Sprawozdanie - Autoenkoder

Elementy Inteligencji Obliczeniowej

Autorzy: Piotr Tomaszewski
Damian Tabaczyński

1. Opis problemu

Celem projektu jest rozwiązanie problemu kolorowania obrazów czarno-białych za pomocą sztucznej sieci neuronowej. Podstawowym założeniem było zastosowanie głębokich sieci neuronowych przyjmujących na wejściu jeden kanał informacji reprezentujący skalę szarości poszczególnych pikseli. Wyjściem takiej sieci neuronowej miał być obraz 3-kanałowy w dowolnym modelu barw, będący w pełni pokolorowany.

W rozwiązaniu została użyta architektura typu *Autoencoder* składająca się z modelu *Encodera* oraz *Decodera*. Jest to sieć, w ogólności, służąca do jak najlepszego odwzorowywania swojego wejścia. Działanie *Autoencodera* opiera się wpierw na skompresowaniu informacji do najważniejszych cech przetwarzanych danych za pomocą *Encodera*, następnie zostaje użyty Decoder do jak najlepszego zrekonstruowania danych wejściowych za pomocą cech przekazanych z *Encodera*.

W rozwiązywanym problemie zostały również poruszone następujące kwestie:

- badanie wpływu różnych rozmiarów sieci neuronowych
- badanie wpływu poszczególnych elementów regularyzacji na wynik, np.:
    - operacja Batch Normalization
    - parametr dropout rate
    - weight decay - dodanie członu regularyzacji L2 do funkcji straty

W ramach projektu należało stworzyć i dobrać odpowiednie dane wejściowe, zaprojektować model sieci oraz za pomocą dostępnych narzędzi przetrenować stworzoną sieć.

2. Opis kroków wykonanych do rozwiązania problemu

Pierwszym krokiem w przygotowywanym projekcie było stworzenie odpowiedniego zbioru danych treningowych oraz testowych. W tym celu została użyta biblioteka ***tensorflow_datasets***, z której wykorzystano początkowo dataset ***downsampled_imagenet/64x64***. Zawierają się w nim zdjęcia o dowolnej tematyce w rozmiarze 64x64 piksele. Jak się później okazało było to złym rozwiązaniem. Ogromna różnorodność zdjęć uniemożliwiła skuteczną naukę sieci neuronowej w stosunkowo krótkim czasie. W związku z tym, po przetestowaniu kilku różnych zbiorów danych, zdecydowano się

na wykorzystanie zbioru *Landscapes dataset*. Zbiór ten zawiera zdjęcia krajobrazów - zarówno naturalnych, jak i miejskich. Ze względu na znaczące różnice w kolorystyce obu typów krajobrazów, zdecydowano się na wykorzystanie jedynie krajobrazów naturalnych. Zawężenie zbioru uczącego poprzez wyselekcjonowanie konkretnej specyfiki zdjęć umożliwiło lepszą naukę sieci. W procesie uczenia wykorzystano 2173 zdjęcia, 80% z nich przeznaczone zostało na zbiór treningowy, pozostałe na testowy.

Równoległym etapem było zaprojektowanie sieci w architekturze Autoencoder. Po głębszej analizie zrezygnowano całkowicie z warstw w pełni połączonych na rzecz warstw splotowych. Takie rozwiązanie pozwoliło na dokładniejsze wychwycenie cech obrazów oraz odwzorowanie ich oryginalnych kolorów. Początkowo użyto trzech warstw sieci splotowych w Encoderze mających odpowiednio 8, 16, 32 filtry, natomiast w Decoderze 4 warstwy z filtrami odpowiednio 32, 16, 8, 3. Wszystkie warstwy posiadały kernele o wielkości 3x3, padding=same, funkcje aktywacji relu. W ostatniej warstwie sieci splotowej w Decoderze została użyta funkcja aktywacji tanh. Do obliczania gradientu wykorzystano funkcję straty Mean Squared Error.

W późniejszej fazie testów zwiększono liczbę filtrów oraz zmieniono strukturę sieci. Ostatecznie zdecydowano się na czterowarstwową strukturę Encodera, z kolejno 16, 32, 64, 64 filtrami. Z kolei decoder otrzymał 5 warstw, z kolejno 64, 64, 32, 16, 3 filtrami. W pierwotnej wersji każda warstwa Encodera posiadała strides o rozmiarze 2x2, natomiast po każdej warstwie Decodera następował upsampling. Powodowało to znaczący spadek jakości wyjściowego obrazu, więc w finalnej wersji ograniczono ich występowanie do minimum, wykorzystując strides i upsampling jednokrotnie. Całkowita rezygnacja z nich nie była możliwa, gdyż sieć nie mieściłaby się w pamięci RAM kart graficznych w komputerach autorów. Wymagałoby to więc zmniejszenia liczby warstw lub filtrów, co jak pokazały przeprowadzone próby skutkowałoby większym spadkiem jakości. Podczas analizy problemu jako algorytm optymalizujący został użyty algorytm ADAM.

W trakcie trenowania modelu została również zastosowana standaryzacja danych wejściowych. Wszystkie wartości zostały przeskalowane, by ich wartości należały do przedziału [-1,1]. Poprawiło to wyniki, względem zastosowanego pierwotnie przedziału [0, 1] - co nie jest niczym zadziwiającym, biorąc pod uwagę zastosowanie na wyjściu funkcji aktywacji tanh. Dodatkowo do reprezentacji danych obrazu użyto modelu barw YCbCr. Zamodelowana przestrzeń kolorów wykorzystuje trzy typy danych: Y – składową luminancji, Cb – składową różnicową chrominancji Y-B, stanowiącą różnicę między luminancją a niebieskim, oraz Cr – składową chrominancji Y-R, stanowiącą różnicę między luminancją a czerwonym. Kolor zielony w tym modelu jest uzyskiwany na podstawie tych trzech wartości. Dzięki takiemu użyciu kanał Y (odpowiadający tak naprawdę za skalę szarości) został wykorzystany jako wejście dla sztucznej sieci neuronowej. Przy takim podejściu, wyjście sieci neuronowej mogłoby mieć zaledwie dwa wymiary: Cb, Cr, natomiast czarno-biały obraz wejściowy mógłby być bezpośrednio wykorzystany jako kanał Y. W projekcie zdecydowano się jednak na trójwymiarowe wyjście, aby spełnić wymagania.

Ostatnim krokiem było dodanie regularyzacji do zaprojektowanej sieci neuronowej. Użyto do tego między innymi:

- warstwy normalizujące dane w obrębie jednego batcha (batch normalization)

- warstwy Dropout, która losowo wyłącza z obliczeń losowe neurony
- dodatkowego członu regularyzacji L2 dodawanego do funkcji straty

Wszystkie wymienione techniki regularyzacji mogą potencjalnie uchronić uczoną sieć przed przeuczeniem (*ang. overfitting*).

3. Analiza zastosowania metod regularyzacji

Testy były przeprowadzane na zbiorze obrazów o rozdzielczości 128x128 pikseli w 20 epokach. Liczba epok została dobrana w ten sposób, gdyż przy tej rozdzielczości obrazów, nieregularyzowana sieć dawała po niej akceptowalne rezultaty na zbiorze testowym i proces uczenia był zazwyczaj ręcznie przerywany.

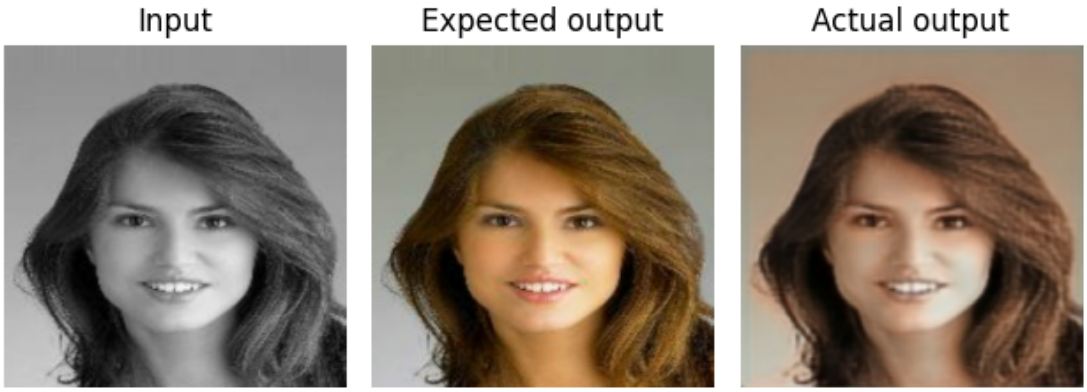
Batch normalization	Rozmiar batcha	Momentum	Dropout rate	Weight decay (L2)	MSE po ostatniej epoce	MAE po ostatniej epoce
Nie	60	-	0	0	0.00989	0.06867
Nie	60	-	0.2	0	0.01656	0.09676
Nie	60	-	0.35	0	0.02775	0.13007
Nie	60	-	0.5	0	0.02149	0.11307
Nie	60	-	0	0.0001	0.03213	0.07798
Nie	60	-	0	0.0005	0.06018	0.07088
Nie	60	-	0	0.0008	0.06686	0.07575
Tak	20	0.5	0	0	0.00908	0.06897
Tak	40	0.5	0	0	0.01525	0.08407
Tak	60	0.5	0	0	0.01016	0.07004
Tak	60	0.2	0	0	0.01080	0.07485
Tak	60	0.7	0	0	0.01478	0.08115
Tak	60	1.0	0	0	0.06373	0.16454

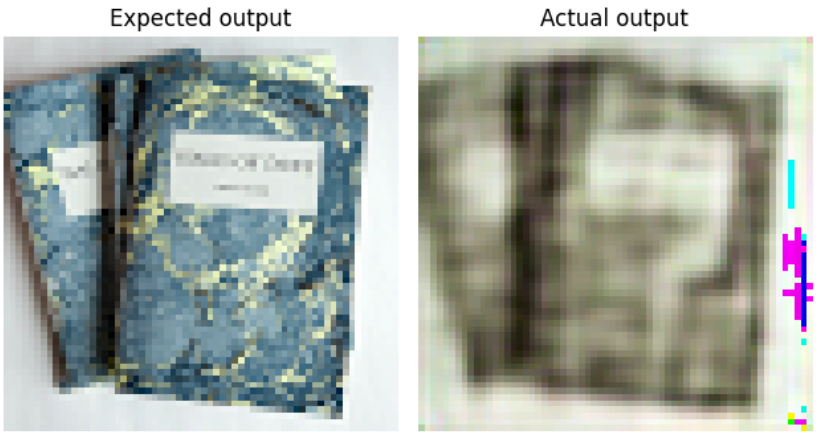
W pierwszym wierszu tabeli zaprezentowane są wyniki dla nieregularyzowanej sieci neuronowej. Jak można zauważyć, był to jeden z najlepszych uzyskanych wyników dla zbioru testowego. Sugeruje to, że na tym etapie nauki sieć nie ulegała jeszcze zjawisku przeuczenia i wykorzystanie metod regularyzacji jedynie spowalniało proces uczenia. Być może zawdzięczać to można stosunkowo dużej liczności zbioru treningowego. Najprawdopodobniej, gdyby zbiór treningowy był mniej liczny lub istniała konieczność

trenowania sieci przez większą liczbę iteracji zastosowane metody regularyzacji przyniosłyby oczekiwane skutki i spowodowałyby zmniejszenie błędu generalizacji, kosztem błędu treningowego.

4. Problemy w implementacji

Dużym wyzwaniem okazało się sprawienie, by sieć neuronowa stosowała różnorodne kolory. W przypadku zbyt obszernej dziedziny przykładów uczących, sieć nie była w stanie nauczyć się struktury przedstawianych jej obiektów, co powodowało wypełnianie całego obrazu jednym kolorem, który w ogólności minimalizuje funkcję straty (sepia, odcienie brązu). Jednym z rozważanych potencjalnych rozwiązań było rozbudowanie struktury sieci neuronowej. Przeprowadzone próby zdawały się potwierdzać tę tezę, niestety techniczne ograniczenia w postaci niewielkiego rozmiaru pamięci RAM kart graficznych w komputerach obu autorów nie pozwoliły na znaczną rozbudowę sieci. Ostatecznie, problem został rozwiązany poprzez wspomniane już ograniczenie dziedziny problemu.
Poniżej przedstawione zostały próby wykorzystania tworzonej w ramach projektu sieci na zbiorze twarzy celebrytów oraz ***downsampled_imagenet/64x64***.





5. Przykładowe wyniki

Przykładowe wyniki dla zbioru testowego przy wykorzystaniu nieregularyzowanej sieci neuronowej działającej na obrazach 256x256 pikseli uczonej przez 40 epok.

Input

Expected output

Actual output

Input

Expected output

Actual output