BME, VIK, AUT
Témalaboratórium 2017, ősz
Dokumentáció

téma: **Aláíráshitelesítés, Classification**

konzulens:
Kővári Bence Dr.

csapat:
Hanusch Róbert, GIM5E5
Iványi Béla, HLUNLA
Pávlicz András, BTT3C1

2017.12.10.

Tartalomjegyzék

Bevezetés

Az aláíráshitelesítés egy jelenleg is erősen kutatott téma. Cél, hogy az aláírásszakértők általi 1-2%-os hibaarányt elérjük.
Az aláíráshitelesítésen belül két típusról beszélhetünk. Az offlineról, és az onlineról.
Online aláíráshitelesítésről akkor beszélhetünk, amikor a digitális eszközökre tollal, vagy ujjal írunk alá például futárok esetén. Itt már ma is elértük az 1%-os hibaarányt, mivel nem csak az aláírásból készült képfájlból nyerhetünk információkat, de az aláírás folyamatáról (sebesség, lenyomás ereje...). Így az online aláírásban eléggé jól áll az informatika. Az egyetlen probléma, hogy az aláírások döntő hányada papíron van tárolva, és papírra írták. Nem ilyen eszközökre.
Offline aláírásnak nevezünk minden olyan aláírást, aminél csak a beolvasott képfájl elérhető számunkra. Ezeknél nehezítés még, hogy kissebb-nagyobb mértékben zajos a forrásunk. Van, hogy a lapon van egy gyűrődés, amit nem sikerült Simára vasalni, és van, hogy az aláírással megegyező színű nyomda került az aláírásra. Jelenleg 10-20% hibaarány körül mozognak a legjobb algoritmusok, amikről tudunk.

Az osztályozásról

Tehát az offline aláírás az, ami igazán érdekel minket. A probléma egy elég hatékony felosztása a következő:

- ➔ Előfeldolgozás: Ebben a fázisban minimalizáljuk a zajt. A nyomdáktól a gyűrődéseken át a kávéfoltokig a lehető legtöbb fölösleges elváltozást letakarítjuk a képről.
- ➔ Jellemző kinyerés: Itt nyerjük ki az adatokat a képből, és próbáljuk függvényekkel, stratégiákkal számértékenként megadni a kép jellemzőit.
- ➔ Párosítás: Az adott mintákhoz tartozó jellemzőket megpróbáljuk összepárosítani.
- ➔ Osztályozás: Olyan algoritmusok keresése, ami a kinyert adatok alapján jó hatékonysággal tud tippelni.

Az osztályozási problémáknak elég nagy irodalma van. A mi esetünk azért speciális, mert bár 2 osztály közül kell választanunk(eredeti, hamisítvány), csak az eredetiből van mintáink, és abból is kevés (esetünkben 10 tanítómintával dolgoztunk).
Leggyakrabban a spam szűréshez hasonlítják a problémát, érdemes itt kezdeni a keresést.

A lehetőségek közül a mi csapatunk 2 neurális háló, és egy statisztikai alapú osztályozót próbáltunk megvalósítani, majd az eredményüket merge-elni.

Forrásfájl

Forrásfájlként egy Excel táblát kaptunk, melyben 20 aláíróhoz 20 eredeti, és 20 hamis aláírás adatai tartoztak.
Egy aláíráshoz 149 algoritmus kimenet volt megadva, és ezeket a kimeneteket kellett felhasználni az osztályozáshoz.
Az aláírások aláírónként sorba voltak rendezve, 1 aláírónak első 20 aláírása eredeti, a második 20 hamis volt. Az első 10 aláírást kaptuk meg betanításra, a többit tesztként használtuk.

Osztályozás neurális hálóval.

(Iványi Béla)

Bevezetés

Ahogy sok más osztályozást igénylő feladatban, az aláírás hitelesítésben is helyet kap a neurális háló.
Én ebben az irányban indultam el, és úgy döntöttem saját implementációval próbálkozok. Ez lehetőséget adott számomra, hogy a neurális hálókat az alapjaitól megismerjem, azonban így bonyolultabb architektúrák építése túlmutatott volna a saját erőforrás kereteimen.

Első lépések

Első dolgom az volt, hogy létrehozzak egy bármilyen, de működő neurális hálót. Aki a témakörben már keresgélt az interneten, tudja, hogy nagyon sok forrás és segédanyag van, amik közül válogathat. Azonban ezek között vannak ellentmondásosak is.
Első működő neurális háló megtanulja az XOR függvényt.

Mnist és az aláírás hitelesítés

András csapattársam szintén neurális hálókkal foglalkozott a tárgy keretében, de más megközelítésből. Rajta keresztül szereztem tudomást az mnist nevezetű adatbázisról, ami kézzel írott számokról tartalmaz képeket, mind tanítás és tesztelés céljából. Remek lehetős volt számomra, hogy tovább fejlesszem a már kialakított megvalósítást.
Sikerült a számok felismerése és olyan hiba arányokat értem el, amik az mnist esetében elvártak egy jól működő neurális hálótól. Az eredményen felbuzdulva kipróbáltam az aláírásokra is. Egyelőre 10 pozitív és 10 negatív mintán tanítva, de igazán jó eredményt nem sikerült elérni.
Az aláírások esetében ugyanis nincs sok tanító minta és ez korlátozza a neurális háló általánosító képességét. A valódi cél az lenne, hogy 10 pozitív mintán tanuljon.

A tanító halmaz generálása

Feltételezhetünk olyat, hogy az aláírásokban megfigyelhető tulajdonságok eloszlása normál eloszlást követ. Azt is feltételezhetjük, hogy a hamis aláírások, hasonló eloszlást követnek mint a pozitívak, csak nagyobb szórással. Ezekkel a feltételezésekkel élve már könnyű tanító mintákat generálni.

1000 pozitív és 1000 negatív mintát generálunk magunknak minden aláíróra az első 10 pozitív aláírás alapján és Andrással közösen használjuk. Az eredményeim nem bíztatóak, sajnos. A tesztminták közel 50%-át osztályozza helyesen, egy-két aláíró esetén figyelhető meg 55%. Elég egyértelműen az látszik, hogy egyöntetűen dönt egy aláíró esetében az összes tesztre.

Amin lehetne fejleszteni

Azt teljesen figyelmen kívül hagytam, hogy a forrásfájl néhány oszlopa összetartozó tulajdonság. Ezt úgy lehetne kihasználni neurális hálóval, hogy minden tulajdonság oszlopait külön-külön kiértékelném, majd azok eredményeiből állna elő egy globális eredmény.

Osztályozás Neurális Hálóval (Pávlicz András)

Bevezetés

Mint ahogy minden osztályozási problémát, ezt is meg lehet próbálni neurális hálóval megoldani. A kérdés csak az, hogy mennyire lesz pontos és hatékony a többi osztályozóval szemben.
Kezdésnek Horváth Ádám szakdolgozatát olvastam el. Ebből megismertem a témához kapcsolódó fogalmakat és a program alapvető működésének elemeit. A számomra releváns információ az alapokon kívül két/háromféle osztályozó megismerése volt: neurális háló és statisztikai alapú és nagyvonalakban SVM. Ebből általánosan azt a következtetést vontam le, hogy a neurális háló alapú osztályozó nem a leghatékonyabb ezen a szakterületen.
Ennek két fő oka, hogy nagyon kevés számú tanítóminta áll rendelkezésünkre, ezért mesterségesen előállított tanítómintákat kell alkalmaznunk, illetve legtöbbször hamis tanítóminta nem áll rendelkezésünkre, azonban a hálónak szüksége lenne rá a tanuláshoz. Ennek következménye, hogy még pontatlanabb hamis aláírásokat tudunk csak generálni.
Én mégis ezt választottam. A neurális háló szépsége, hogy nem látunk bele a modellbe, amit épít, illetve számos szabad paraméter és változtatható módszer létezik, amikkel finomhangolni lehet a működését, azaz szinte mindig „van hova fejlődni”.

Használt eszközök

A feladat megoldására a Google által fejlesztett TensorFlowt használtam Python nyelv felett. Mindkettő új volt számomra, ezért eleinte csak ismerkedtem a környezettel.
A kódszerkesztéshez Visual Studio Code-ot használtam.

TensorFlow

Ami miatt ezt választottam, az a hatékonysága és támogatottsága, azaz kompetens.
A TensorFlow alapvető elemei a tensorok, ezek egy gráf csúcsaiként képzelhetők el. Több dimenziósak is lehetnek, ez azt jelenti, hogy több adaton képesek végrehajtani a műveleteket.
A programkód két fő részre különül el: először felépítjük a számítási gráfot, utána pedig egy ún. sessionben futtatjuk az adatainkat rajta. Ezt a futtatást a TensorFlow nagyon hatékonyan végrehajtja (GPU támogatással is lehet).

Python

Nem részletezném nagyon, a nyelvnek így is csak az alapjait ismertem meg. Eleinte kisebb nehézséget okozott, hogy nem erősen típusos, és ezért nehezen értettem meg mások kódját, ehhez idővel azonban hozzászoktam.

Első neurális háló: MNIST

Első próbálkozásnak az MNIST adatbázissal próbálkoztam meg. Ebben egyjegyű írott számok (fekete-fehér) képei vannak pixeladatok formájában tárolva (0 és 1 közötti érték: pixel mennyire fekete). Ez be van építve a TensorFlowba, ezért választottam első próbálkozásnak.
Itt lényegében saját kódot nem írtam, csak végigkövettem az útmutatókat. A fő célom a megértés volt, és erre tökéletesen megfelelt.
Maga a probléma nagyon egyszerű, egy lineáris regresszióval már 90% feletti pontosságot lehet elérni. Neurális hálóval (3 rejtett réteggel, rétegenként 500 neuronnal) csak pár százalékot sikerült javítani az eredményen.

Második neurális háló: Boolean függvény

Miután már képes voltam megírni magamtól egy neurális hálót, a következő feladat az volt, hogy a saját adataimmal is tudjam használni. Hogy ne bonyolítsam a feladatot fölöslegesen, egy egyszerű 3 bementes logikai függvényt választottam.
A feladat nehezebbnek bizonyult, mint amire számítottam. Itt mutatkozott meg a Pythonos tudásom hiánya leginkább. A fájlból való adatok beolvasása és ebből a jellemzők („feature”-ök) és elvárt kimenetek („label”-ök) különválasztása nem jelentett különösebb gondot. Ami viszont igen, az a TensorFlownak való átadás. A megoldás egy megfelelő méretű és dimenziójú numpy tömbbé alakítás volt. Miután ez sikerült, a háló sikeresen lefutott.

Harmadik neurális háló: Aláírás hitelesítés

Minden kellő tudás a rendelkezésemre állt, hogy nekilássak az eredeti feladatnak.
A neurális háló lényegében adott volt, felhasználtam a korábbiakat. Pár kisebb módosítás és kiegészítés kellett csak a kódba, hogy fusson. Ezen felül még egy kimeneti fájlt kellett létrehoznom Hanusch Róbert számára, akinek egy algoritmusa a neurális hálók kimenete.
A feladatot az adatok és azokkal való dolgozás tette nehézzé.
Első körben létre kellett hoznunk azokat a generált aláírásokat, amikkel majd a hálót tanítani tudjuk. Ehhez azt feltételeztük, hogy az aláírásból kinyert jellemzők normál eloszlást követnek, illetve hogy a hamis aláírások jellemzői ugyanazzal a várható

értékkel, de kétszeres szórással írhatók le. Ezt Excelben csináltam kézzel, de mivel ketten is használtuk ezeket az adatokat, ezért nem egyedül állítottam elő az egész adatbázist. Egy aláíráshoz négy szöveges fájl tartozik: eredeti/hamis tanító/tesztelő minták.
A következő probléma a hiányos vagy nem értelmes értékek (végtelen). Első futtatáskor emiatt NaN eredményeket kaptam. Úgyhogy megpróbáltam az első aláírással tesztelni úgy, hogy az első 12 jellemzőt veszem, itt mindegyik sorban hasznos adatok voltak, és egészen jó eredményt sikerült elérni (80% körül). Ezzel a problémával sajnos később sem tudtam többet foglalkozni. A nehézség nyilván az, hogy mindegyik aláírásnál más jellemzők vesznek fel hasznos értékeket, még akár mintánként is változhat.

A végleges háló paraméterei:

- 3 db rejtett réteg, egyenként 500 neuronnal
- az első 30 jellemző felhasználása (nem számértékek helyett 0)
- 100-as batch méret
- 30 epoch
- Adam optimalizáló 0.005-ös tanulási rátával

Az eredmények aláírásonként nagyon változók. Például a 9. aláíráson sikerült 97.5%-os pontosságot elérni, azonban a 14. aláíráson csak 47.5%-ot. Ennek fő okai a generálásban és a jellemző kihasználásban lehetnek.

Amiben még lehetne fejlődni

Az időkeret miatt sajnos csak egy kísérleti próbálkozást sikerült létrehozni, semmiképpen sem optimális paraméterekkel.
Jó pár dolgot ezért lehetne javítani a projektben.
A fejlesztés során:

- kódszervezés, újrafelhasználhatóság
- adatok szervezése
- aláírások automatikus generálása

Az eredmények javításához:

- aláírásgenerálási módszerek:
    - jellemzők eloszlásának vizsgálata
    - faktoranalízis: egyes jellemzők közötti korreláció vizsgálata
- jellemzők kihasználása: Egyes aláírásokra számos jellemző értelmetlen (pl. körök mérete), ezeket ki lehetne szűrni aláírónként, hogy azokat a jellemzőket használja, ahol nem csupa 0 érték vagy csupa végtelen szerepel.
- bonyolultabb neurális háló
- neurális háló paraméterezése

Statisztikai osztályozás, “MetaClassification”

(Hanusch Róbert)

Bevezetés

Alapvetően annak szerettem volna utánajárni, hogy több algoritmus eredményéből milyen módon lehet egy végső következtetést levonni.
Ehhez több osztályozó kellett, hogy legyen aminek összevonni az eredményét. Így nekiálltam mini osztályozófüggvényeket csinálni.

Statisztikai osztályozók

A statisztikai osztályozóimnál alapelvnek vettem, hogy oszloponként legyen eredményem. Minden oszlopra adtam egy tippet, ami azt mondta meg, hogy az adott minialgoritmus az alapján az egy oszlop alapján eredetinek tartja-e az aláírást, vagy sem.

Miniosztályozók

A miniosztályozóknál intervallumvizsgálatokat végeztem.
A tanítominták alapján meghatároztam egy intervallumot, majd a tesztelés alatt az adott oszlopra egy igen, nem választ adott vissza a függvény.
A vizsgált intervallumok közt volt egy minimum maximum, ezen kívül középértéktől való maximum távolság is (A középértéknél felfele, lefelé is ugyanaz a d távolság jelentette az intervallum határokat).A középértékek között vizsgáltam a számtani-, mértani-, harmonikus-, és négyzetes átlagot, valamint a mediánt.

A miniosztályozók közül megfelelő súlyozással a mediánnal, és az átlaggal értem el. Ezen felül még az is nagyon pozitív ebben a két algoritmusban, hogy bár nagyon hasonló a pontosságuk, más arányban találják el az eredetit, és a hamisat. Ennek köszönhetően pedig nem kell annyira félnünk a túlzott átfedéstől.

Eredmények:

_____GeometricAverageDistanceB_____	_____HarmonicAverageDistanceB_____
Eredeti: 146/200 73	Eredeti: 158/200 79
Hamis: 256/400 64	Hamis: 249/400 62,25
Avg's AVG: 68,5	Avg's AVG: 70,625

_____MinMax_____ Eredeti: 160/200 80 Hamis: 245/400 61,25 Avg's AVG: 70,625	_____NegyzetesAverageDistanceB_____ Eredeti: 157/200 78,5 Hamis: 259/400 64,75 Avg's AVG: 71,625
_____AverageDistanceB_____ Eredeti: 164/200 82 Hamis: 265/400 66,25 Avg's AVG: 74,125	_____MedianDistanceB_____ Eredeti: 157/200 78,5 Hamis: 280/400 70 Avg's AVG: 74,25

H_1 táblázat

Ami nem fért bele a félévbe, de érdemes utánanézni:

- Szórás
- Lefele, és felfele is az adott irányban max határozza meg az intervallumot

Súlyozás

Elöljáróban annyit tartanék fontosnak elmondani, hogy miniosztályozónként csináltam egy osztályozóalgoritmust. Ezeknél ugyanazt a súlyozást alkalmaztam, így egy elég általános súlyozófüggvény jött létre.
Mivel a forrásban oszloponként más és más az információnyereség, valamilyen módon súlyoznunk kell a kapott eredményeket. Ehhez még mindig csak az a 10 tanítómintánk van, mint a miniosztályozók kalibrálására.

Első próbálkozásom az volt, hogy x mintából számolok intervallumot, majd 10-x értékeli a az oszlop információnyereségét. Így egyedül az átlag alapján számoló osztályozó ért el 68 körüli pontosságot.
Ezt követően az első kettő mintát használtam az intervallumhoz. Majd mintánként teszteltem, és újraszámoltam az intervallumot. Így a tanítás végére 10 mintából számolt intervallum, és 8 mintából kapott súly állt rendelkezésemre.
A következő rész adatbázis specifikus rész, így torzíthatta az eredményeket, és ezért nem is részletezném túl ezt a részt:
Miután megvoltak a súlyok, és a betanított algoritmus, a túlságosan pontatlan eredményeket nem vettem figyelembe, a többit a korábban kiszámolt súllyal vettem figyelembe. Az így elért eredmények a H_1 táblázat tartalmazza.

Lehetséges stratégiai irány (ezek minden aláírás hitelesítő osztályozóra vonatkoznak):

Lehetnek oszlopok, amik nem tárolnak információt, mert pl az előfeldolgozásnál igazították az aláírást, a tulajdonságkinyerésnél pedig pont ezt méri az adott algoritmus.

Lefutásonként lehetne 1-2 mintaaláíró, ami az ilyen oszlopokat kiszűri, így a falspozitívak nagy arányát lehetne mérsékelni. Valamint az osztályozó eredménye is kissebb mértékben függne az előtte található lépések milyenségétől.

MetaClassification (vagy más néven fúziós algoritmus)

A félév végére 3 osztályozónk készült el. Kettő statisztikai alapú (átlag, medián), és egy neurális háló. Mivel a félév végére lettek kész ezek az algoritmusok, nem volt túl sok időm ezzel a témával foglalkozni.
Előre megbeszélt módon, fájlon keresztül kommunikáltak az algoritmusok, mivel így bármilyen számunkra, vagy az adott téma számára kényelmes környezetben tudtunk dolgozni. A fájlban az algoritmus kimenete található, ami aláírónként soronként, vannak a hozzá tartozó aláírások értékelései. A
A konzulens javaslatára a bayes tétellel, és az adatbázison mért pontossággal számoltam, mint valószínűség. Mivel jelenleg 3 eredményt kellett mergelni, nem túl sok látszódik az algoritmus hatékonyságán. Gyakorlatilag a többség dönt-tel egyenértékű, viszont látszik, hogy így is elértünk egy minimális javulást a korábbiakhoz képest.

eredmény:

_____MERGE_____

Eredeti:	159/200	79,5
Hamis:	277/400	69,25
Sum:	436/600	72,6666666666667
Avg's AVG:		74,375

Összegzés

Hanusch Róbert

Az eredményekből érezhető, hogy a statisztikai osztályozással elég egyszerűen, és gyorsan lehet egész jó eredményeket elérni. Viszont egy ponton túl finomítani elég nehézkes, és lassú folyamat.

Iványi Béla

Az eredményeim nem mutatnak áttörést, sőt leginkább semmit nem mutatnak, egy egyszerű pénz feldobással hasonló eredményeket lehetne elérni.
Úgy gondolom azonban, hogy sikerült a neurális hálókról alkotott elképzelésemet bővíteni, és az aláírás hitelesítéssel is megismerkedni.

Pávlicz András

A megoldásom a mostani formájában nem tud konzisztensen jó eredményt produkálni, ezért ilyen formában még nem lehetne alkalmazni semmiképpen sem. A kiegészítésekkel együtt azonban nagyon sokat lehetne fejlődni szerintem.
A cél azonban az volt, hogy megismerjem a neurális hálókat, és ez sikerült, méghozzá a TensorFlowt, ami kompetensnek mondható. Ezen felül Python tudásra is szert tettem, ami mindenképpen hasznos.