



Zadania Olimpijskie

Michał Karp i Bartosz Trojan

Pizza time

Autor: Bartosz Trojan





Założenia:

Bardzo proste zadanie

Z wektora cech wybrać 5 najważniejszych i użyć ich do klasyfikacji

Hiperparametry - jedna z trudności, to serio bardzo istotne

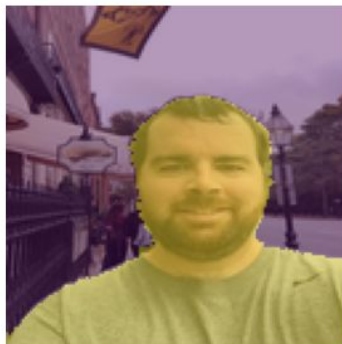
Punktacja: max 100pkt za accuracy ≥ 0.87 , min 0pkt za accuracy ≤ 0.72 - pomiędzy tymi wartościami wynik rośnie liniowo



Bracie, wytrenuj sobie segmentację

Autor: Michał Karp

Czym jest segmentacja?





MSE Loss

$$l_n = (x_n - y_n)^2$$

VS

$$l_n = -w_n [y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)]$$

$$H(P) = - \sum_{x \in C} P(x) \log P(x)$$

BCE Loss



Założenia:

2 warianty - ViT lub bez ViT'a

Dice coefficient do oceny

Dataset: [hngnngn/portrait-segmentation-128x128](#)

Bez ViT-a - bardzo proste, pobaw się gotowymi architekturami

Z ViTem - sprawa się komplikuje

Ocenianie

Metryką jest Dice coefficient. Wzór na Dice coefficient to:

$$\text{Dice} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Gdzie:

- TP – True Positives (prawidłowo przewidziane piksele obiektu)
- FP – False Positives (piksele tła przewidziane jako obiekt)
- FN – False Negatives (piksele obiektu przewidziane jako tło).

Jeśli Dice coefficient jest poniżej 0.7, otrzymasz 0 punktów. Jeśli powyżej 0.98, otrzymasz 100 punktów. Pomiędzy tymi wartościami wynik rośnie liniowo.

Inne popularne metryki to F1 score, accuracy itp, ale my posłużymy się tą.

Biedne reprezentacje

Autor: Bartosz Trojan





Założenia:

14% accuracy - dużo? Niedużo?

5 obrazów na klasę, 100 klas i dużo obrazów bez etykiety.

Semi supervised learning - FixMatch

Unsupervised learning + prosty klasyfikator - SimCLR

Klasy **[0-4, 5-9, 10-14, 15-19, 20-24, 25-29]** muszą być oddalone

Sposób oceny

Twoje rozwiązanie będzie oceniane na podstawie dwóch metryk:

1. Accuracy - dokładność przyporządkowania klas do obrazków. **Accuracy musi być osiągnięte tylko i wyłącznie z użyciem wygenerowanych przez ciebie embeddingów. W przeciwnym razie, wygenerowane embeddingi byłby bezużyteczne.**
2. Average Groups Embeddings Distance - czyli metryka którą wymyśliłem na potrzeby zadania, polegająca na tym, że dla każdej klasy liczymy uśredniony embedding, potem dla wcześniej wskazanych grup klas: [0-4, 5-9, 10-14, 15-19, 20-24, 25-29], liczymy *cosine similarity* pomiędzy każdą jedną parą embeddingów klas w danej grupie i uśredniamy to cosine similarity dla każdej grupy osobno, a potem uśredniamy te wyniki dla każdej grupy pomiędzy sobą co daje nam liczbę z zakresu $[-1,1]$ reprezentującą uśrednione *cosine similarity* pomiędzy grupami. My chcemy by embeddingi w każdej z grup były możliwie jak najbardziej różne, niezależne, więc w tym zadaniu, będziecie dążyli by ta metryka była możliwie bliska zera.

Metryki oceny rozwiązania

1. Accuracy Score (A)

Accuracy przekształcana jest do wartości w przedziale $[0, 1]$ według zasad:

- Jeśli $accuracy < 0.14$, to $A = 0$
- Jeśli $accuracy > 0.19$, to $A = 1$
- Jeśli $0.14 \leq accuracy \leq 0.19$, to wartość rośnie liniowo

2. Average Groups Embeddings Distance (G)

Mierzy średnią *cosine similarity* wewnątrz zdefiniowanych grup klas. Dążymy do wartości bliskiej 0 (czyli do wewnętrznej ortogonalności embeddingów).

Niech `AGED` oznacza obliczoną średnią wartość *cosine similarity* w grupach, a `|AGED|` to jej wartość bezwzględna.

Score przekształcany jest do wartości w przedziale $[0, 100]$ według zasad:

- Jeśli `|AGED| > 0.7`, to `G = 0`
- Jeśli `|AGED| < 0.2`, to `G = 100`
- Jeśli `0.2 ≤ |AGED| ≤ 0.7`, to wartość rośnie liniowo

3. Ostateczny wynik

Końcowy wynik `S` to iloczyn obu skalowanych wyników:

$$S = A * G$$

gdzie:

- (`A` należy do: $[0, 1]$)
- (`G` należy do: $[0, 100]$)
- (`S` należy do: $[0, 100]$)

Spokojnie, dostarczyłem funkcję liczącą wynik na dole notebooka

Te małe szkodniki powróciły

Autor: Michał Karp





Ocenianie

Metryką oceny jest accuracy: cp/ap Gdzie cp to liczba prawidłowo odgadniętych tokenów w całym zbiorze, a ap to liczba wszystkich tokenów do odgadnięcia.

Dla accuracy poniżej 0.7 otrzymasz 0 punktów, a powyżej 0.9 otrzymasz 1 punkt. Pomiędzy tymi wartościami wynik rośnie liniowo.

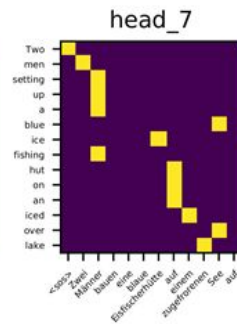
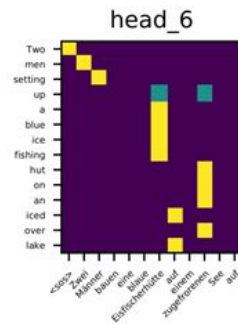
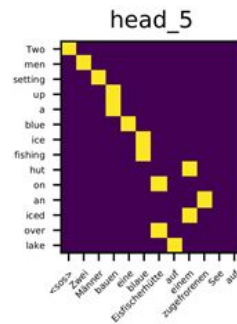
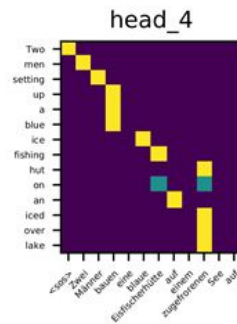
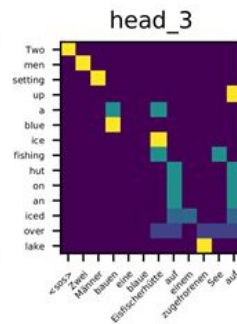
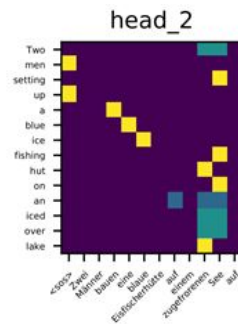
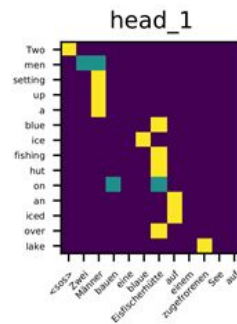
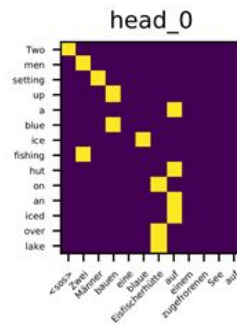
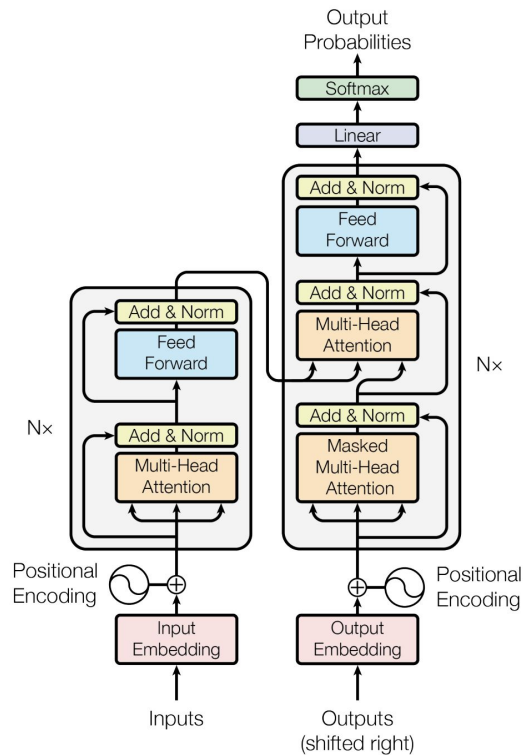


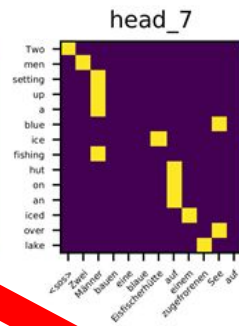
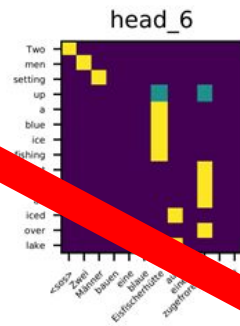
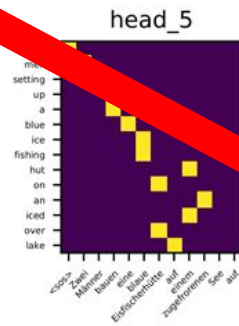
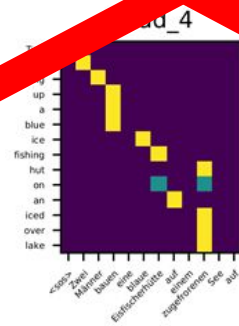
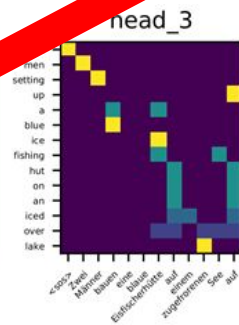
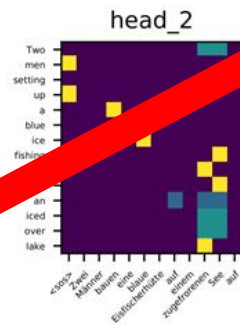
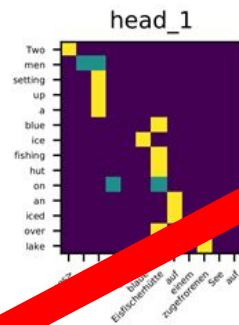
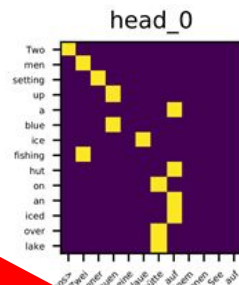
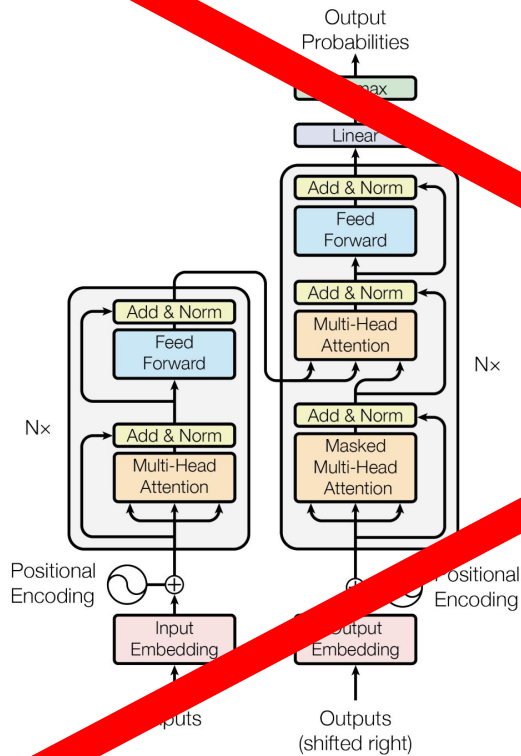
Założenia:

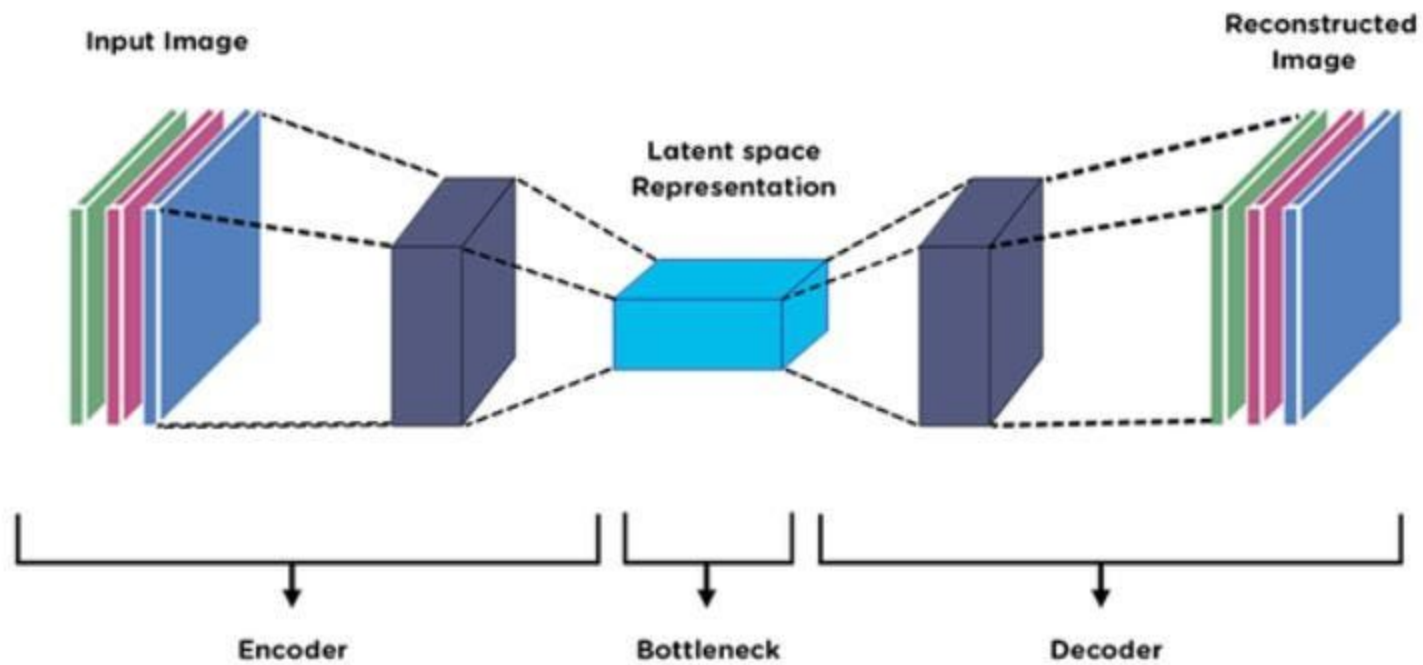
Fairseq2 - straszna biblioteka

Encoder - decoder, ale czy na pewno?

Odgadnąć, jakie słowa zostały zamaskowane







[github.com/MrFishPL/
AIOlympiadClub-final-project](https://github.com/MrFishPL/AIOlympiadClub-final-project)

Powodzenia!