1 Eksperyment 1: Architektury sieci splotowej dla MNIST (Jakub Pawlak)

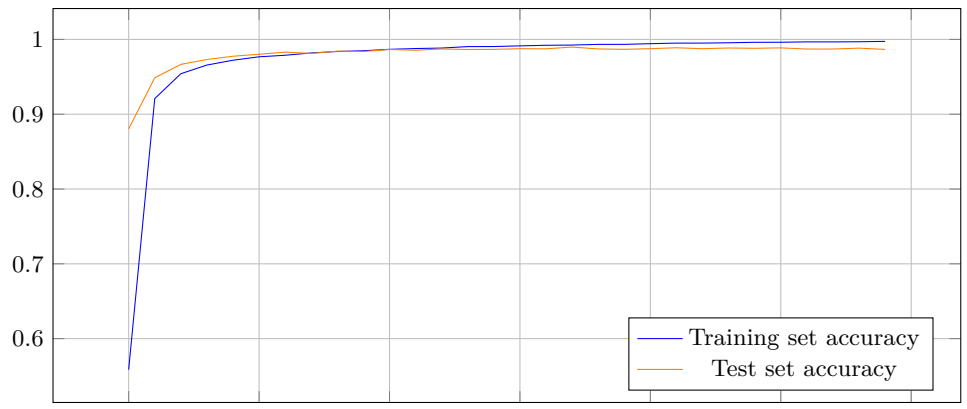
Pierwsza architektura

Pierwsza architektura składa się z warstwy splotowej z jądrem 5×5, na wyjściu której obraz ma 6 kanałów, następnie jest przepuszczony przez funkcję sigmoid oraz max pooling. Zamierzeniem jest, aby ta warstwa wykrywała krawędzie lub rogi cyfr. Z tego też powodu, użyto funkcji aktywacji sigmoid, ponieważ funkcja ReLU całkowicie zeruje ujemne wartości. W przypadku wykrywania krawędzi, ujemne wartości mogą być używane do reprezentacji kierunku.

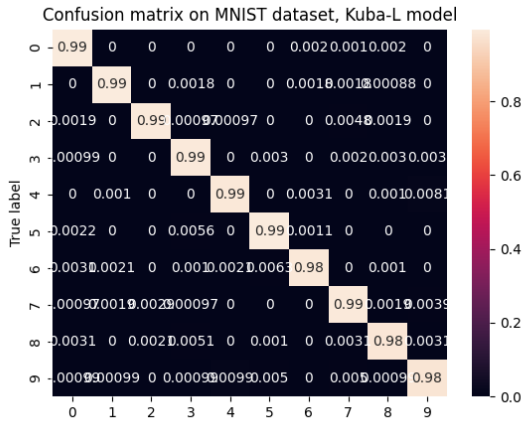
Drugi splot prowadzi do 16 kanałów i ponownie jest używana funkcja sigmoid oraz max pooling. Zadaniem tej warstwy jest rozpoznanie większych, bardziej złożonych kształtów. Finalnie ekstraktor prowadzi do tensora $16 \times 5 \times 5$, który zostaje spłaszczony.

Klasyfikator to MLP z rozmiarami warstw odpowiednio $400, 120, 84, 10$, używający sigmoidy jako funkcji aktywacji.

```
(feature_extractor): Sequential(
    (0): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): Sigmoid()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
    (4): Sigmoid()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Flatten(start_dim=1, end_dim=-1)
)
(classifier): Sequential(
    (0): Linear(in_features=400, out_features=120, bias=True)
    (1): Sigmoid()
    (2): Linear(in_features=120, out_features=84, bias=True)
    (3): Sigmoid()
    (4): Linear(in_features=84, out_features=10, bias=True)
)
```



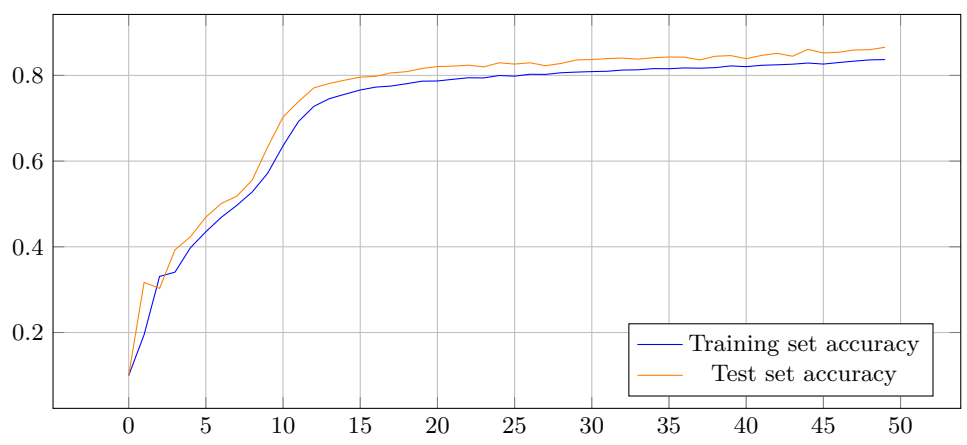
(a) Wykres zmian accuracy



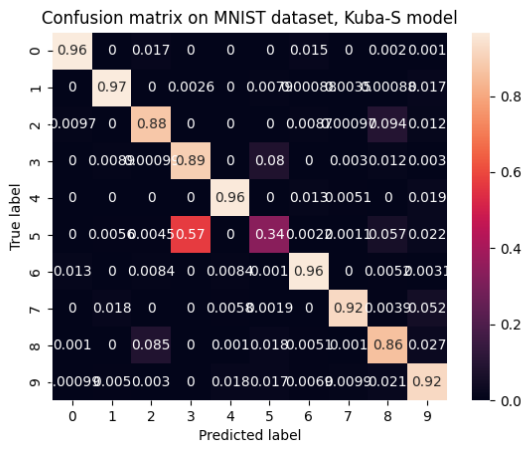
(b) Macierz pomyłek

Rysunek 1: Wyniki dla 1 architektury

Druga architektura, prowadząca do ekstrakcji 2 cech



(a) Wykres zmian accuracy

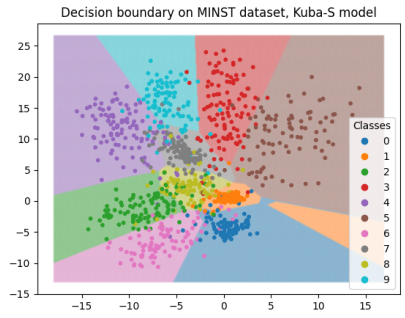


(b) Macierz pomyłek

Rysunek 2: Wyniki dla 2 architektury

W drugiej architekturze warunkiem koniecznym jest rozmiar wektora cech, wynoszący 2 cechy. Jest to umotywowane chęcią stworzenia wizualizacji granic decyzyjnych. Ekstraktor cech składa się z następujących warstw:

```
(feature_extractor): Sequential(
    (0): Conv2d(1, 10, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): ReLU()
    (2): Dropout2d(p=0.1, inplace=False)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(10, 5, kernel_size=(5, 5), stride=(1, 1))
    (5): ReLU()
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (7): Conv2d(5, 2, kernel_size=(3, 3), stride=(1, 1))
    (8): ReLU()
    (9): Flatten(start_dim=1, end_dim=-1)
    (10): Linear(in_features=18, out_features=2, bias=True)
)
(classifier): Sequential(
    (0): Linear(in_features=2, out_features=5, bias=True)
    (1): ReLU()
    (2): Linear(in_features=5, out_features=10, bias=True)
)
```



Rysunek 3: Granica decyzyjna dla arch. 2

W tym modelu, aby jeszcze bardziej zmniejszyć ilość cech, zastosowano jeszcze jedną warstwę konwolucyjną. Finalnie, ekstraktor cech zawiera jedną w pełni połączoną warstwę, w celu redukcji do 2 cech.

W tym modelu postanowiono również przeprowadzić eksperyment z użyciem warstwy dropout w celu poprawienia odporności na overfitting. Pomimo, że z racji na dużą liczność zbioru treningowego, w eksperymencie 1 nie ma dużego ryzyka przetrenowania, sytuacja może ulec zmianie w eksperymencie 2. Negatywnym efektem użycia warstwy dropout jest spowolnienie procesu uczenia, co jest dobrze widoczne na wykresie.

Warstwy konwolucyjne prowadzą do finalnego tensora o wymiarach $2 \times 3 \times 3$, 10więc na końcu ekstraktora cech znajduje się jeszcze jenda warstwa linear, tworząca 2-wymiarowy wektor cech.

Klasyfikator zawiera jedną warstwę ukrytą o rozmiarze 5 neuronów, z funkcją relu, a następnie warstwę wyjściową o rozmiarze 10 neuronów.

2 Eksperyment 1: Architektury sieci splotowej dla MNIST (Magdalena Pakuła)

Pierwsza architektura

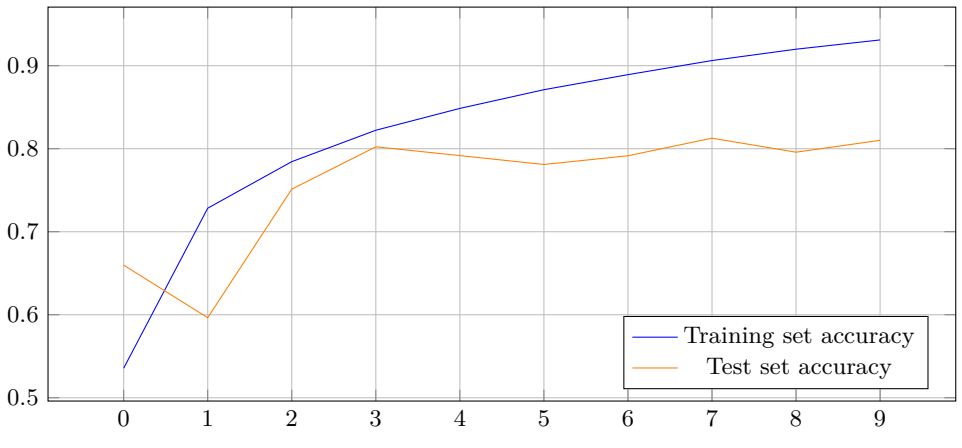
Pierwsza architektura sieci składa się z trzech warstw splotowych, po każdej następuje warstwa aktywacji ReLU i warstwy maksymalnego łączenia, a także w pełni połączona warstwa do klasyfikacji. Wybór trzech warstw splotowych zapewnia równowagę pomiędzy wystarczającą głębokością do uchwycenia złożonych cech a uniknięciem nadmiernej złożoności, która może prowadzić do nadmiernego dopasowania lub wysokich kosztów obliczeniowych. Po każdej warstwie splotowej następuje aktywacja ReLU i warstwa maksymalnego łączenia. ReLU pomaga we wprowadzeniu nieliniowości, niezbędnej do uczenia się złożonych wzorców, podczas gdy max-pooling zmniejsza wymiary przestrzenne, zmniejszając w ten sposób obciążenie obliczeniowe i pomagając w wyodrębnieniu dominujących cech. Zastosowanie jąder 5×5 z wypełnieniem w pierwszych dwóch warstwach zapewnia zachowanie wymiarów przestrzennych, umożliwiając sieci nauczenie się lepszych hierarchii przestrzennych w obrazach wejściowych. Architektura ta osiąga dokładność 98,96% w przypadku danych testowych przy minimalnej stracie testowej wynoszącej 0,0337 (dla najlepszego modelu, tzn. w 5 epoch).

Druga architektura, prowadząca do ekstrakcji 2 cech

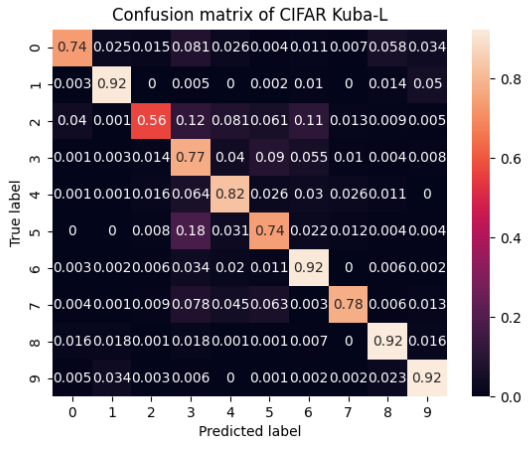
Druga architektura to bardziej kompaktowy CNN zaprojektowany w celu zmniejszenia wymiarowości do 2 cech przed klasyfikacją. Ten prostszy model ma na celu sprawdzenie, jak dobrze może działać wysoce skompresowana reprezentacja. W tym przypadku zostały zastosowane. Architektura rozpoczyna się od dwóch warstw splotowych, w których zastosowano mniej filtrów (16 i 8) w porównaniu do pierwszego modelu. Ta redukcja ma na celu zmusić sieć do wyodrębnienia najważniejszych informacji. W warstwie w pełni połączonej elementy (392) są skompresowane do dwóch, co testuje zdolność sieci do destylacji informacji w minimalną reprezentację. Następnie te dwie cechy są odwzorowywane na dziesięć wyników klas, przy czym warstwa aktywacyjna ReLU pomaga zachować nieliniowość w procesie transformacji cech. Architektura osiąga dokładność danych testowych wynoszącą 81,07%, przy stracie testowej wynoszącej 0.7146 (dla najlepszego modelu, tzn. w 5 epoch), oferuje ona wgląd w skuteczność redukcji wymiarowości.

3 Eksperyment 1: Architektury sieci splotowej dla CIFAR10 (Jakub Pawlak)

Pierwsza architektura



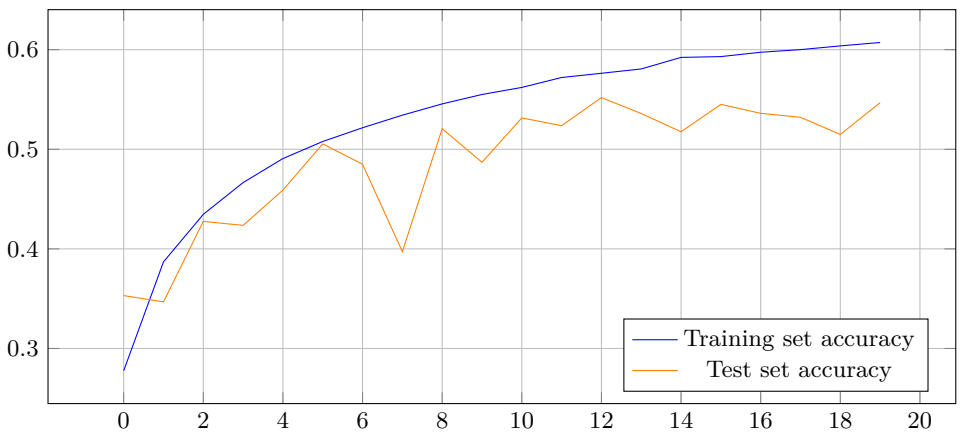
(a) Wykres zmian accuracy



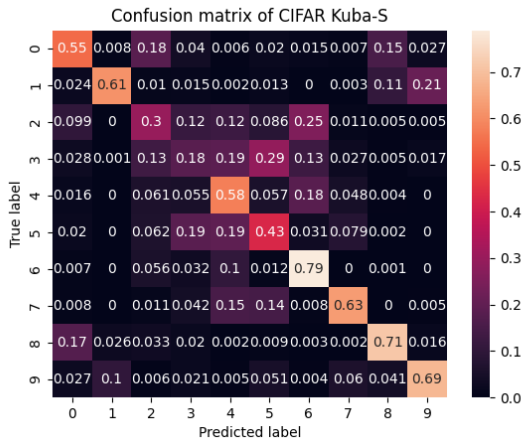
(b) Macierz pomyłek

Rysunek 4: Wyniki dla 1 architektury

Druga architektura prowadząca do ekstrakcji 2 cech



(a) Wykres zmian accuracy



(b) Macierz pomyłek

Rysunek 5: Wyniki dla 2 architektury

```
(feature_extractor): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(64, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (8): Conv2d(32, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU()
    (10): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (11): Conv2d(8, 2, kernel_size=(4, 4), stride=(1, 1))
    (12): Flatten(start_dim=1, end_dim=-1)
)
(classifier): Sequential(
    (0): Linear(in_features=2, out_features=16, bias=True)
    (1): ReLU()
    (2): Linear(in_features=16, out_features=10, bias=True)
)
```

4 Eksperyment 1: Architektury sieci splotowej dla CIFAR10 (Magdalena Pakuła)

Pierwsza architektura W przypadku tej architektury zostały wybrane dwie warstwy splotowe, po których następują warstwy maksymalnego łączenia w celu zmniejszenia próbkowania map obiektów. Pierwsza warstwa splotowa posiada 16 filtrów o wymiarach 3×3. Druga warstwa splotowa posiada 32 filtry o wymiarach 3×3. Funkcje aktywacji ReLU są stosowane po każdej warstwie splotowej. Po warstwach splotowych następuje warstwa w pełni połączona z 64 neuronami, po której następuje końcowa warstwa w pełni połączona do klasyfikacji z liczbą neuronów równą liczbie klas (10 dla CIFAR-10). Wybór takiej architektury pozwala na uchwycenie na obrazach zarówno cech niskiego, jak i wysokiego poziomu. Głębsze warstwy mogą uczyć się bardziej złożonych wzorów i reprezentacji. Funkcje aktywacji ReLU są używane po każdej warstwie splotowej w celu wprowadzenia nieliniowości, umożliwiając sieci poznanie złożonych relacji między cechami w danych. W pełni połączona warstwa przed końcową warstwą klasyfikacyjną ma mniej neuronów w porównaniu z warstwą poprzednią. To zmniejszenie złożoności pomaga w nauce bardziej zwartej reprezentacji przestrzeni cech, potencjalnie redukując nadmierne dopasowanie i obciążenie obliczeniowe.

Druga architektura prowadząca do ekstrakcji 2 cech Druga architektura jest dosyć podobna do pierwszej, tzn. pierwsza wartwa wygląda tak samo, lecz druga warstwa posiada 16 filtrów o wymiarach 3×3. Po warstwach splotowych następuje warstwa w pełni połączona, zawierająca tylko 2 neurony. Następnie jest końcowa, w pełni połączona warstwa do klasyfikacji z liczbą neuronów równą liczbie klas

5 Eksperyment 2: Wyniki dla MNIST

Najlepsza architektura

Najelpsza architektura prowadząca do ekstrakcji 2 cech

6 Eksperyment 2: Wyniki dla CIFAR10

Najlepsza architektura

Najelpsza architektura prowadząca do ekstrakcji 2 cech

7 Analiza i wnioski

Porównanie architektur sieci splotowych

Wpływ augmentacji danych