Gépi tanulás

Képfelismerés gépi tanulásos módszerek alkalmazásával python nyelven

írta: Köő József Témavezető: Czúni lászló

2018

1. Tartalom

[1 Bevezetés 2](#_Toc513584953)

[2 Projekt kezelés 2](#_Toc513584954)

[3 Python programnyelv 2](#_Toc513584955)

[3.1 Python verzió 3](#_Toc513584956)

[3.2 Fejlesztői környezet 3](#_Toc513584957)

[4 Képadatbázis 3](#_Toc513584958)

[4.1 ImageNET 3](#_Toc513584959)

[4.2 CIFAR-10 4](#_Toc513584960)

[5 K-legközelebbi szomszéd 6](#_Toc513584961)

[6 Lineáris osztályozás 6](#_Toc513584962)

[6.1 Optimalizációs eljárások 6](#_Toc513584963)

[7 Neurális hálózatok 6](#_Toc513584964)

[8 Hiba visszaterjesztés 6](#_Toc513584965)

# Bevezetés

Célom a projekt során megismerni, és bemutatni a gépi tanulás néhány módszerét és alkalmazását. Néhány gyakorlati példán keresztül ismertetném az egyes megoldásokat adott problémára.

Forrásként főként a Stanford egyetem egy online elérhető, a CS231n nevű kurzusának anyagát használtam. A kurzus alapvetően a konvolúciós neurális hálókat hivatott bemutatni, de emellett bemutat még más tanulási módszereket is.

A projekt során a fő probléma, amin keresztül bemutatom az egyes tanulási algoritmusokat, egy osztályozás jellegű probléma. Egyszerű képek besorolása néhány kategóriába a tartalmuk alapján. Például egy macskát ábrázoló képet a macskák kategóriába kell besorolnunk.

# Projekt kezelés

A projektet a könnyed kezelhetőség, és követhetőség érdekében Git verziókezelővel menedzseltem. A forrás elérhető a <https://github.com/koojozsef/AI-project-for-university> url címen nyilvánosan. Az alap struktúra szerint az *src/* mappában található a részfeladatokat tartalmazó Python szkriptek. A projekt fő könyvtárában található egy *Readme.md* fájl, amely egy rövid összefoglalót tartalmaz a projektről. Jelen dokumentáció a *data/* mappában található.

# Python programnyelv

Mivel a forrásként megjelölt kurzusban is a Python programnyelvet használják, valamint napjainkban elég előrehaladott, és népszerű nyelv ez, ezért a problémák megoldásait én is Python nyelven prezentálom.

A Python egy általános célú szkript nyelv, amely fordítási időben futtatja a programot. A feladatainkra az teszi igazán alkalmassá, hogy igen erős támogatottsága miatt léteznek olyan könyvtárak hozzá, amelyek igen hatékonyan kezelik a tudományos számítási műveleteket. A numpy könyvtár tartalmazza azokat az eszközöket, amelyekkel igen egyszerű és hatékony kezelni a mátrix műveleteket. Könnyen dolgozhatunk vele többdimenziós adattömbökön, adathalmazokon is.

## Python verzió

A python két fejlesztési ága terjedt el: a 3.x és a 2.7.x verziószámozással jelöltek. A projektem során a 2.7 verziót használtam. A két verzió között néhány függvényben van csupán különbség, valamint kompatibilitási problémák is előfordulhatnak. A[*http://sebastianraschka.com/Articles/2014\_python\_2\_3\_key\_diff.html*](http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html)weboldal rövid áttekintést ad a két irányvonal főbb különbségeiről.

## Fejlesztői környezet

Az Anaconda csomagkezelő asztali grafikus program tartalmaz többféle fejlesztői környezetet. Az Anaconda kezeli a parancssori interfészt, így egyszerűen használhatóvá teszi a fejlesztést, adott fejlesztési környezetekben. Támogatja a JupyterLab, Jupyter Notebook, QTConsole, Spyder, VSCode valamint sok egyéb környezetet alapértelmezetten.

Projektem fejlesztéséhez Spyder környezetet használtam. Számomra kényelmes, és igen hamar tanulható alkalmazás. A felület bal oldalán található alapértelmezetten a szövegbeviteli rész. Itt a készítendő szkript fájlokat lehet látni és szerkeszteni. A jobb oldalon helyet kapott a parancssori kimenet, ahol a program futtatásának eredményét, valamint a hibaüzeneteket figyelhetjük. A jobb oldalon még nyomon követhetőek az aktuális futás során létrehozott változók értékei és szerkezeti felépítésük, valamint típusaik is.

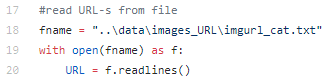
# Képadatbázis

A gépi tanulásos módszerek alapját a tanítási halmaz képezi. Szükséges a megléte egy olyan adathalmaznak, amely már osztályozott elemeket tartalmaz, és rendelkezik olyan címkével, amivel majd az új adatokat az algoritmusunk fog ellátni.

## ImageNET

A képek forrásaként először az Image-net.org weboldalt használtam. Itt elérhető több mint 14 millió kategorizált kép, és kategóriánként van lehetőség a letöltésükre. Van lehetőség továbbá a képek URL-jének letöltésére is. Ezek szintén kategóriák szerint érhetőek el egy-egy szövegfájlban felsorolásként. Ez utóbbi forrást választottam, amihez szükségeltetett egy feldolgozó program, ami letölti automatizáltan a képeket az URL-ek alapján.

A projektemben a *00\_image\_download\_from\_URL.py* fájl tartalmazza a kódot, ami a képek letöltéséért felelős. Az első részletben beolvastam a sorokat a fájlból egy tömbbe.



Így az URL tömb tartalmazza az összes elérési utat, ami macskát ábrázoló képekre mutat.



Egy ciklussal végighaladva a tömbön egyesével letöltöttem a képeket egy meghatározott mappába. A képek nevei és a mappa neve reprezentálta a rendszeremben a címkéket.

A képek letöltésekor azonban probléma lépett fel, ugyanis a fájl tartalmazott sok olyan elérési utat ahol már vagy nem létezett a kép, vagy nem volt elérhető. Így a teljes adathalmaznak csak egy részét tudtam ezzel a módszerrel megszerezni. Továbbá a sikeresen letöltött képek is tartalmaztak fals értékeket, ugyanis néhány oldalon a megszűnt képet egy hibát reprezentáló képpel helyettesítették, így ezt a programom már nem tudta kiszűrni, és címkézett adatként le is töltötte.

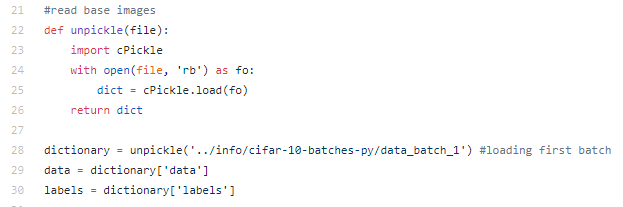
## CIFAR-10

Egy másik megoldást, a CIFAR-10 adatbázis jelentett. Ez egy speciális adathalmaz, ami 60000 darab 32x32 pixel méretű színes képet tartalmaz. A képek tíz osztályba vannak sorolva:

* repülőgép
* autó
* madár
* macska
* szarvas
* kutya
* béka
* ló
* hajó
* teherautó

Ezeket a képeket speciális formátumban lehet elérni. A letöltést követően 6 kiterjesztés nélküli fájlhoz jutunk hozzá. A 6 fájl összekeverve tartalmazza a képeket, mindegyikhez hozzárendelve egy osztályt jelentő számot. Öt fájlt jelöltek meg tanító halmazként, és egy fájlt tesztelő halmazként. Minden fájl azonos hosszúságú, és a következőképpen tárolja a kép adatokat:

Egy fájl 10000 képet tartalmaz. Egy képet egy „sor” reprezentál. Az első bájt az osztályt jelölő címkét tartalmazza (0-tól 9-ig), az utána következő 1024 bájt jelenti a kép vörös pixelkomponenseit, majd 1024 bájt a zöld, és újabb 1024 bájt a kék komponenseket tartalmazza.



A batch fájlok tartalmát a fenti módon olvastam be. Definiáltam egy *unpicle* nevű függvényt, amely bemeneti paraméterként a fájl nevét és elérési útvonalát kapja meg, majd a beolvasott fájlt eltárolja egy programváltozóban és ezt adja vissza függvényhíváskor. Az adathalmazt tartalmazó *dictionary* változó egy speciális adatstruktúra, amely elemeire kulcsszavakkal is lehet hivatkozni. Ez a típus automatikusan, a fájl beolvasásakor generálódik.

# K-legközelebbi szomszéd

A k-legközelebbi szomszéd osztályozó algoritmus egy elég egyszerűen használható módszer az osztályozásra. Az alapvető logikája, hogy tanulás során felveszünk bizonyos pontokat a paramétertérben. A betanított algoritmus ezután a beérkező új adatot elhelyezi a paramétertérben, és megnézi a hozzá legközelebb eső k darab pontot. Ezen pontok, mivel címkézett adatokat használtunk a betanítás során, meghatározzák az új pont osztályát.

# Lineáris osztályozás

Tbd

SVM

## Optimalizációs eljárások

Tbd

Minimalizálni kell a veszteségfüggvényt, ez a tanítás

gradiens alapú minimalizálás

# Neurális hálózatok

Tbd

# Hiba visszaterjesztés

Tbd