AutoKeras review

Adam Frej

1 Wstęp

Code review funkcji trenującej zbiór danych przy pomocy AutoKerasa przygotowanej przez grupę Tojada.

2 Ocena

2.1 Czy ten kod osiąga cel, który postawiono?

Funkcja przyjmuje zbiór danych objaśniających i kolumnę objaśnianą oraz zwraca model, wynik na zbiorze testowym i czas trenowania. A więc podstawowa funkcjonalność jest spełniona.
Kod dzieli dane na train, test losowo. Brak możliwości własnego zdefiniowania podziału, kroswalidacji, albo chociaż random state'a (jest ustawiony jeden na stałe).
Nie ma też możliwości wyboru metryk. Funkcja zwraca accuracy, recall, precision i F1 score. Jednak wszystkie wyniki są takie same, więc prawdopodobnie poprawnie liczone jest tylko accuracy.
Funkcja nie wykonuje żadnego preprocessingu, co bezpośrednio nie jest potrzebne do frameworku. Jednak odkryłem, że jeżeli zmienna celu jest typu 'category' to AutoKeras się nie uruchamia. Po ręcznej zmianie typu na 'object' kod zadziałał jednak zwrócił wyjątkowo niski wynik 0.47 'accuracy'. Po zamianie wszystkich zmiennych objaśniających typu 'category' na 'object' funkcja polepszyła wynik do 0.6, co sugeruje, że framework (oraz oceniany kod) zupełnie nie radzi sobie z typem 'category'.

Podsumowując najważniejszy cel jest spełniony. Jednak brakuje podstawowych funkcjonalności, metryki źle działają oraz preprocessing nie jest wystarczający do każdorazowego uruchomienia frameworka.

2.2 Czy w kodzie są jakieś oczywiste błędy logiczne?

Wszystkie metryki wyliczane są przy pomocy wbudowanych funkcji sklearn'a z argumentem *average='micro'*, co prawdopodobnie jest przyczyną błędnego liczenia. Poza tym kod zbudowany jest logicznie.

2.3 Czy patrząc na wymagania zawarte podczas prezentacji są one w pełni zaimplementowane?

Autorzy chcieli przeprowadzić benchmark z openml i im się to udało, więc osiągnęli podstawowy cel. Byli w stanie zwalidować wyniki przy pomocy accuracy i ocenić framework mierząc czas. Jednak nie przeprowadzili zadanej kroswalidacji z benchmarku.

2.4 Czy kod jest zgodny z istniejącymi wytycznymi stylistycznymi? (czy kod jest zgodny z PEP 8)

Kod jest w pełni zgodny z PEP 8. Jedyne błędy stylistyczne to argument 'X' funkcji oraz zmienne 'train_X' i 'test_X' napisane dużą literą (co stoi w sprzeczności z ogólnymi standardami, ale jest często stosowane w uczeniu maszynowym).

2.5 Czy są jakieś obszary, w których kod mógłby zostać poprawiony? (skrócić, przyspieszyć, itp.)

W funkcji są dwie podane linijki:

```
test_y = test_y.apply(pd.to_numeric, errors='coerce').fillna(test_y)
prediction = prediction.apply(pd.to_numeric, errors='coerce').fillna(prediction)
```

Gwarantują one zamianę typów danych liczbowych na 'numeric' przy jednoczesnym pozostawieniu danych kategorycznych nienaruszonych. Następuje to już po użyciu frameworka wyłącznie w celu liczenia metryk. Nie jestem pewien czy jest to w ogóle potrzebne, ale bez wątpienia jest to nieczytelna część kodu obarczona jedynie komentarzem:

```
# następne trzy linijki bo były problemy z typami
```

Być może framework zwracał inne typy od prawdziwych danych. Jednak mi osobiście trochę zajęło zrozumienie tej części kodu.

2.6 Czy dokumentacja i komentarze są wystarczające?

Brak jakiejkolwiek zewnętrznej dokumentacji. Jednak generalnie kod jest czytelny. Poza wymienionym wyżej brak również objaśniających komentarzy. Dla osoby z doświadczeniem i zapleczem do uruchomienia kod jest dosyć łatwy do zrozumienia. Jednak nie wiem czy każdy by go tak odebrał.

2.7 Czy udało się odtworzyć zamieszone przykłady w kodzie?

Odtworzyłem wyniki na zbiorze 'phoneme'. Funkcja bezproblemowo zadziałała. Uzyskałem wynik 0.7557 accuracy, co jest porównywalne z uzyskanym przez autorów 0.7683. Natomiast nie jestem pewien jak interpretować czas. W moim przypadku trenowanie trwało 77.44 s. Autorzy podali czas 2.32 bez jednostki. Jest to dosyć nieprawdopodobny wynik dla sekund

(mimo wszystko uruchamialiśmy na tej samej maszynie). Być może zamienili czas na minuty, co też daje dziwny wynik, ale bardziej akceptowalny. Ja uruchamiałem kod z w pełni zwolnionymi zasobami. Może autorzy trenowali wiele zbiorów jednocześnie.

2.8 Czy udało się użyć przygotowanych kodów na nowych danych?

I tak, i nie. Wytrenowałem framework na nowym zbiorze z openml. Jest to dataset o id=4 i nazwie 'labor'. Tak jak już pisałem w sekcji 2.1 musiałem go trenować dwukrotnie ze względu na dane typu 'category'. Bez zamiany zmiennej celu na 'object' kod nie był w stanie się uruchomić. Następnie uzyskałem accuracy 0.4667 - dosyć niski wynik. Po zamianie wszystkich zmiennych uzyskałem równo 0.6. Zbiór zawierał braki danych jako NaN oraz zmienne liczbowe. Więc bez zewnętrznych operacji można podsumować, że funkcja się nie uruchomiła.

3 Podsumowanie

Funkcja spełnia podstawowy cel, jednak wymaga poprawek. Obsługa danych 'category' (i być może innych niesprawdzonych typów) jest podstawą. Wymaga też poprawy liczenia metryk i podziału na dane testowe. Kod jest napisany czytelnie, ale brakuje komentarzy/dokumentacji. Przynajmniej przy pobieżnym sprawdzeniu wyniki autorów są odtwarzalne.