Projekt 4

Uczenie maszynowe 2021

W tym projekcie będziemy trenować sieć typu transformer do zadania *language modeling* na Panu Tadeuszu. Skorzystamy z modelu HerBERT, pretrenowanego na dużym korpusie języka polskiego, który będziemy fine-tunować na naszym zbiorze danych.
Dozwolone jest korzystanie z dowolnych bibliotek, w szczególności PyTorch, transformers i datasets.
Do kodu należy dołączyć oświadczenie o samodzielności rozwiązania.
Deadline: ~~**10 maja, 23:59**~~ **20 maja, 23:59**.

Pretrenowany model

Należy skorzystać z następującego kodu, by załadować pretrenowany model wraz z tokenizerem:

```
from transformers import RobertaForCausalLM, AutoTokenizer

model = RobertaForCausalLM.from_pretrained("allegro/herbert-klej-cased-v1", is_decoder=True).to("cuda")
tokenizer = AutoTokenizer.from_pretrained("allegro/herbert-klej-cased-tokenizer-v1")
```

Dane

Obowiązuje zbiór danych z Pana Tadeusza:

- Trening: Scalony tekst 10 ksiąg (ponad 370k znaków)
- Walidacja: 11 księga
- Test: 12 księga

Uwaga: tym razem nie polecam korzystać z kodu do wczytywania danych który wrzucałem ostatnio, lepiej wykorzystać funkcjonalności bibliotek datasets oraz transformers.
Do wstępnego załadowania i stokenizowania tekstu należy użyć poniższego kodu. Uwaga: jeśli wczytają Państwo dane inaczej (np. nie dzieląc na linijki, co robi funkcja load_dataset), istnieje szansa że tokenizacja będzie błędna i wyniki będą nieporównywalne!
Po użyciu poniższego snippetu należy jeszcze skleić i odpowiednio pobatchować dane.

```
ds = datasets.load_dataset("text", data_files={
   "train": "pan_tadeusz_1_10.txt",
   "validation": "pan_tadeusz_11.txt",
   "test": "pan_tadeusz_12.txt",
})

def tokenize_function(examples):
   return tokenizer(examples["text"])

tokenized_datasets = ds.map(tokenize_function, batched=True, num_proc=1, remove_columns=["text"])
```

Wymagania:

- Po każdej epoce należy odnotować (w logach tekstowych lub wykresach) aktualne perplexity na zbiorze treningowym i walidacyjnym i wypisać próbkę tekstu zainicjalizowanego kontekstem: “Jam jest Jacek”. Samplowanie powinno być stochastyczne. Mogą się państwo zainspirować linkiem i poeksperymentować z metodami samplowania.
- Wymagane perplexity na zbiorze testowym: **168** (ten próg nie jest bezpośrednio porównywalny z poprzednim zadaniem; tym razem w alfabecie mamy 50560 tokenów).

Uwaga: Zastosowany przez nas tokenizer zamienia końce linii na specjalny token `tokenizer.sep_token`. Proszę zwrócić uwagę na to przy wypisywaniu wygenerowanego tekstu i odpowiednio przetworzyć ten token.