W ćwiczeniu do stworzenie enkodera i dekodera dla danych obrazkowych ze zbioru MNIST najpierw wykorzystujemy sieć gęstą, podzieloną na sekwencję kodującą i dekodującą, w trakcie kodowania zmniejszamy w kolejnych warstwach liczbę wyjść, w warstwie dekodującej zwiększamy ją systematycznie do rozmiaru początkowego:

Wyniki:

Po 10 epokach:

 > 

Po 20 epokach:

 > 

Po 30 epokach:

 > 

Model nauczył się dekodować zakodowane obrazy jednak widać utratę informacji, przez co obrazki zdekodowany wydają się rozmyte, nie widać też znaczącej poprawy przy dłuższym treningu.

W kolejnej części należało wykorzystać sieci splotowe do tego samego zadania, tak samo zmniejszamy a następnie zwiększamy rozmiar obrazu, w tym przypadku należy uwzględnić odpowiednie wzory na zmianę rozmiaru wyjściowego dla warstw:

Conv2d – zmniejszających rozmiar, stosowane już we wcześniejszych laboratoriach.

ConvTranspose2d: powiększające rozmiar, zgodnie ze wzorem:

Output Height=(H−1)×stride−2×padding+kernel\_size Output Width=(W−1)×stride−2×padding+kernel\_size

Wyniki:

Po 10 epokach:

 > 

Po 20 epokach:

 > 

Po 30 epokach:

 > 

Dla sieci koewolucyjnych skuteczność jest zauważalnie lepsza, jednak widać artefakty na zdekodowanych obrazach.

Ostatnim zadaniem było wytrenowanie modelu na innych wybranych danych, wybrałem zbiór danych CIFAR10 i przystosowałem model splotowy do wymiarów 32 na 32, odpowiednio manipulując rozmiarem filtra, oraz parametrami: padding i stride, zgodnie ze wzorem podanym wyżej.

Wyniki:

Po 10 epokach:

 > 

Po 20 epokach:

 > 

Po 30 epokach:

 > 

Dla zbioru cifar wyniki są raczej słabe, widać tylko plamy kolorów, ciężko doszukać się kształtów nie znając oryginalnych obrazów, dla bardziej złożonego problemu powinno się zastosować większą sieć z większą liczbą wag, powinno to pomóc, ale nie należy oczekiwać że na wyjściu otrzymamy dokładnie te same dane bez żadnej straty.