

HitPredict, a zenei népszerűséget előrejelző mélytanulási háló

HitPredict, the music popularity predicting deep learning network (2019 ősz)

Tóth Dániel, Gyurós Péter

Kivonat—Olyan mélytanulási modell létrehozása Keras és Python alapon, mely egy zeneszám zenei tulajdonságai alapján képes pontosan előrejelezni annak jövőbeni népszerűségét.

Abstract—Creation of a neural network with Python and Keras which can accurately predict the future popularity of a song based only on its musical properties and features.

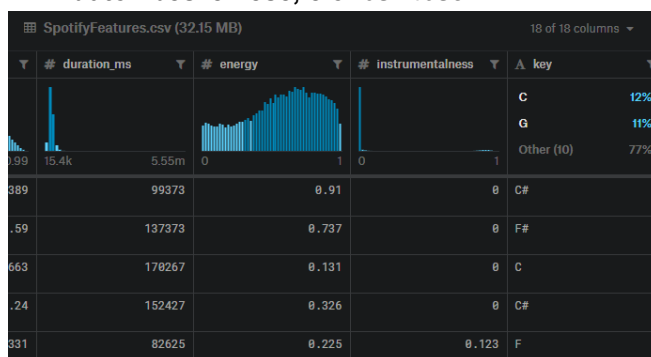
Kulcsszavak—deep learning, Spotify, zene, népszerűség, regresszió, Keras, Python,

1 BEVEZETŐ

A HitPredict a BME VIK „Deep Learning a gyakorlatban Python és LUA alapon” c. tárgyának nagyházifeladat projektjeként készült. Az ajánlott témák közül a „zenei toplista előrejelző”-t választottuk. A zenék toplistán való elhelyezkedését a Spotify, a világ legnépszerűbb zenei streaming szolgáltató alapján mértük. A szolgáltatást annak széleskörű elterjedtségének köszönhetően választottuk. Továbbá a Spotify rendelkezik a legtöbb előadóval és zeneszámmal az iparban, valamint ehhez elérhetőek a legjobb minőségű adataik. A megoldandó probléma alapvetően regressziós, a zeneszámok csupán zeneiségüket tükröző adatai alapján próbáljuk a zeneszám jövőbeni várható népszerűségét előrejelezni. A népszerűséget jelen esetben a várható streamelések számában mérjük, mely alapján sorolják be toplistát is.

2 MEGVALÓSÍTÁS

2.1 Adatok beszerzése, előkészítése



Az adatokat a Kaggle.com honlapról szereztük be, mely a legnagyobb adattudós és gépi tanulási közösség az interneten. A „Spotify Tracks DB” nevű adatbázist választottuk Zahen Hamidani összeállításában. Ez az adatbázis több, mint 230,000 zeneszámot tartalmaz, 26 különböző műfajból, műfajonkénti eloszlása egyenletes. Az szerző a Spotify saját API-jából nyeri ki az adatokat így biztosítva a megfelelő minőséget. Az adatbázis .csv formátumú, melyet a pandas könyvtárral egyszerűen tudunk olvasni és kezelni. A zeneszámok sok tulajdonságát tartja számon a

Spotify, ezek között például: hossz, zenei hangnem, tempó, akusztikusság, hangosság, hangszeresség. A zenei információt nem hordozó adatokat, például a zeneszám belső kulcsát eldobtuk az adatbázisból. Az említett adatok nagy része már float típusú, így ezeket már nem kellett tovább feldolgozni. Az előre jelezni kívánt popularity érték a zeneszámokat streamelési számuk alapján előállított eloszlás alapján sorolja be, tehát egy 0-100-ig tartó érték. A kategorikus adatokat például a hangnemet vagy a műfajt one-hot-encoding-nek vetettük alá, így ezekből a műfajok számosságával megegyező méretű vektort készítünk. Ellenőriztük az adatbázist, van-e benne hiányzó adatpont, melyet be kellene helyettesíteni, szerencsére nincs ilyen. Előfordul azonban zeneszám, mely többször is szerepel, ezek esetén mindig egyet tartottunk csupán meg. Az adatbázist a tanítási, teszt és végül validációs részekre bontottuk. 60%-20%-20% arányban osztottuk fel. Az sklearn preprocessing könyvtárából a StandardScaler függvénnyel standardizáltuk az összes adatoszlopot.

2.2 Tanítás

Első lépésként felépítjük a hálót. Ehhez a Keras keretrendszerben a szekvenciális könyvtárat használtuk. A háló először 4 rétegből áll. A bemenet dimeziója az adatpontok dimenziójának felel meg. A kimenet egy skalár, mely a népszerűséget jelzi elő. A loss függvény regressziós feladat révén a mean squared error-t választottuk. A tanítás megkezdése előtt az adatbázis egy töredékével teszteltük a háló helyes működését. A túltanítás elkerülése végett early stopping módszerrel tanítottuk a hálót 200 epochon keresztül. Az aktuális legjobban teljesítő modell súlyait elmentettük, ha az javult az előzőhöz képest. A háló vázaltos felépítése a következő ábrán látszik:

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 40)	2040
activation_1 (Activation)	(None, 40)	0
dense_2 (Dense)	(None, 30)	1230
activation_2 (Activation)	(None, 30)	0
dense_3 (Dense)	(None, 15)	465
activation_3 (Activation)	(None, 15)	0
dense_4 (Dense)	(None, 1)	16
Total params: 3,751		
Trainable params: 3,751		
Non-trainable params: 0		

2.3 Kiértékelés

A háló első eredményei a 92.5 értéket kaptunk, azaz a modell a kívánt értéktől átlagosan 9.7-es eltérést tapasztaltunk. Ez nem nagyszerű, de valós összefüggést mutat a zenei adatok és a népszerűség között. Ezután hiperparaméter optimalizálásával próbálkoztunk, hogy javítani tudjuk a modell eredményeit. Mivel a témához leginkább egy adatfeldolgozási feladatként álltunk hozzá fontosnak tartottuk, hogy a modell minél pontosabban mutassa a különböző oszlopok közötti összefüggéseket. Azonban a deep learning tudomány jelenlegi állása szerint rendkívül nehéz intuitív magyarázatot adni arra, hogy egy adott bementre miért éppen azt a kimenetet kapjuk, amit. Az viszont biztos, hogy a modell felépítése, a különböző optimalizációs algoritmusok és egyéb hiperparaméternek nevezett változó rendkívül erősen korrelál a neurális háló által szolgáltatott eredményekkel. Azonban ezek megfelelő meghatározása sem triviális feladat. Sőt az egyik legelterjedtebb módszer a különböző hiperparaméterek kipróbálása, az általuk adott eredmények összevetése és ez alapján a legjobb modell megalkotása. A feladat megvalósítása során a hyperas python csomagot használtuk ennek a folyamatnak az automatizálására. Ennek a használatához két függvényt kellett létrehozni, az egyiket az adatok betöltésére a másikat

a modell megalkotására. A másodikban a modell felépítésénél különböző változókat vezettünk be, amelyek közül a hyperas minden tanításhoz mást választott. A hiperparaméterek között szerepelt a rejtett rétegek száma, a rejtett rétegekben lévő neuronok száma, a dropout valószínűsége rétegenként, az aktivációs függvény (ReLU, Leaky ReLU), az optimizer függvény (rmsprop és adam) és a batchek mérete. A hiperoptimalizáció eredményeit loggoltuk egy csv file-ba mely segítségével optimalizálhatjuk a háló felépítését:

l	n_layers	n_layer1_dropout_1	n_layer2_dropout_2	n_layer3_dropout_3	act	optim	n_batch	best_loss
2	40	0.3462695171578595	40	0.3462695171578595	leakyrelu	rmsprop	64	90.3523535114517
3	40	0.21261430843422813	40	0.21261430843422813	relu	rmsprop	128	89.254243335144132
13	40	0.3637788628842422	40	0.3637788628842422	leakyrelu	adam	256	89.5588835877381
12	50	0.2075403595424816	50	0.2075403595424816	leakyrelu	adam	128	89.871548371305127
15	50	0.18781759268378335			leakyrelu	adam	256	189.81245982743796
17	30	0.3441846753786239	30	0.3441846753786239	leakyrelu	rmsprop	256	89.8336632986676
18	30	0.37858227718867114			leakyrelu	adam	64	89.88233387489383
19	30	0.1294828578148945	30	0.1294828578148945	leakyrelu	rmsprop	128	89.8559338868687
10	50	0.83264337582447834	50	0.83264337582447834	leakyrelu	rmsprop	256	85.97862487609776
11	30	0.868518959675492			relu	adam	128	87.3453884130713
12	40	0.3462695171578595	40	0.3462695171578595	leakyrelu	rmsprop	64	90.81512857558268
13	40	0.21261430843422813	40	0.21261430843422813	relu	rmsprop	128	89.8337288688198
13	40	0.3637788628842422	40	0.3637788628842422	leakyrelu	adam	256	89.4182683524513
12	50	0.2075403595424816	50	0.2075403595424816	leakyrelu	adam	128	89.7287891868256
16	30	0.18781759268378335			leakyrelu	adam	256	189.38887178618854
17	30	0.3441846753786239	30	0.3441846753786239	leakyrelu	rmsprop	256	81.21832528885884
18	30	0.37858227718867114			leakyrelu	adam	64	89.8828824813215
19	30	0.1294828578148945	30	0.1294828578148945	leakyrelu	rmsprop	128	89.7951782172385
10	50	0.83264337582447834	50	0.83264337582447834	leakyrelu	rmsprop	256	85.111825987294

3 JÖVŐBELI TERVEK, ÖSSZEFOGLALÁS

A jövőben szeretnénk kiegészíteni a modellt Natural Language Processing hálózattal, mely a zeneszámok dalszövege alapján jelzi előre a népszerűséget. A legtöbb dalszöveg API azonban nem ingyenes vagy nagyon limitált, így ezt végül nem valósítottuk meg.

REFERENCES

- [1] F. Chollet, „Deep Learning with Python”
- [2] Santanu Pattanayak „Pro Deep Learning with Tensorflow”
- [3] Jojo Moolayil, „Learn Keras for Deep Neural Networks”
- [4] Tobias Domhan “Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves”
Available:
<https://www.aaii.org/ocs/index.php/IJCAI/IJCAI15/paper/view/11468/11222>

Gyurós Péter A BME VIK harmadéves villamosmérnök hallgatója,
az EET tanszéken Mikroelektronikai tervezés specializáción

Tóth Dániel A BME VIK BSc Mérnökirformatikus végzős hallgatója,
Rendszertervezés specializáció, AUT tanszék