

Zastosowanie Curriculum Learning dla dotrenowywania ResNet18 (ResearchProposal)

13 czerwca 2024

1.

Hipoteza

ResNet18, dotrenowana na zbiorze RSSNC7 za pomocą Curriculum Learning będzie lepsza niż ResNet18, dotrenowana na zbiorze RSSNC7 za pomocą Transfer Learningu. Prawdopodobnie uczenie z użyciem Curriculum Learning będzie szybsze niż z użyciem Transfer Learningu.

2.

Jakie eksperymenty

*Resnet18(ImageNet) - to Resnet18 przetrenowana na ImageNet

*ResNet18(DTD) - to Resnet18 przetrenowana na DTD.

(a) Dotrenujemy ResNet18(ImageNet) i ResNet18(DTD) na zbiorze RSSNC7 za pomocą Curriculum Learning na kilka sposobów:

i. z ręcznym podziałem zdjęć na 3 kategorie (Easy, Medium, Hard).
Dzielić będziemy się starać dość równomiernie, czyli żeby w Easy było po około 135 zdjęć każdej klasy (Grass, Field, Industry ...), w Medium po około 160 - 170 z każdej klasy oraz w Hard - około 90 - 100.

Następnie będzie 3 części uczenia sieci konwolucyjnej: najpierw do osiągnięcia wystarczającej dokładności dotrenujemy naszą sieć na zdjęciach kategorii Easy, potem analogicznie na Medium, i finalnie na Hard.

ii. z zastosowaniem jednej z wersji self-paced learning, czyli sieć automatycznie będzie wybierała ze wszystkich zdjęć w zbiorze treningowym tylko te, które są wystarczająco łatwe według niej na tym etapie uczenia się. Ta faza uczenia się będzie regulowana za pomocą parametru λ . Na samym początku lambda będzie się równać 0.1. Potem w każdej epoce sieć na początku w trybie eval będzie każdemu batchu z całego zbioru treningowego przyporządkowywać *loss* i sortować według *loss*. Potem na podstawie parametru lambda zostanie stworzony nowy zbiór treningowy

(odpowiedni dla danej fazy uczenia się), który będzie zawierał λ % najłatwiejszych zdjęć (według loss). W procesie uczenia λ będzie zwiększana do 1, czyli do trenowania sieci na całym zbiorze. * jeszcze na razie nie możemy dokładnie określić "jak często" i o ile trzeba będzie zwiększać λ , to trzeba dostosować do naszych danych.

- (b) dotrenujemy jeszcze raz nasze ResNet18(ImageNet) i ResNet18(DTD) na RSSNC7 w takich samych warunkach jak i za pomocą Curriculum Learningu. Trzeba będzie dodać obliczenie metryki-KNN (o której poniżej) po każdym kilku epokach trenowania.

3.

W jaki sposób zmierzyć czy się udało?

- (a) będziemy trenować wszystkie 6 sieci w tych samych warunkach (na przykład na kaggle stosując GPU T4x2), żeby mieć możliwość porównywać czasy trenowania.
- (b) dla każdej sieci będziemy po epokach liczyć train/test loss, train/test accuracy oraz metrykę-KNN
 * metryka-KNN polega na tym, że będziemy brać na przykład zdjęcie i , szukać dla niego k najbliższych sąsiadów i patrzeć ile z tych k sąsiadów ma taki samy label (oryginalny), jak i zdjęcie i (niech takich sąsiadów będzie m). I będziemy zdjęciu i przyporządkowywać $p_i = \frac{m}{n}$. Analogicznie zrobimy dla wszystkich zdjęć, które były użyte w trenowaniu sieci w danej epoce. Obliczymy metrykę-KNN jako średnią p_i po wszystkich zdjęciach, uczestniczących w trenowaniu w danej epoce. Jeszcze nie wiemy dokładnie jaką metrykę zastosujemy dla szukania najbliższych sąsiadów, ale może euklidesową.

4. wstępny podział pracy:

- (a) podział zdjęć ręcznie - dziewczyny i chłopaki
- (b) skrypt do podziału danych ręcznych - Rafał
- (c) ogarnąć metrykę-KNN - Kuba
- (d) transfer learning (dodać metrykę) - dziewczyny
- (e) curriculum learning z ręcznie podzielonymi danymi - dziewczyny/chłopaki
- (f) curriculum learning (self-paced learning) - Kasia