

Длабоко учење со Keras и Tensorflow

Со користење на Keras со Tensorflow како backend engine решете ја следната задача.

Задача – Класификација на квалитет на вино

Креирајте модел на длабоко учење од целосно поврзани слоеви за класификација на квалитет на вино. Податочното множество преземете го од [UCI страната](#), односно од следниот [линк](#). За оваа задача потребна е да се предвиди квалитетот на Vinho verde белото вино, односно потребна е датотеката **winequality-white.csv**. Ова множество содржи 4 898 инстанци со 12 карактеристики. 11те карактеристики вклучуваат објективни тестови (на пример, вредности на pH), а резултатот за квалитетот е базиран на сензорски податоци (средна вредност од најмалку 3 оценки направени од експерти за вино). Секој експерт дава оценка за квалитетот на виното помеѓу 0 (многу лошо) и 10 (многу добро).

Очекуваниот излез за ова множество е реален број од 0 до 10. За да се трансформира проблемот во проблем на класификација потребно е да го промените очекуваниот излез во категориска вредност така што ќе креирате три класи: лош (0-5), нормален (6), одличен (7-10) квалитет (препорака: по вчитување на карактеристиките нормализирајте ги во ранг од 0 до 1). Почетен код за креирање на класите е даден во скриптата **starter_code.py** на [GitHub репозиториумот](#). Потребно е да се подели податочното множество на множество за тренирање, валидација и тестирање во соодносот 70%:10%:20%.

Бројот на скриени слоеви, структурата на мрежата, бројот на неврони и сите останати параметри потребно е самите да ги дефинирајте. Користете Adam оптимизатор со основните вредности на параметрите за тренирањето.

По тренирањето моделот зачувајте го во датотеката **wine_model.h5**. Дополнително во датотеката **wine_model_weights.h5** сочувајте ги тежините на моделот. Истренираниот модел евалуирајте го со метриките: точност (**accuracy_score**), прецизност (**precision_score**), одзив (**recall_score**) и F1-мерка (**f1_score**).

A.

Како се менуваат перформансите на моделот доколку се промени бројот на неврони? Нацртајте график на кој ќе се прикажат функциите на загуба при тренирање и валидација (во скриптата `visualize.py` на [GitHub репозиториумот](#) дадена е функција за цртање на графикот на функција на загуба).

B.

Како се менуваат перформансите на моделот доколку се промени бројот на слоеви? Нацртајте график на кој ќе се прикажат функциите на загуба при тренирање и валидација.

C.

Како се менуваат перформансите на моделот доколку се додадат слоеви за регуларизација? Нацртајте график на кој ќе се прикажат функциите на загуба при тренирање и валидација.