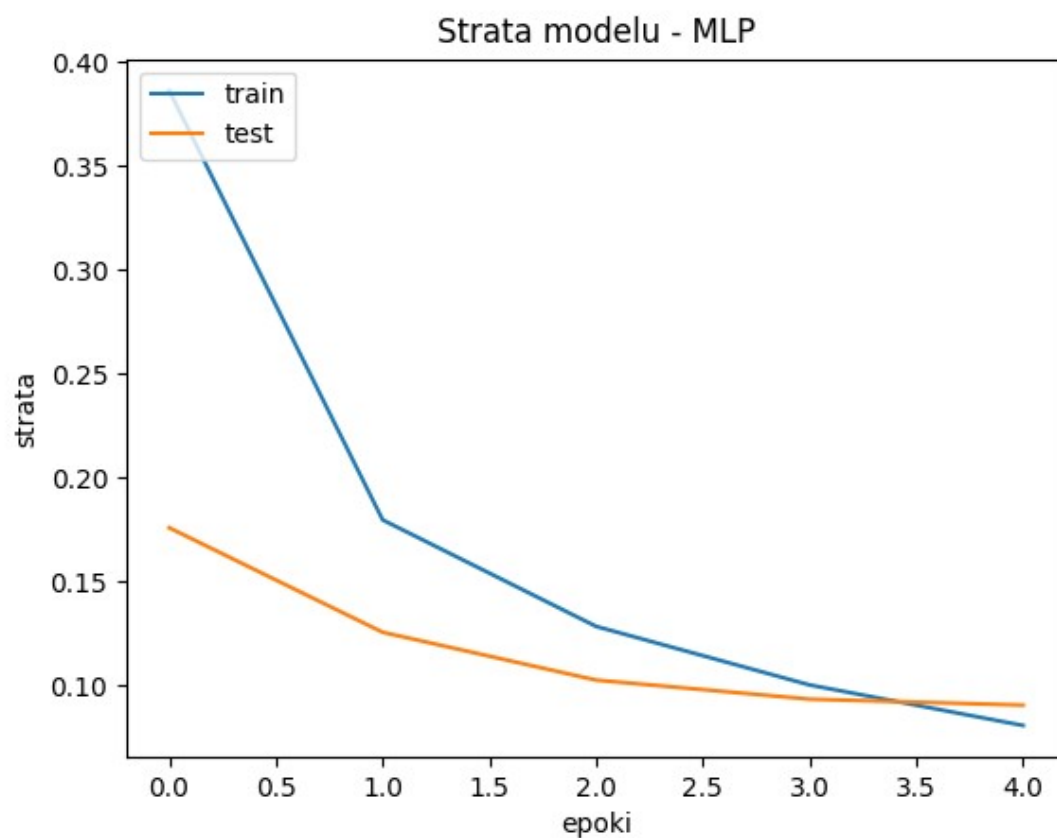
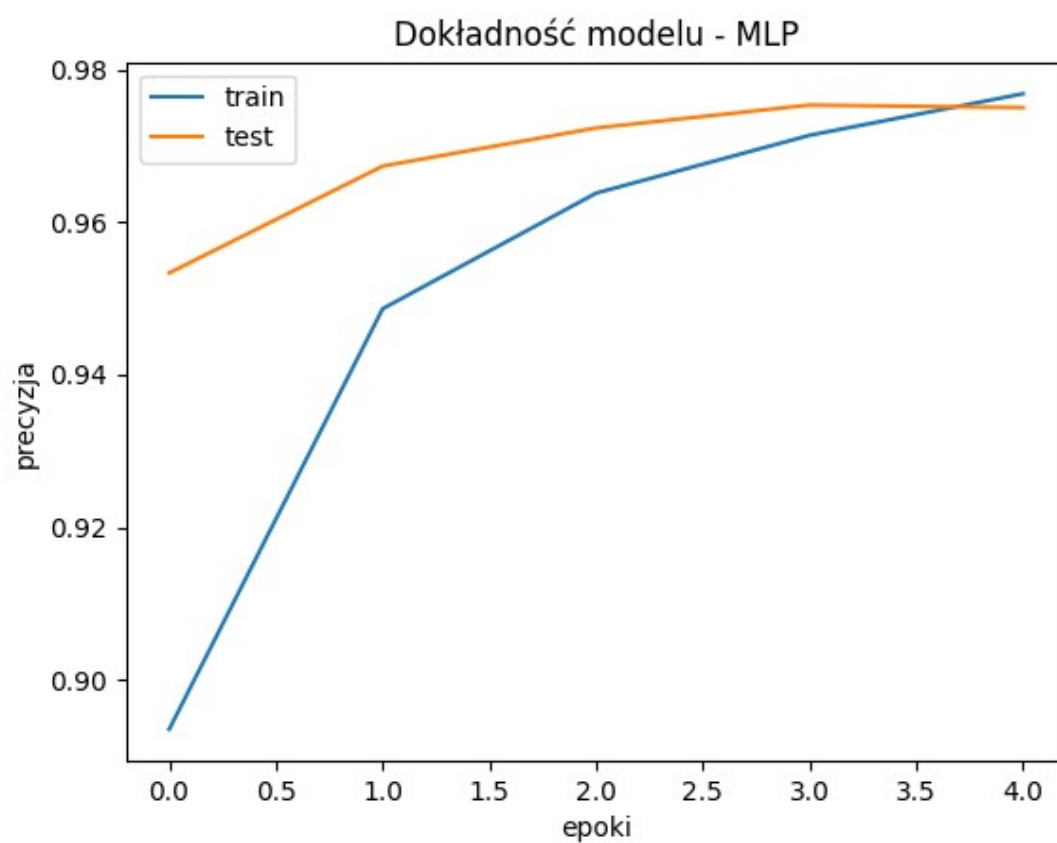
Liczba epok - 3



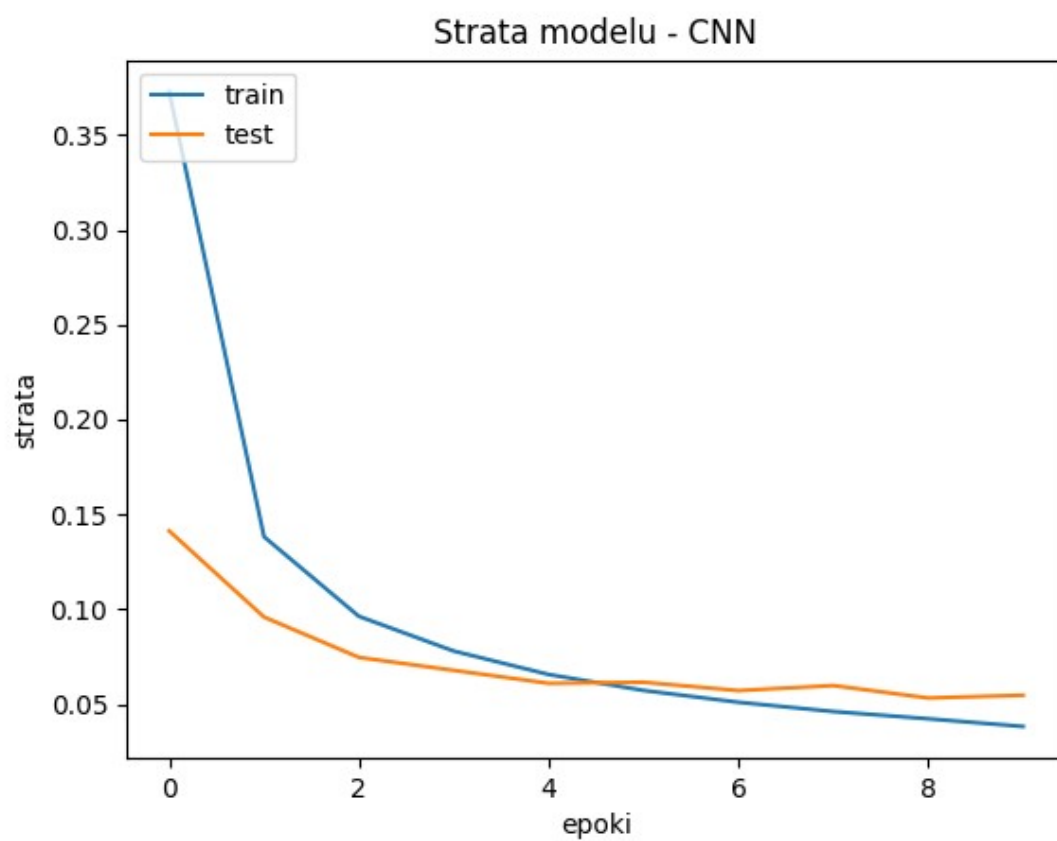
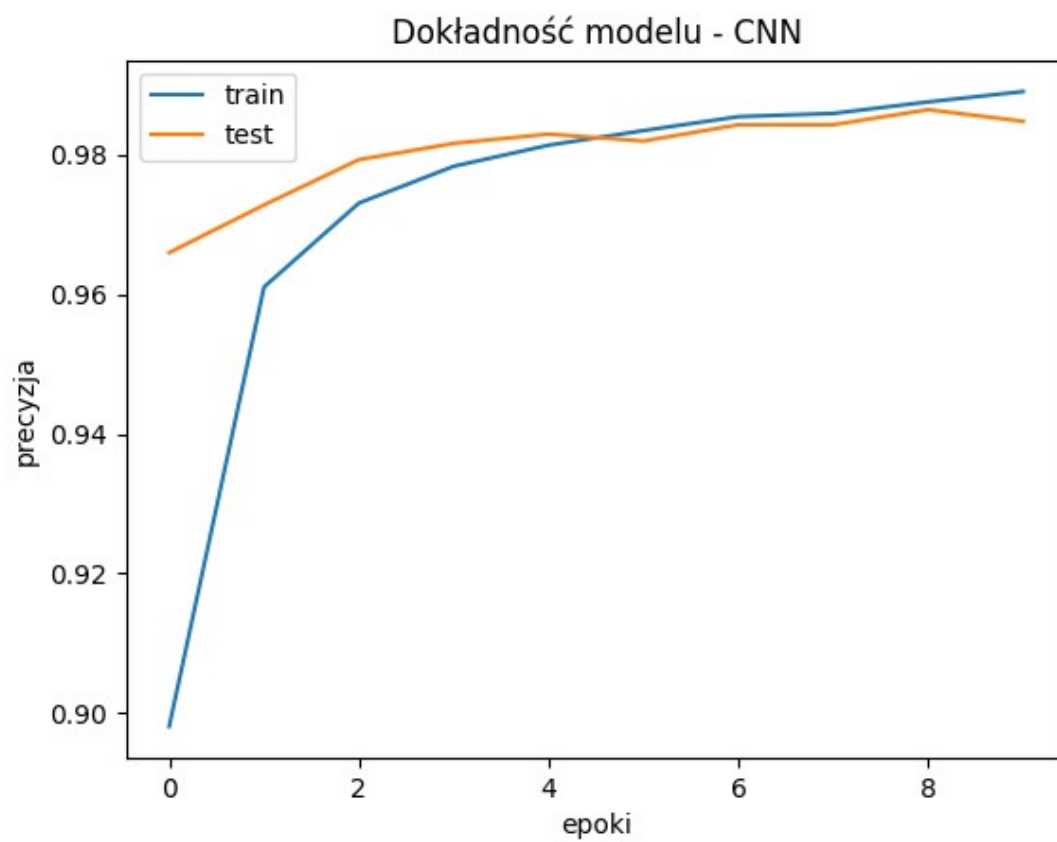


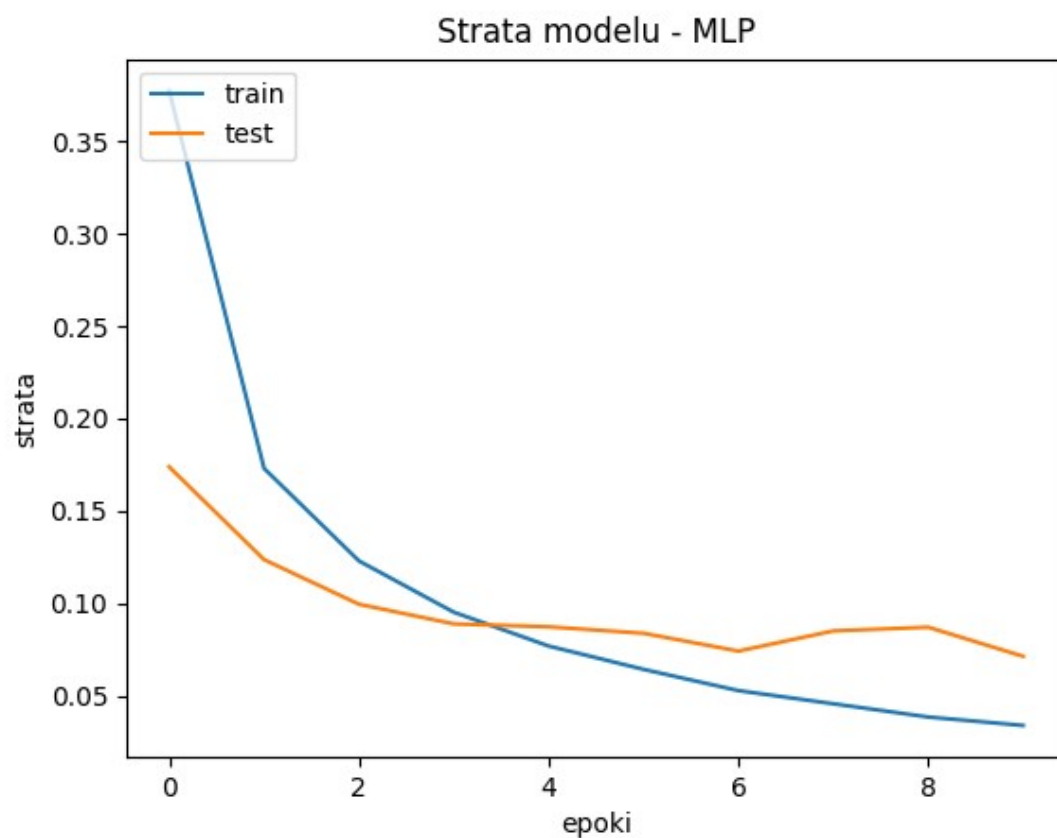
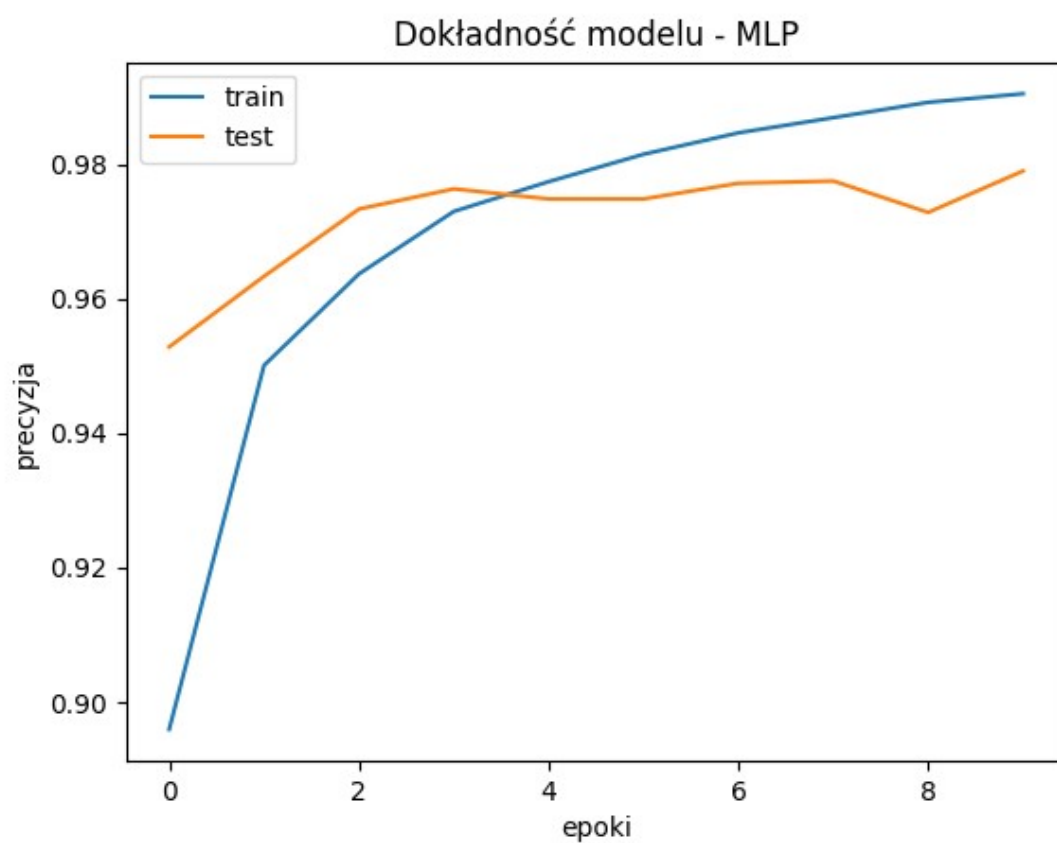
Liczba epok - 5



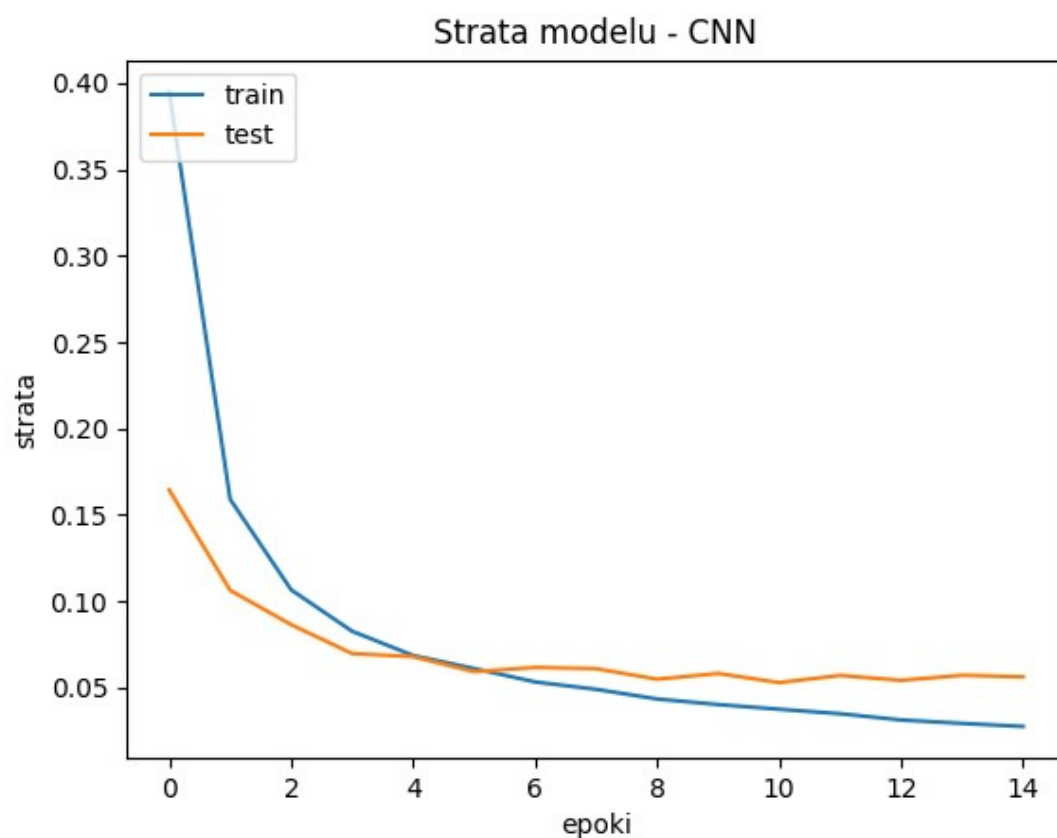
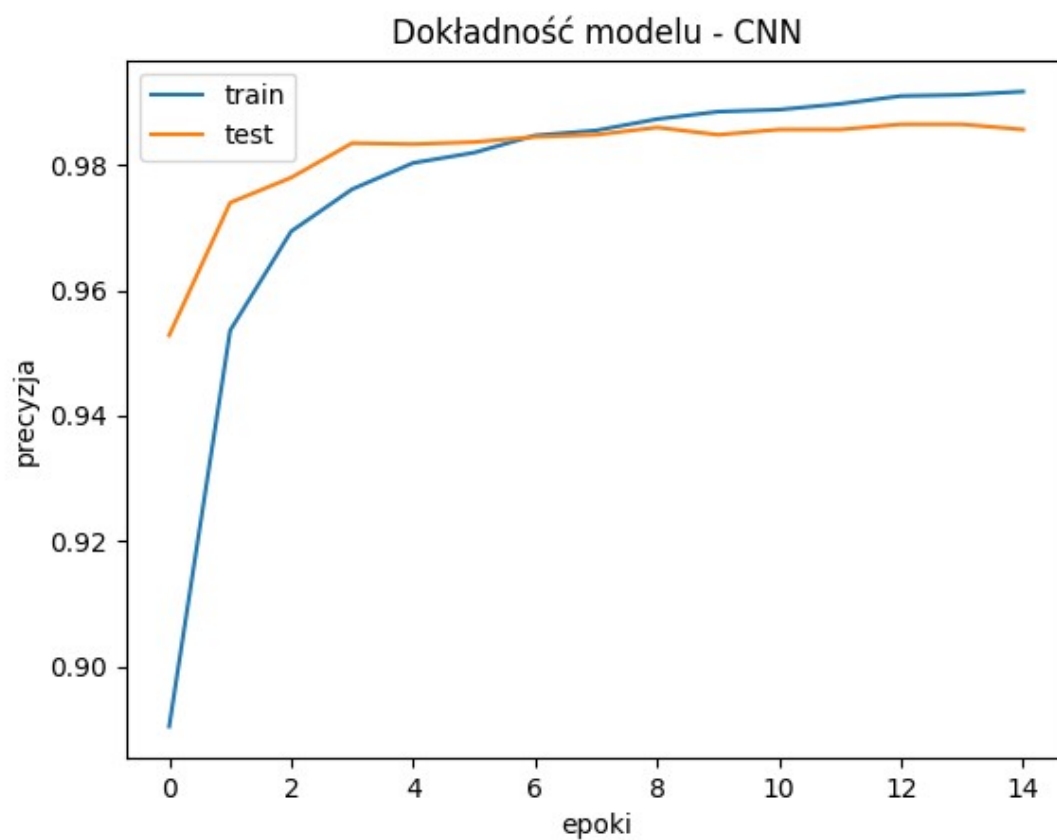


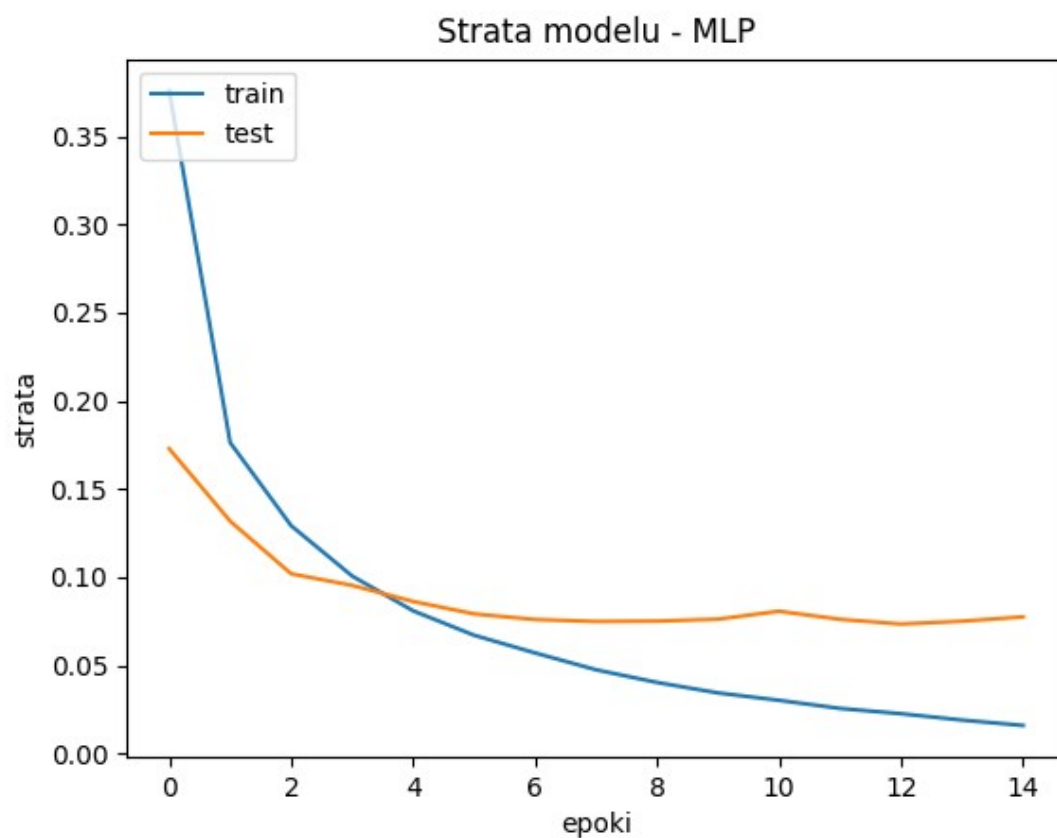
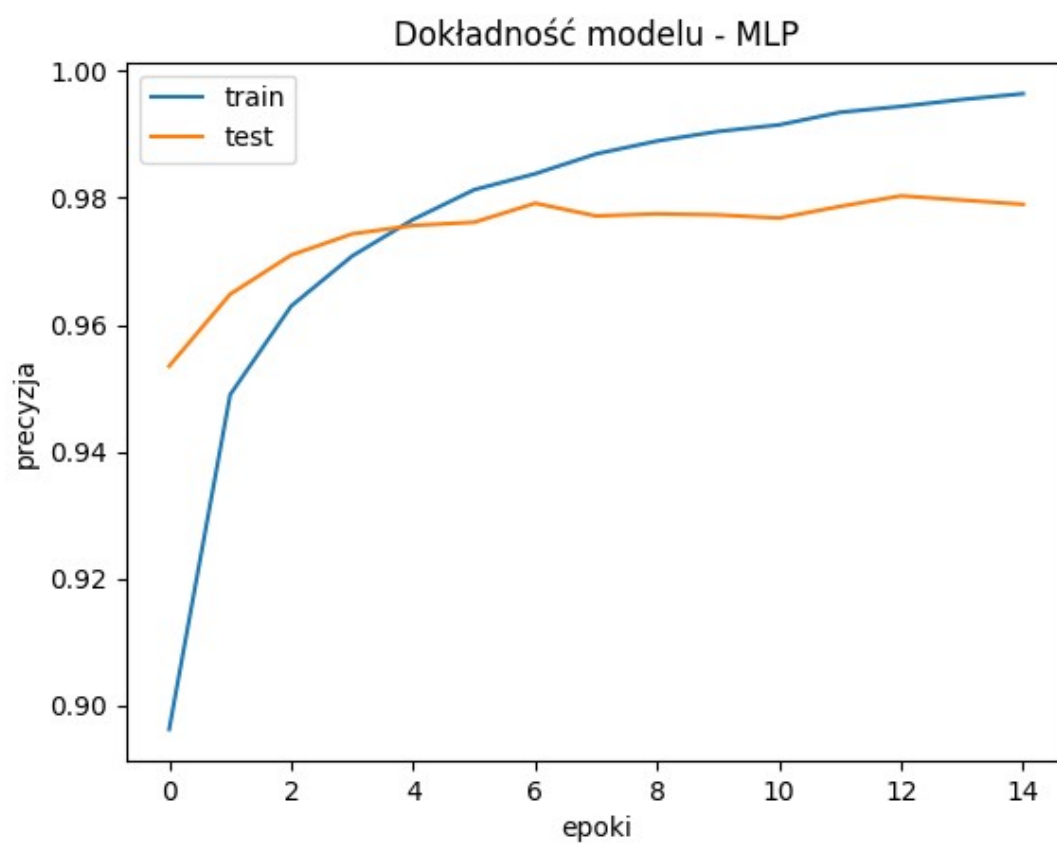
Liczba epok - 10



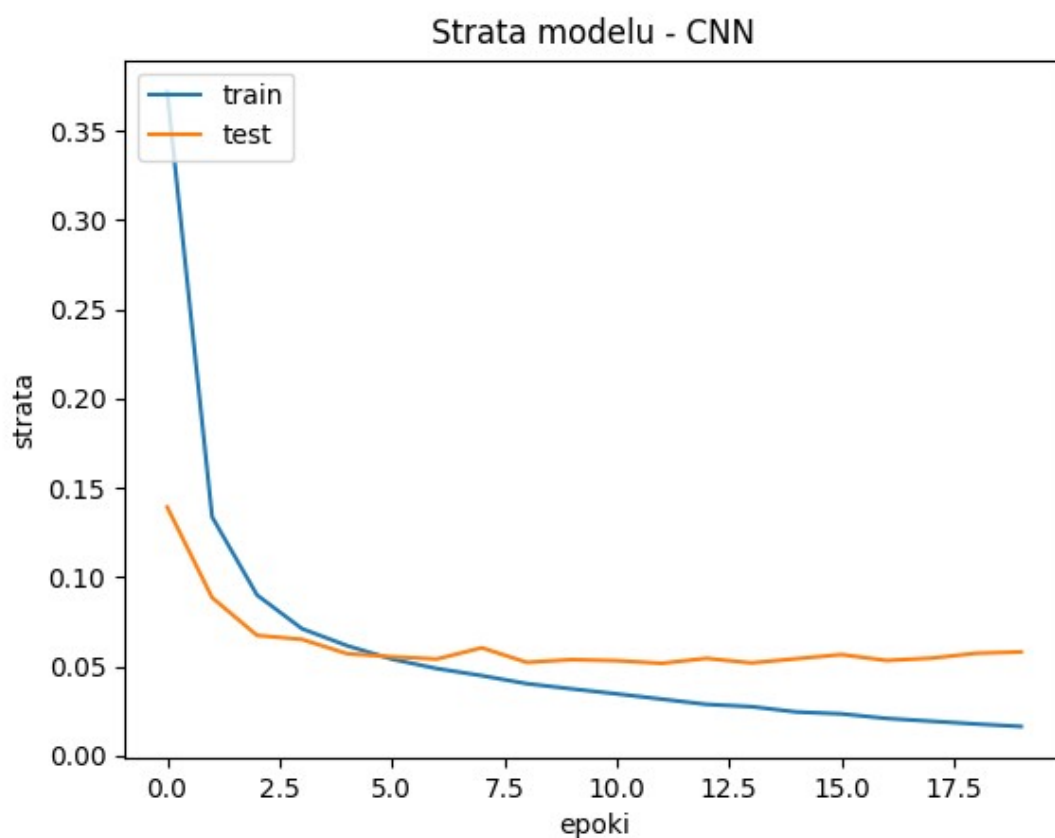
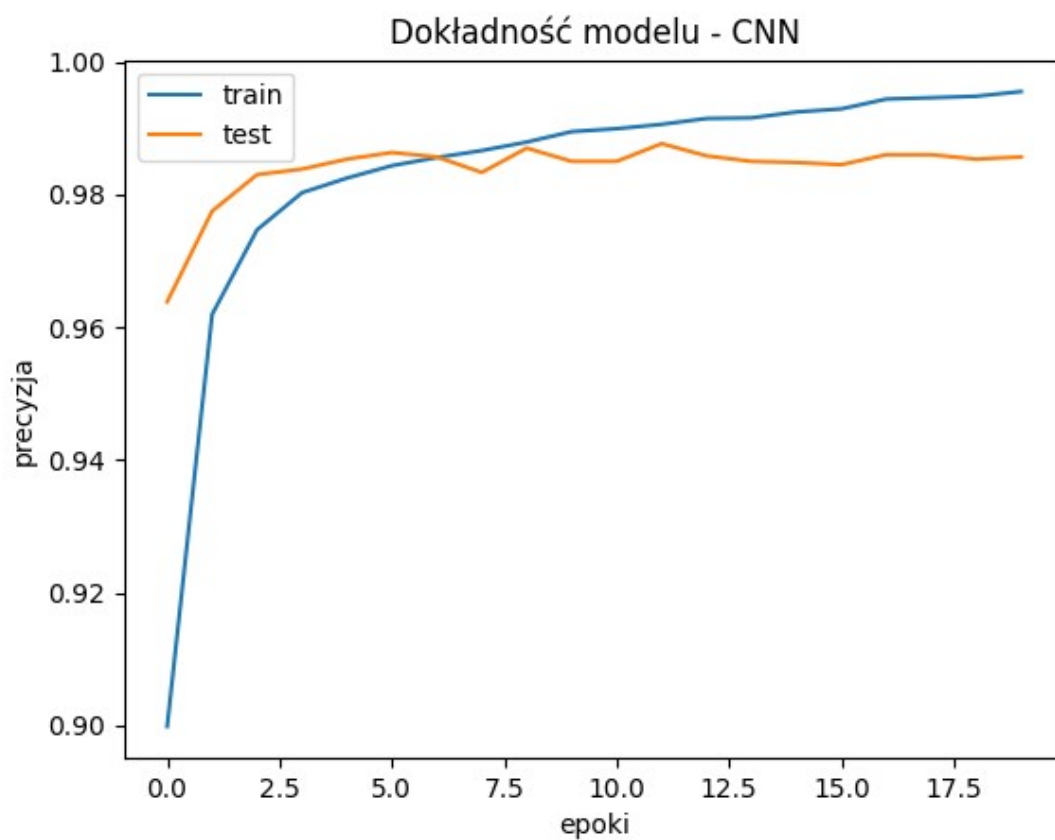


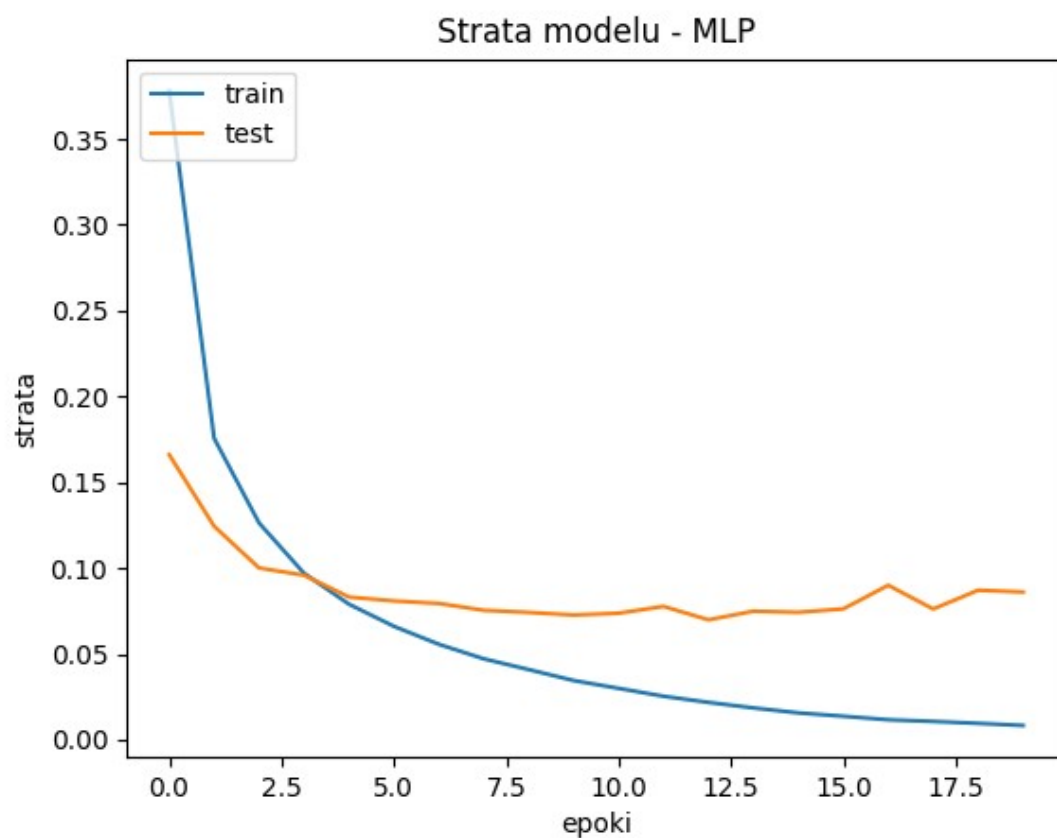
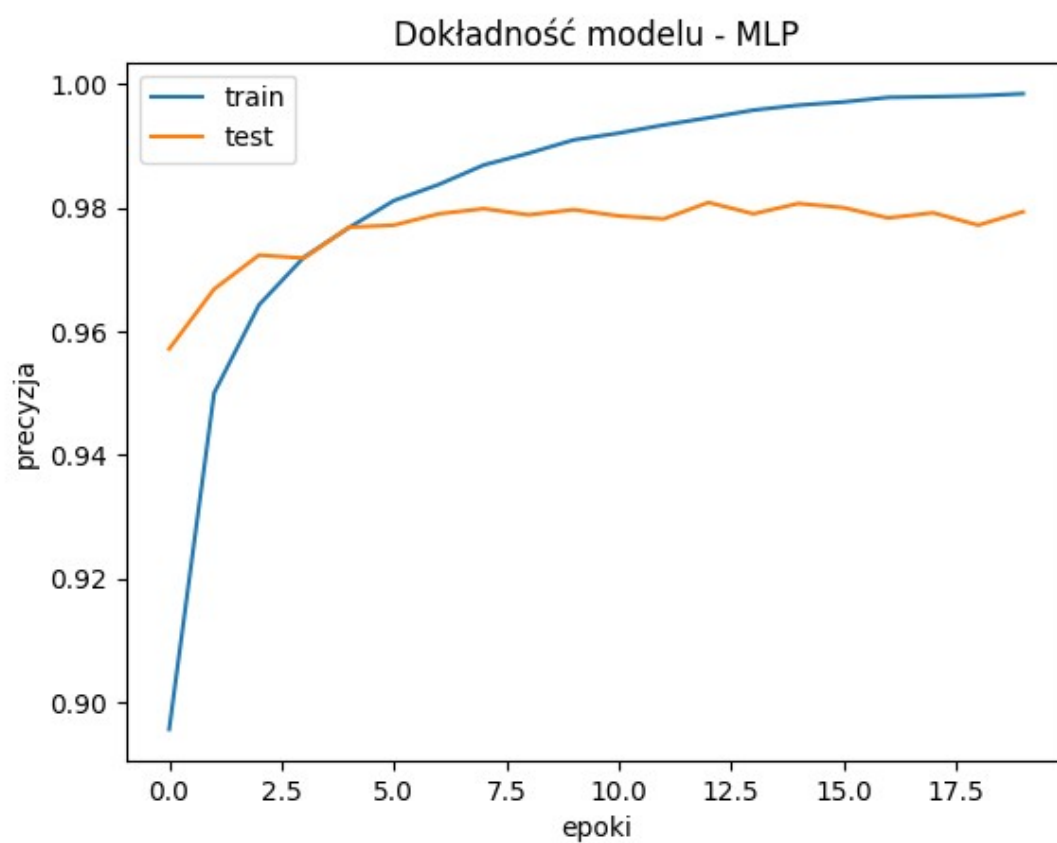
Liczba epok - 15





Liczba epok - 20



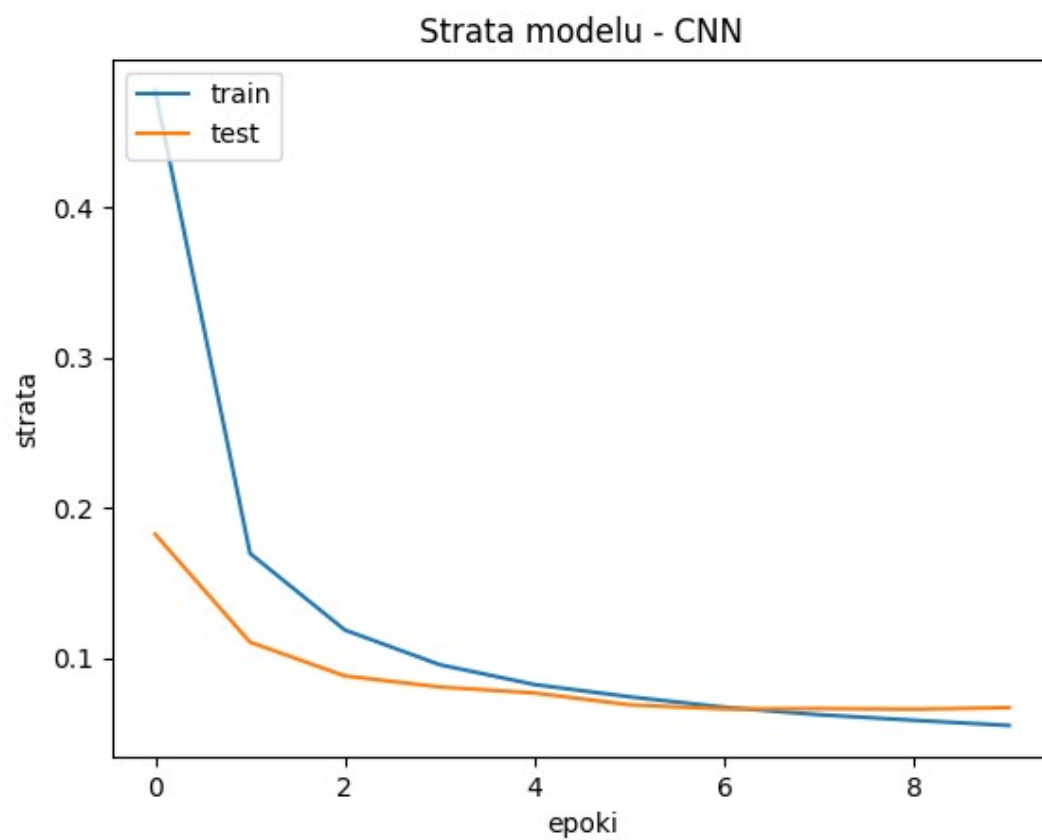
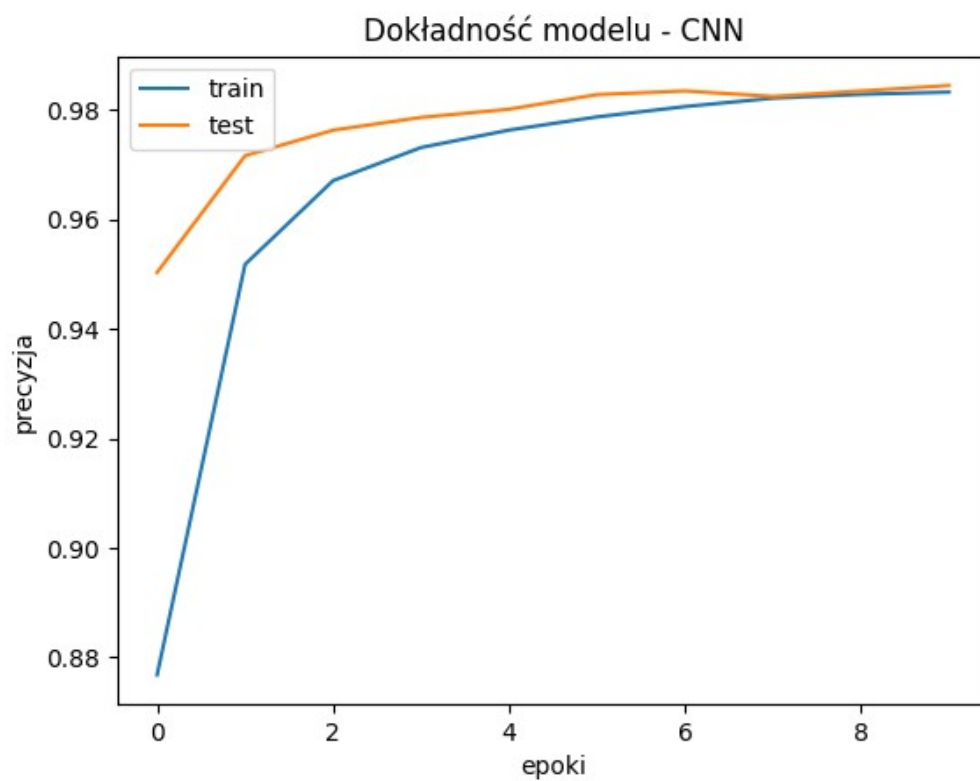


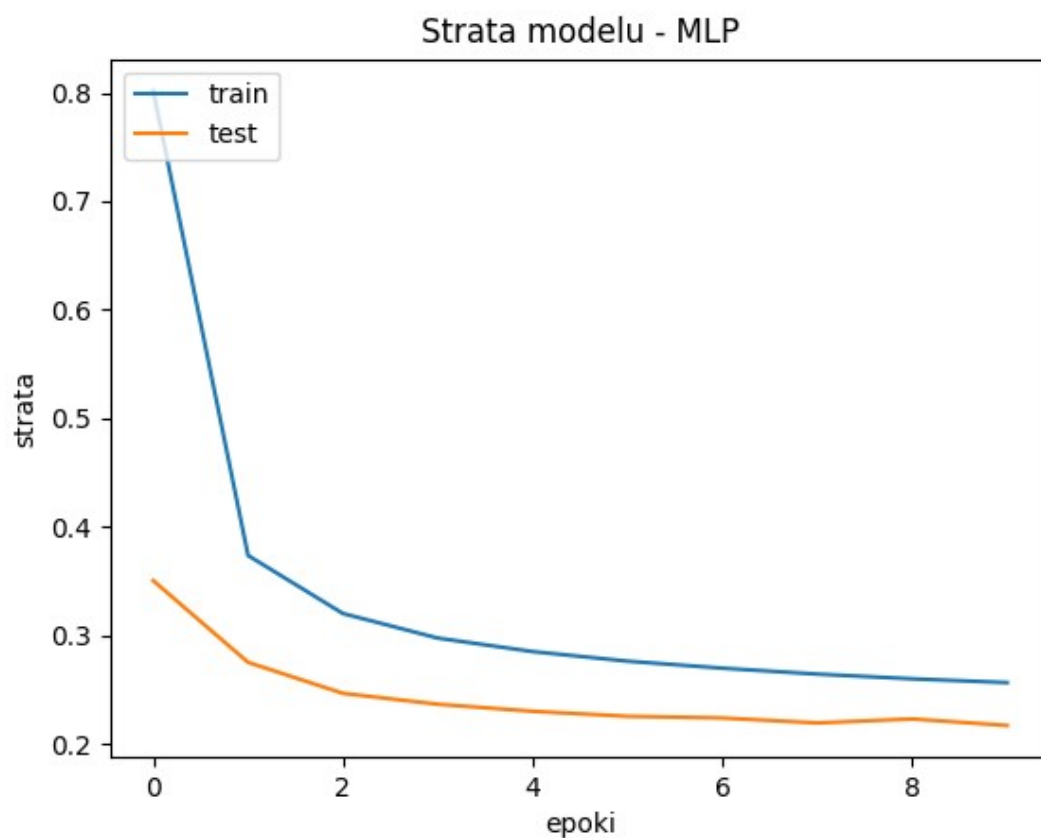
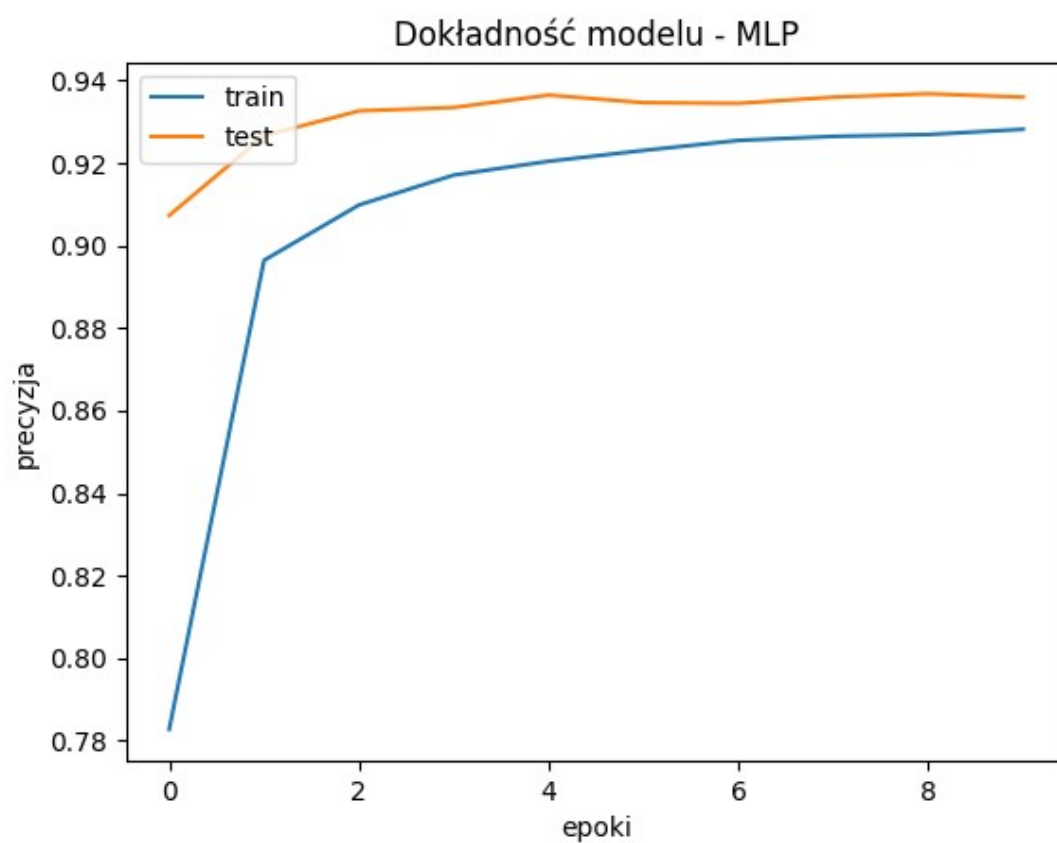
Wnioski

Jak można zauważyć, dla ostatniej wartości liczby epok (20), oba modele podlegają zjawisku overfittingu, które pojawia się w podobnym momencie (około piątej epoki). Warto jednak zauważyć, że sieć CNN zdecydowanie szybciej osiąga próg skuteczności rzędu 98%. Dodatkowo, dokładność modelu na zbiorze testowym w przypadku CNN wydaje się oscylować w okolicach 99%, natomiast dla MLP w okolicach 98% co jest, nieznacznie ale jednak, poprawą.

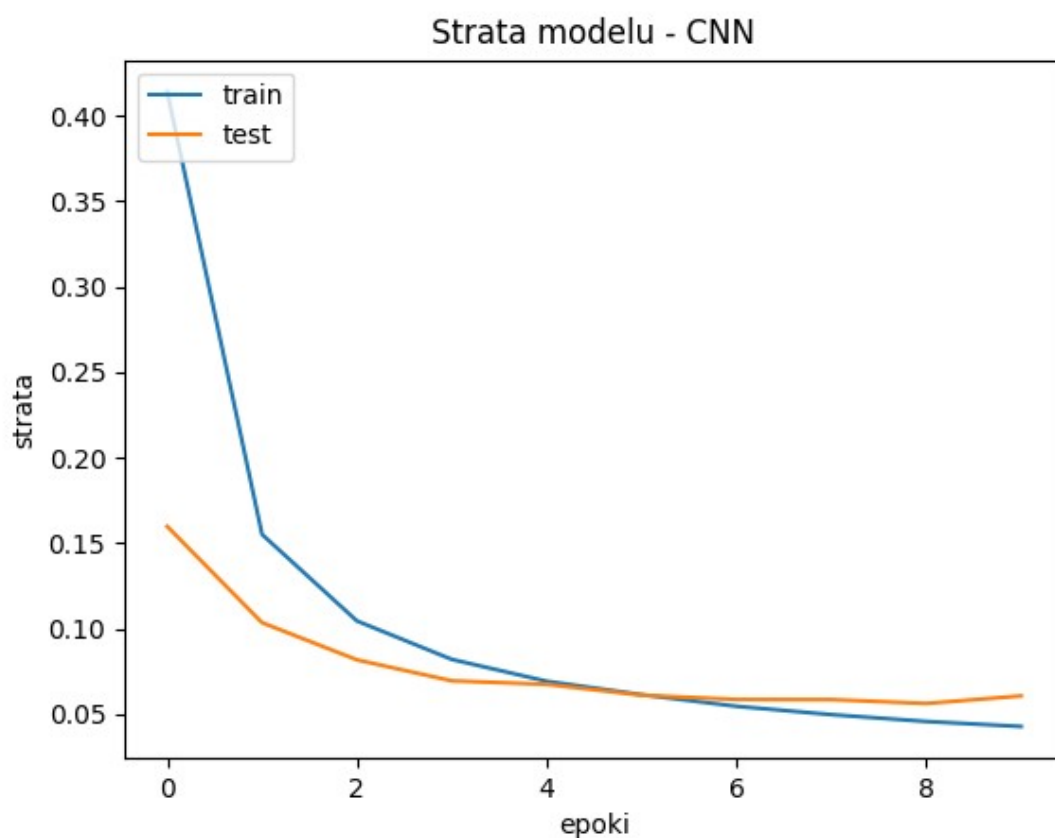
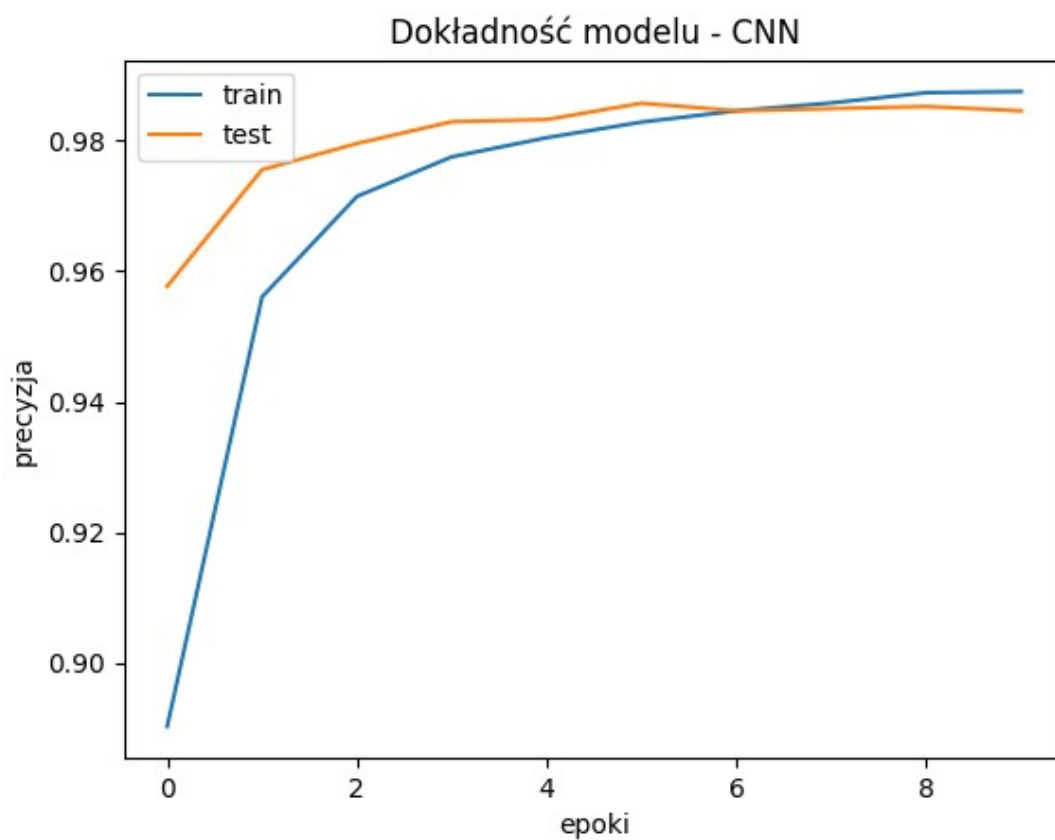
Liczba neuronów / filtrów

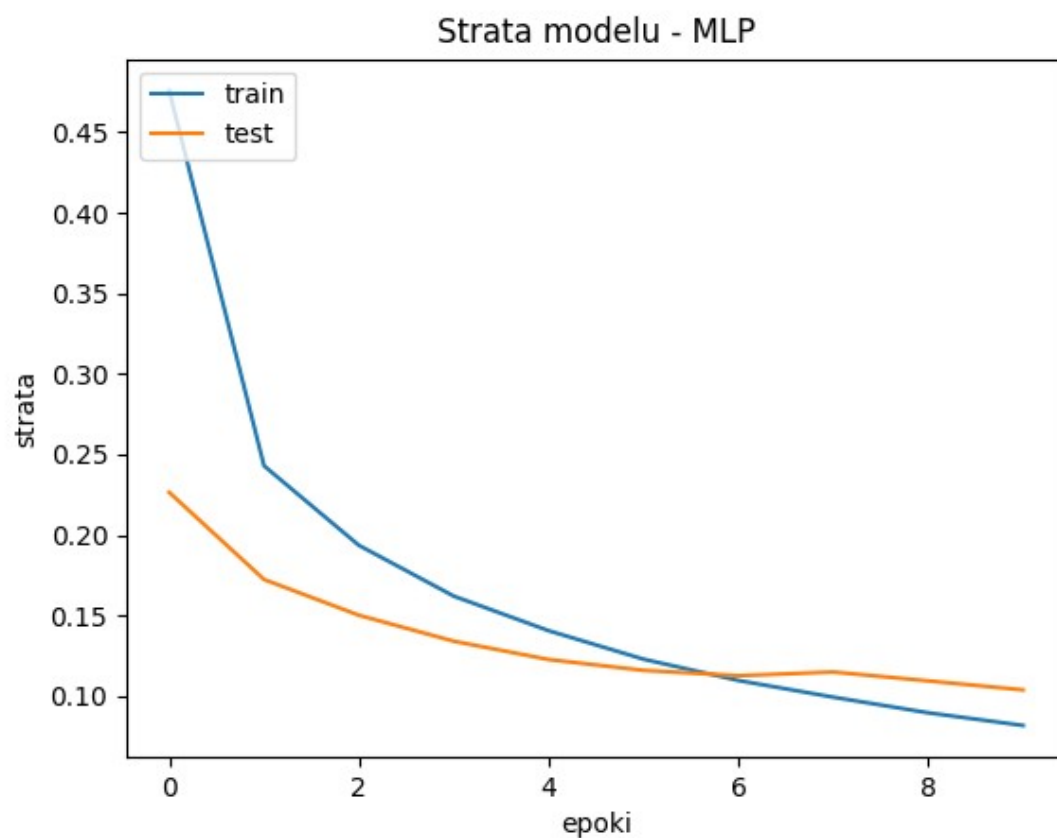
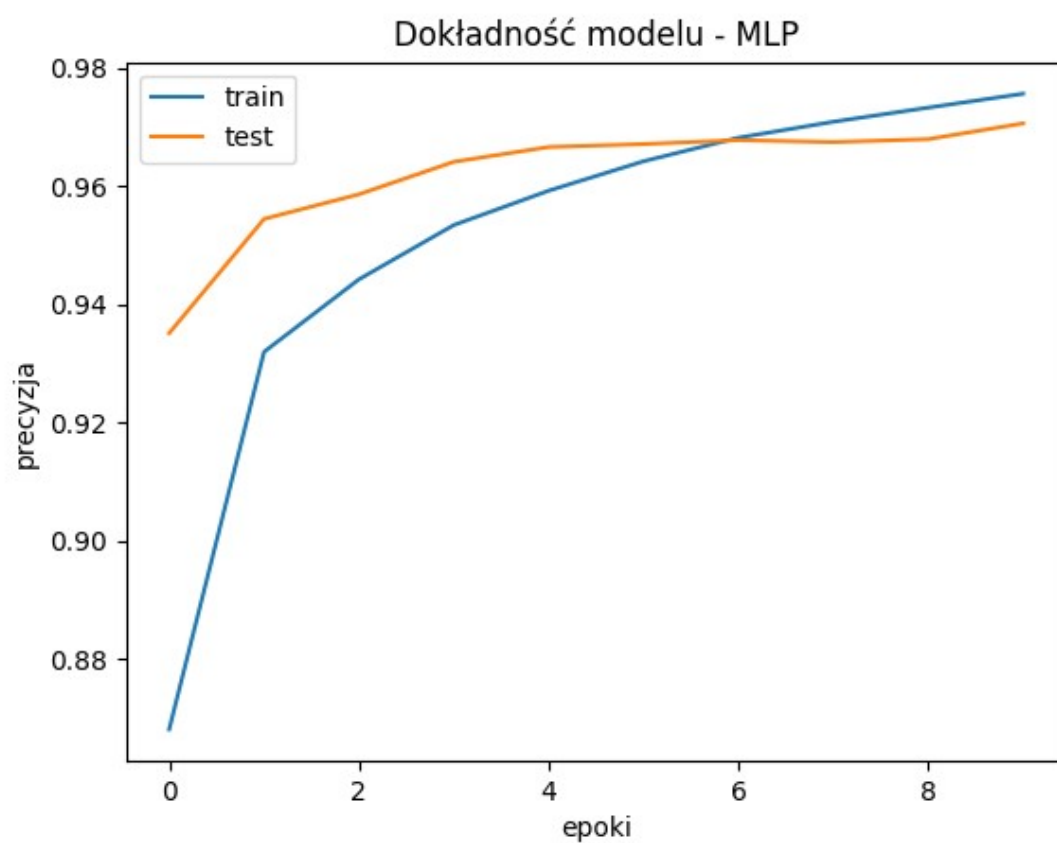
Liczba neuronów / filtrów – 10/10



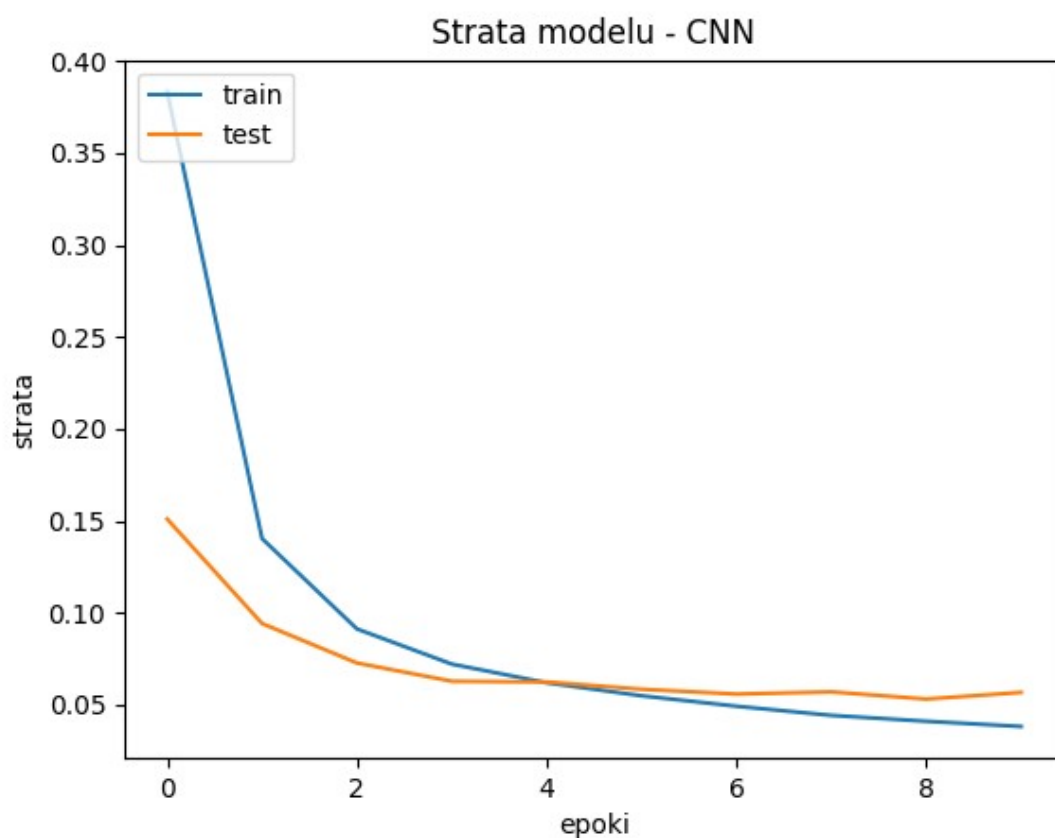
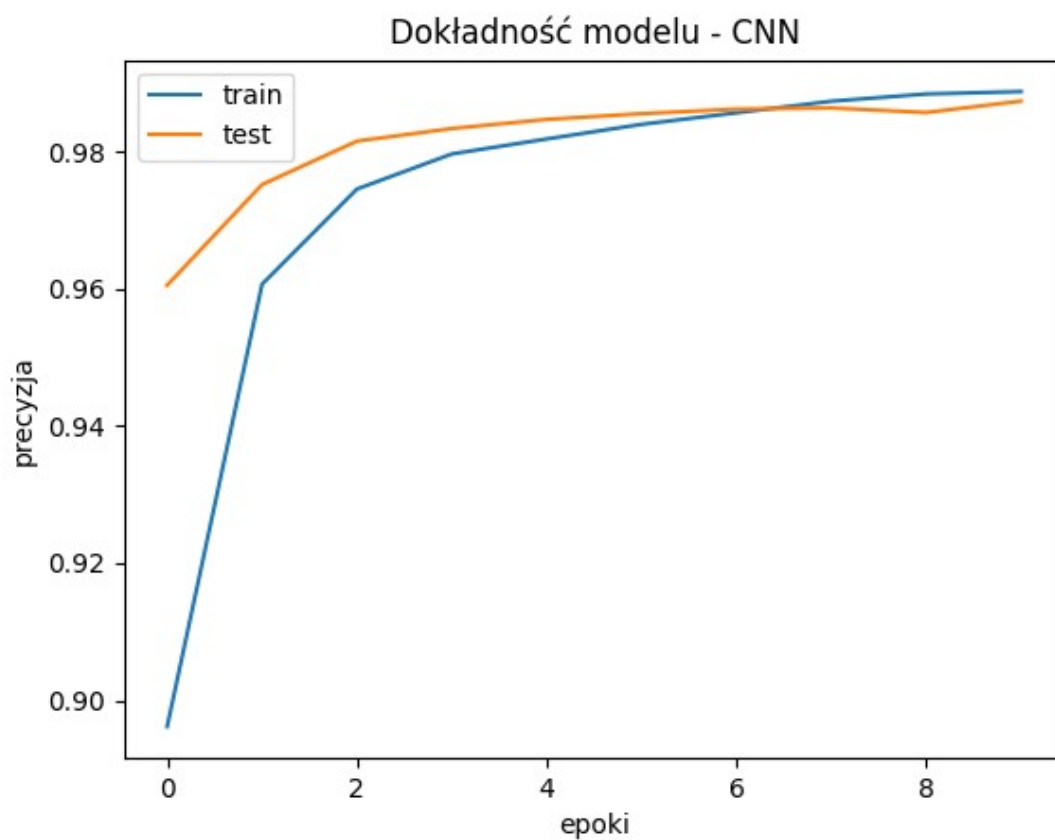


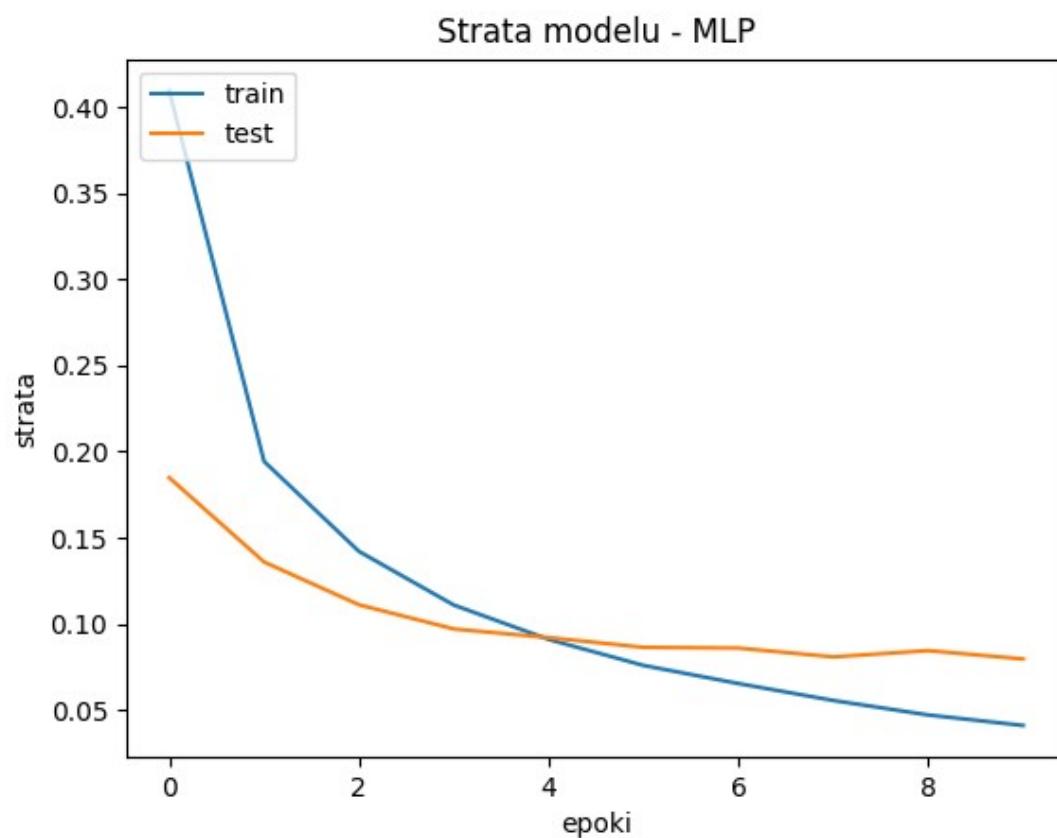
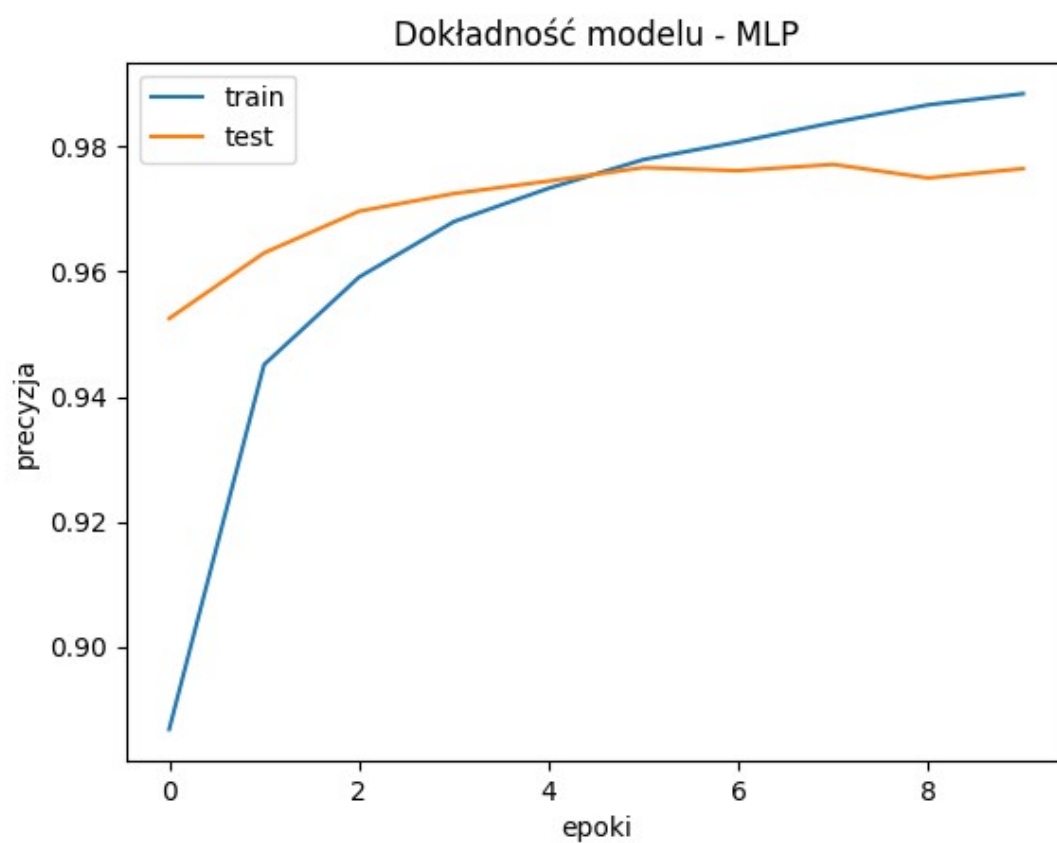
Liczba neuronów / filtrów - 50/20



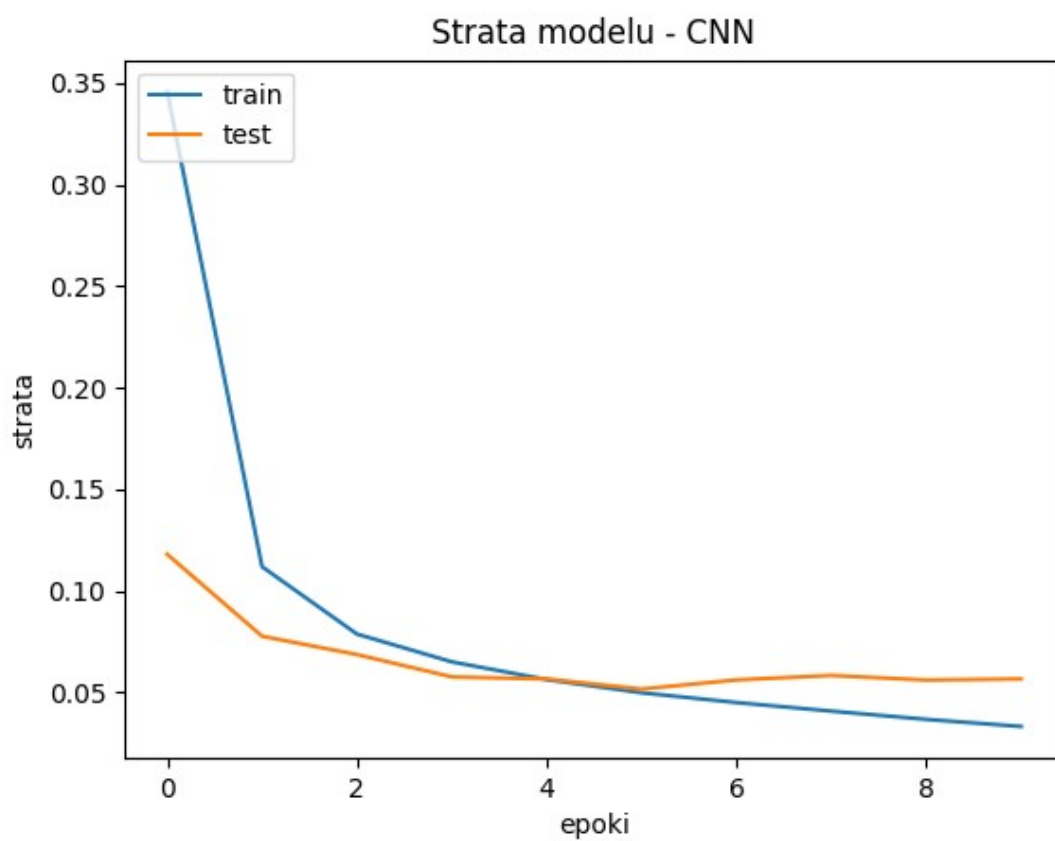
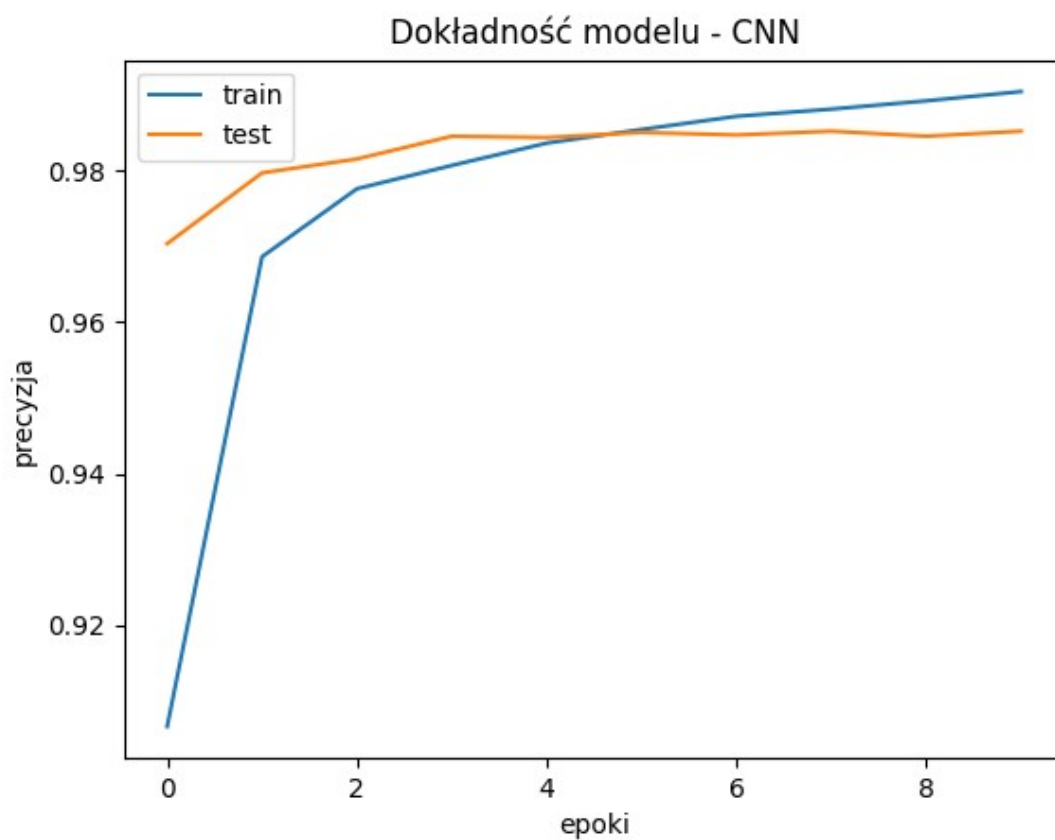


Liczba neuronów / filtrów - 100/30

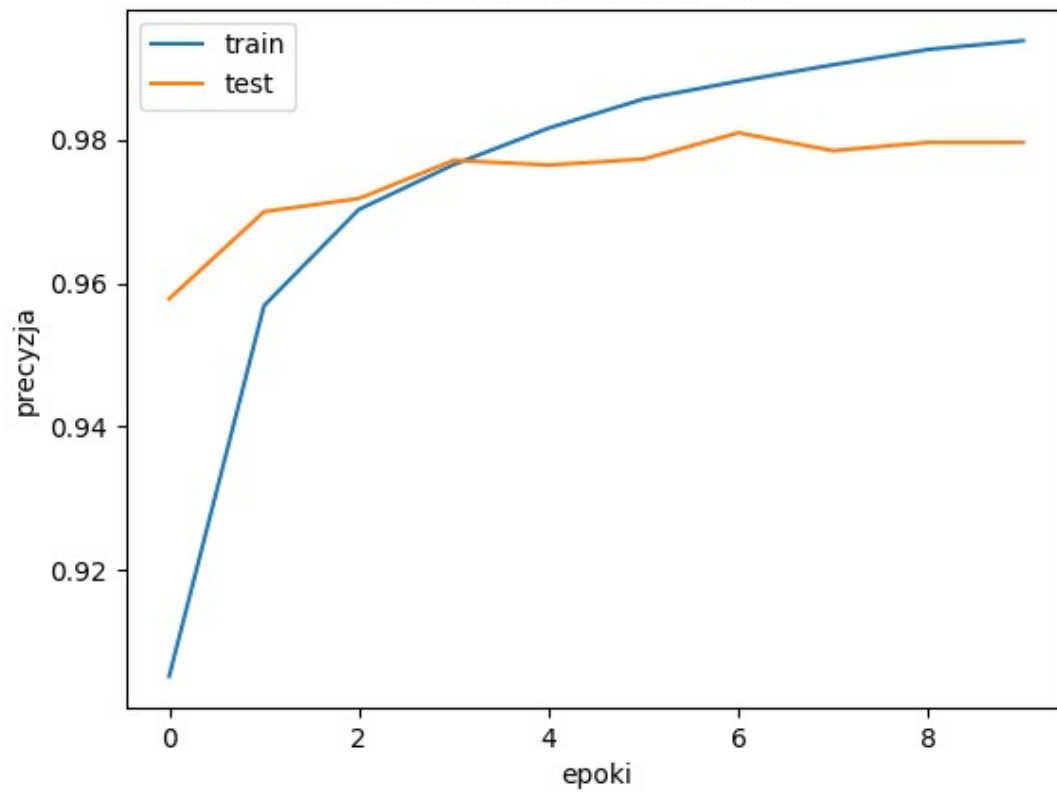




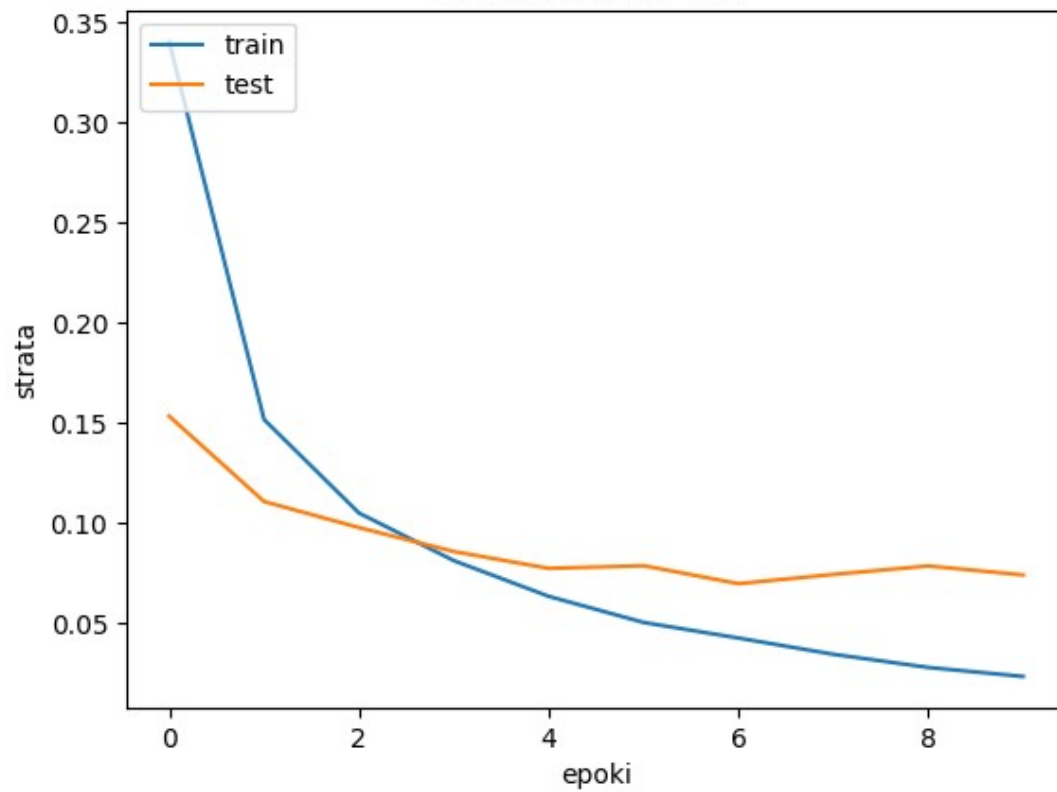
Liczba neuronów / filtrów - 200/40



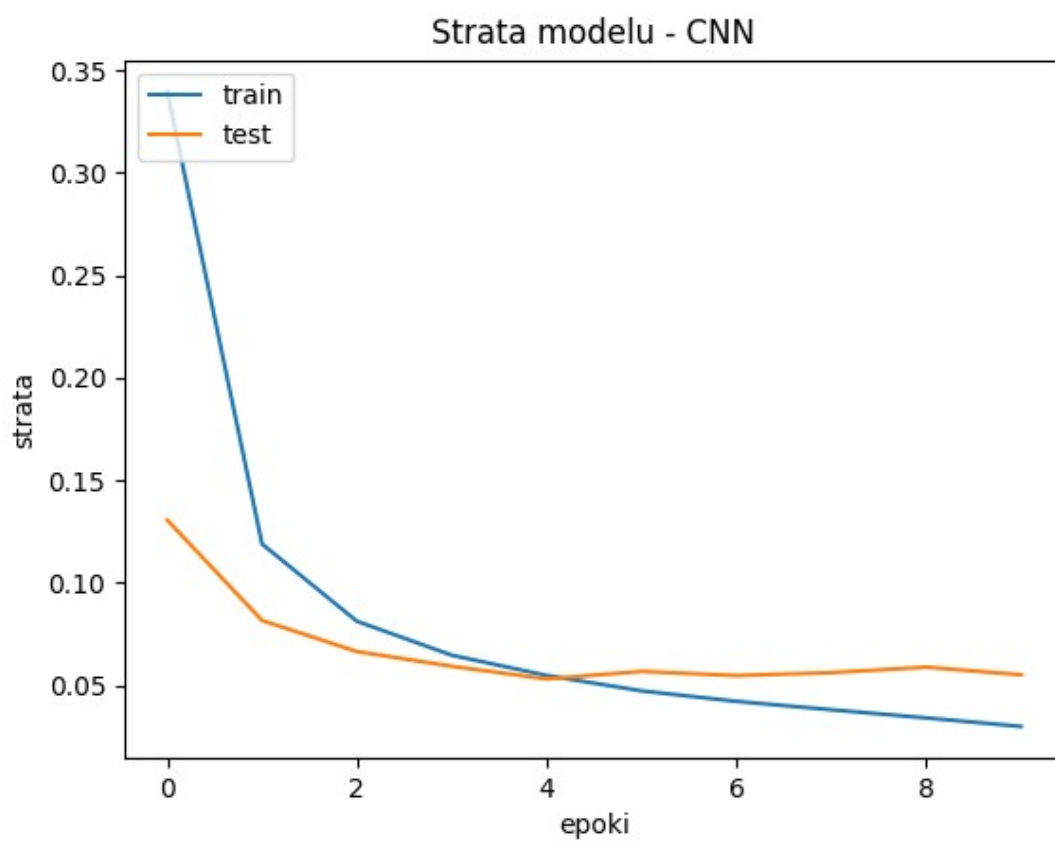
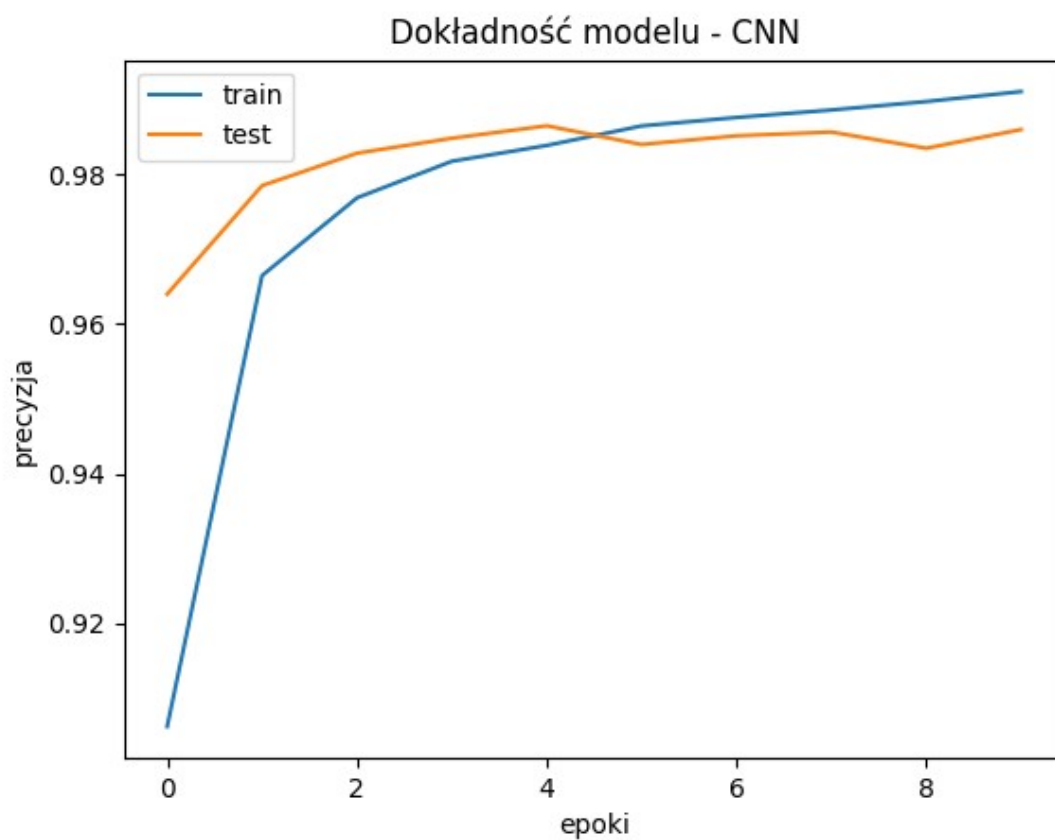
Dokładność modelu - MLP

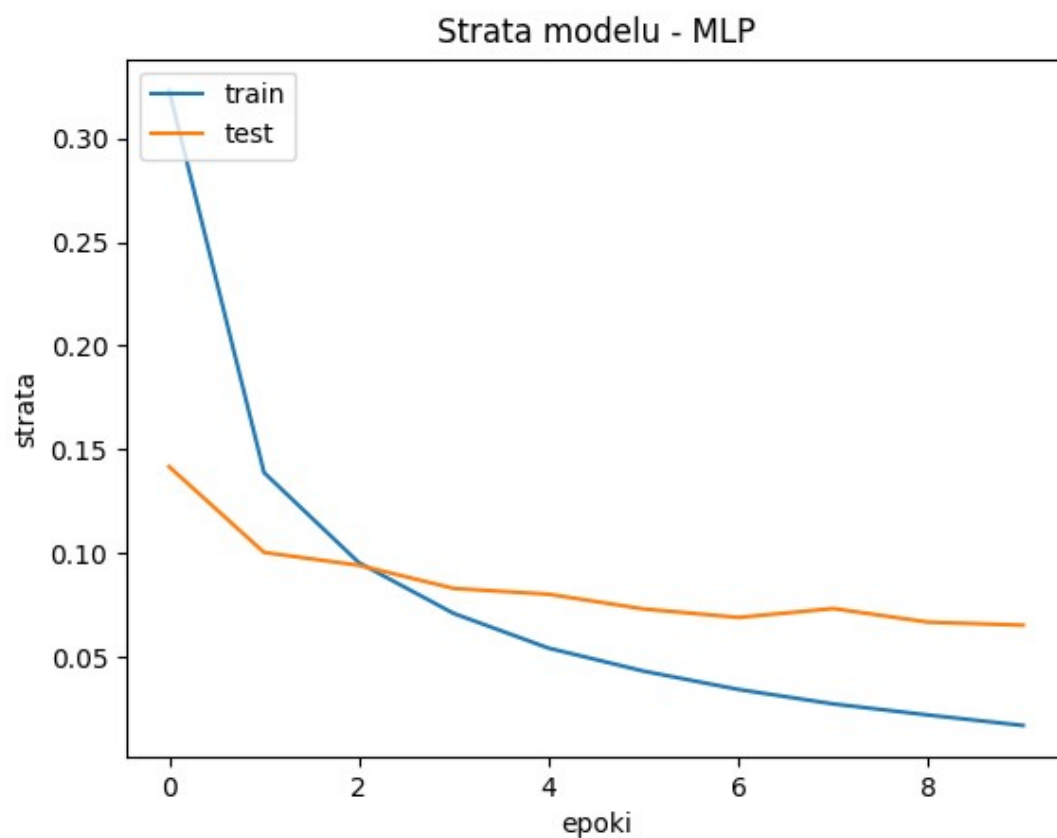
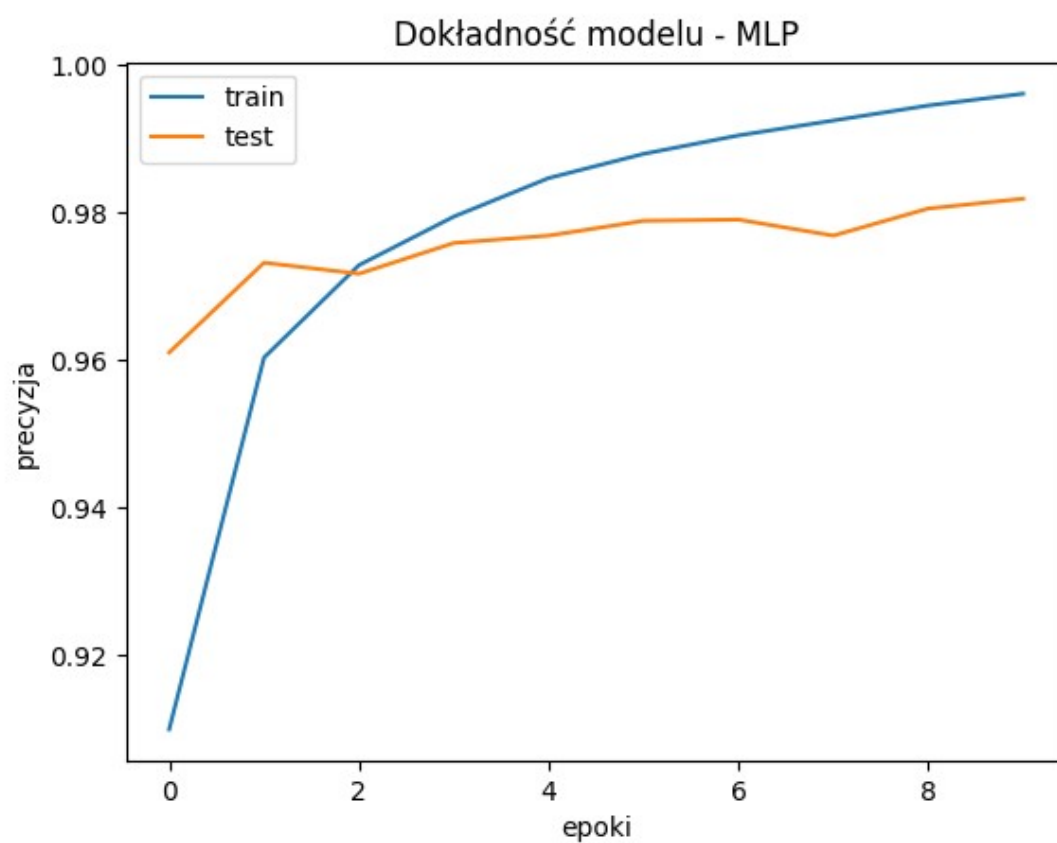


Strata modelu - MLP



Liczba neuronów / filtrów - 250/50





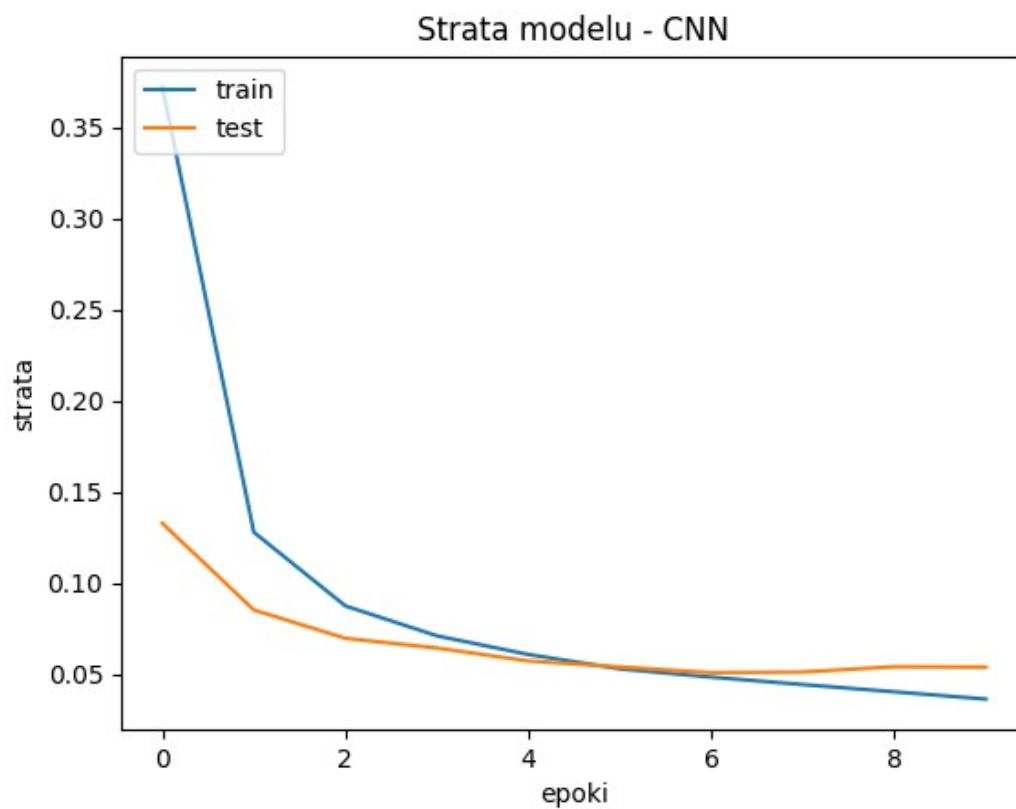
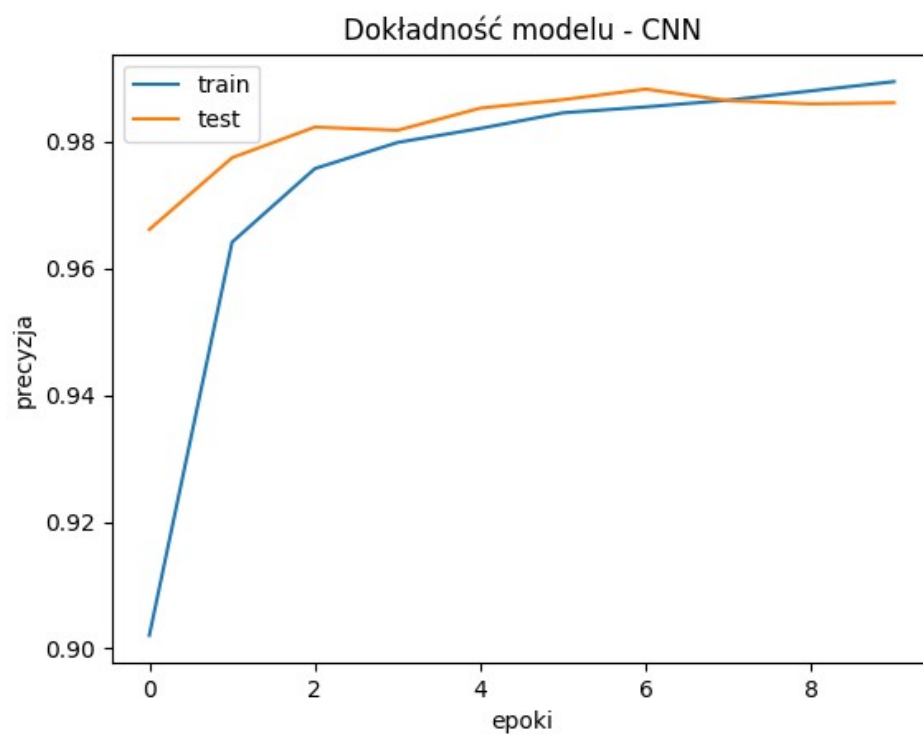
Wnioski

Zwiększanie liczby filtrów w przypadku sieci CNN wydaje się mieć dużo mniejsze znaczenie niż zwiększanie liczby neuronów dla sieci MLP. Skuteczność dla 10 filtrów CNN, jest osiągana przez sieć MLP dopiero przy wykorzystaniu 100 neuronów. Dalsze zwiększanie liczby neuronów nie prowadzi jednak do podniesienia skuteczności modelu (jedynie do szybciej występującego i większego overfittingu). By osiągnąć podobne wyniki prawdopodobnie należałoby dodać dodatkowe warstwy do sieci MLP, tak by mogła ona 'wychwycić' pewne wzorce z obrazów. Wiązałoby się to oczywiście z dużo większą złożonością obliczeniową. Sieć konwolucyjna natomiast jest stworzona do tego typu zadań (klasyfikacji obrazów) ze względu na wykorzystanie filtrów. Skutkuje to dużo szybszym i wydajniejszym procesem nauki.

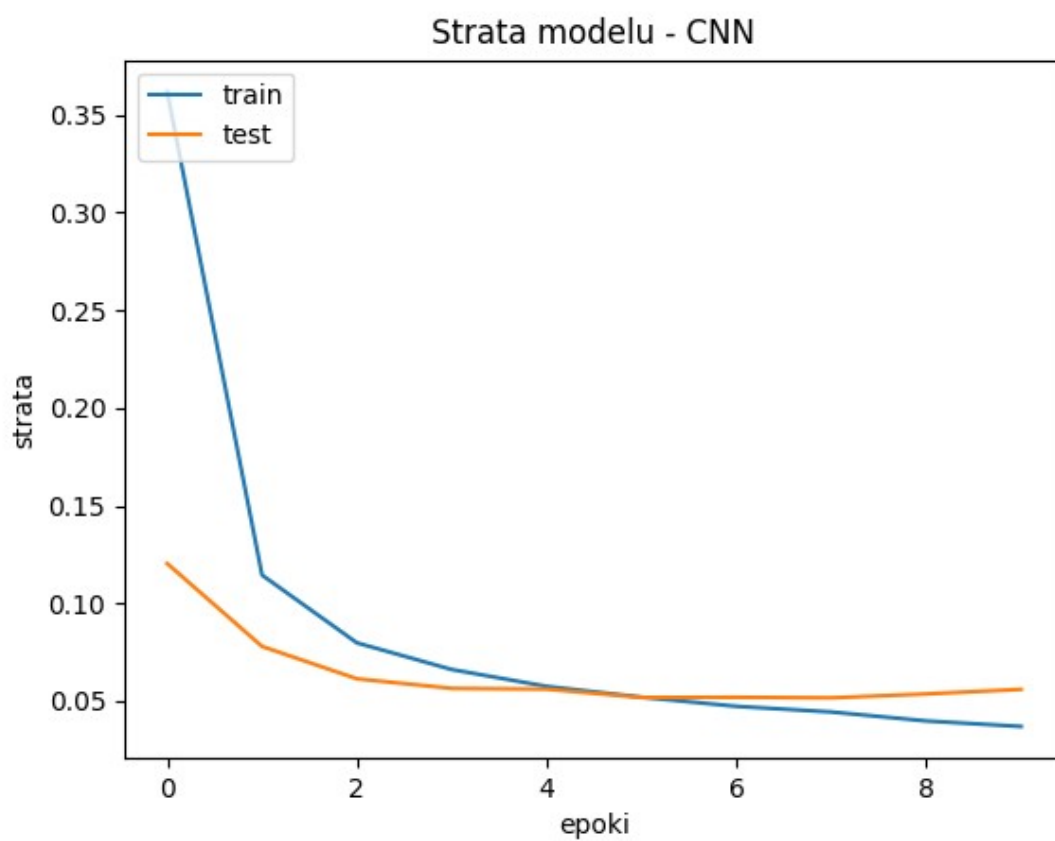
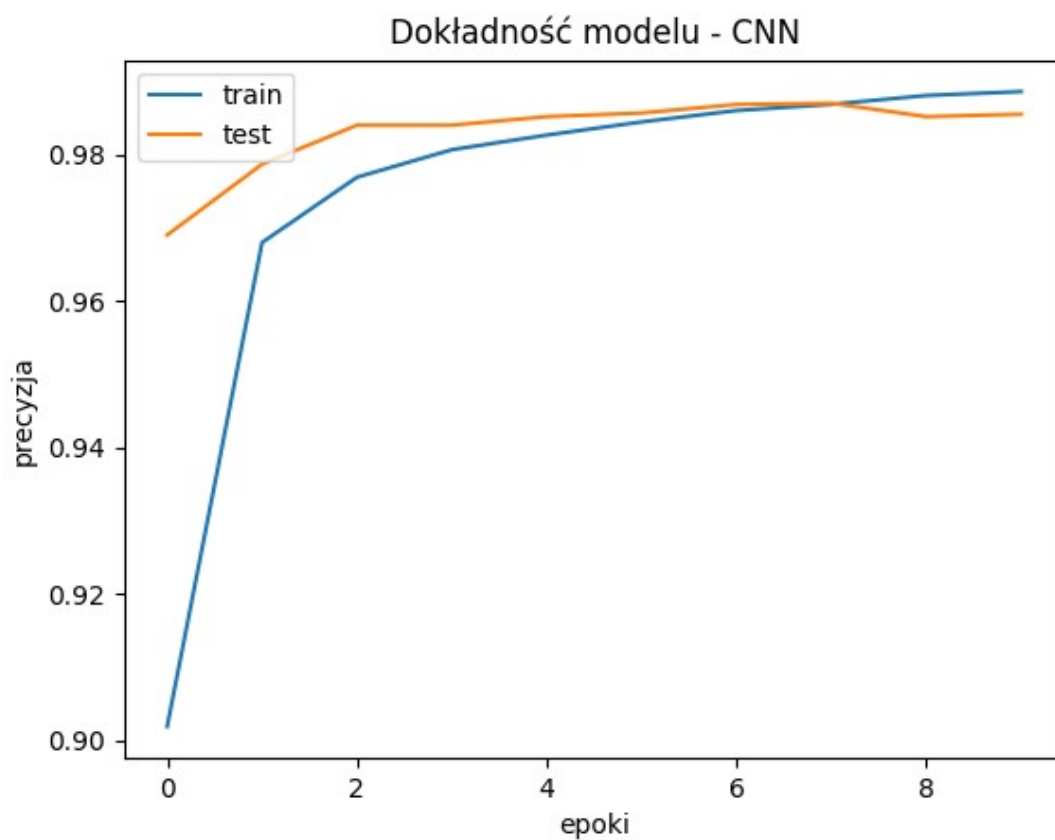
Wpływ rozmiaru okna pooling'u oraz rodzaju pooling'u

Kwadratowe okno pooling'u

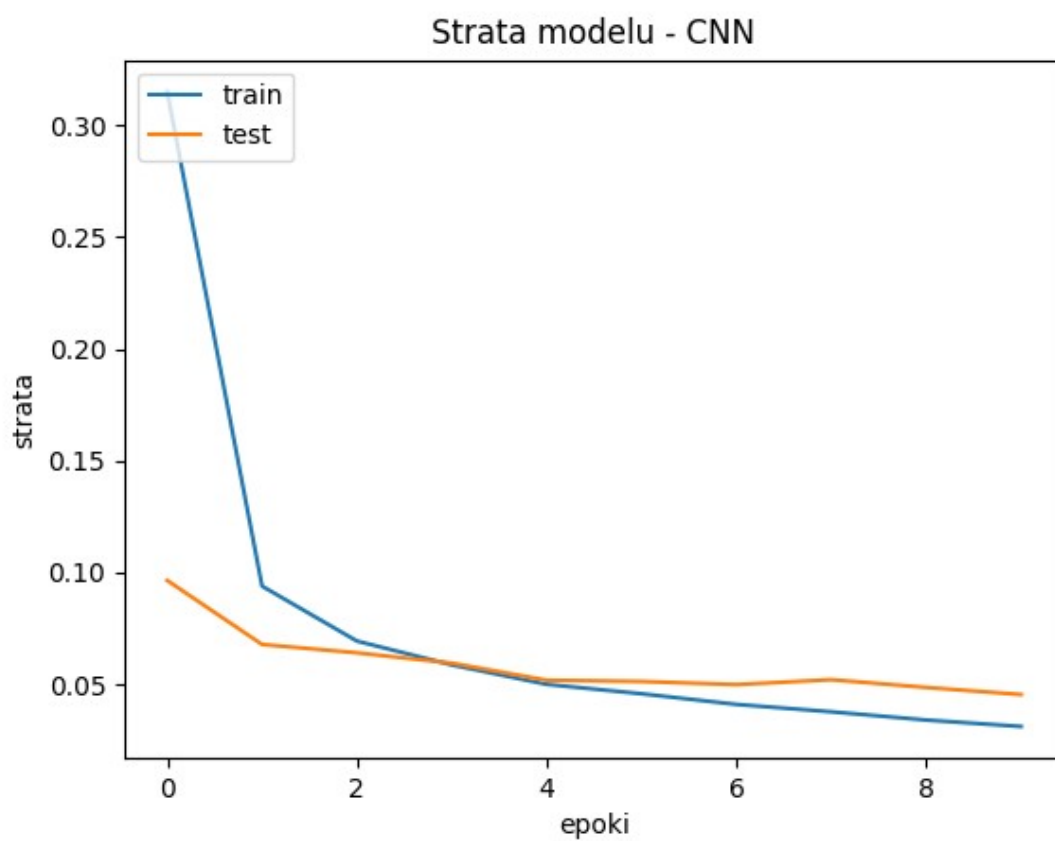
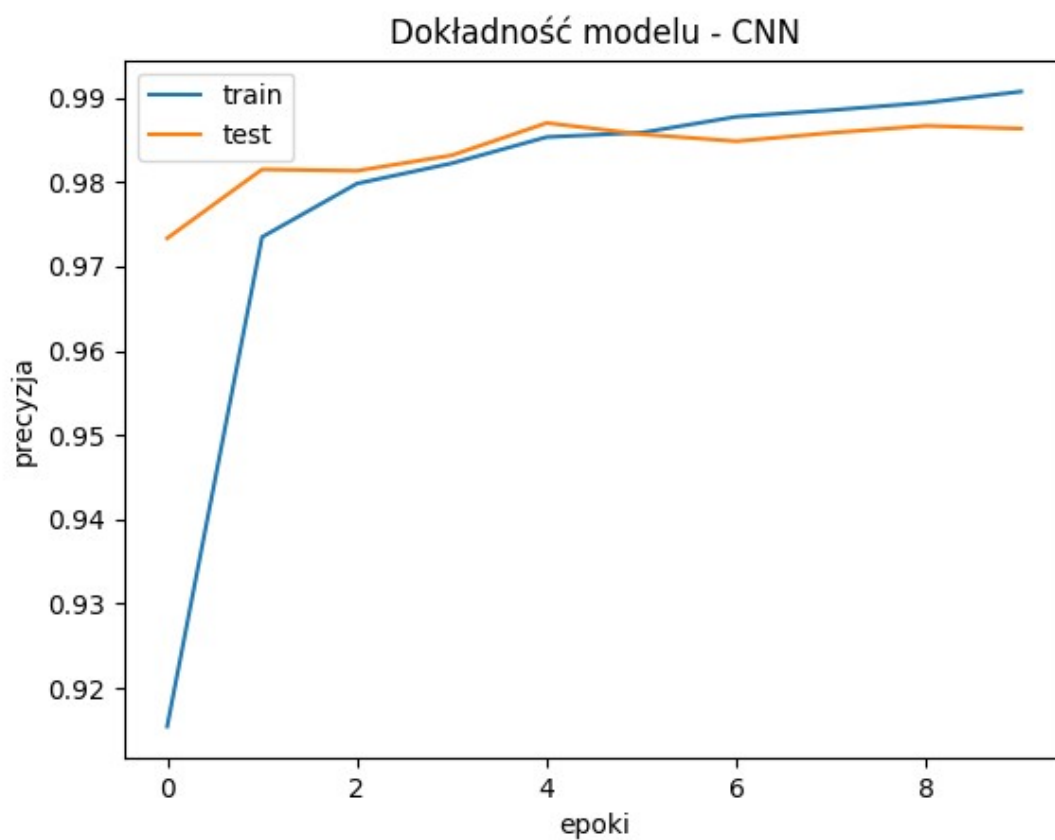
Okno pooling'u - 2x2



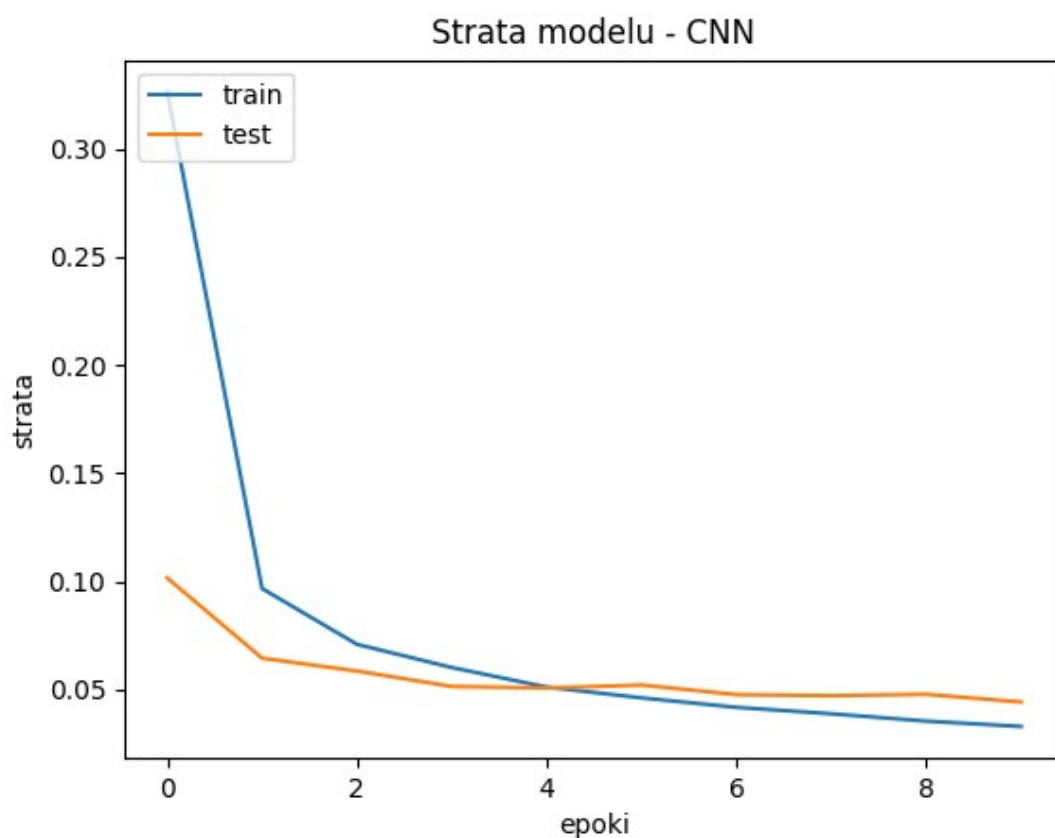
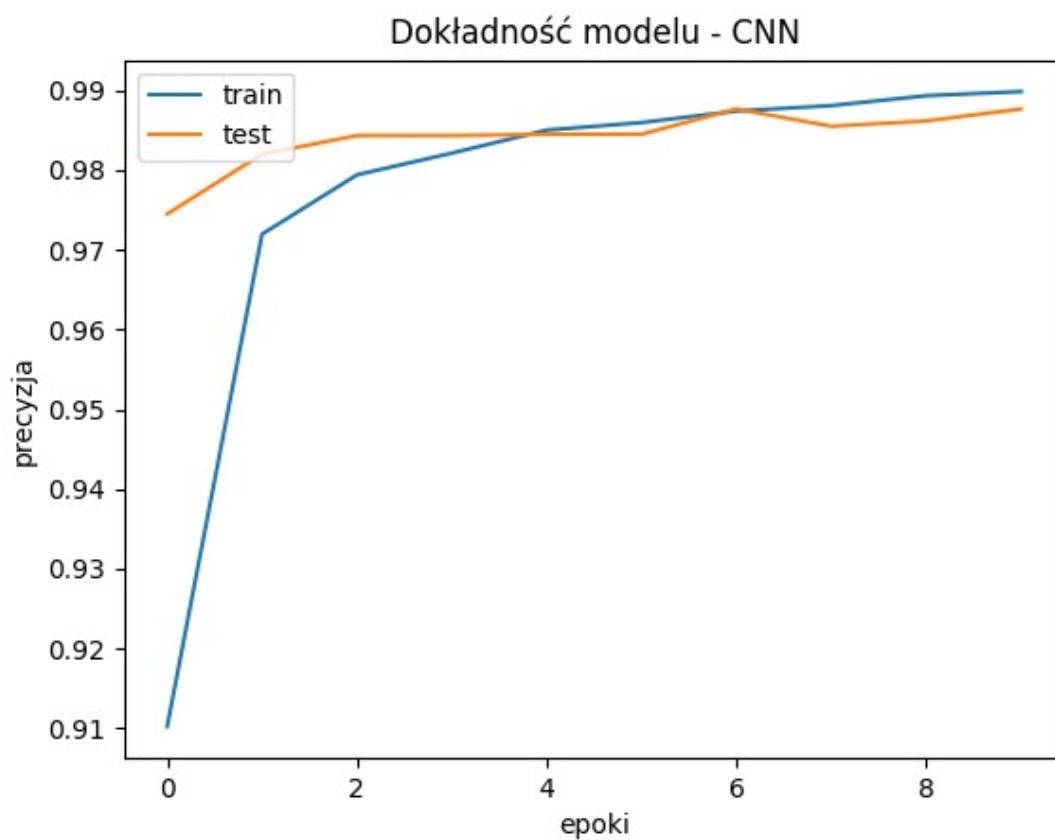
Okno pooling - 3x3



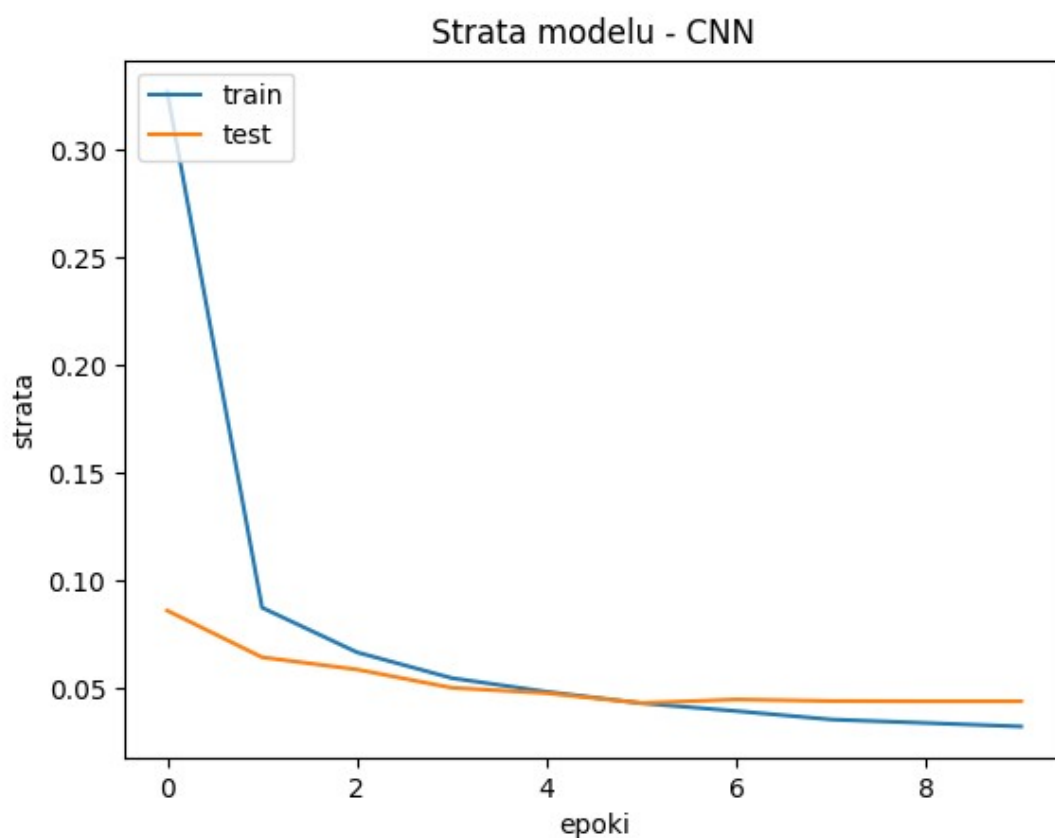
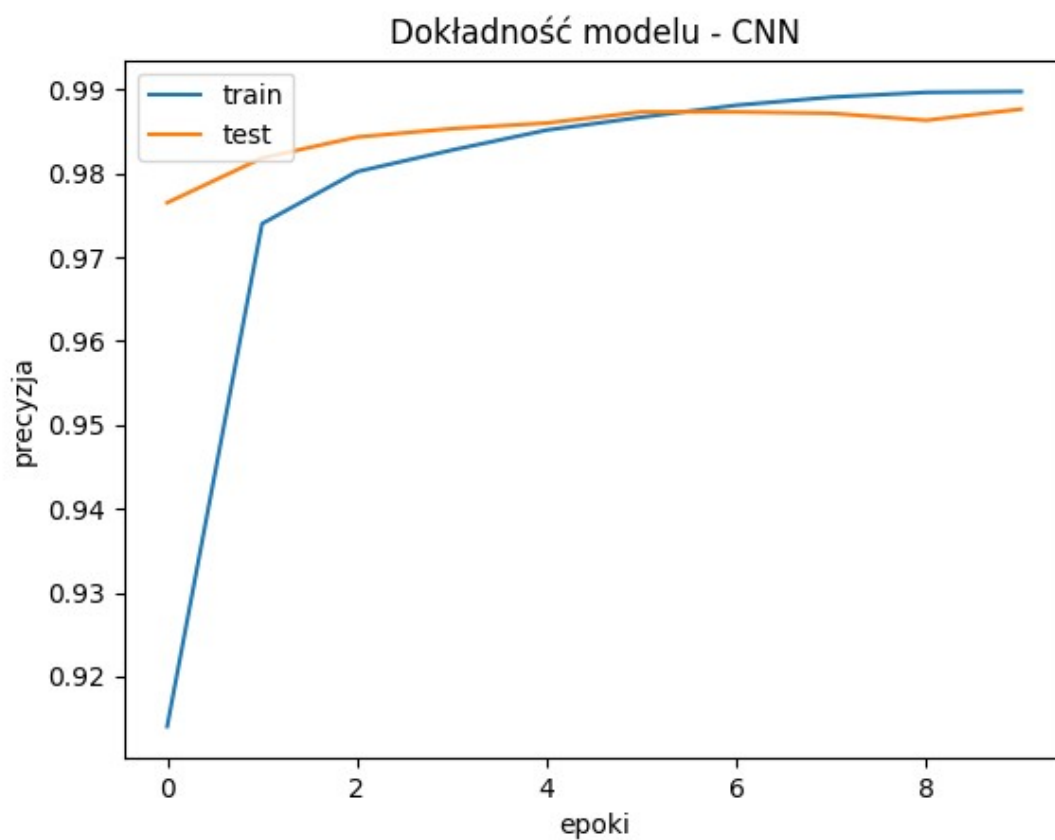
Okno pooling - 4x4



Okno pooling - 5x5



Okno pooling - 6x6

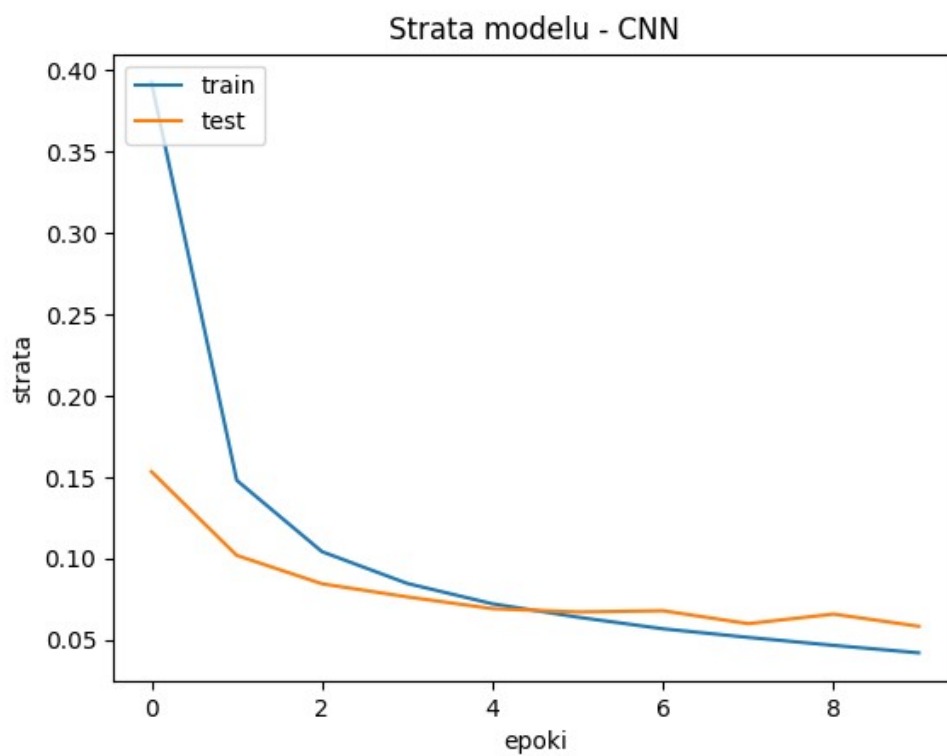
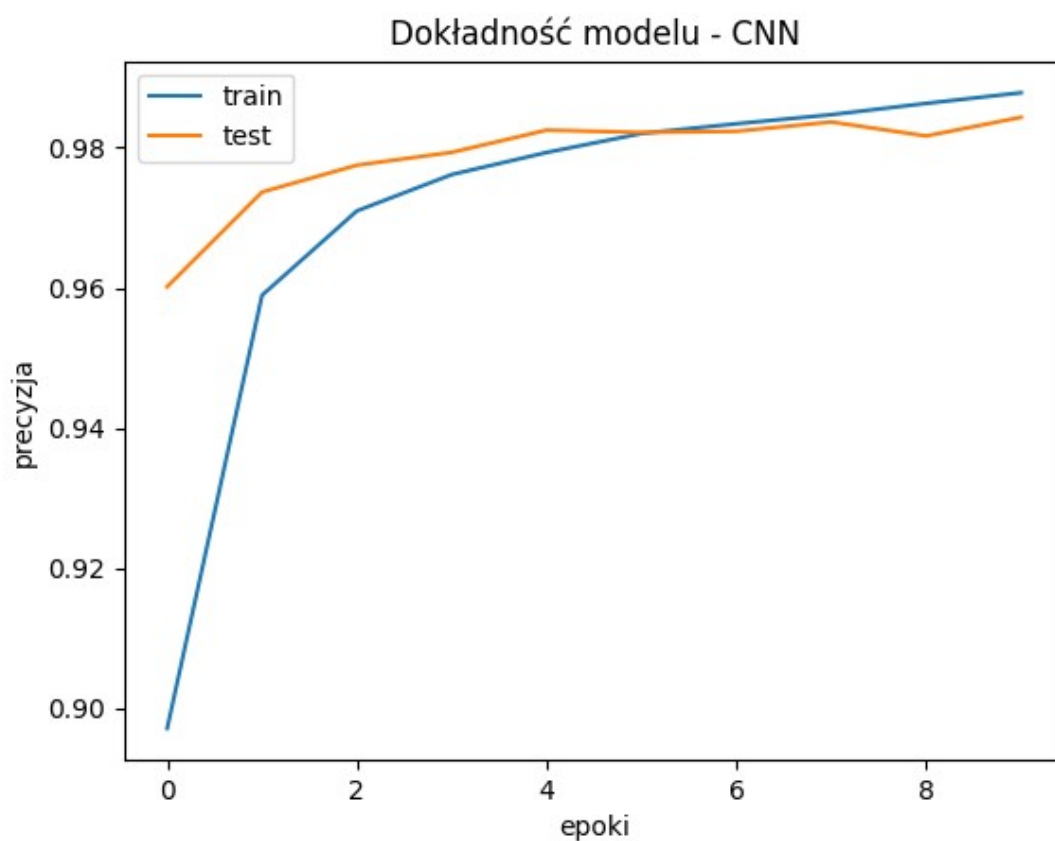


Wnioski

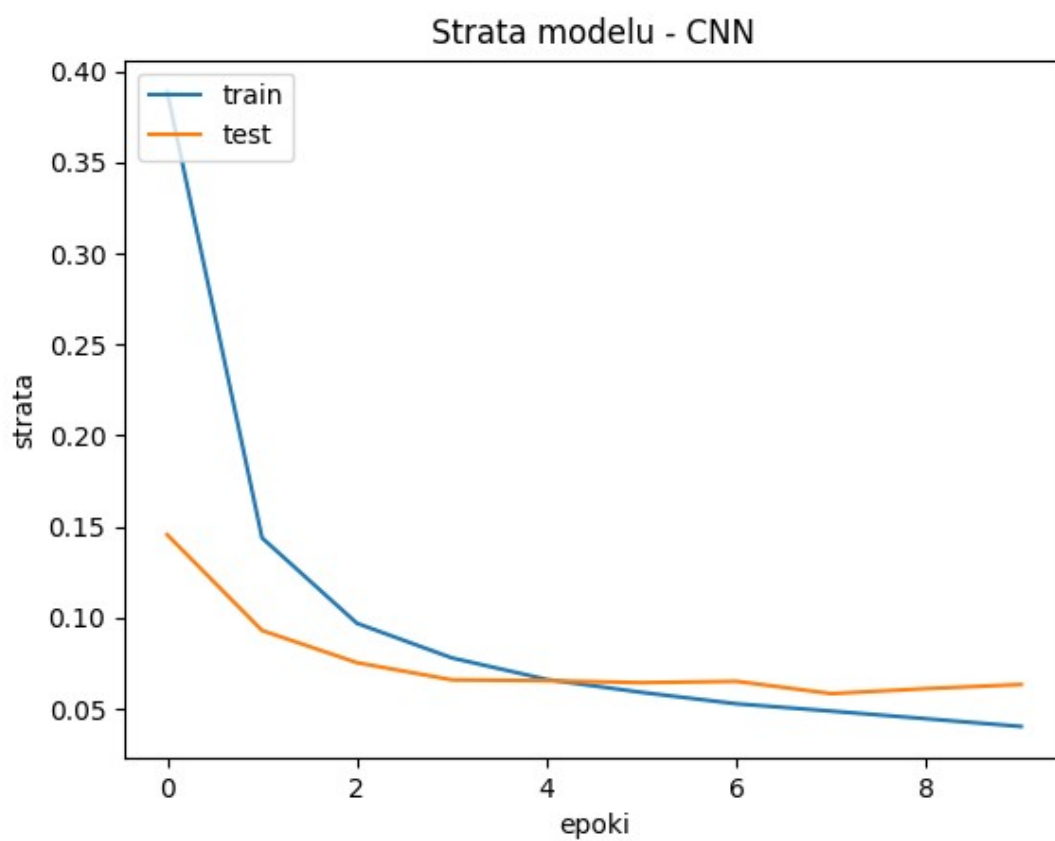
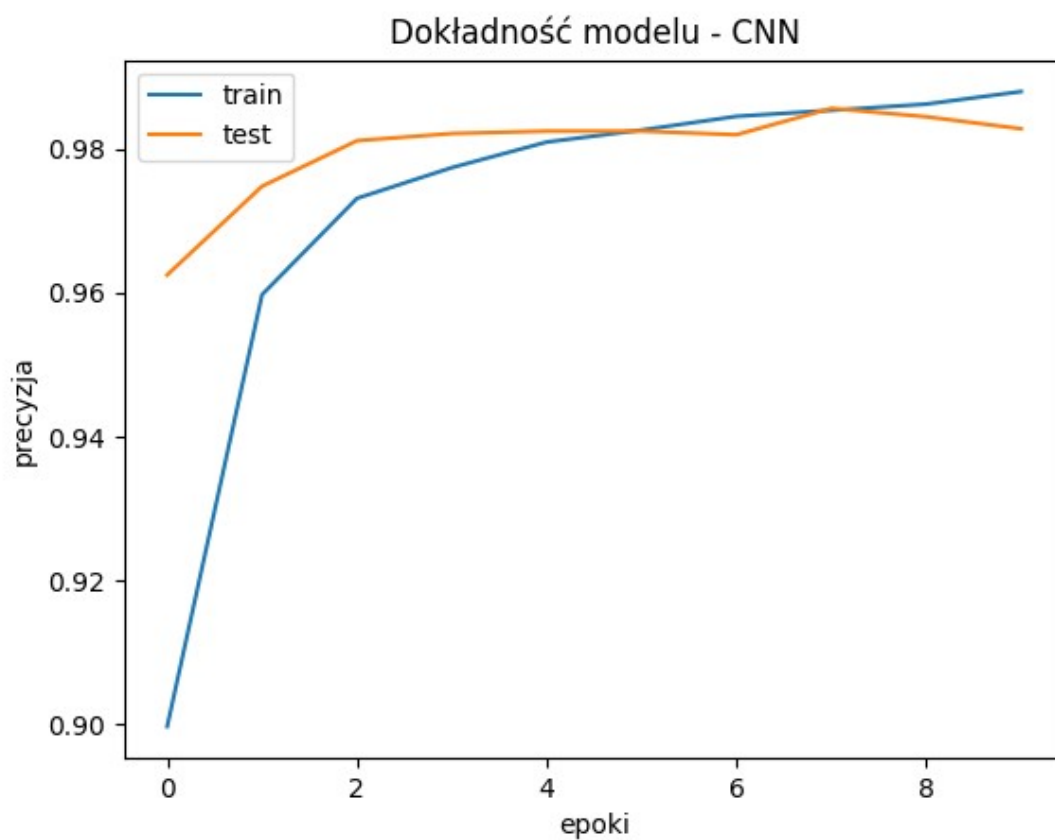
Niezależnie od rozmiaru kwadratowego okna pooling, skuteczność na zbiorze testowym wydaje się być bardzo podobna. Może to wynikać z samej postaci danych, ponieważ są to obrazy o stosunkowo niewielkiej rozdzielczości z wyraźnymi pikselami cyfr na czarnym tle. W związku z tym upscaling/downscaling wynikający z nakładania filtrów i pooling może nie być aż tak istotnym elementem w całym procesie nauki.

Prostokątne okno pooling

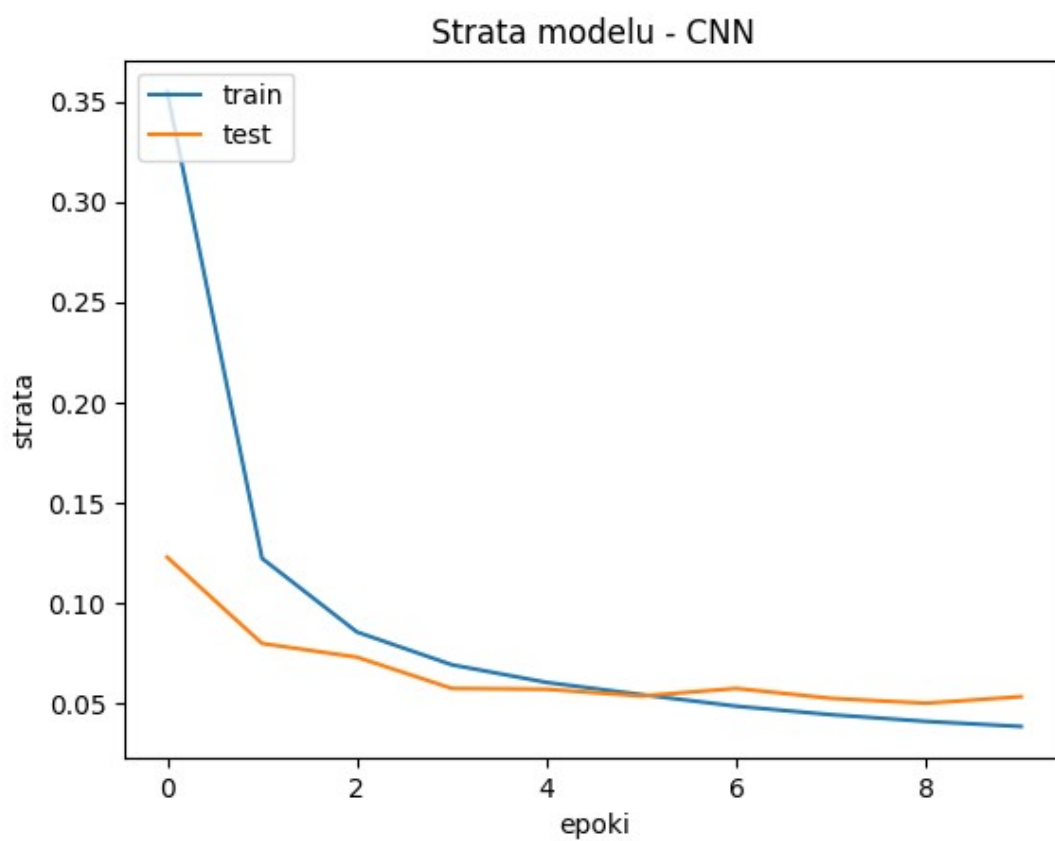
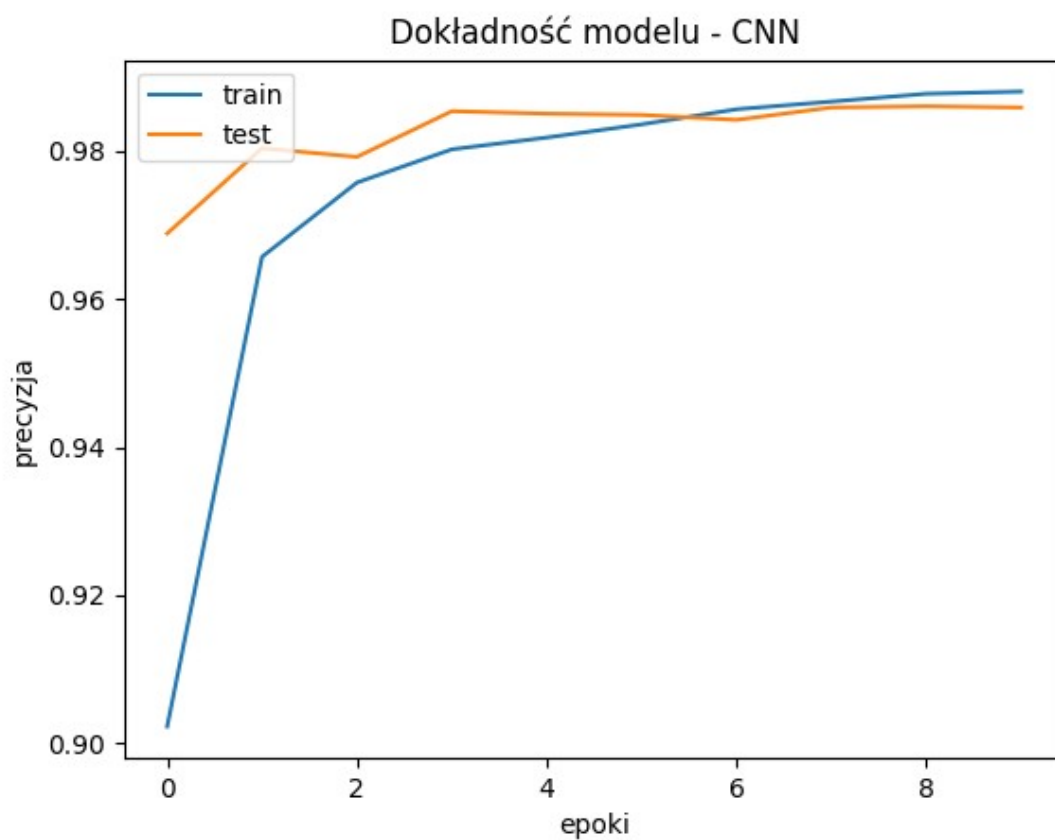
Okno pooling - 2x1



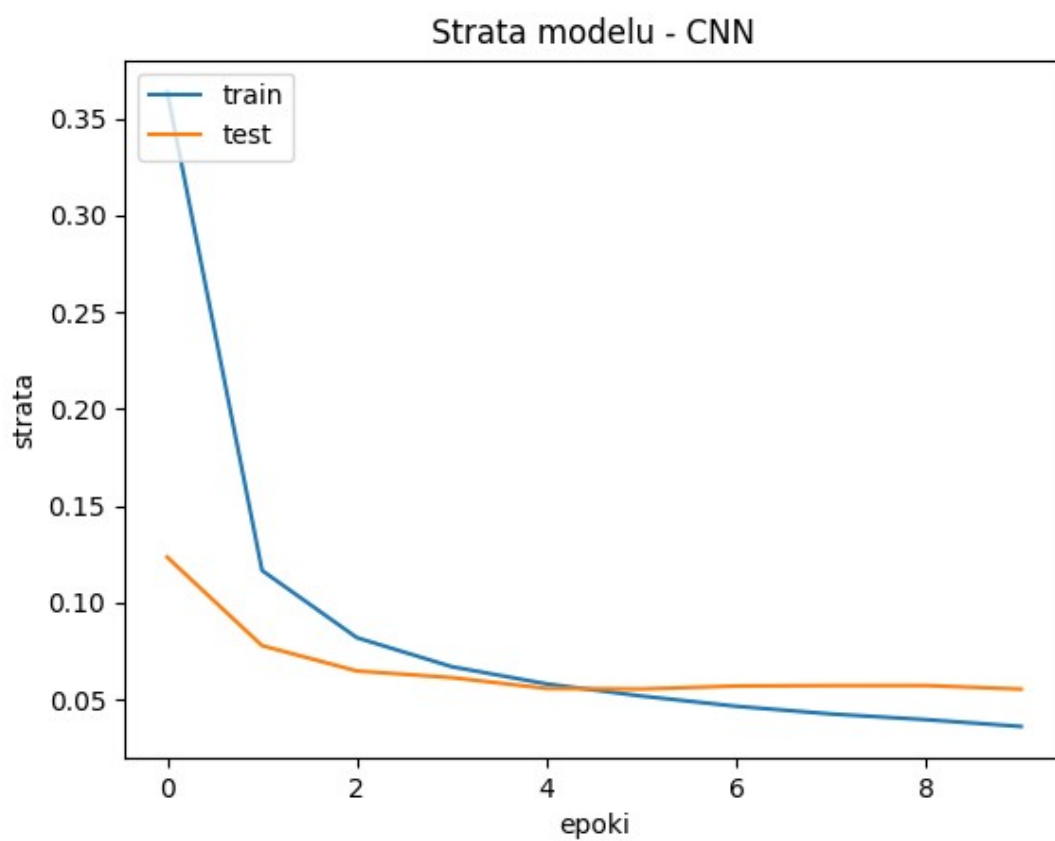
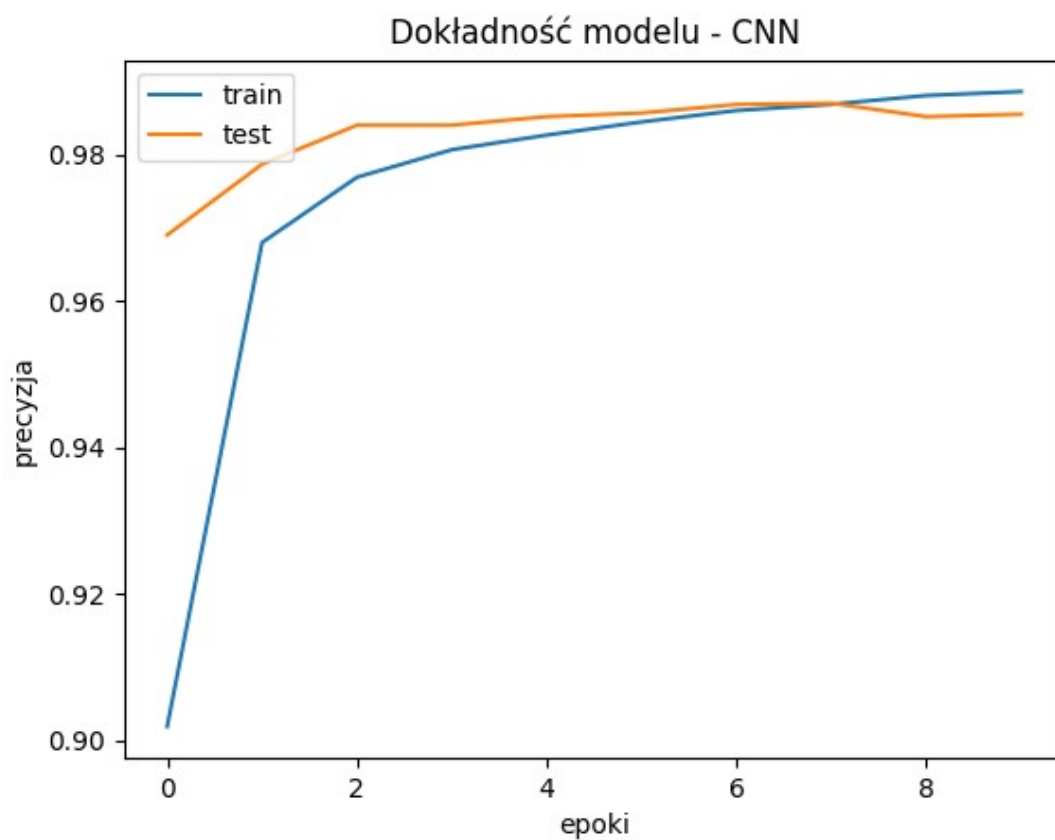
Okno pooling - 1x2



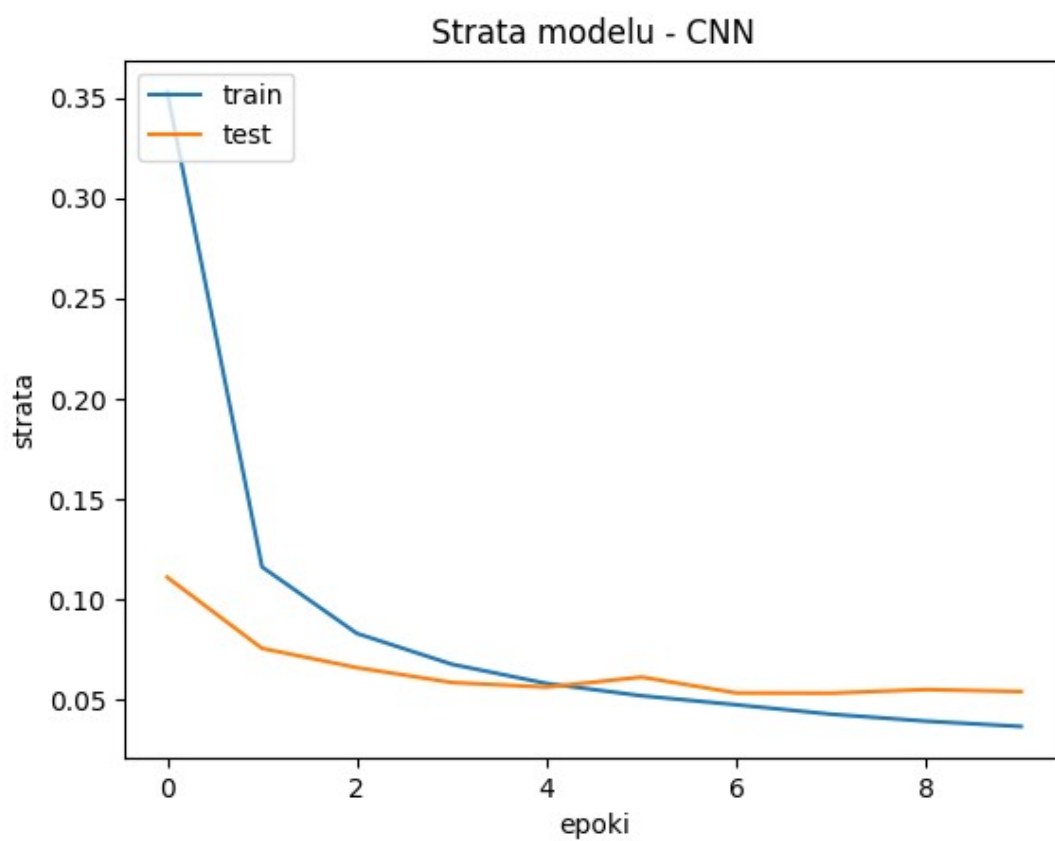
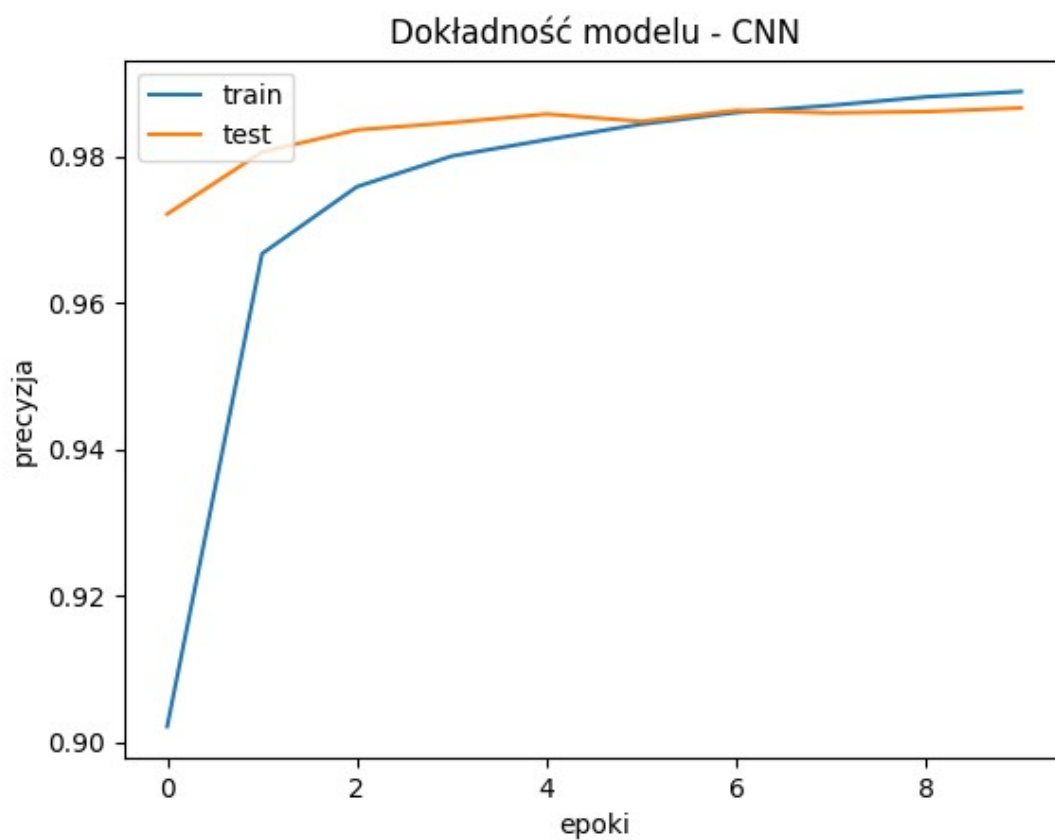
Okno pooling - 2x3



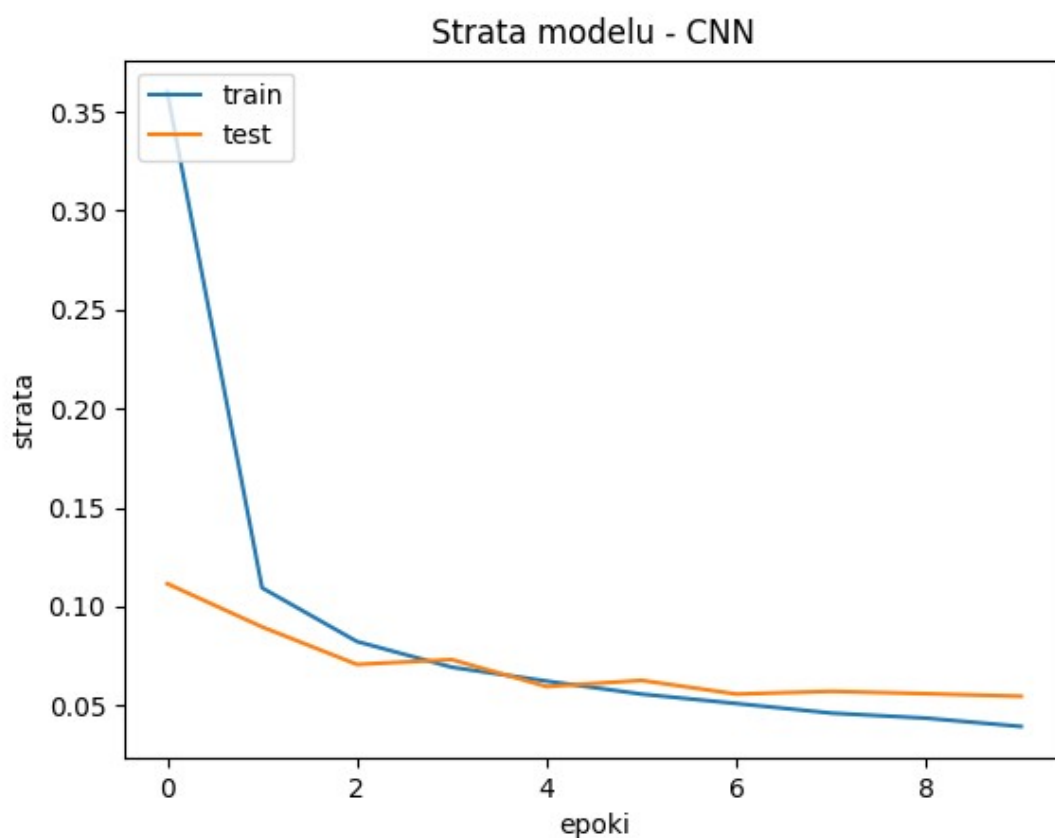
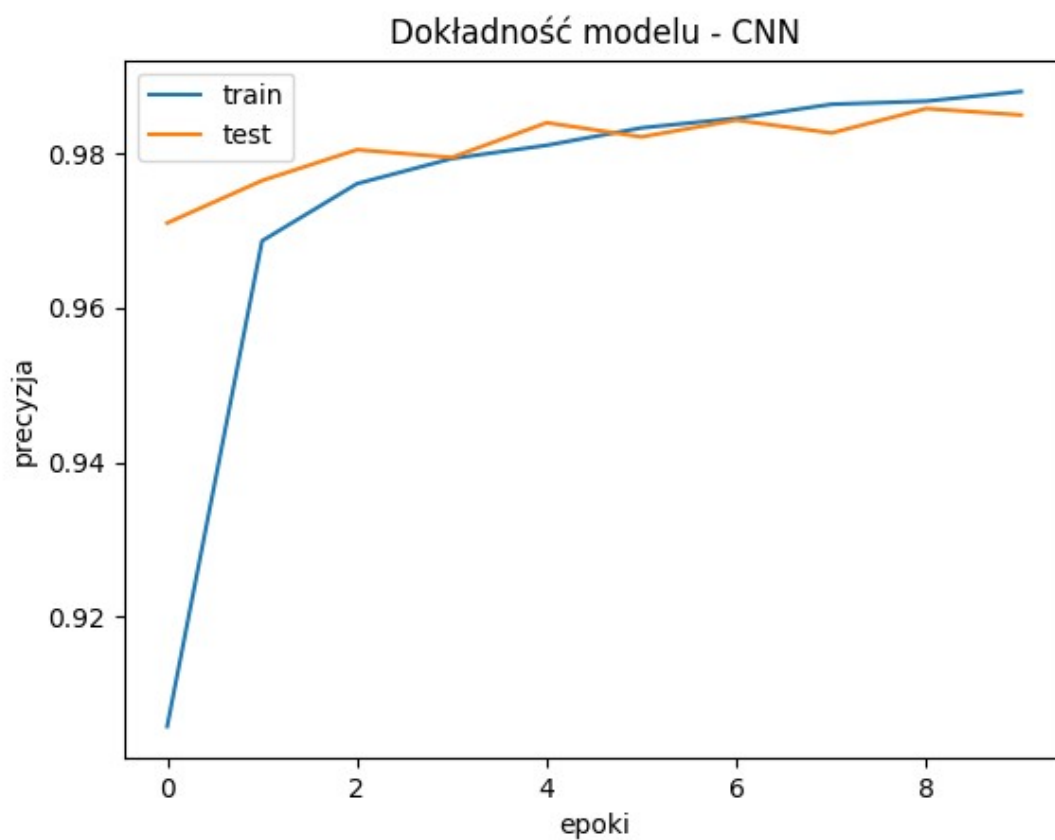
Okno pooling - 3x2



Okno pooling - 1x5



Okno pooling - 5x1

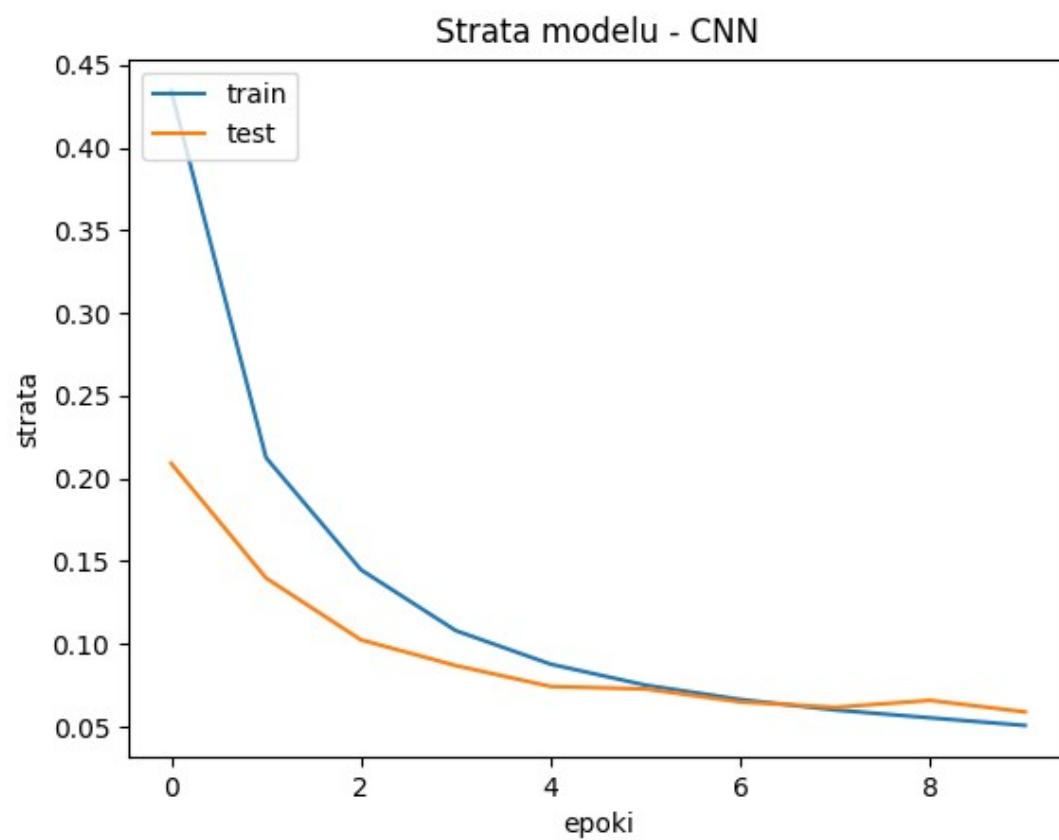
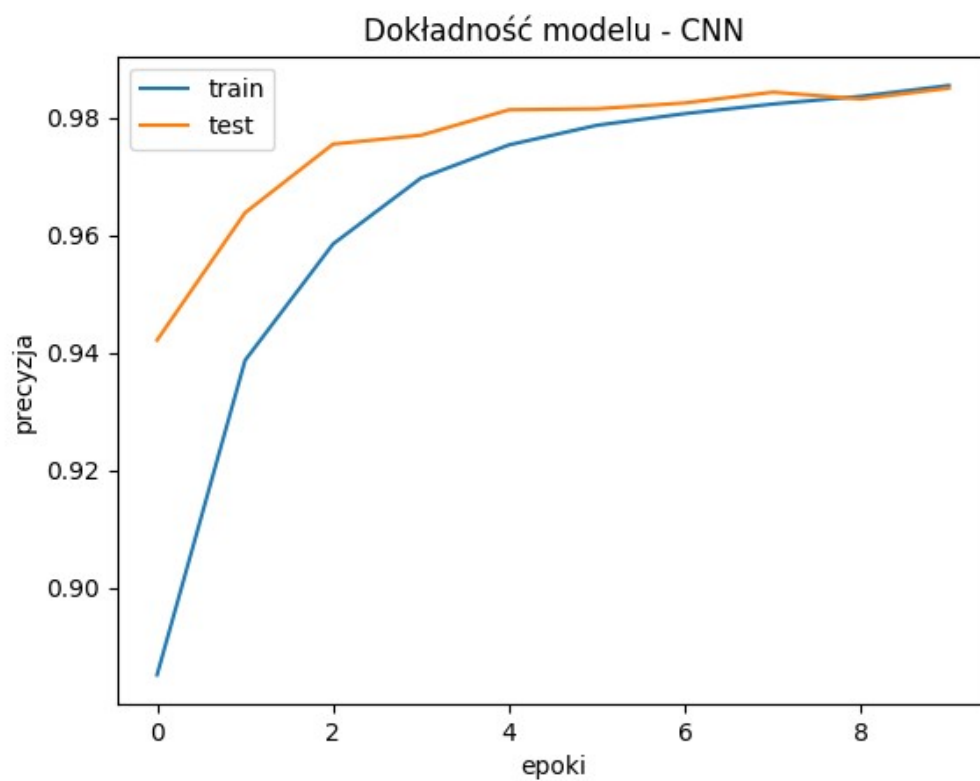


Wnioski

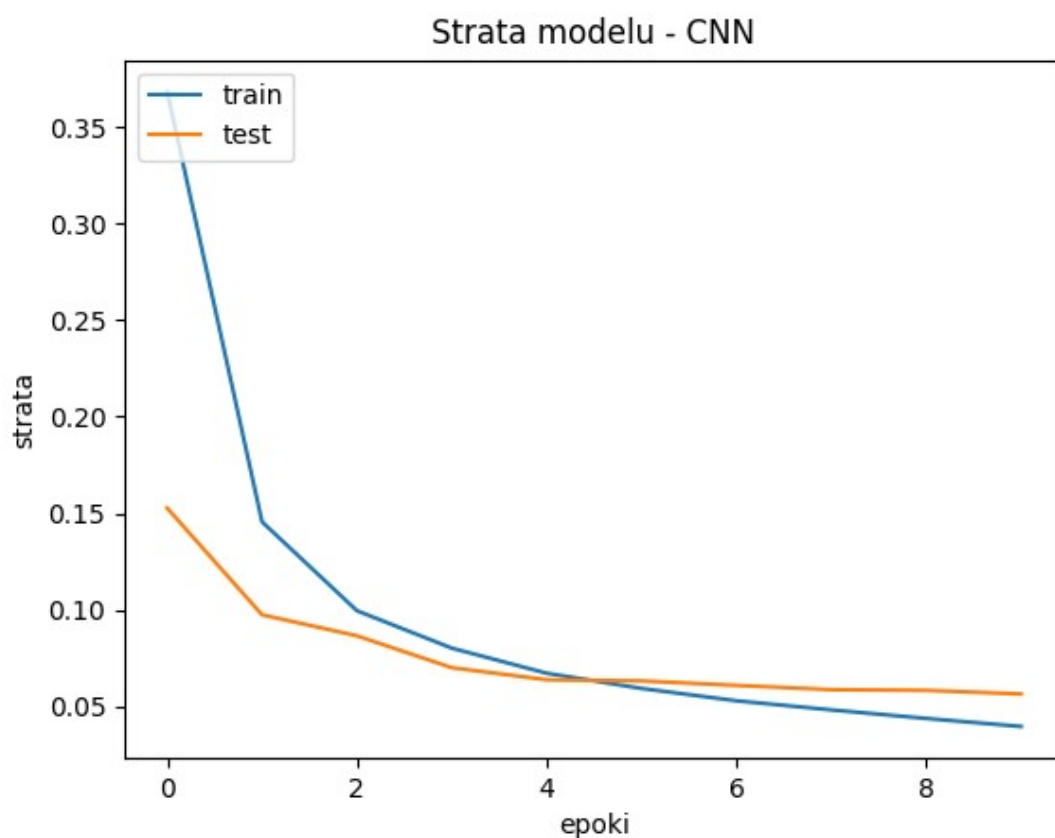
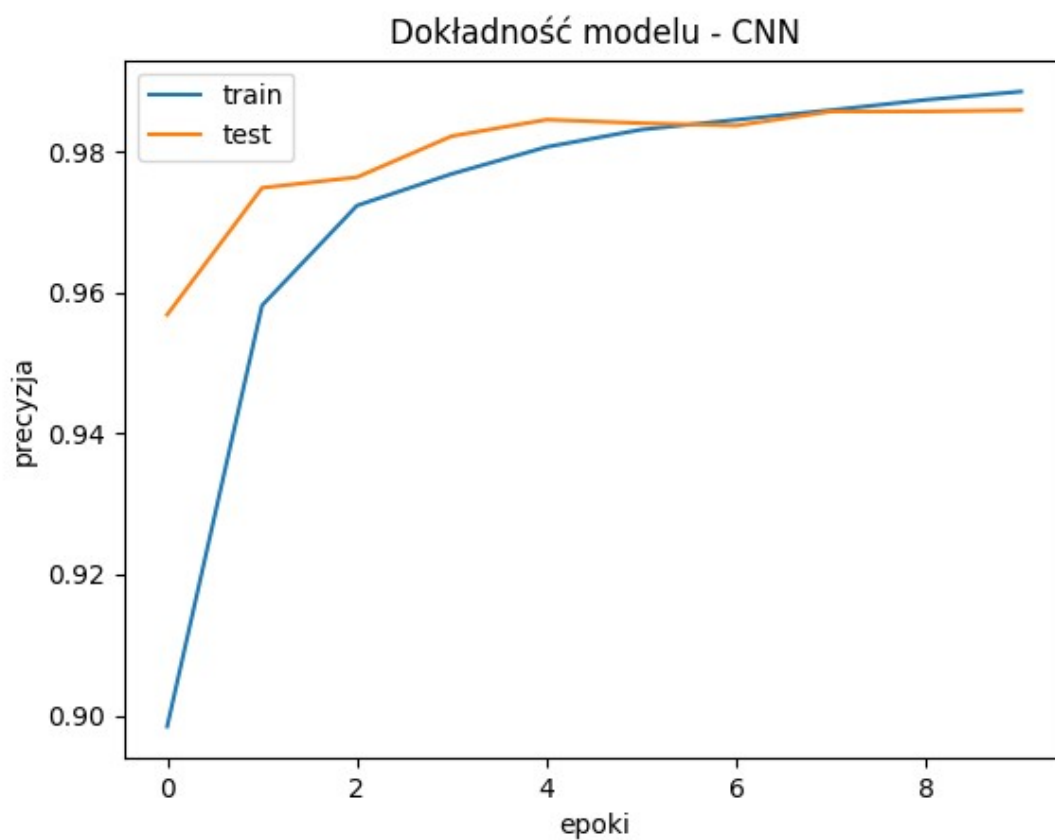
Sytuacja w zasadzie analogiczna do poprzedniego eksperymentu. Różnice są nieznaczne, czasem można zaobserwować gorsze lub lepsze oscylowanie dokładności na zbiorze testowym natomiast ostatecznie osiągnany jest podobny wynik.

Typ poolingu

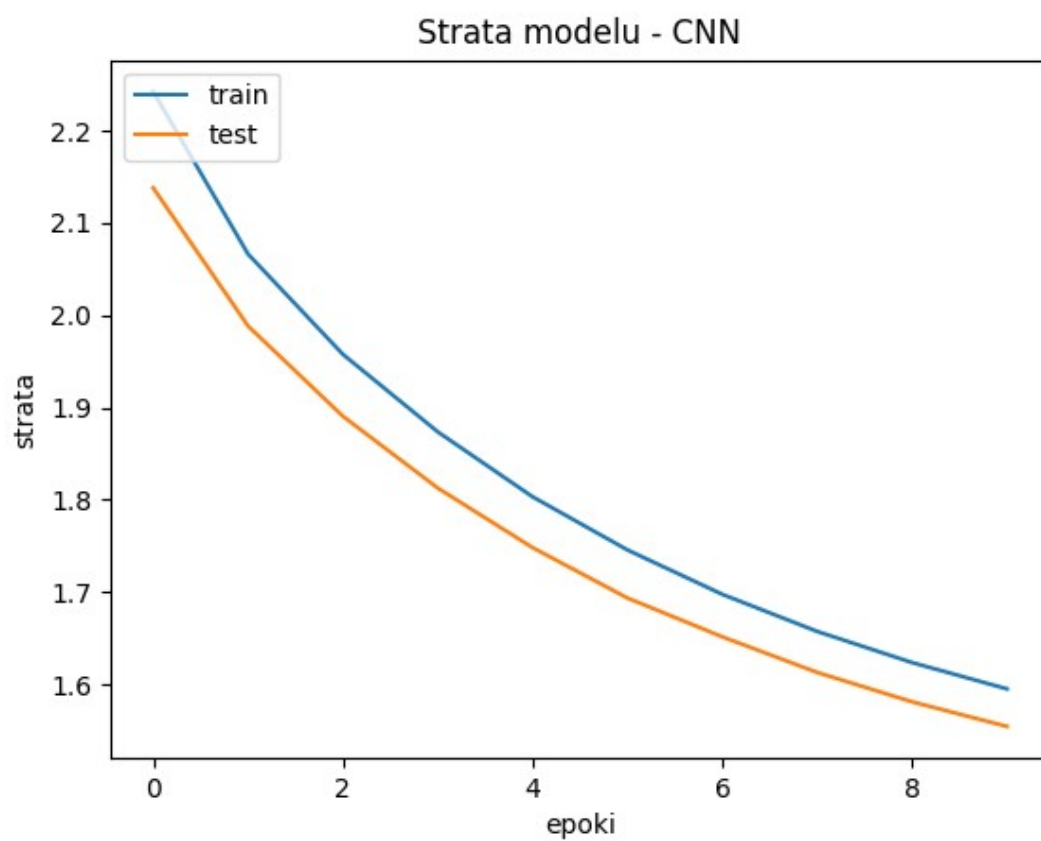
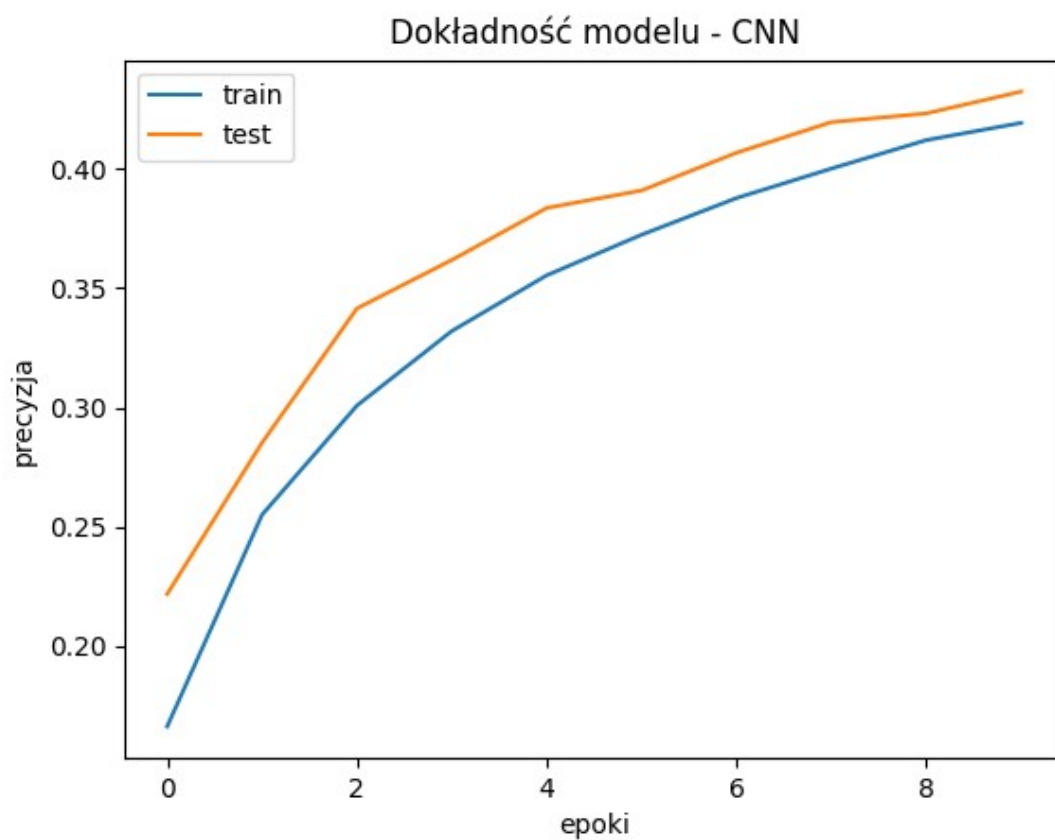
Average pooling



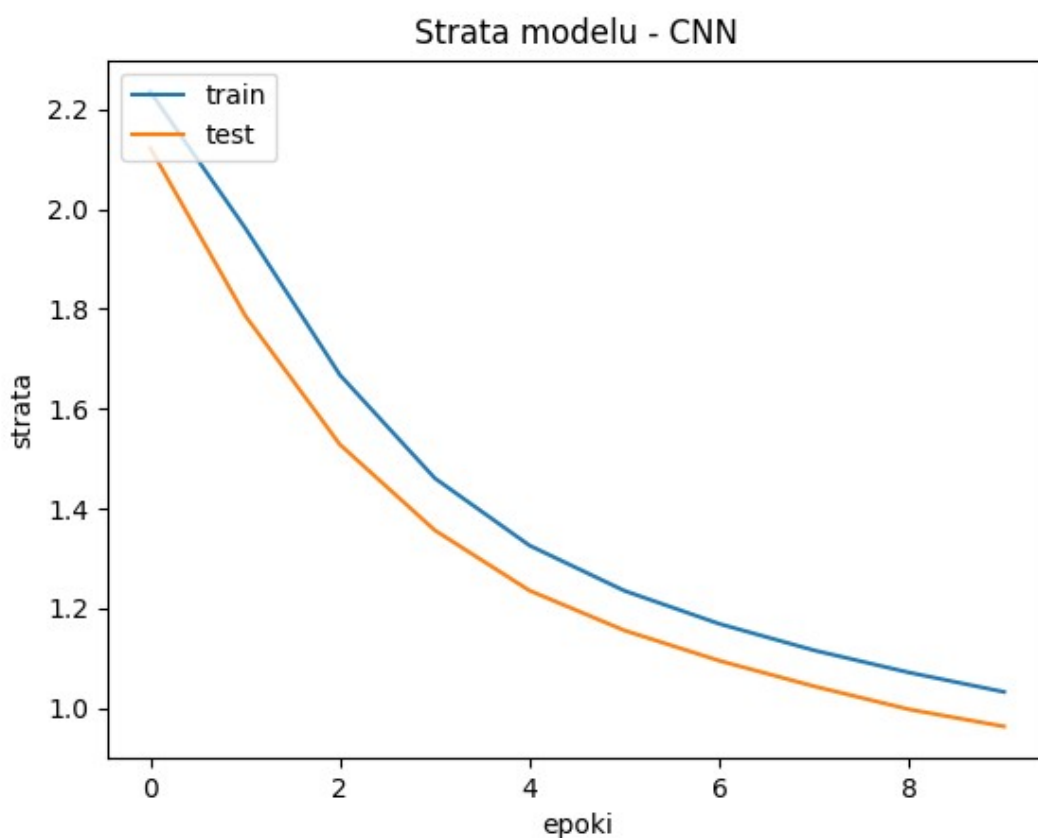
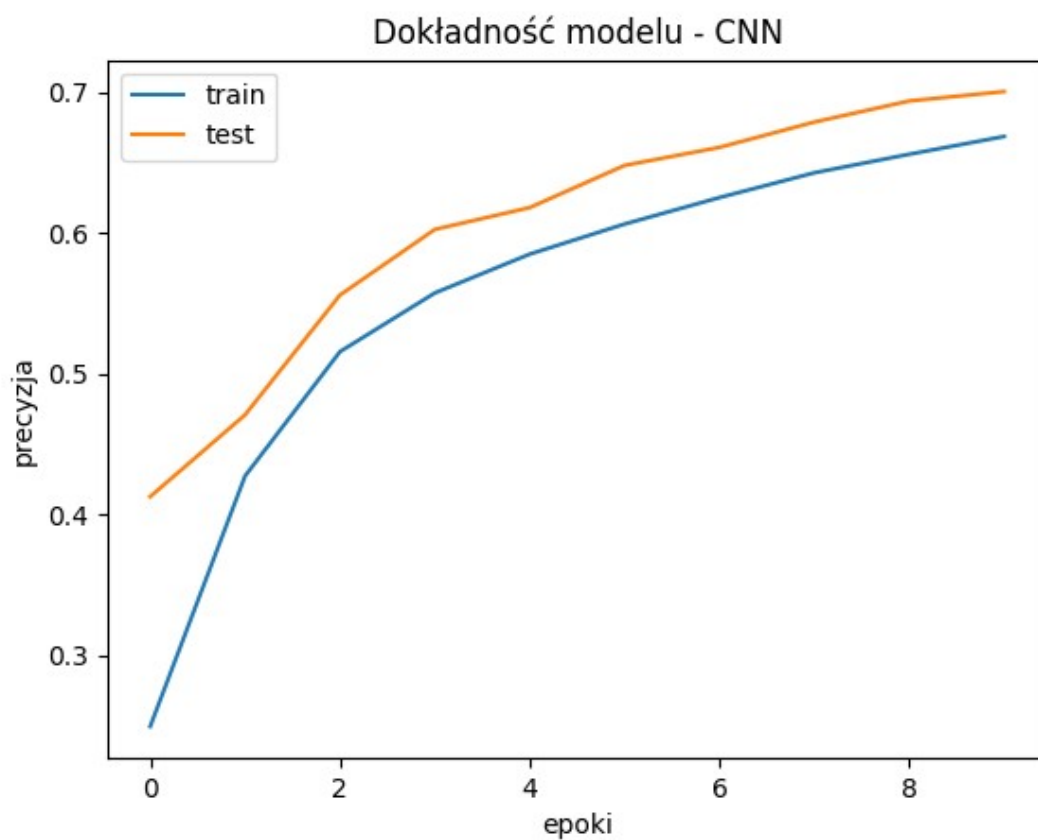
Max pooling



Global average pooling



Global max pooling

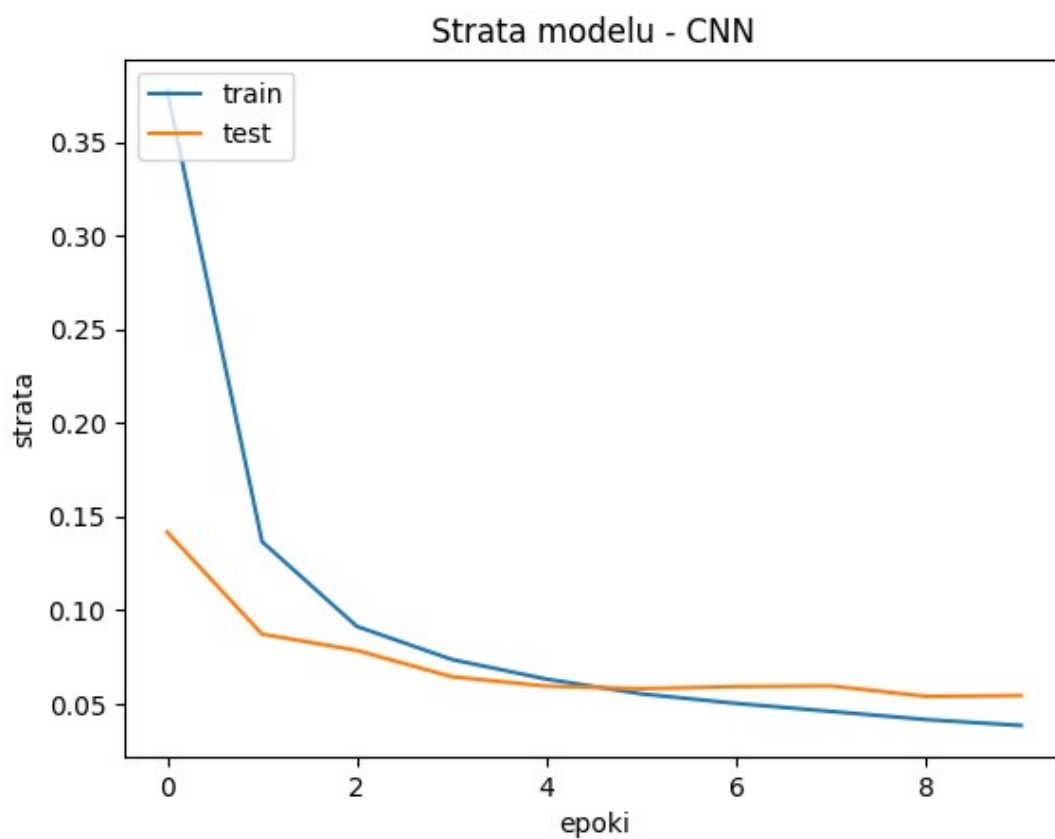
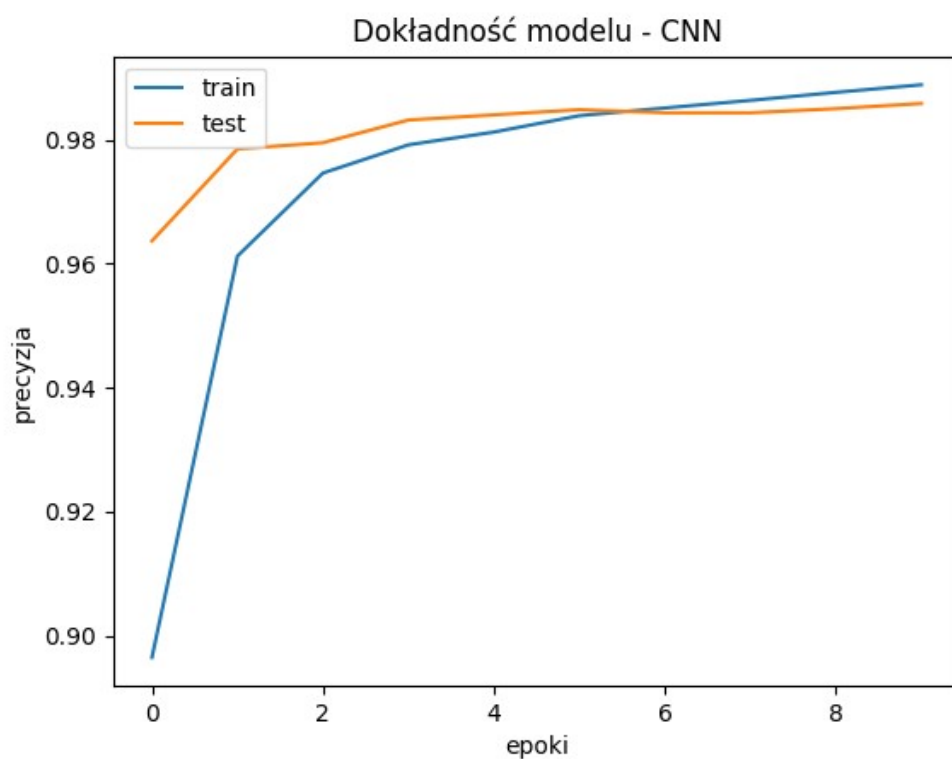


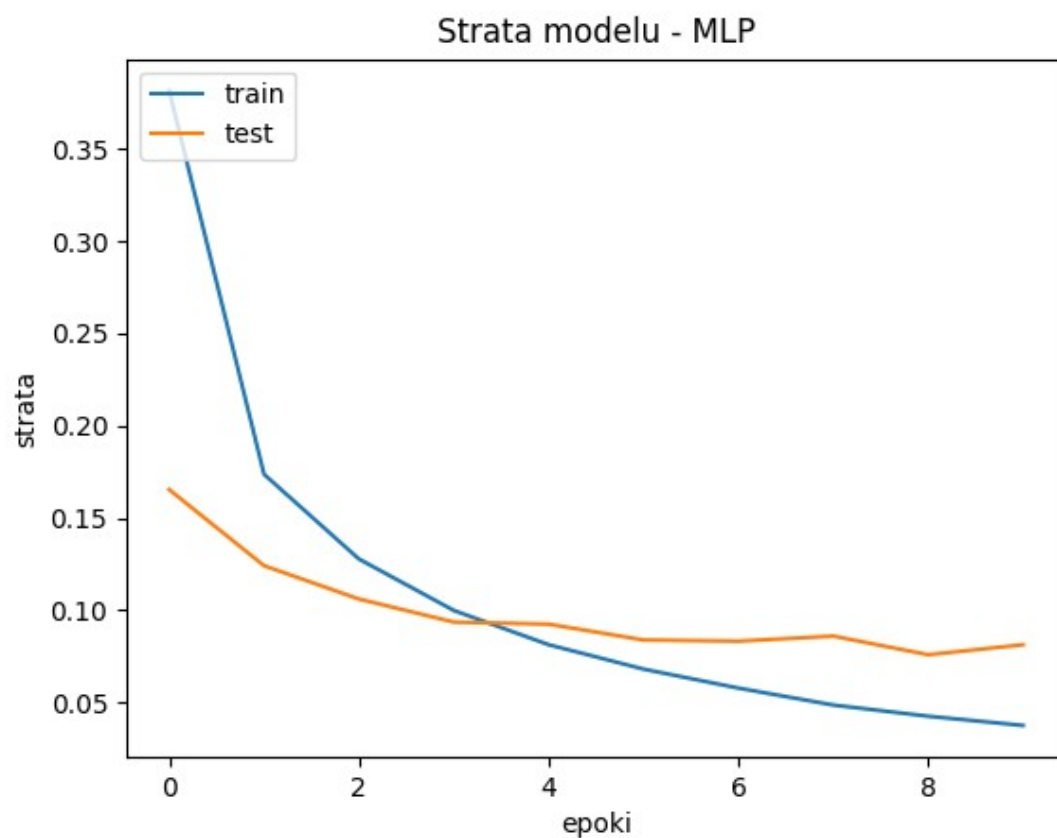
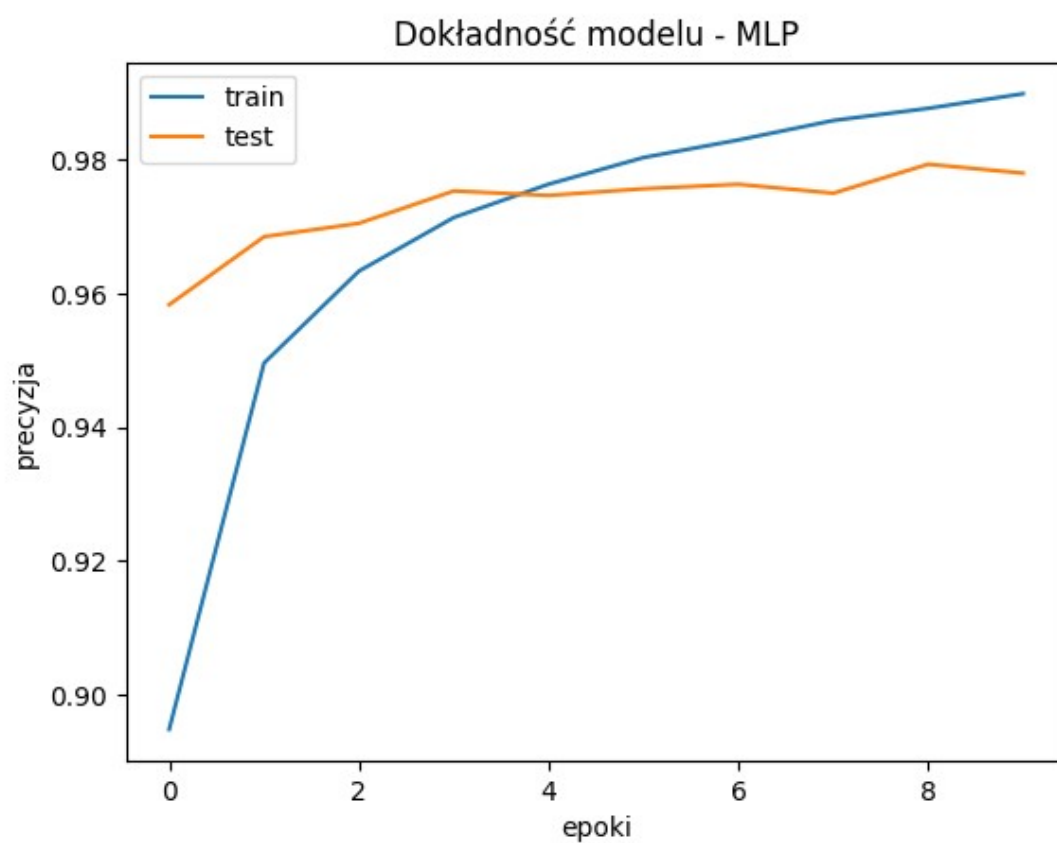
Wnioski

Max pooling wydaje się doprowadzać do szybszego wyuczenia modelu niż avg pooling (na przestrzeni mniejszej liczby epok dociera do około 98% skuteczności). Ostatecznie oba podejścia wydają się skuteczne. Wybrałem do badań jeszcze metody global avg pooling i global max pooling. Od zwykłego avg/max pooling różni się tym, że rozmiar okna to zawsze cały obraz. Jak widać na powyższych wykresach, przy wykorzystaniu tych technik model również się uczy, natomiast proces ten przebiega zdecydowanie wolniej. Ostatecznie oba podejścia nie osiągną takich samych rezultatów, ponieważ gubią bardzo dużo szczegółów.

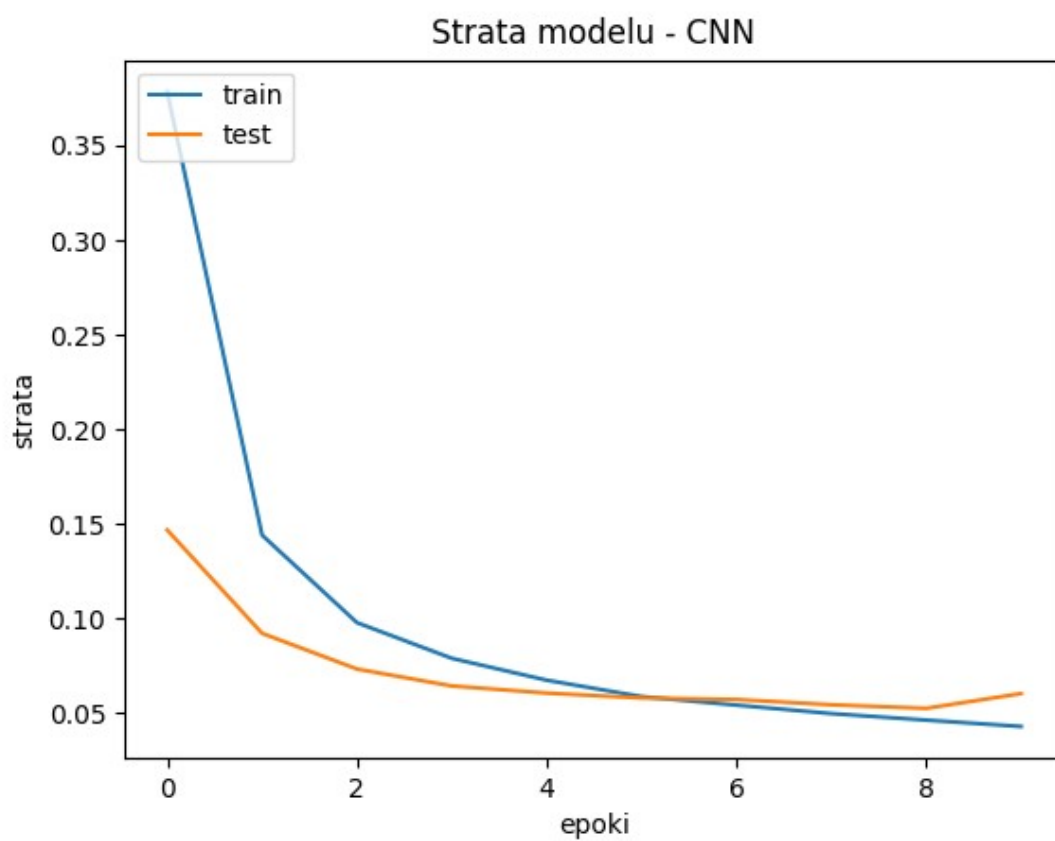
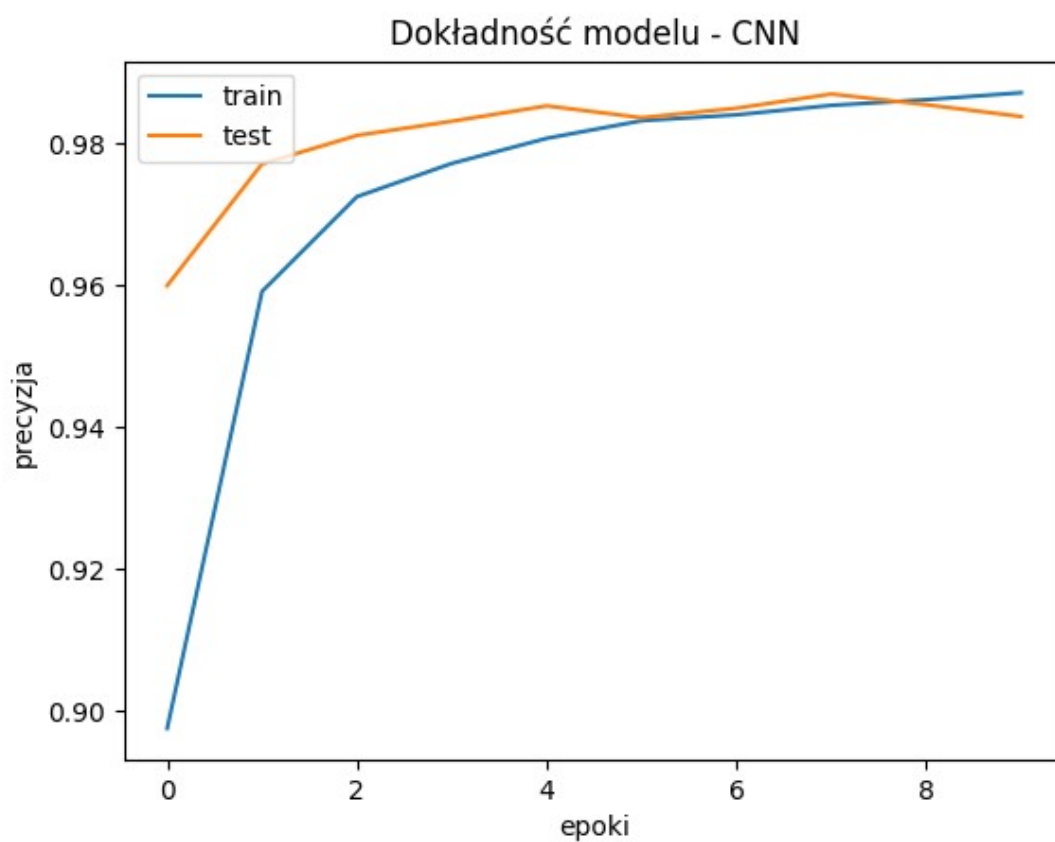
Ocena efektywności techniki 'dropout'

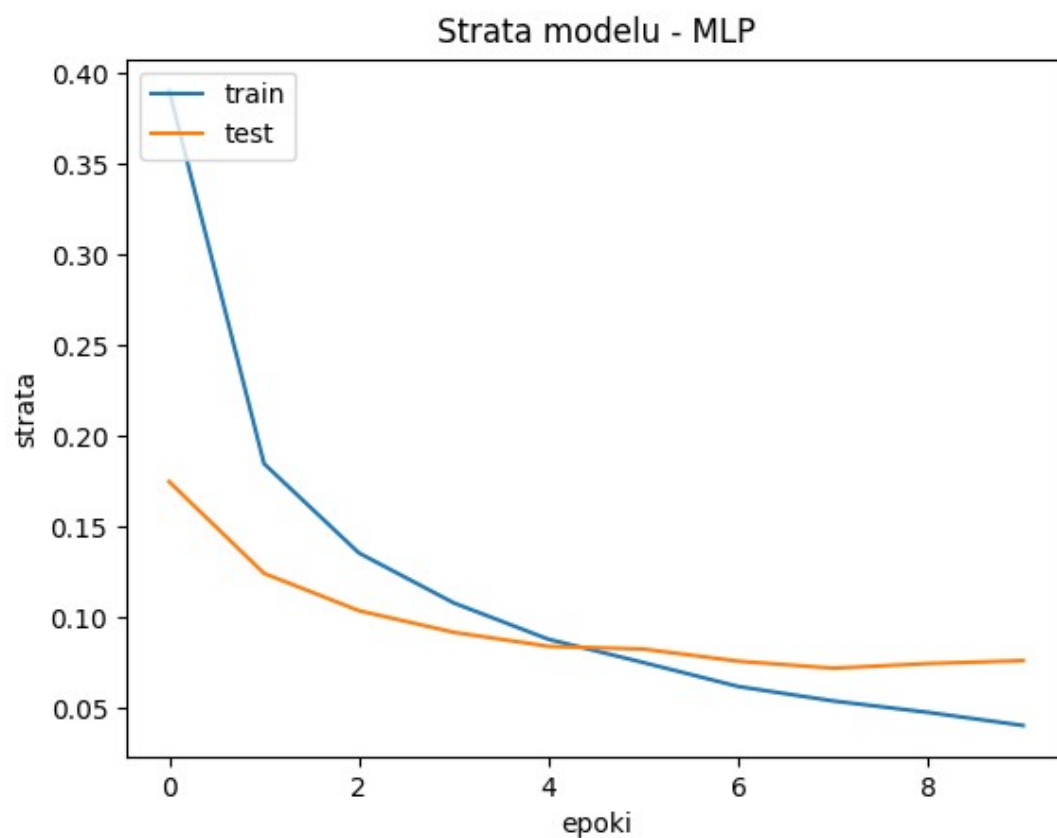
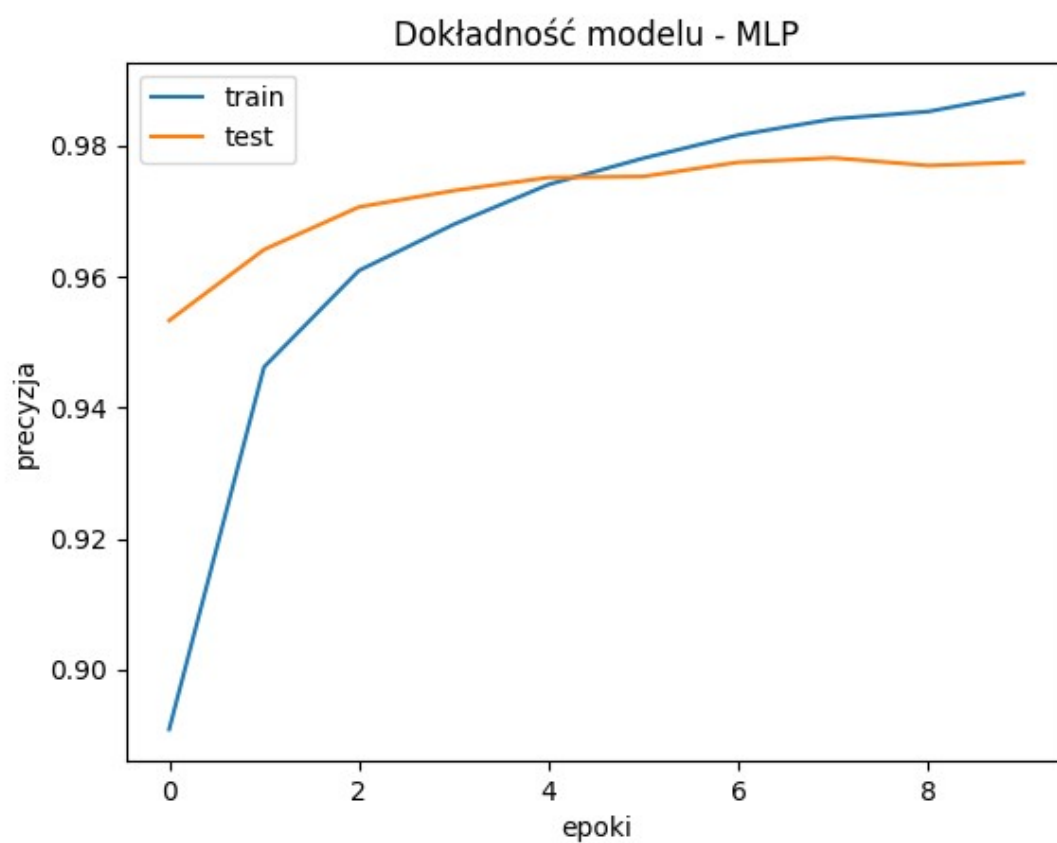
Prawdopodobieństwo dropout'u - 0.01



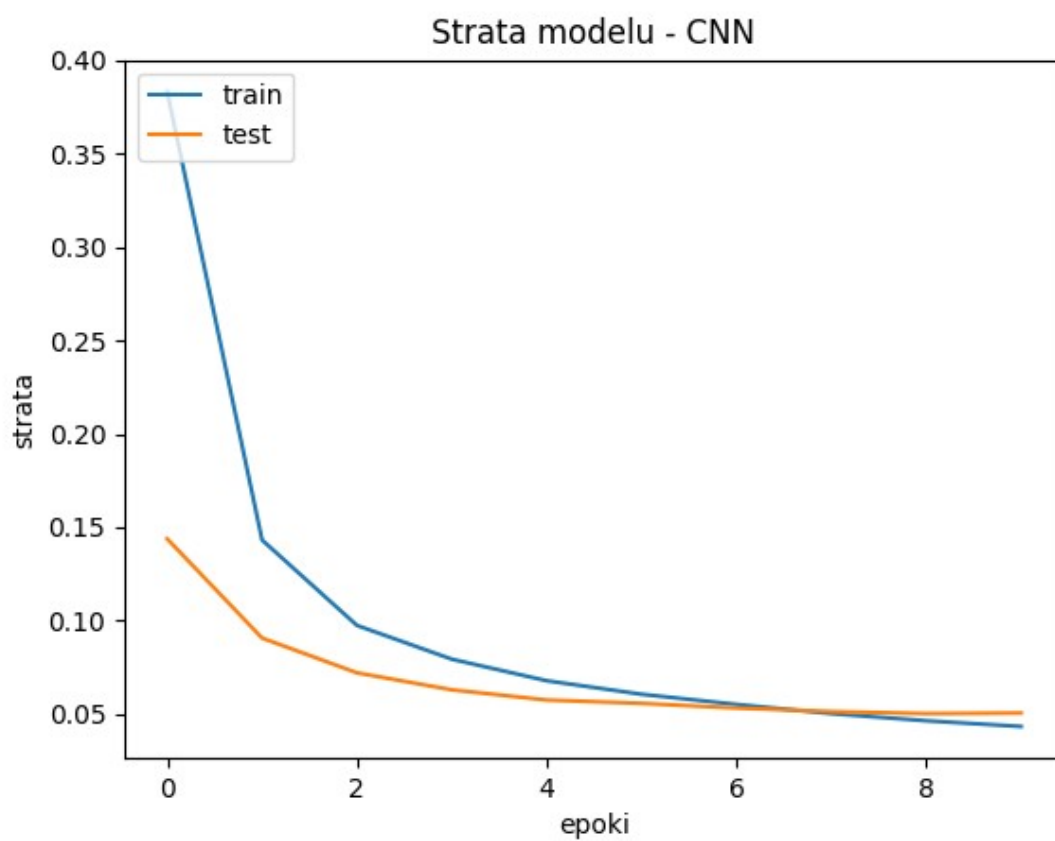
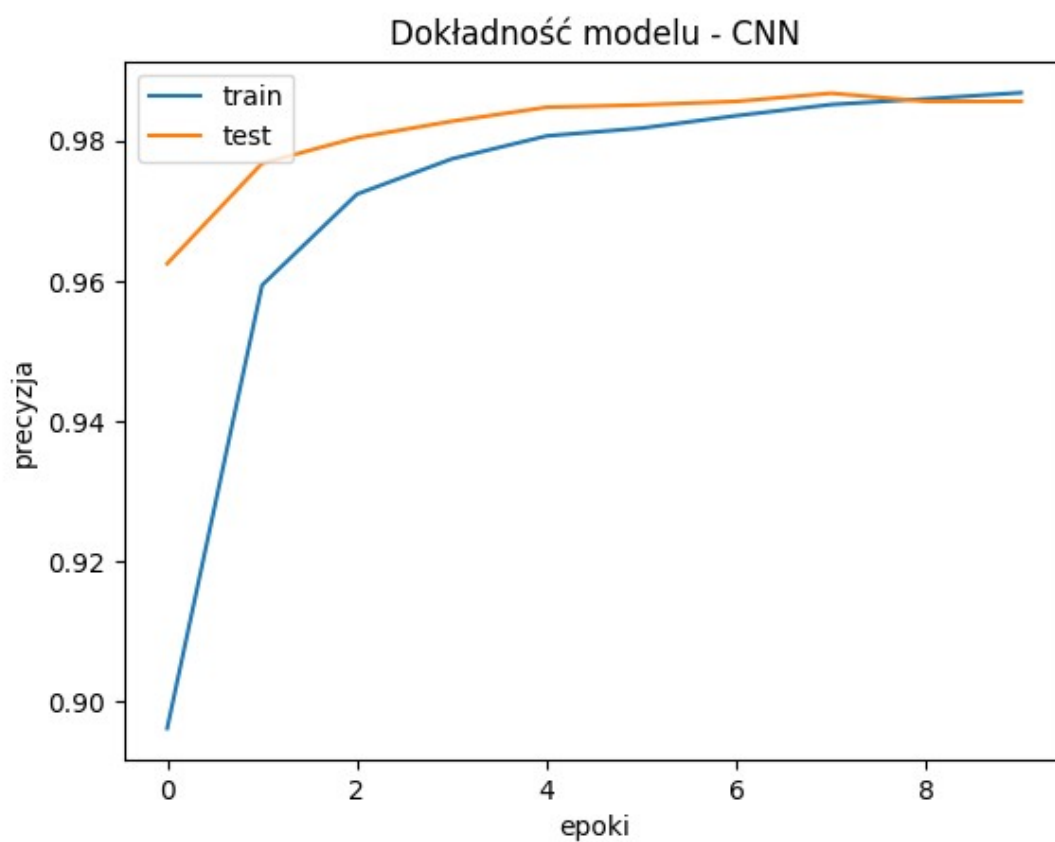


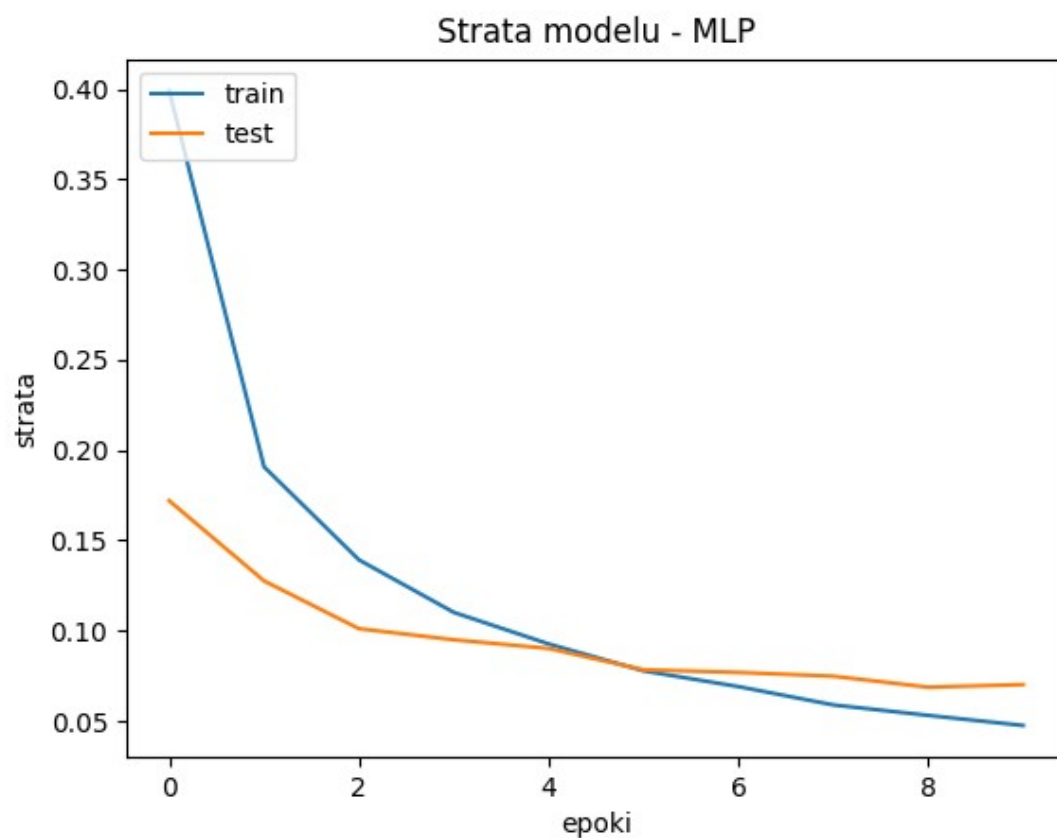
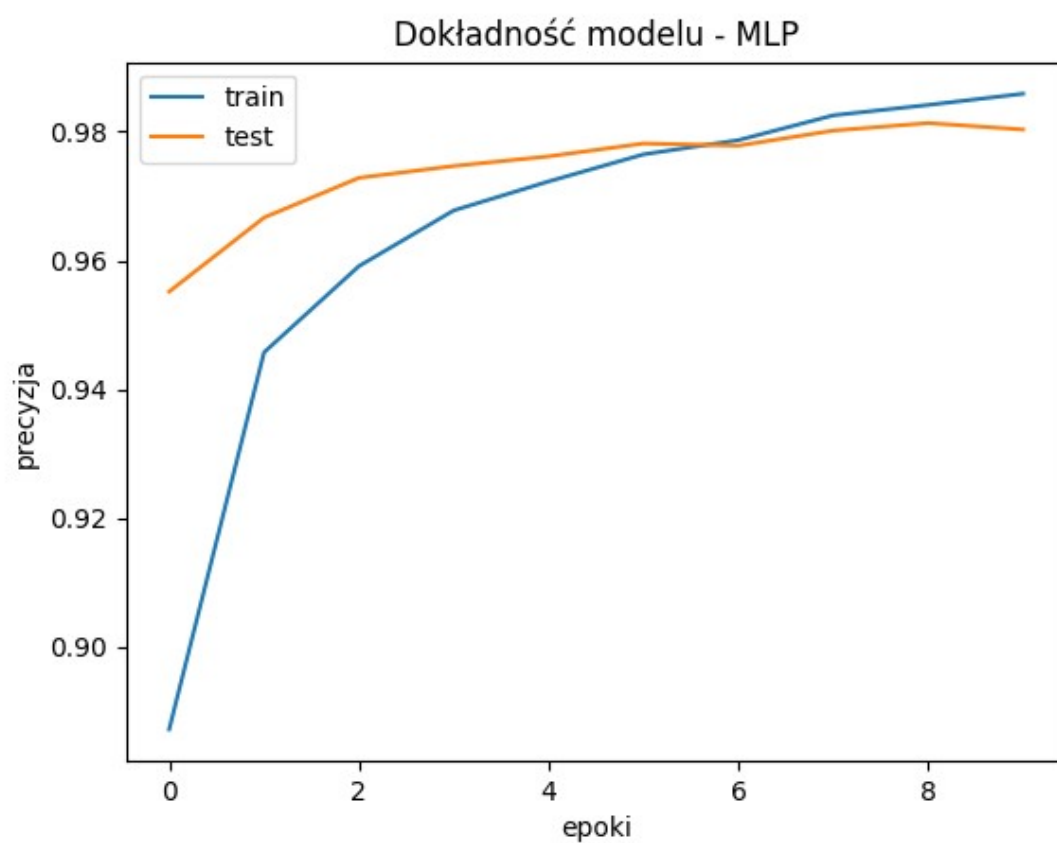
Prawdopodobieństwo dropout'u - 0.05



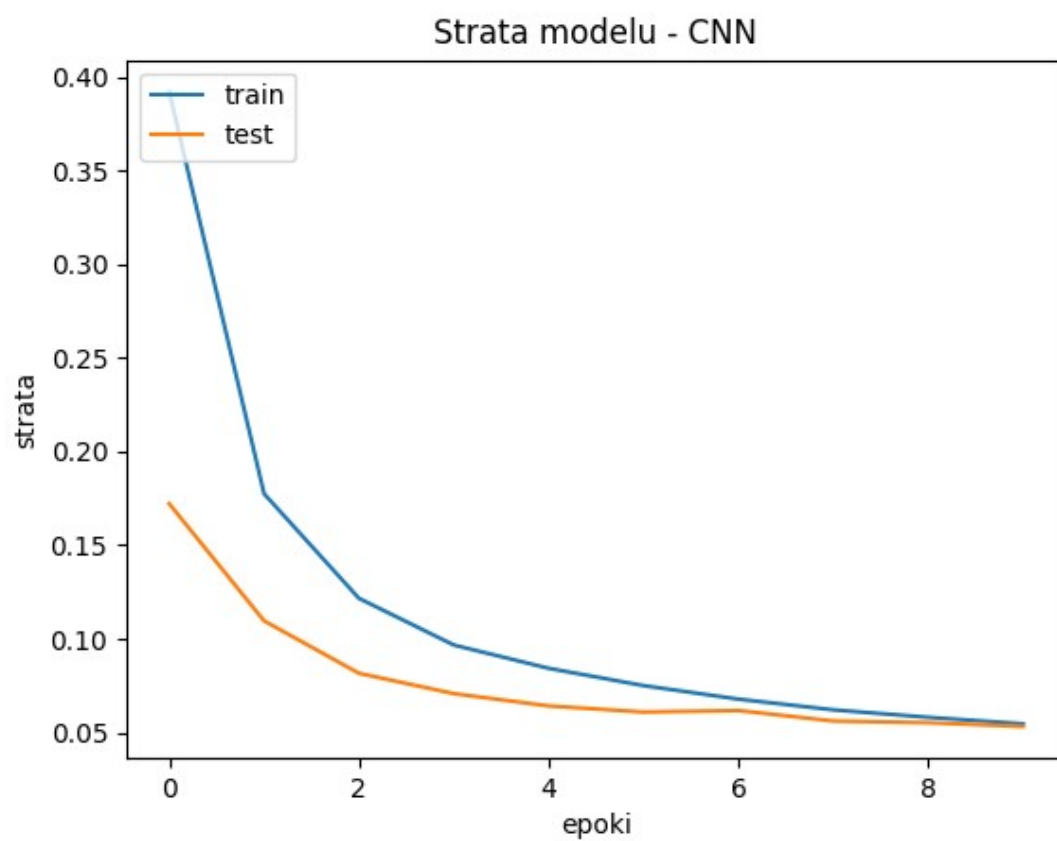
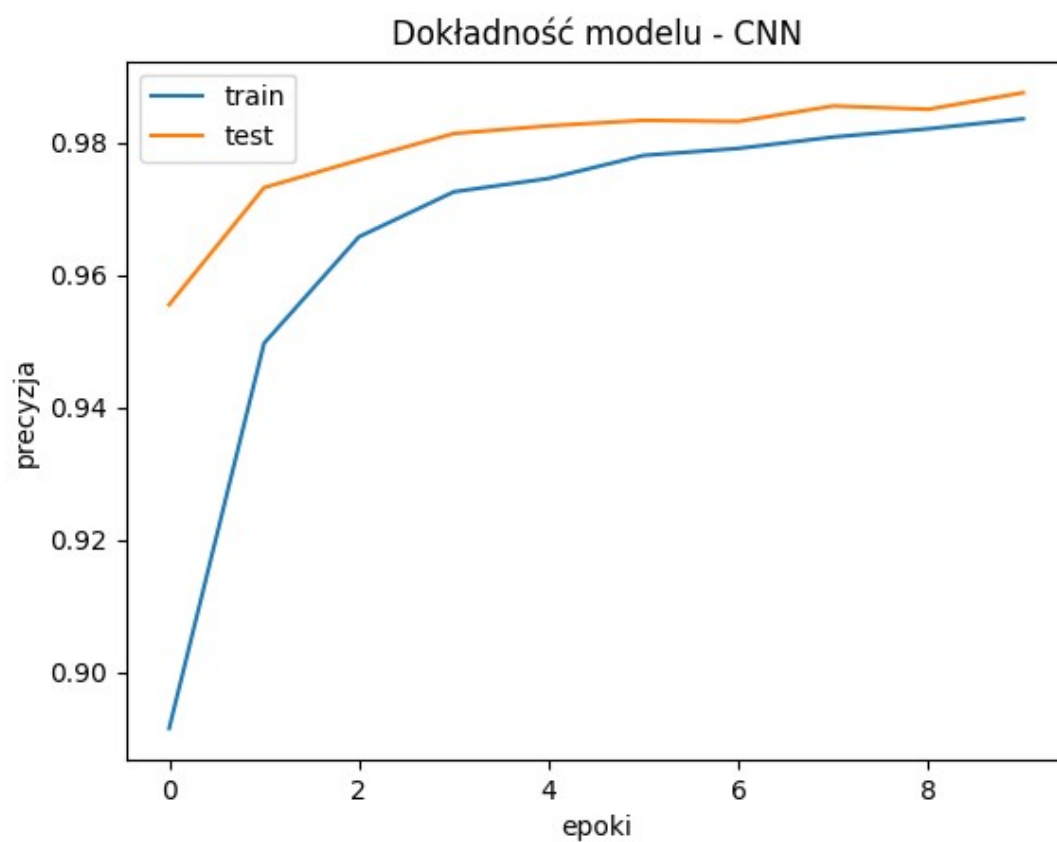


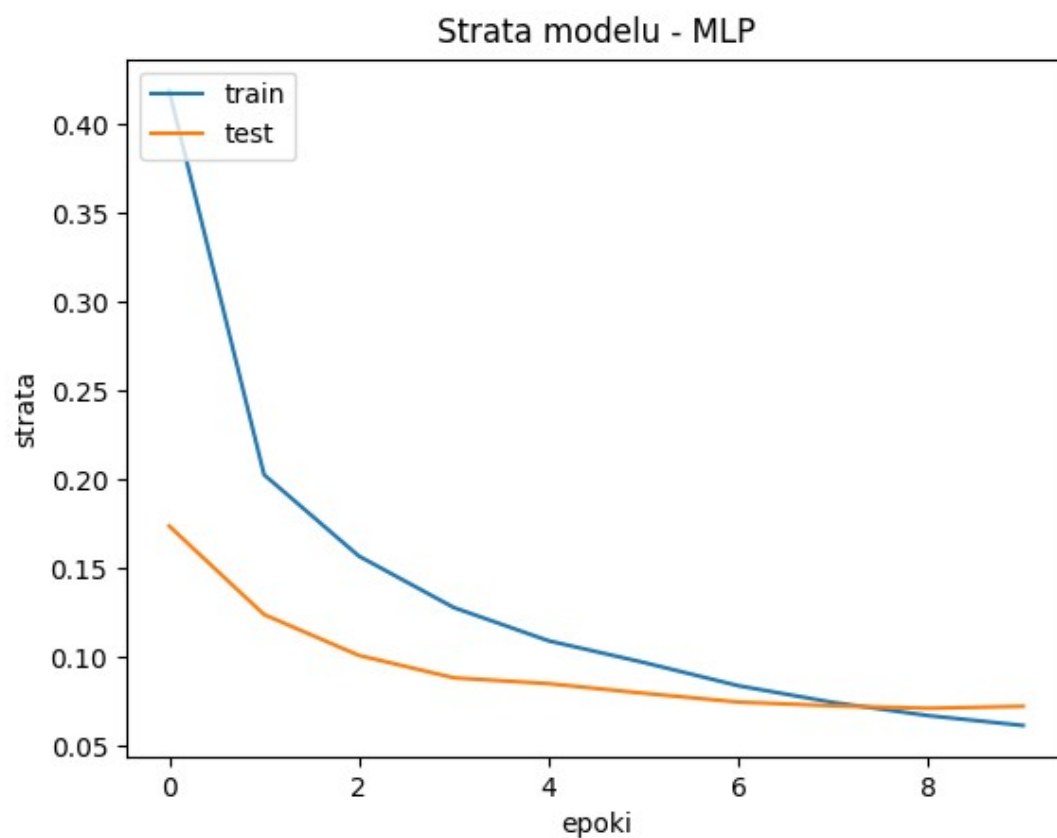
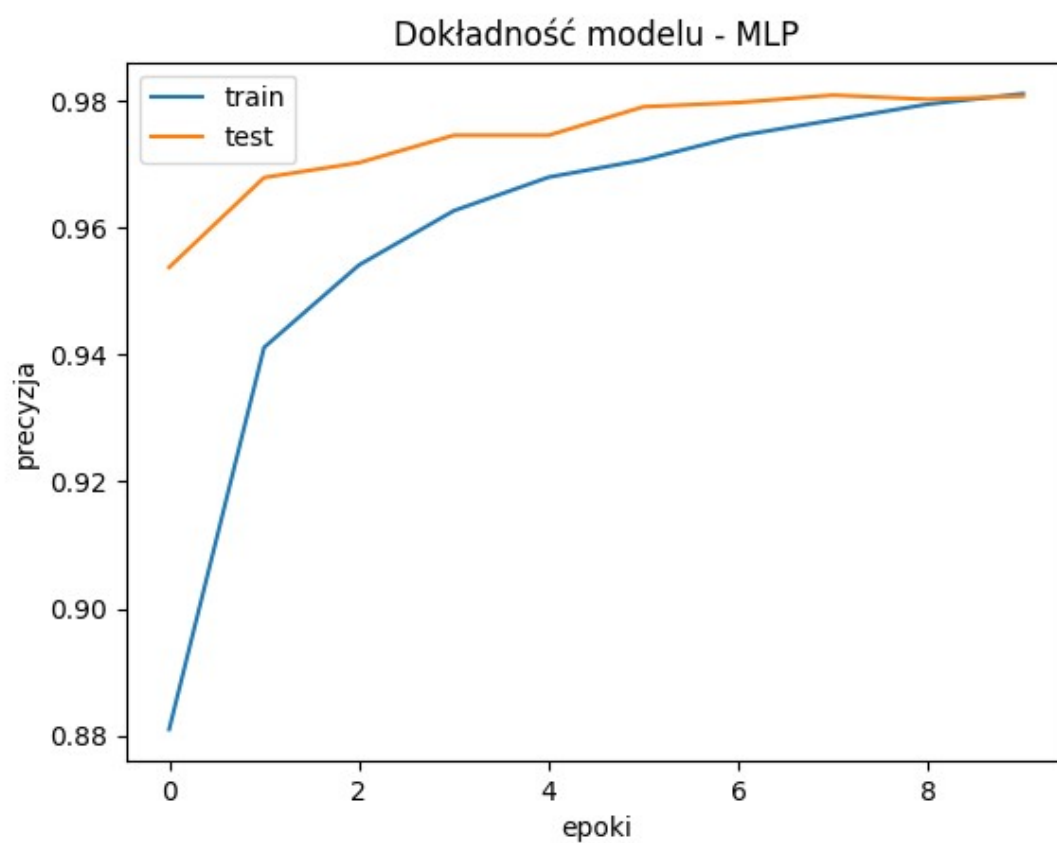
Prawdopodobieństwo dropout'u - 0.1



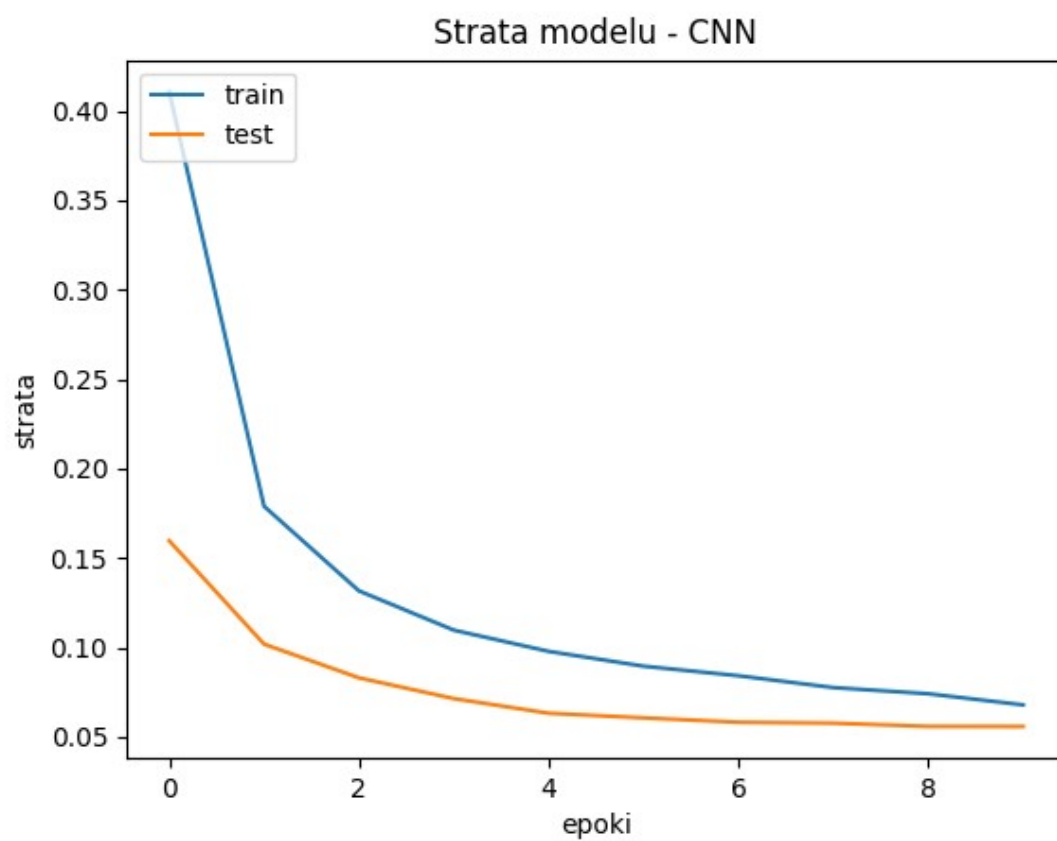
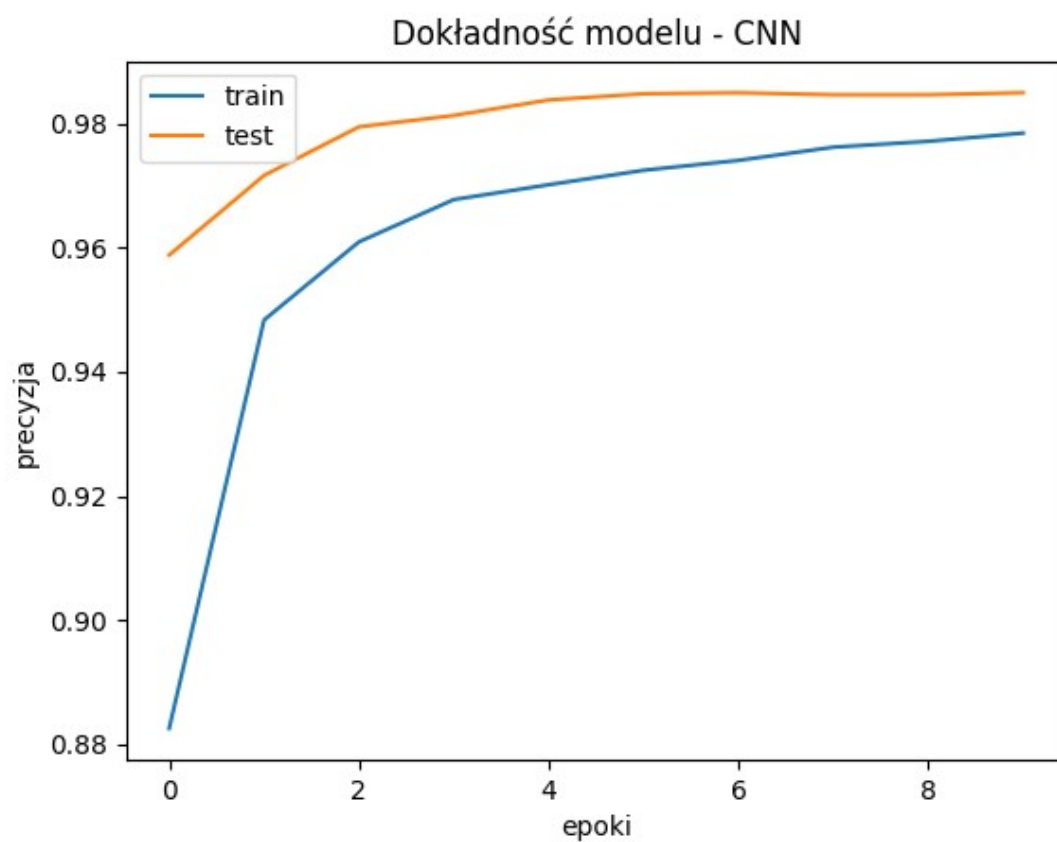


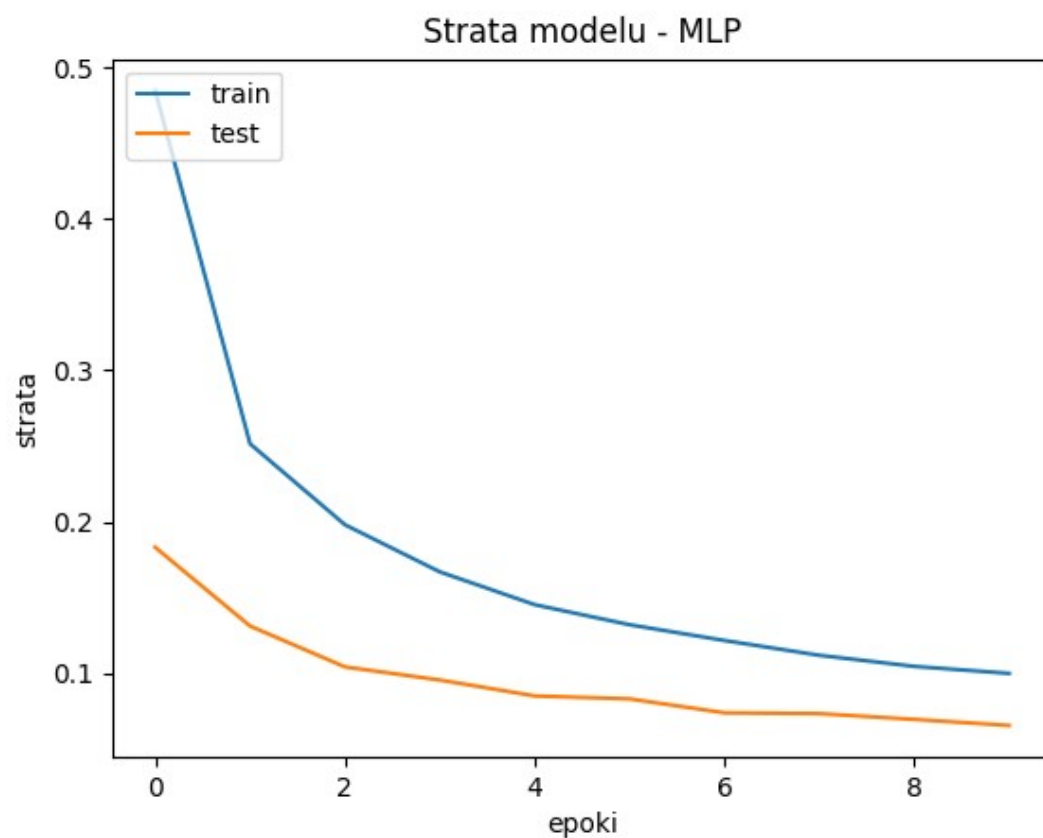
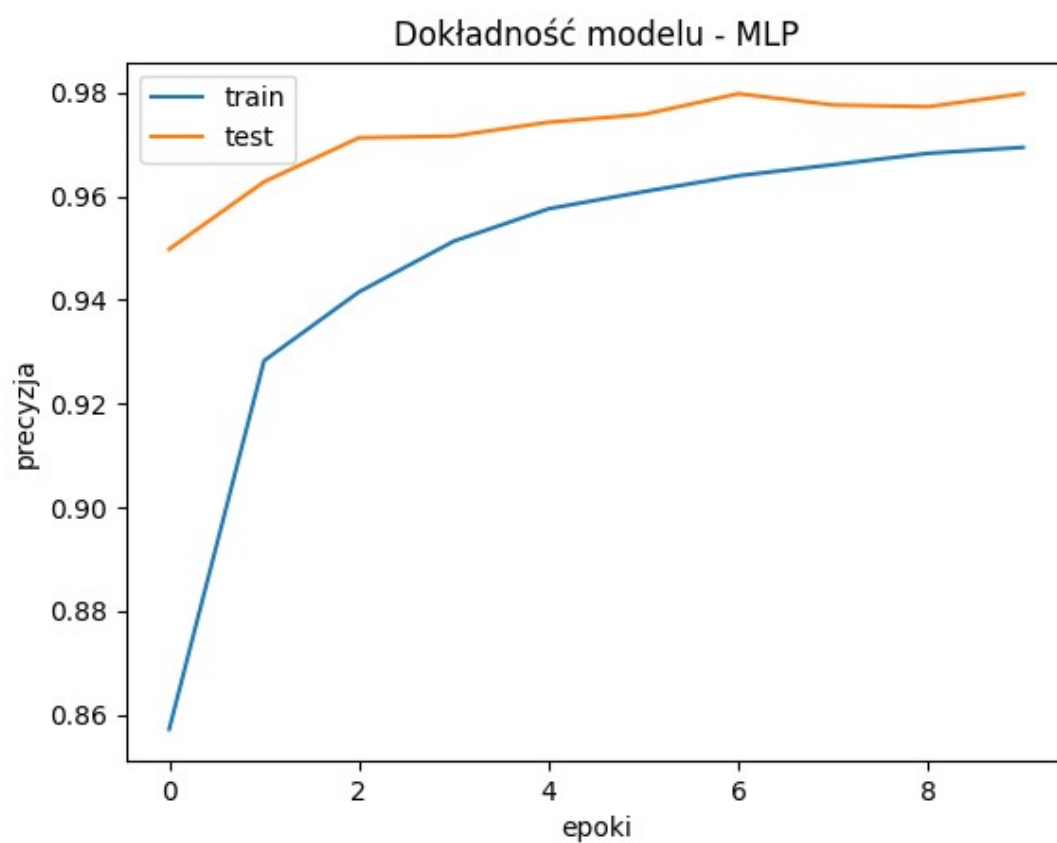
Prawdopodobieństwo dropout'u - 0.2





Prawdopodobieństwo dropout'u - 0.4





Wnioski

Technika 'dropout' ma służyć przede wszystkim do eliminowania zjawiska przeuczenia, niezależnie od modelu. Patrząc na powyższe wykresy (szczególnie wykresy straty) można stwierdzić, że jest ona efektywna i spełnia zamierzony cel. Im większe prawdopodobieństwo wystąpienia dropout'u podałem, tym mniej skuteczność na zbiorze treningowym 'odskakiwała' od skuteczności na zbiorze testowym. Przy zbyt dużym prawdopodobieństwie dochodzi do sytuacji, w której model zdecydowanie wolniej się uczy (widać to na ostatnich wykresach). Wiąże się to z wyłączeniem dużej liczby neuronów, co z jednej strony prowadzi do generalizacji ale z drugiej strony dłużej zajmuje modelowi 'wypracowanie' bardziej złożonych wzorców.

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 usuwania '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.