

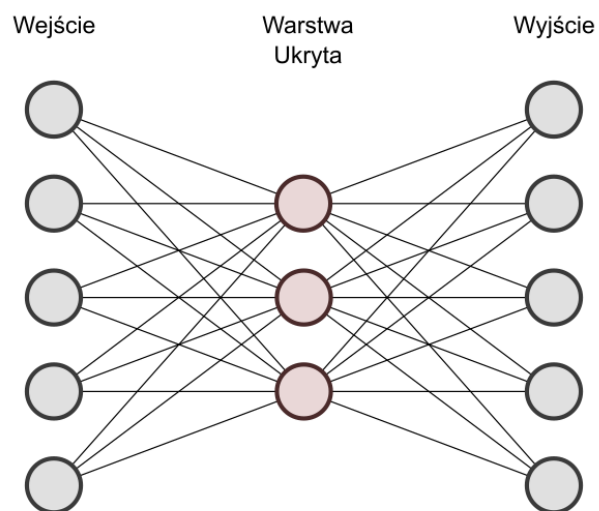
1. Elementy rozwiązania:

Przeprowadzenie badań w celu określenia zasad i zakresu wykorzystywania funkcji autoenkodera z oprogramowania MATLAB.

2. Wprowadzenie teoretyczne; charakterystyka autoenkodera:

Autoenkoderem nazywa się sposób budowania sieci neuronowych szkolonych w celu odtworzenia na wyjściu danych podanych na wejściu. Kompresja danych przy użyciu autoenkodera polega na znalezieniu możliwie najmniejszej ilości cech jak najlepiej reprezentujących treść danych na wejściu drogą uczenia maszynowego i zakodowanie danych jako wartości tych cech. Autoenkoder jest też w stanie odkodować obraz korzystając z przekształceń odwrotnych do tych, które zostały użyte do kodowania.

Neurony w sieciach neuronowych są podzielone na warstwy; zbiory neuronów o połączeniach reprezentujących liczne przekształcenia liniowe, które w przypadku autoenkodera pozwalają na transformację danych z wielu neuronów na wejściu sieci na możliwie niewielką ilość neuronów w warstwie ukrytej.



Ilość neuronów warstwy ukrytej autoenkodera decyduje o ilości cech, którymi opisany będzie obraz. Większa ilość neuronów warstwy ukrytej pozwala na zwiększenie celności, z jaką autoenkoder jest w stanie odtworzyć obraz po kodowaniu, ale zwiększa zarówno rozmiar zakodowanych danych (ilość cech) jak i rozmiar samej sieci. Dla kompresji obrazu rozmiar warstwy ukrytej powinien pozwalać na odtworzenie obrazu o możliwej jakości z jak najmniejszej ilości zakodowanych cech.

Jakość odtworzonego obrazu można też poprawić innym sposobem. W ramach uczenia maszynowego autoenkoder generuje jak najlepsze połączenia w sieci neuronowej poprzez proces optymalizacji – minimalizacji różnicy między danymi na wejściu i wyjściu sieci. Kolejne iteracje autoenkodera będą poprawiać wynik działania, jednak efektywność takiego treningu w upływie czasu spada w tempie zbliżonym do wykładniczego.

3. Autoenkoder w programie MATLAB

Toolbox Deep Learning programu Matlab zawiera wszystkie narzędzia potrzebne do zaprezentowania działania autoenkodera. W celu przyspieszenia obliczeń użyto dodatkowo Parallel Computing Toolbox. W programie zostały wykorzystane poniższe funkcje:

- trainAutoencoder()

Pozwala na wygenerowanie autoenkodera wytrenowanego określonymi danymi. Można dostosować ilość iteracji, ilość neuronów w warstwie ukrytej i inne parametry. Z użyciem Parallel Computing Toolbox parametr 'UseGPU' ustawiony na 'true' pozwala na generację nowych iteracji sieci neuronowej w szybszym tempie, z wykorzystaniem zasobów karty graficznej komputera.

- encode()

Pozwala na zamianę danych wejściowych na zbiór cech. Ta funkcja zostanie użyta do kompresji obrazu.

- decode()

Pozwala na zamianę zakodowanych cech na odtworzony obraz. Ta funkcja zostanie użyta do dekompresji obrazu.

4. Wykorzystanie autoenkodera w programie MATLAB

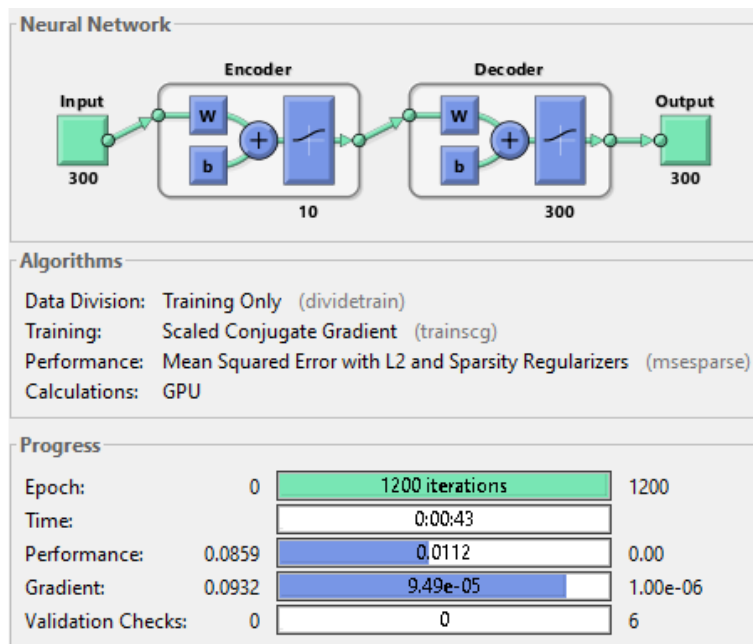
Do wytrenowania sieci neuronowej potrzebna jest możliwie największa ilość względnie różnorodnych danych. Ponadto, trenowanie sieci elementami o dużym rozmiarze zwiększa ilość neuronów w sieci, co pogarsza jakość kompresji danych przy założeniu, że autoenkoder jest dołączony do danych skompresowanych.

Rozważając powyższe ograniczenia opracowano sposób pozyskiwania danych do kompresji. Pozyskano pojedynczy obraz i podzielono go na elementy o rozmiarze 10x10 pikseli. Taka metoda pozwala na łatwe uzyskanie dużej ilości wzorców o indywidualnie małym rozmiarze.

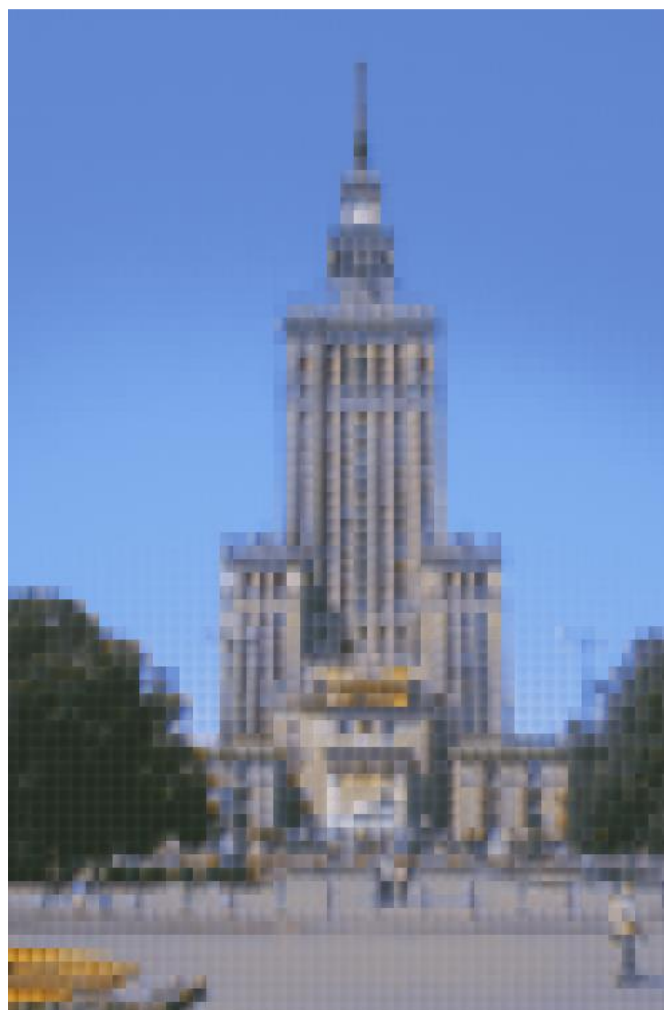


Obraz oryginalny

Autoenkoder został wytrenowany powyższymi danymi. Każdy element 10x10 pikseli został zakodowany, a następnie odkodowany.

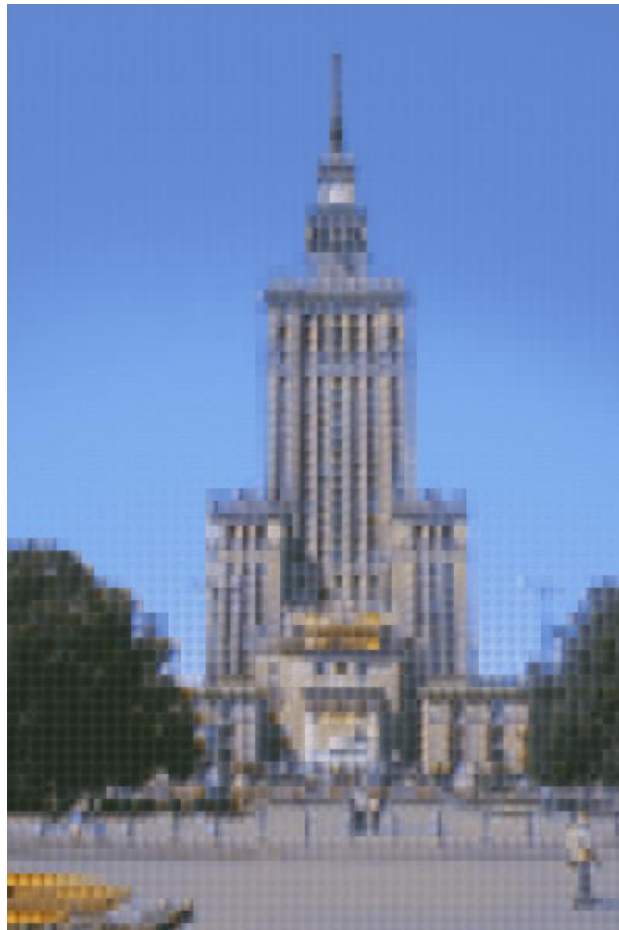


Wykaz wyników z treningu sieci neuronowej



Odkodowany obraz

W celu poprawienia jakości obrazu każdy z odkodowanych elementów został poddany filtracji zaostrażającej, a następnie zrekonstruowany obraz został rozmyty metodą Gaussa.



Odkodowany obraz po filtracji

Za pomocą komendy whos... w konsoli Matlab sprawdzono rozmiar danych zakodowanych, wejściowych i autoenkodera. Poniżej podano otrzymane rozmiary w bajtach:

Obraz wejściowy: 9000000

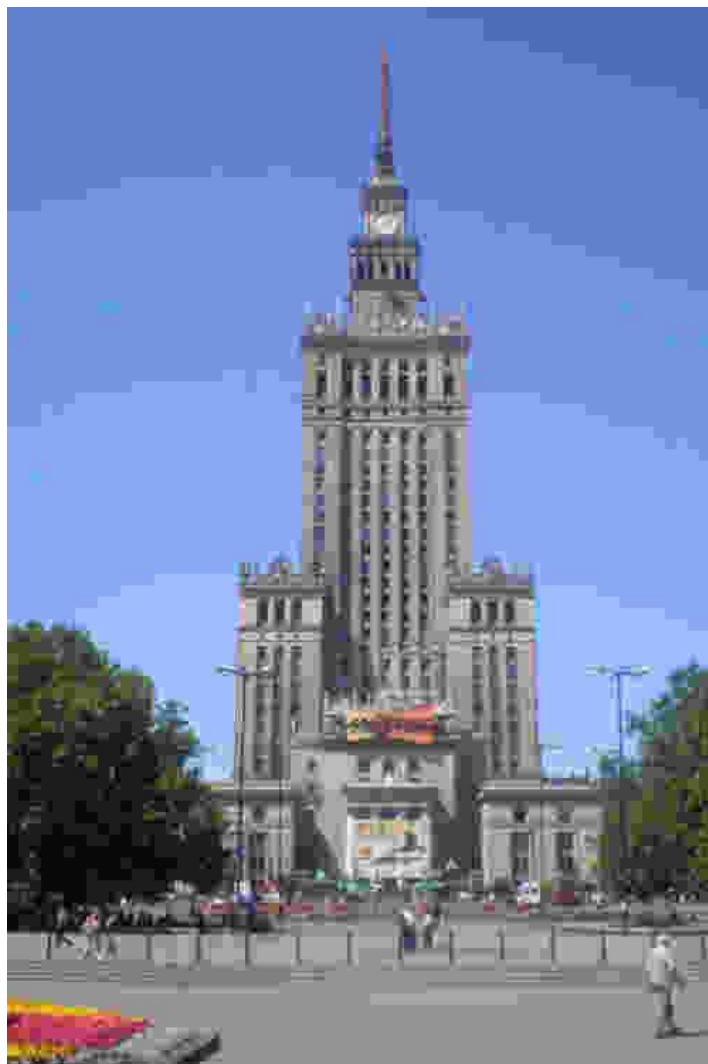
Dane zakodowane: 300000

Autoenkoder: 186331

Kod wykorzystany w zadaniu zawarty jest w pliku Projekt_Jan_Zygadło_KDM.m.

5. Wnioski

Stosunek rozmiaru obrazu wejściowego do rozmiaru danych zakodowanych w zadaniu wynosi **30:1**. Stosunek rozmiaru obrazu wejściowego do rozmiaru danych zakodowanych i autoenkodera wynosi poniżej **20:1**. Jakość obrazu po dekompresji nie jest dobra. Pojawiły się liczne artefakty, ale mimo to, obraz jest rozpoznawalny. Sieć neuronowa nauczyła się odtwarzać elementy obrazu z podanych dziesięciu różnych cech dla każdego bloku 10x10 pikseli. W celu względnej oceny jakości kompresji zapisano obraz w popularnym formacie .jpeg o poziomie kompresji odpowiadającym jakości obrazu zdekodowanego autoenkoderem.



Obraz zapisany w formacie .jpeg

Obraz .jpeg ma rozmiar 8659 bajtów. Stosunek danych przed i po kompresji dla obrazu wynosi **ponad 1000:1**. Na obrazie skompresowanym w ten sposób lepiej widoczne są obrysy mniejszych kształtów, chociaż nieco gorsze jest odwzorowanie kolorów.

Porównując kompresję za pomocą autoenkodera i standardową metodą .jpeg można stwierdzić, że kompresja sztuczną inteligencją jest metodą o niszowym zastosowaniu. Kompresja za pomocą autoenkodera rzadko będzie skuteczniejsza od innych metod o podobnej stratności. Metody kompresji .jpeg i podobne zostały stworzone z myślą o postrzeganiu obrazów przez ludzi i sposobów, w jaki interpretujemy detale, za to autoenkoder nie zawsze optymalizuje swój trening w kierunku kodowania wszystkich niezbędnych cech obrazu.