

Debreceni Egyetem
Informatikai Kar

Fazekas István

Neurális hálózatok

Debrecen, 2013

Szerző:
Dr. Fazekas István
egyetemi tanár

Bíráló:
Dr. Karácsony Zsolt
egyetemi docens

A tananyag a
TÁMOP-4.1.2.A/1-11/1-2011-0103
azonosítójú pályázat keretében valósulhatott meg.



SZÉCHENYI TERV



MAGYARORSZÁG MEGÚJUL



A projekt az Európai Unió támogatásával,
az Európai Szociális Alap
társfinanszírozásával valósul meg.

Tartalomjegyzék

Előszó	7
1. Bevezetés	9
2. A perceptron	15
2.1. A neuron sémája	15
2.2. Aktivációs függvények	18
2.3. A perceptron tanítása	21
2.4. A perceptron algoritmusának egy változata	25
2.5. A perceptron algoritmusának duális alakja	28
2.6. Lineáris regresszió	28
2.7. Feladatok	30
3. A többrétegű perceptron	31
3.1. A többrétegű perceptron felépítése	31
3.2. A többrétegű perceptron tanítása	33
3.3. A back-propagation eljárás variánsai és tulajdonságai	37
3.3.1. Az aktivációs függvény deriváltjáról	38
3.3.2. Az η tanulási paraméter és a momentum konstans	39
3.3.3. Soros és kötegelt tanítás	39
3.3.4. Heurisztikus megjegyzések a back-propagation algorit- musról	40
3.3.5. Osztályozás több halmaz esetén	40
3.3.6. Az MLP mint univerzális approximátor	41
3.3.7. Általánosítás	41
3.3.8. A konjugált gradiens módszer	42
3.3.9. Kvázi Newton-módszerek	44
3.3.10. A Levenberg–Marquardt-eljárás	44
3.4. A hálózat metszése	46

3.4.1. Regularizáció	46
3.4.2. A Hesse-mátrixon alapuló metszés	47
3.5. Numerikus példák	51
3.6. Feladatok	57
4. A radiális bázis függvények	59
4.1. A szeparálás Cover-féle elmélete	59
4.2. Interpoláció radiális bázis függvényekkel	63
4.3. A Tyihonov-féle regularizáció	66
4.3.1. A regularizációs hálózat	74
4.4. Az általánosított RBF hálózat	75
4.4.1. A regularizációs paraméter meghatározása	78
4.4.2. Tanítási stratégiák	81
4.5. Magfüggvényes becslések	84
4.5.1. A sűrűségfüggvény magfüggvényes becslése	84
4.5.2. A regressziós függvény magfüggvényes becslése	85
4.6. Numerikus példák	86
4.7. Feladatok	90
5. A tartó vektor gépek	93
5.1. A tartó vektor gépek kialakulása	93
5.2. SVM osztályozásra	94
5.2.1. Az optimális hipersík a lineárisan szeparálható esetben	94
5.2.2. Az optimális hipersík a nem szeparálható esetben	101
5.2.3. Az SVM használata nem-lineáris szeparálásra	106
5.2.4. Az SVM tanítása	110
5.3. SVM regressziószámításra	115
5.3.1. Veszteségfüggvények	115
5.3.2. A lineáris regresszió	116
5.3.3. Nem-lineáris függvények közelítése	119
5.3.4. Többváltozós függvények közelítése	123
5.4. Numerikus példák	123
5.5. Feladatok	131
6. Appendix	135
6.1. Néhány matematikai fogalom	135
6.1.1. Vektorok és mátrixok	135
6.1.2. Differenciálszámítás	136
6.2. Mátrixok általánosított inverze és kvadratikusság mini- muma	138

6.3. Optimalizációs technikák	145
6.3.1. A gradiens módszer	145
6.3.2. A Newton-módszer	148
6.3.3. Kvázi Newton-módszerek	150
6.3.4. Levenberg–Marquardt-módszerek	154
6.3.5. A lineáris modell	157
6.3.6. A Gauss–Newton-módszer	158
6.3.7. Lineáris legkisebb négyzetes módszer	160
6.3.8. A Least-Mean-Square (LMS) módszer	161
6.3.9. A konjugált gradiens módszer	162
6.4. Feltételes szélsőérték problémák	168
6.5. Feladatok	171
7. Feladatok megoldása, útmutatások	173
2.7. Feladatok megoldása	173
3.6. Feladatok megoldása	175
4.7. Feladatok megoldása	178
5.5. Feladatok megoldása	183
6.5. Feladatok megoldása	192
Irodalomjegyzék	195
Tárgymutató	201

Előszó

Ez a jegyzet a Debreceni Egyetemen informatikus és matematikus diákok számára tartott Neurális hálók tantárgyhoz készült. Lényegében az előadások anyagát tartalmazza, de a gyakorlatok jelentős részét is lefedi.

A jegyzet által tartalmazott anyag az alábbi (fejezetenként haladva). A Rosenblatt-féle perceptron; a többrétegű perceptron (Multi Layer Perceptron, MLP); a radiális bázis függvények (Radial Basis Function, RBF); a tartóvektor gépek (Support Vector Machine, SVM). Az Appendix a matematikai háttérrel, főleg az optimalizálási módszereket foglalja össze. Számos ábra és kidolgozott példa is segíti a megértést. A fejezetek végén több kitűzött feladat áll, amelyek megoldása, vagy megoldási útmutatója a jegyzet végén található.

Az előadásaim megindulásuktól kezdve elsősorban Haykin átfogó, világszerte használt [18] könyvén alapulnak, így a jegyzet is ezt a művet követi. Viszont az SVM oktatásában folyamatosan az elmélet megalkotójának, Vapniknak [49] monográfiáját követtem, így a jegyzet SVM fejezete is erre támaszkodik. Kisebb speciális részek tárgyalásához más könyvek is szükségesek voltak, illetve pár esetben az eredeti cikkekhez kellett visszanyúlni. A matematikai háttér tárgyalása standard matematika és statisztika könyveken alapul, amelyek közül Fletcher optimalizációról szóló [14] könyvét kell kiemelni. A neurális hálózatok az utóbbi évtizedekben rendkívül gyorsan fejlődő témakör, de a jegyzet keretei között csak a már letisztult, elfogadott módszerek tárgyalására van mód.

A jegyzetben a numerikus példákat a Matlab programcsomag (l. [29]) segítségével dolgoztam ki. Az SVM regresszióhoz használtam a libsvm program-könyvtárat (l. [5] és [6]). A szövegszerkesztés nagy részét magam végeztem \LaTeX -ben. Az ábrák többsége Matlab-bal készült, néhány pedig \LaTeX -ben. A \LaTeX -ben végzett munkában Tómacs Tibor jelentősen kivette a részét.

Köszönetnyilvánítások. Hálás vagyok kollégámnak, Jeszenszky Péternek, hogy évekkkel ezelőtt rábeszélte a Neurális hálók tantárgy elindítására. Több éven keresztül Ő vezette a tárgyhoz tartozó gyakorlatokat. Köszönet illeti a Debreceni Egyetem azon hallgatóit, akiknek a tantárgy iránti érdeklődése fenntartotta bennem is a lelkesedést. Az érdeklődés különösen a témavezetéssel szakdolgozatot vagy diplomamunkát író diákok részéről nyilvánult meg. Ebben a tekintetben ki kell emelnem három volt diákomat.

Horváth Roland matematikus hallgató „A neurális hálók alapjai és alkalmazásai” címmel (lásd [21]), Szép Gábor matematikus hallgató pedig „A neurális hálók egy osztálya: az SVM” címmel (lásd [44]) írt diplomamunkát a témavezetéssel. Mindkettőjüktől az volt a kérésem, hogy a dolgozatuk legyen alkalmas oktatási segédanyagnak is. Ennek megfelelően Horváth Roland a dolgozata első részében a perceptronról, az MLP-ről és az RBF-ről tartott előadásaim anyagát foglalta össze, Szép Gábor pedig az SVM-ről szólókat. Mivel az előadásaim anyaga napjainkban is tartalmazza ezeket a standard anyagrészeket, így a fenti két dolgozatból (a szerzők előzetes jóváhagyásával) több részletet alkalmasan átdolgozva és kibővítve, beillesztettem a jegyzetbe. Kovács György matematikus hallgató „A Support Vector Regresszió elmélete, implementációjának elemzése és alkalmazása” címmel (lásd [25]) szintén a témavezetéssel írt diplomamunkát. Ez a munka már kutatói szintű részeket is tartalmaz. Kovács György ma már a kollégám, ebbe a jegyzetbe Ő ajánlotta fel az 5.8. Példát. A fentiekért ezúton fejezem ki köszönetemet Horváth Rolandnak, Szép Gábornak és Kovács Györgynek.

Köszönet illeti a Debreceni Egyetem Alkalmazott Matematika és Valószínűségszámítás Tanszékének azon munkatársait, akik segítettek a kéziratban lévő nyomdahibákat felfedezni. Sok köszönet jár Tómacs Tibornak a technikai szerkesztésért, Karácsony Zsoltnak pedig a lektorálásért.

A szerző

Debrecen, 2013.

1. fejezet

Bevezetés

Neurális hálózatokat széles körben alkalmaznak tudományos és műszaki feladatok megoldására. Többek között karakterfelismerésre, képfeldolgozásra, jelfeldolgozásra, adatbányászatra, bioinformatikai problémákra, mérés technikai és szabályozástechnikai feladatokra használnak különböző neurális hálózatokat. Meg lehet velük oldani olyan összetett problémákat, amelyek visszavezethetők két alapvető feladatra: osztályozásra (azaz szeparálásra) és függvény közelítésre (azaz regresszió számításra).

Ebben a műben mesterséges neurális hálózatokról lesz szó, de a „mesterséges” jelzőt csak akkor fogjuk kitenni, ha a természetes, azaz biológiai neurális hálózatoktól való megkülönböztetést hangsúlyozni kell. A mesterséges neurális hálózatok kialakítása valójában a biológiai neurális hálózatok tanulmányozására vezethető vissza. Ugyanis a biológiai rendszerek sok területen rendkívül hatékonyan működnek. Például az emberi idegrendszer jeleket (kézírást), képeket (arcokat, embereket), valamint egyéb alakzatokat nagyon sikeresen ismer fel. Vagy gondolhatunk a denevérek kiváló tájékozódási képességére.

A biológiai rendszerek analógiájára próbáltak meg mesterséges rendszereket alkotni. A biológiai rendszerek olyan sajátosságait vették alapul, mint a nagy számú, de egyenként kicsiny alapegységből való felépítés, valamint az egységek közötti sok kapcsolat, és nem utolsósorban a tanulás képessége. Valójában azonban az élő természetből csak bizonyos általános elveket sikerült ellesni, a ténylegesen megvalósított mesterséges neurális hálózatok nem rendelkeznek a biológiai rendszerek mélyebb tulajdonságaival. Egy közvetítőleges meghatározás lehet a következő. *A mesterséges neurális hálózat az idegrendszer felépítése és működése analógiájára kialakított számítási mechanizmus.* Hiszen a fő cél nem elvi, hanem ténylegesen működő modell

létrehozása. Ezt pedig tipikusan valamilyen elektronikai eszközzel (általában számítógéppel) és valamilyen tudományos eljárással (matematikai jellegű modellel) lehet elérni.

Tehát a biológiai elvek (felépítési és működési elképzelések) alapján megalkottak bizonyos matematikai jellegű modelleket. Ezeket elméleti matematikai módszerekkel pontosították, alkalmazott matematikai (numerikus, operációkutatási, statisztikai) módszerekkel számításokra alkalmassá tették, majd számítógépen realizálták. Azonban a matematikai módszerek mellett sokszor heurisztikus megfontolásokra és számítógépes kísérletezésre is szükség van.

Amikor valamilyen tudományos vagy műszaki feladatot akarunk megoldani, akkor bizonyos törvényszerűségekre, illetve modellekre szeretnénk támaszkodni. Ilyen ismert törvényszerűség pl. a tömegvonzás törvénye, a gáztörvények, az atomok felépítésének modellje és természetesen minden fizikai, kémiai, biológiai, ... törvény és modell. Ezek bizonyos határok között jól írják le a jelenségeket. Ezek alapján tudjuk megérteni a jelenségeket, azok jellemzőit kiszámítani, és főleg berendezéseket, folyamatokat tervezni. Vannak azonban olyan esetek, amikor nincsenek használható modellek, nem ismertek a pontos törvényszerűségek, vagy ha igen, akkor azok (pl. bonyolultságuk miatt) számításokra nem alkalmasak. De vannak adataink (méréseink, megfigyeléseink). Ezek alapján pedig megpróbálunk mégis eredményt elérni.

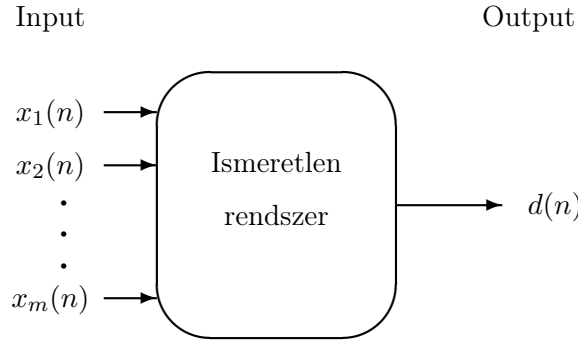
Egy neurális hálózatot érdemes úgy felfogni, hogy nem kívánja a jelenséget modellezni, arra törvényszerűségeket megállapítani (legalábbis a hagyományos természettudományos, matematikai értelemben nem). Hanem a jelenséget fekete dobozként kezeli, csak a bemenő (input) és a kimenő (output) adatokat tekinti. Legyenek tehát a bemeneti adatok

$$\mathbf{x}(n) = (x_1(n), \dots, x_m(n))^T$$

m -dimenziós vektorok, ahol $n = 1, 2, \dots$ az időpillanatokot jelenti. Itt és a továbbiakban a vektorokat oszlopvektornak fogjuk tekinteni, ezekből a transzponálás (jele T) segítségével kapunk sorvektort. Az ismeretlen rendszer minden $\mathbf{x}(n)$ -hez megad egy $d(n)$ kimeneti értéket (lásd 1.1. ábra).

A jó neurális hálózat olyan, hogy ugyanarra az inputra hasonló outputot ad, mint a vizsgált jelenség. De a fekete dobozban működő mechanizmust nem tárja fel, maga a neurális hálózat pedig nem „használt” a jelenségre. Ez a felfogás persze más területen sem ismeretlen, gondoljunk a nem-paraméteres statisztikai módszerekre. És ráadásul bizonyos problémákra nagyon hatékony neurális hálózatos megoldást lehet adni. Tehát a jelenségek fekete dobozként való kezelése a neurális hálózatoknak részben hátrányuk, de részben előnyük

is. Hiszen működésükhöz csak adatokra van szükség. (Természetesen a jelenség bizonyos fokú ismerete és némi tervezési tapasztalat nagyon hasznos jó neurális hálózat megalkotásához.)



1.1. ábra. A modellezendő rendszer

A neurális hálózatok tudományának fejlődése nagyon tanulságos. Mi ebből csak néhány lépést sorolunk fel, a részletek megtalálhatóak pl. Haykin [18] átfogó művében. A kialakuláshoz szükséges volt az idegrendszer felépítésének feltárása. Ezen a téren megemlíthető McCulloch és Pitts [30] neuron modellje, továbbá Hebb [19] tanulással kapcsolatos eredményei. Az idegrendszerrel kapcsolatos kutatások inspirálták a kibernetika (informatika) fejlődését is.

A neurális hálózatok közismert kezdeti típusa az egyetlen neuronból (idegsejtből) álló *perceptron* volt, Rosenblatt [42]. A Rosenblatt–Novikoff-féle perceptron konvergencia tétel (lásd Novikoff [35]) azt állítja, hogy a perceptron képes elválasztani két lineárisan szeparálható halmazt. A következő lépés az *Adaline* megalkotása volt. Ez úgy tekinthető, mint egy lineáris függvény illesztésére alkalmas eszköz. Ennek tanítása a Widrow–Hoff-algoritmus [50], más néven a *Least mean square* eljárás.

Kiderült azonban, hogy több neuront egy rétegbe rendezve sem oldható meg lineárisnál bonyolultabb feladat, lásd Minsky és Papert [32]. Bonyolultabb elrendezést pedig nem tudtak betanítani. Áttörést a *többrétegű perceptron* (Multi Layer Perceptron, MLP) tanítására szolgáló eljárás, a *hiba visszaáramoltatása* (hiba visszaterjesztése, error back-propagation) felfedezése hozott, lásd Rumelhart, Hinton, Williams [43]. Azóta a neurális hálózatok elmélete és alkalmazásai hatalmas fejlődésen mentek keresztül.

Ebben a jegyzetben csak a neurális hálózatok bizonyos típusait mutatjuk be, a gazdag tárházba való további betekintésre ajánljuk Haykin [18] könyvét. Haykin könyve általános szemléletű és átfogó alkotás, számos szakterü-

let művelői forgathatják haszonnal. Magyar nyelven elérhető Horváth Gábor és szerzőtársai [20] műve, amely elsősorban mérnöki szemlélettel rendelkezőknek szól. Ebben a jegyzetben olyan modellekre koncentrálunk, amelyek általánosan használhatóak, ténylegesen megvalósíthatóak, valamint létezik elérhető számítógépes implementációjuk.

A *többrétegű perceptron* (MLP) talán a legismertebb modell, a legtipikusabb neurális hálózat. Rétegekbe rendezett perceptronokból áll. Előrecsatolt hálózat (feedforward network). Igen flexibilisen alakítható ki a struktúrája. Számos numerikus eljárással betanítható. Megoldja a szeparálás és az approximálás feladatát (nem csak lineáris esetben) és az ezekre alapozó számos gyakorlati problémát. Ezek miatt széles körben alkalmazott.

A *radiális alapfüggvényekből álló hálózat* (Radial Basis Function, RBF) az MLP alternatívája (lásd [3]). Az RBF elméleti alapja a matematikai analízisben igen fontos *Tyihonov-féle regularizáció*. Azaz a számos problémakörben jól használható *büntető függvények* segítségével ad használható eljárást. A megoldásban szereplő Radial Basis Function valójában egy magfüggvény. Az RBF mind a célját, mind a megoldás módját tekintve nagyon közel áll a nem-paraméteres statisztikai függvény-bebecslésekhez.

Szintén statisztikai hátterű a *tartó vektor gép* (Support Vector Machine, SVM). Ez nem véletlen, hiszen megalkotója, Vapnik maga is statisztikusként indult. Az SVM bizonyos értelemben megoldja az optimális szeparálás és az optimális approximálás feladatát. Azonban az elméletileg optimális megoldás numerikus megtalálása nagy méretű feladatok esetén problémát okozhat. Az SVM elmélete szervesen kapcsolódik a statisztikus tanulás-elméletéhez, lásd Vapnik [49].

A neurális hálózat alkalmazásának fő fázisai: (1) a modell megalkotása, (2) betanítása, (3) használata. A modellt a konkrét probléma (pl. rendszám-felismerés, képfeldolgozás) mibenléte alapján választjuk az ismert modell osztályokból a saját tapasztalataink alapján. Ha pl. MLP-t használunk, akkor meg kell adnunk annak konkrét specifikációit is. Ezután a modellt betanítjuk, azaz a konkrét adatokhoz illesztjük. A betanított hálózatot gyakran tesztadatokon ellenőrizzük. Ha sikerült betanítani, akkor alkalmazhatjuk nem ismert adatokra. Ez pl. rendszám-felismerés esetén azt jelenti, hogy a műszer előtt elhaladó autók rendszámát outputként szolgáltatja a neurális hálózat.

A neurális hálózatok tanítása alapvető fontosságú. Mi a *tanítóval történő tanulással* (ellenőrzött tanulás, felügyelt tanítás, supervised learning) foglalkozunk. Ez azt jelenti, hogy vannak *tanító pontjaink* (training points), azaz bizonyos $\mathbf{x}(1), \dots, \mathbf{x}(N)$ input adatokhoz tudjuk a $d(1), \dots, d(N)$ out-

put értékeket is. Ezek alapján illesztjük a modellt (ami azt jelenti, hogy megkeressük azon paraméter értékeket, amelyekre a neurális hálózat egy-egy inputra hasonló outputot ad, mint amelyet a tanító pontok alapján az általunk megfigyelt jelenség ad). Ha sikerült a hálózatot jól betanítani, akkor használjuk olyan inputra, amire az outputot nem tudjuk. Éppen a hálózat által adott outputot fogjuk igaznak tekinteni.

Érdemes elidőzni a szóhasználatnál, különösen a tanulás-tanítás kettősége miatt. A jelenségről az $(\mathbf{x}(1), d(1)), \dots, (\mathbf{x}(N), d(N))$ adatok állnak rendelkezésre, azaz a jelenség az n időpillanatban az $\mathbf{x}(n)$ inputra a $d(n)$ outputot szolgáltatva. Ezeket az input-output adatokat a statisztikában mintának nevezik, a természettudományokban ezek a mérési, megfigyelési eredmények. A neurális hálózatok angol nyelvű irodalmában ezt „training set”-nek nevezik, amit egyaránt fordíthatunk tanító halmaznak, ill. tanuló halmaznak. A hálózat szempontjából tekinthetjük úgy, hogy a hálózat tanul a tananyag alapján. Viszont a tanulás mögött mindig van egy számítási mechanizmus, amit az ember alkotott. Így a tervezői, felhasználói aktivitást hangsúlyozandó, a jegyzetben a „tanítás” és a „tanító pontok” elnevezés mellett maradtunk.

A fentiek alapján a tanítás (tanulás) tehát nem valamilyen természeti jelenség, hanem kemény numerikus feladat, amit számítógépen kell realizálni. Tehát a tanító (tanuló) algoritmusok az ismert numerikus matematikai módszerek (pl. gradiens módszer, Newton-módszer) szofisztikált megvalósításai, amik szoftveres implementációi is komoly tapasztalatot igényelnek. A neurális hálózatokkal megoldható feladatok jelentős része felmerült mint klasszikus matematikai feladat (pl. a numerikus matematika, az operációkutatás, ill. a statisztika témakörében), azonban a neurális hálózatok módszerével sok probléma jobban kezelhető, mint ha csak szigorúan a matematika eszközeire támaszkodnánk.

A jelen műben a neurális hálózatok tudományának egy közép szintjét célozzuk meg. Az egyes modelleket részletesen és pontosan megadjuk. Megvilágítjuk a matematikai hátteret, a főbb tulajdonságokat leírjuk. Elemezzük a betanításhoz szükséges numerikus eljárásokat. Konkrét kidolgozott példákat is nyújtunk. Végül feladatokat adunk (megoldással vagy útmutatóval).

Az igazán mély matematikai elemzés azonban kívül esik jegyzetünkön. Az általunk bemutatott modellek viselkedésének leírása a napjainkban gyorsan fejlődő statisztikus tanulási elmélet témája, ezzel kapcsolatban csak utalunk Vapnik [49], Devroye, Györfi, Lugossy [28] és Hastie, Tibshirani, Friedman [17] könyvekre. A főbb numerikus módszereket megadjuk, ezek hátterét is

leírjuk az Appendixben. A numerikus módszerek tulajdonságait nincs lehetőség elemeznünk, csak utalunk Fletcher [14] könyvére.

A jegyzetben talált leírás elegendő lenne neurális hálózatok egyszerű számítógépes programjainak megírására. Azonban igazán jó implementációhoz további ismeretek szükségesek (lásd [16]). Valójában léteznek és elérhetőek alkalmas programok és programcsomagok, ilyen pl. a Matlab és az R (lásd [29], ill. [45]; a programok leírásáról lásd pl. [15], ill. [9]). Ezen programok alkalmazása konkrét adatokra nagyon tanulságos. A jelen jegyzet numerikus példái is Matlab-bal készültek (kivéve az 5.8. Példát, amely R-ben).

A jegyzet fejezetei a következőket tárgyalják: perceptron, MLP, RBF, SVM, illetve az Appendixben a matematikai háttér. Helyhiány miatt számos hálózat típus kimaradt (ezekről lásd [18]), így nem szerepelnek az önszervező hálózatok (pl. a Kohonen-háló [24]), és kimaradt a nem-ellenőrzött tanulás teljes elmélete is.

2. fejezet

A perceptron

2.1. A neuron sémája

Ebben a fejezetben elsősorban Haykin [18] könyvére támaszkodva ismertetjük a perceptront.

Induljunk ki abból, hogy adataink (méréseink, megfigyeléseink) vannak egy jelenségről. Legyenek a bemeneti adatok $\mathbf{x}(n) = (x_1(n), \dots, x_m(n))^T$ m -dimenziós vektorok, ahol $n = 1, 2, \dots$ az időpillanatokat jelenti. A számunkra ismeretlen rendszer minden $\mathbf{x}(n)$ -hez megad egy $d(n)$ kimeneti értéket. Az egyszerű esetekben $d(n)$ skalár. (Itt és a továbbiakban a vektorokat oszlopvektornak fogjuk tekinteni, ezekből a transzponálás (jele T) segítségével kapunk sorvektort.) A célunk az, hogy az ismeretlen rendszert az input és output adatok alapján leírjuk egy olyan modellel, amely majd olyan input adatokra, amelyekre vonatkozó outputot nem ismerjük, megadja (legalábbis közelítőleg) az outputot. Például egy betanított karakter felismerő program egy kamera felvételeiből tudja azonosítani a megfigyelt jeleket.

A *neurális hálózat* nem más, mint egymással összekötött neuronok összessége. A *neuron* (idegsejt) egy információ feldolgozó egység, a neurális hálózat alapegysége. A legegyszerűbb neurális hálózat egyetlen neuronból áll.

A 2.1. ábrán látható egy neuron, a *perceptron* általános sémája. Ennek részei az alábbiak.

(1) Bemenet (input). Az $\mathbf{x}(n) = (x_1(n), \dots, x_m(n))^T$ ismert m -dimenziós vektor a bemenet. Azaz az n -edik időpontban érkezik m számból álló bemenő jel. Ezek a számok számunkra ismertek (mérhető, megfigyelhető értékek). Minden $n = 1, 2, \dots, N$ időpontban érkezik egy bemenő jel.

(2) Súlyok (synaptic weight). Az igazi w_1, \dots, w_m súlyok nem ismertek. Éppen ezek meghatározása a feladat. A $w_1(n), \dots, w_m(n)$ mennyiségek az n -

edik időpillanatban használt súlyok. Ezek a valódi súlyok n -edik közelítései. Ezeket a közelítéseket lépésenként finomítjuk. A $\mathbf{w} = (w_1, \dots, w_m)^\top$ vektor az igazi, $\mathbf{w}(n) = (w_1(n), \dots, w_m(n))^\top$ pedig a közelítő súlyvektor.

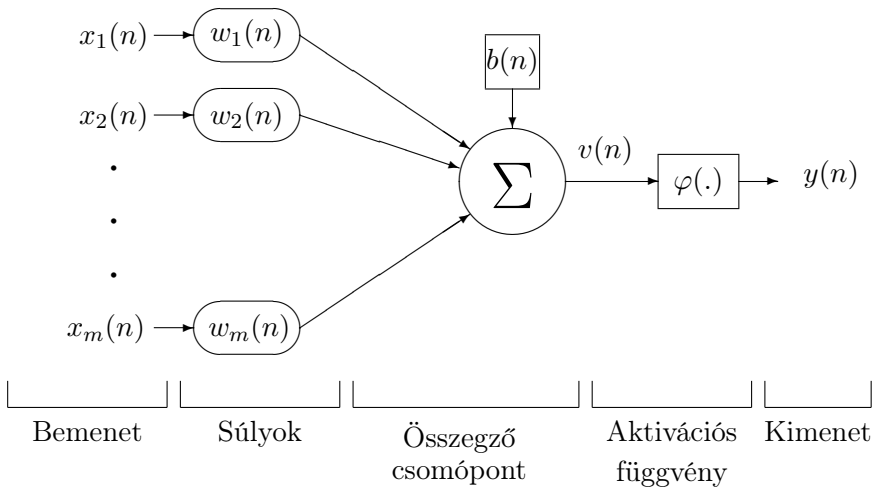
(3) Torzítás (bias). A b torzítás igazi értéke szintén nem ismert. Ennek az n -edik közelítését jelöli $b(n)$. A torzítás meghatározása is cél. (Általában b skalár mennyiség.)

(4) Összegző csomópont (summing junction). Ez a bemenet adatainak alábbi súlyozott összegét képezi:

$$v(n) = b(n) + \sum_{i=1}^m w_i(n)x_i(n). \quad (2.1)$$

(5) Aktivációs, más szóval transzformációs vagy egyszerűen transzfer függvény (activation function, transfer function). Ez a problémának megfelelő φ függvény. Az előző $v(n)$ összegzett értéket alakítja át, értéke pedig a feladat szempontjából alkalmas intervallumba esik. Az alkalmas transzfer függvényt nekünk magunknak kell megadnunk.

(6) Kimenet (output). Ez a neuron által a bemeneti $\mathbf{x}(n)$ értékhez rendelt érték. Ez nem más, mint $y(n) = \varphi(v(n))$.



2.1. ábra. A perceptron felépítése

A jelölés egyszerűsítése érdekében ki szoktuk egészíteni a bemenő jelek és a súlyok vektorát egy nulladik koordinátával, konkrétan $x_0(n) = 1$ és $w_0(n) = b(n)$ értékekkel. Tehát a $b(n)$ torzítást is súlyként kezeljük. Mind a

bemeneti, mind a súlyvektorból $(m+1)$ -dimenziós lesz. Vektoros alakban:

$$\mathbf{x}(n) = \begin{pmatrix} 1 \\ x_1(n) \\ \vdots \\ x_m(n) \end{pmatrix} = \begin{pmatrix} x_0(n) \\ x_1(n) \\ \vdots \\ x_m(n) \end{pmatrix}, \quad \mathbf{w}(n) = \begin{pmatrix} b(n) \\ w_1(n) \\ \vdots \\ w_m(n) \end{pmatrix} = \begin{pmatrix} w_0(n) \\ w_1(n) \\ \vdots \\ w_m(n) \end{pmatrix}.$$

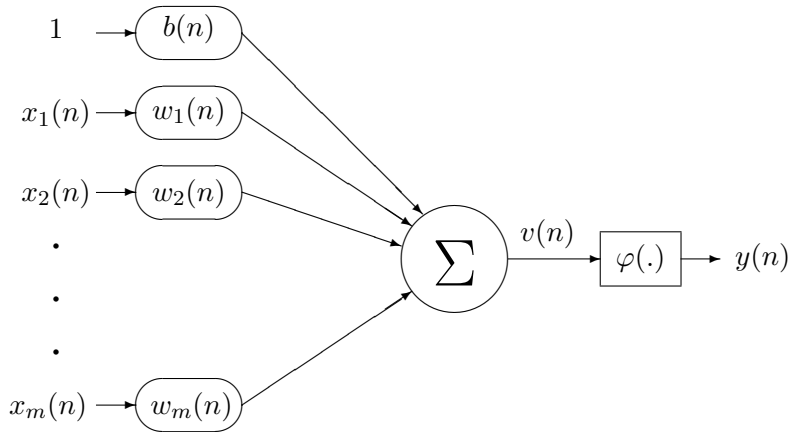
(Ez nyilván visszaélés az $\mathbf{x}(n)$ és $\mathbf{w}(n)$ jelöléssel.) Ezzel a jelöléssel:

$$v(n) = \sum_{i=0}^m w_i(n)x_i(n) = \mathbf{w}(n)^\top \mathbf{x}(n), \quad (2.2)$$

továbbá

$$y(n) = \varphi(v(n)) = \varphi\left(\sum_{i=0}^m w_i(n)x_i(n)\right) = \varphi\left(\mathbf{w}(n)^\top \mathbf{x}(n)\right).$$

Ezt a kissé módosított sémát a 2.2. ábra szemlélteti.



2.2. ábra. A módosított perceptron

A neurális hálózatok használatának fő szakaszai: a konkrét hálózat séma megadása, a hálózat betanítása, valamint a hálózat használata. Tegyük most fel, hogy egy konkrét jelenség modellezését perceptronnal kíséreljük meg, amihez kiválasztunk egy aktivációs függvényt. Ezzel a sémát meg is adtuk. Szükségesek olyan adatok, amelyek esetén ismerjük az $\mathbf{x}(n)$ input vektorokat és a hozzájuk tartozó $d(n)$ kimeneti értékeket. Ezek a *tanító pontok* (training point). A perceptron által az $\mathbf{x}(n)$ input vektorhoz kiszámolt $y(n)$ output

értékeket hasonlítsuk össze a tényleges $d(n)$ -ekkel. Ez a hálózat (négyzetes) hibája:

$$\mathcal{E}(n) = (d(n) - y(n))^2.$$

A cél ezt a hibát minimalizálni. Ezt a súlyok változtatásával tudjuk elérni. Ezt a folyamatot nevezzük a *hálózat tanításának* (tanulás, training). Ha sikerül a hálózatot betanítani, azaz a hálózatunk a tanító pontok esetén (megközelítőleg) „helyes” értékeket szolgáltat, akkor kezdődhet a használata. Ez azt jelenti, hogy olyan input adatokra alkalmazzuk, amelyekre az output ismeretlen. Éppen a hálózat által szolgáltatott output értékeket fogadjuk el igaz értékeknek. Természetesen nincs arra garancia, hogy ezek mindig a valóságos értékekkel egyeznek meg. Megjegyezzük, hogy a betanított hálózat működésének ellenőrzése gyakran tesztadatokon történik.

Látható, hogy az igazi feladat a modell megadása és betanítása, a betanított modell alkalmazása általában rutinszerű.

2.2. Aktivációs függvények

Leggyakrabban olyan aktivációs függvényeket használunk, amelyek monoton növekvők, jobbról folytonosak, határértékük a $-\infty$ -ben 0, a $+\infty$ -ben pedig 1. Az ilyen φ tekinthető mint egy ξ valószínűségi változó $\varphi(x) = P(\xi \leq x)$ eloszlásfüggvénye. (Balról folytonosság esetén pedig $\varphi(x) = P(\xi < x)$ lenne.) Leggyakrabban az aktivációs függvények *szigmoid*, azaz S alakú függvények (pl. logisztikus, tangens hiperbolikus, ...).

1. Logisztikus függvény:

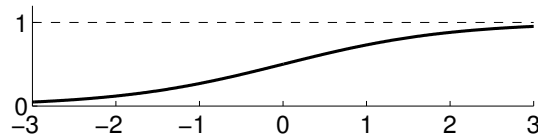
$$\varphi(x) = \frac{1}{1 + \exp(-ax)}, \quad x \in \mathbb{R},$$

ahol $a > 0$ konstans. Bizonyos művek csak a logisztikus függvényt értik szigmoid függvény alatt.

Erre a függvényre nyilván $\varphi(0) = 1/2$, $\varphi(\cdot)$ növekvő, $\varphi(-\infty) = 0$, $\varphi(\infty) = 1$. A deriváltja pedig

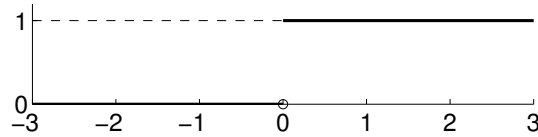
$$\varphi'(x) = \frac{a \exp(-ax)}{(1 + \exp(-ax))^2}.$$

Innen $\varphi'(0) = a/4$, azaz nagyobb a érték esetén φ meredekebb a 0 közelében. A 2.3. ábrán $a = 1$ esetén ábrázoltuk a logisztikus függvényt.

2.3. ábra. A logisztikus függvény $a = 1$ esetén

2. Küszöb függvény (threshold function, Heaviside function, hard limit):

$$\varphi(x) = \begin{cases} 1, & \text{ha } x \geq 0, \\ 0, & \text{ha } x < 0. \end{cases}$$

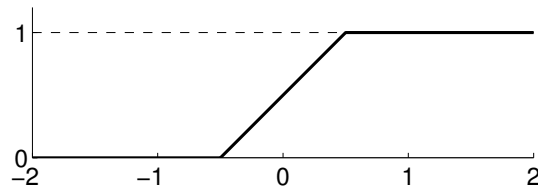


2.4. ábra. A küszöb függvény

3. Szakaszonként lineáris függvény (telítéses lineáris):

$$\varphi(x) = \begin{cases} 0, & \text{ha } x < -0.5, \\ x + 0.5, & \text{ha } -0.5 \leq x < 0.5, \\ 1, & \text{ha } x \geq 0.5. \end{cases}$$

Tehát φ a $[-0.5, 0.5]$ intervallumon egyenletes eloszlásfüggvény.

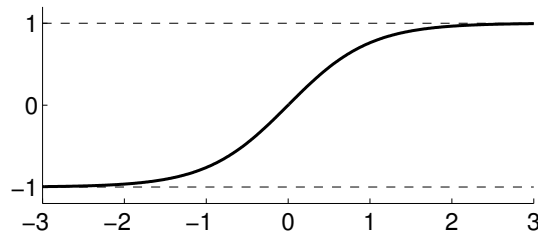


2.5. ábra. Szakaszonként lineáris függvény

A fenti függvényeket néha úgy módosítják, hogy a $-\infty$ -ben a határértékük -1 legyen. Ilyen tulajdonságú a tangens hiperbolikus és az előjel függvény is.

4. Tangens hiperbolikus függvény:

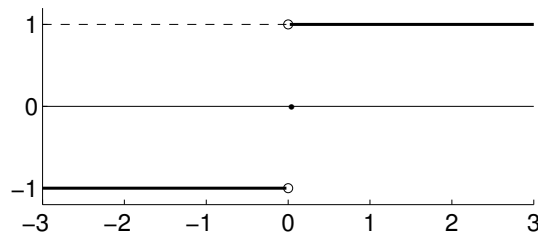
$$\varphi(x) = \frac{2}{1 + \exp(-2x)} - 1 = \tanh(x).$$



2.6. ábra. A tangens hiperbolikus függvény

5. Szignum függvény (speciális küszöb függvény):

$$\varphi(x) = \begin{cases} -1, & \text{ha } x < 0, \\ 0, & \text{ha } x = 0, \\ 1, & \text{ha } x > 0. \end{cases}$$



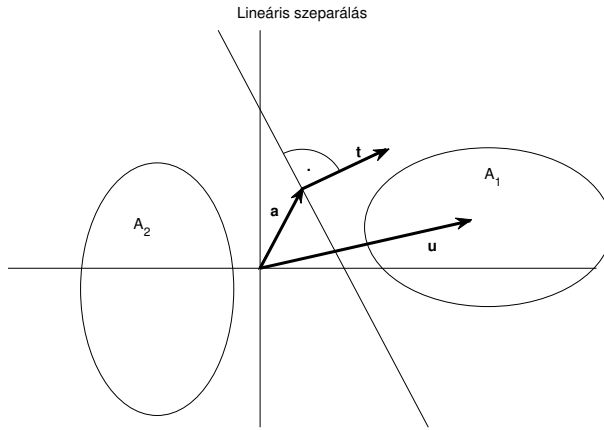
2.7. ábra. A szignum függvény

A logisztikus, a küszöb és a szakaszonként lineáris aktivációs függvények tekinthetők eloszlásfüggvényeknek is. A fenti függvényeket szokták telítődő (saturated) függvényeknek is nevezni.

Megjegyezzük továbbá, hogy bizonyos esetekben a fentiekől eltérő tulajdonságú transzfer függvényeket is használnak, pl. a $\varphi(x) = x$, $x \in \mathbb{R}$, lineáris függvényt.

2.3. A perceptron tanítása

Perceptronnal megoldható a lineáris szeparálás feladata. Tegyük fel, hogy a megfigyeléseink két csoportból származnak. A két csoport A_1 és A_2 . Tegyük fel továbbá, hogy a két csoport lineárisan szeparálható, azaz elválasztható egy hipersíkkal. Ezen hipersík megtalálása a cél. A perceptron tehát a statisztikában jól ismert lineáris szeparálást hajtja végre. Kétdimenzióban a helyzetet a 2.8. ábra szemlélteti.



2.8. ábra. Lineáris szeparálás

Természetesen a minta szétválasztása nem csak a síkban működik. Általában a perceptron egy k -dimenziós térben, $(k - 1)$ -dimenziós hipersíkkal való szeparálást hajt végre.

Tekintsük a tanító pontokat, azaz azokat a pontokat, amelyekről tudjuk, hogy melyik osztályba tartoznak. A hovatartozás alapján rendeljük ezen pontokhoz 1-et, vagy -1 -et. Ezek lesznek azok az igazi output értékek, amikhez majd hasonlítani tudjuk a perceptronnak a tanító input pontokra adott választát.

Először azt mutatjuk meg, hogy elegendő az origón átmenő elválasztó hipersíkokkal foglalkozni. Jelölje átmenetileg u az input vektort. Ekkor $u \in A_1 \cup A_2$. Legyen a az elválasztó hipersík rögzített pontjának helyvektora, t pedig az elválasztó hipersík normálvektora (mely A_1 felé néz). $u \in A_1$ akkor és csakis akkor teljesül, ha $(u - a)$ hegyesszöget zár be t -vel, így $(u - a)^T t > 0$, azaz

$$u^T t - a^T t > 0.$$

Most egészítsük ki az input vektort egy 1-essel, a sík normálvektorát pedig $-\mathbf{a}^\top \mathbf{t}$ -vel:

$$\mathbf{x} = \begin{pmatrix} 1 \\ u_1 \\ \vdots \\ u_m \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} -\mathbf{a}^\top \mathbf{t} \\ t_1 \\ \vdots \\ t_m \end{pmatrix},$$

ahol u_1, \dots, u_m az \mathbf{u} , míg t_1, \dots, t_m a \mathbf{t} vektor koordinátái. Az ilyen módon kapott \mathbf{x} lesz a perceptron bemeneti értéke, a \mathbf{w} pedig a súlyvektora. Látható, hogy \mathbf{w} első komponense a torzítással van összefüggésben. Tehát ezen transzformáció után az 1-gyel magasabb dimenziós térben kell az origón átmenő hipersíkot keresni, hiszen

$$\mathbf{u} \in A_1 \iff \mathbf{x}^\top \mathbf{w} > 0, \quad \mathbf{u} \in A_2 \iff \mathbf{x}^\top \mathbf{w} < 0. \quad (2.3)$$

Nyilván ebben a térben \mathbf{w} lesz az elválasztó hipersík normálvektora (amely A_1 felé néz).

Még kérdés az is, hogy ha a magasabb dimenziós mintatérben megtaláltuk az elválasztó hipersíkot, akkor az eredeti problémában is megkapjuk-e azt. A válasz igen. Hiszen \mathbf{w} utolsó m koordinátája éppen a \mathbf{t} vektort adja. Az első koordinátája viszont nem határozza meg \mathbf{a} -t. De magát a hipersíkot igen, mert ha $\mathbf{a}_1^\top \mathbf{t} = \mathbf{a}_2^\top \mathbf{t}$, akkor $(\mathbf{a}_1 - \mathbf{a}_2)^\top \mathbf{t} = 0$, azaz $(\mathbf{a}_1 - \mathbf{a}_2)$ merőleges \mathbf{t} -re, így mind \mathbf{a}_1 , mind \mathbf{a}_2 ugyanannak a \mathbf{t} -re merőleges hipersíknak helyvektora.

Térjünk rá a perceptron tanítására. Az előzőek alapján kiegészített vektorokkal dolgozunk, azaz az origón átmenő hipersíkot keresünk. Jelölje \mathbf{w} az igazi (és az origón átmenő) elválasztó hipersík normálvektorát. \mathbf{w} megtalálása a cél az $\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(N)$ tanító pontok alapján. A tanító pontokról tudjuk, hogy mely osztályba tartoznak. Egyesével dolgozzuk fel a tanító pontokat. Induljunk ki a $\mathbf{w}(1)$ kezdeti értékből. ($\mathbf{w}(1)$ tipikus választása a nullvektor.) Az argumentum itt azt mutatja, hogy hányadik pont feldolgozásánál tartunk. Tegyük fel, hogy már feldolgoztunk $n - 1$ tanító pontot. Adjuk a perceptron bemenetére az $\mathbf{x}(n)$ tanító pontot. Ekkor három eset állhat fenn:

- Ha a pont az aktuális (azaz $\mathbf{w}(n)$ normálvektorú) hipersík jó oldalán van, akkor nem változtatjuk meg az aktuális szeparáló hipersíkot. Tehát, ha $\mathbf{w}(n)^\top \mathbf{x}(n) > 0$ és $\mathbf{x}(n) \in A_1$, vagy pedig ha $\mathbf{w}(n)^\top \mathbf{x}(n) \leq 0$ és $\mathbf{x}(n) \in A_2$, akkor

$$\mathbf{w}(n+1) = \mathbf{w}(n).$$

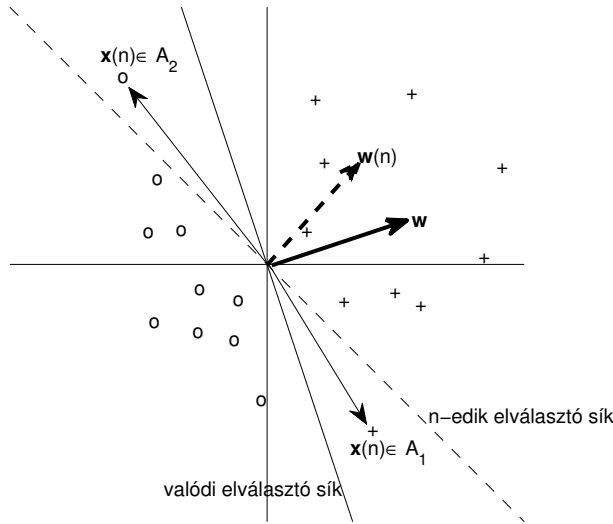
- Ha $\mathbf{w}(n)^\top \mathbf{x}(n) \leq 0$, de $\mathbf{x}(n) \in A_1$, akkor $\mathbf{w}(n)$ -et $\mathbf{x}(n)$ irányába „for-
gatjuk”:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{x}(n),$$

ahol $\eta(n) > 0$.

- Ha $\mathbf{w}(n)^\top \mathbf{x}(n) > 0$, de $\mathbf{x}(n) \in A_2$, akkor $\mathbf{w}(n)$ -et $\mathbf{x}(n)$ -től ellenkező
irányba „for-
gatjuk”:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)\mathbf{x}(n).$$



2.9. ábra. Kétféle korrekciós pont a perceptron tanításánál

Azaz, ha a vizsgált tanító pont az aktuális hipersík rossz oldalán van, akkor a hipersíkot alkalmas irányba forgatjuk. Az utóbbi két esetben $\mathbf{x}(n)$ -et *korrekciós pontnak* nevezzük. A 2.9. ábra mutatja a kétféle korrekciós pontot. $\eta(n)$ jelöli az n -edik lépés *korrekciós tényezőjét*, más szóval tanulási paraméterét (learning rate). $\eta(n) > 0$ skalár.

A tanulás úgy folyik, hogy a tanító pontokat többször átáramoltatjuk. Az összes tanító pont egyszeri átáramoltatását egy *epochnak* nevezzük. Ha valamelyik epochon belül már nem találunk korrekciós pontot, akkor az aktuális hipersík helyesen szeparál, tehát a tanítást befejezzük.

Most belátjuk, hogy realizisztikus feltételek fennállása esetén korlátos sok lépés után nem marad korrekciós pont, azaz leáll az eljárás.

Legyen minden $n \in \mathbb{N}$ esetén a korrekciós tényező ugyanaz a konstans

$$\eta(n) = \eta > 0.$$

Tegyük fel továbbá, hogy a tanító pontok halmaza korlátos, azaz

$$\|\mathbf{x}(n)\|^2 \leq R < \infty, \quad \text{minden } n\text{-re,}$$

és hogy a két halmaz pozitív margóval elválasztható:

$$|\mathbf{w}^\top \mathbf{x}(n)| \geq \delta > 0.$$

(Ha \mathbf{w} egységvektor, akkor fenti feltétel legalább δ nagyságú margót követel meg, azaz δ -nál közelebb nem lehet tanító pont az elválasztó hipersíkhoz.) Ebben a bizonyításban legyen $\text{sgn}(x) = 1$, ha $x > 0$, és $\text{sgn}(x) = -1$, ha $x \leq 0$. Ekkor korrekciós pontok esetén

$$\begin{aligned} \|\mathbf{w}(n+1)\|^2 &= \|\mathbf{w}(n) - \text{sgn}\{\mathbf{w}(n)^\top \mathbf{x}(n)\} \eta \mathbf{x}(n)\|^2 = \\ &= \|\mathbf{w}(n)\|^2 + \eta^2 \|\mathbf{x}(n)\|^2 - \underbrace{2 \text{sgn}\{\mathbf{w}(n)^\top \mathbf{x}(n)\} \eta \mathbf{w}(n)^\top \mathbf{x}(n)}_{\geq 0} \leq \\ &\leq \|\mathbf{w}(n)\|^2 + \eta^2 \|\mathbf{x}(n)\|^2 \leq \dots \leq \\ &\leq \|\mathbf{w}(1)\|^2 + \eta^2 (\|\mathbf{x}(1)\|^2 + \dots + \|\mathbf{x}(n)\|^2) \leq \|\mathbf{w}(1)\|^2 + \eta^2 nR. \end{aligned} \quad (2.4)$$

Másrészt

$$\begin{aligned} \mathbf{w}^\top \mathbf{w}(n+1) &= \mathbf{w}^\top \left[\mathbf{w}(n) - \text{sgn}\{\mathbf{w}(n)^\top \mathbf{x}(n)\} \eta \mathbf{x}(n) \right] = \\ &= \mathbf{w}^\top \mathbf{w}(n) - \underbrace{\eta \text{sgn}\{\mathbf{w}(n)^\top \mathbf{x}(n)\} \mathbf{w}^\top \mathbf{x}(n)}_{=\eta |\mathbf{w}^\top \mathbf{x}(n)| \geq \eta \delta} \geq \dots \geq \mathbf{w}^\top \mathbf{w}(1) + n\delta\eta. \end{aligned}$$

Elég nagy n esetén $\mathbf{w}^\top \mathbf{w}(1) + n\delta\eta$ pozitív lesz. Felhasználva az $|a^\top b| \leq \|a\| \|b\|$ Cauchy-egyenlőtlenséget, a fenti egyenlőtlenségből azt kapjuk, hogy

$$\|\mathbf{w}(n+1)\|^2 \geq \frac{|\mathbf{w}(n+1)^\top \mathbf{w}|^2}{\|\mathbf{w}\|^2} \geq \frac{(\mathbf{w}^\top \mathbf{w}(1) + n\eta\delta)^2}{\|\mathbf{w}\|^2}.$$

Ezt a (2.4) egyenlőtlenséggel összevetve:

$$\|\mathbf{w}(1)\|^2 + n\eta^2 R \geq \|\mathbf{w}(n+1)\|^2 \geq \frac{(\mathbf{w}^\top \mathbf{w}(1) + n\eta\delta)^2}{\|\mathbf{w}\|^2}. \quad (2.5)$$

Ezen egyenlőtlenség bal oldala n -ben elsőfokú, jobb oldala pedig másodfokú. Ez nem állhat fenn végtelen sok n értékre. Tehát csak véges számú korrekciós pont lehet. A többiek már nem változtatnak a szeparáló hipersíkon. Ez azt jelenti, hogy a tanító pontokat többször átáramoltatva a hálózaton, egy idő után nincs több korrekció. Ezzel beláttuk a következő (Rosenblatt és Novikoff nevéhez köthető) tételt.

2.1. Tétel (Perceptron konvergencia tétel, statisztikai tétel a lineáris szeparálásról). *Tegyük fel, hogy A_1 és A_2 úgy szeparálható lineárisan, hogy $\|\mathbf{w}^\top \mathbf{x}(n)\| \geq \delta > 0$ és $\|\mathbf{x}(n)\|^2 \leq R < \infty$ teljesül minden tanító pontra. Legyen a tanulási paraméter $\eta > 0$ állandó. Ekkor a perceptron algoritmus a véges sok lépésben véget ér a tanító pontokat helyesen szeparáló hipersíknál.*

Legyen a kiinduló állapotban $\mathbf{w}(1) = \mathbf{0}$. Ekkor a (2.5) képlet alapján

$$n \leq \frac{\|\mathbf{w}\|^2 R}{\delta^2}.$$

Tehát ennél több korrekciós pontot nem találunk (még akkor sem, ha a már vizsgált tanító pontokat újra inputként használjuk akár többször is). Tehát legfeljebb ennyi lépés után minden tanító pont helyesen van elválasztva.

2.4. A perceptron algoritmusának egy változata

Ebben a részben [10] alapján ismertetjük a perceptron algoritmus egy változatát. Ebben a változatban nem szükséges a feladatot az origón átmenő hipersíkra átfogalmazni. Csupán az itt is érvényes perceptron konvergencia tétel bizonyításában merül fel az eggyel magasabb dimenziós feladattá alakítás. Tehát az elválasztó hipersík egyenlete

$$\mathbf{w}^\top \mathbf{x} + b.$$

$$\mathbf{x} \in A_1 \iff \mathbf{w}^\top \mathbf{x} + b > 0, \quad \mathbf{x} \in A_2 \iff \mathbf{w}^\top \mathbf{x} + b < 0.$$

Legyen ebben a szakaszban

$$S = \{(\mathbf{x}_i, y_i) : i = 1, 2, \dots, N\}$$

a tanító pontok halmaza, ahol $y_i = 1$, ha $\mathbf{x}_i \in A_1$ és $y_i = -1$, ha $\mathbf{x}_i \in A_2$.

Tegyük fel, hogy A_1 és A_2 lineárisan szeparálható. Legyen adott az $\eta > 0$ tanulási paraméter. Ekkor a perceptron tanítási algoritmus a következő.

```

 $\mathbf{w}_0 := \mathbf{0}; b_0 := 0; k := 0$ 
 $R := \max_{1 \leq i \leq N} \|\mathbf{x}_i\|$ 
repeat
  for  $i = 1$  to  $N$ 
    if  $y_i(\mathbf{w}_k^\top \mathbf{x}_i + b_k) \leq 0$  then
       $\mathbf{w}_{k+1} := \mathbf{w}_k + \eta y_i \mathbf{x}_i$ 
       $b_{k+1} := b_k + \eta y_i R^2$ 
       $k := k + 1$ 
    end if
  end for
until egy teljes for ciklus alatt nincs korrekció
return  $k, \mathbf{w}_k, b_k$ 

```

Itt k a szükséges korrekciók száma.

Láthatjuk, hogy helyes osztályozás esetén $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0$.

2.2. Tétel (Rosenblatt–Novikoff). *Legyen S tanító pontok egy nem-triviális (azaz $A_1 \neq \emptyset, A_2 \neq \emptyset$), hipersíkkal elválasztható halmaza. Legyen*

$$R = \max_{1 \leq i \leq N} \|\mathbf{x}_i\|.$$

Tegyük fel, hogy létezik egy \mathbf{w}_{opt} vektor, melyre $\|\mathbf{w}_{\text{opt}}\| = 1$ és egy b_{opt} skalár, hogy

$$y_i(\mathbf{w}_{\text{opt}}^\top \mathbf{x}_i + b_{\text{opt}}) \geq \gamma > 0 \quad (2.6)$$

minden $1 \leq i \leq N$ esetén. Ekkor a fenti perceptron algoritmus legfeljebb

$$\left(\frac{2R}{\gamma}\right)^2$$

korrekciós lépés után leáll a két halmazt helyesen szeparáló hipersíknál.

Bizonyítás. Egészítsük ki az input vektorokat egy extra koordinátával, melynek értéke R legyen. Azaz legyen az új vektor $\hat{\mathbf{x}}_i = (\mathbf{x}_i^\top, R)^\top$. A súlyvektort is egészítsük ki, legyen az új vektor $\hat{\mathbf{w}}_t = (\mathbf{w}_t^\top, b_t/R)^\top$. A $\hat{\mathbf{w}}_{\text{opt}}$ vektor definíciója hasonló. Az algoritmus $\hat{\mathbf{w}}_0 = \mathbf{0}$ -ból indul. Legyen $\hat{\mathbf{w}}_{t-1}$ a $t - 1$ -edik korrekció után a súlyvektor. t -edik korrekció akkor szükséges, ha

$$y_i \hat{\mathbf{w}}_{t-1}^\top \hat{\mathbf{x}}_i = y_i(\mathbf{w}_{t-1}^\top \mathbf{x}_i + b_{t-1}) \leq 0, \quad (2.7)$$

azaz $(\mathbf{x}_i, y_i) \in S$ olyan pont, melyet $\hat{\mathbf{w}}_{t-1} = (\mathbf{w}_{t-1}^\top, b_{t-1}/R)^\top$ helytelenül osztályoz. A súlyvektor aktualizálása az alábbi

$$\hat{\mathbf{w}}_t = (\mathbf{w}_t^\top, b_t/R)^\top = (\mathbf{w}_{t-1}^\top, b_{t-1}/R)^\top + \eta y_i (\mathbf{x}_i^\top, R)^\top = \hat{\mathbf{w}}_{t-1} + \eta y_i \hat{\mathbf{x}}_i. \quad (2.8)$$

Itt kihasználtuk, hogy

$$b_t/R = b_{t-1}/R + \eta y_i R,$$

mivel

$$b_t = b_{t-1} + \eta y_i R^2.$$

A (2.8) és (2.6) miatt teljesül

$$\hat{\mathbf{w}}_t^\top \hat{\mathbf{w}}_{\text{opt}} = \hat{\mathbf{w}}_{t-1}^\top \hat{\mathbf{w}}_{\text{opt}} + \eta y_i \hat{\mathbf{x}}_i^\top \hat{\mathbf{w}}_{\text{opt}} \geq \hat{\mathbf{w}}_{t-1}^\top \hat{\mathbf{w}}_{\text{opt}} + \eta \gamma$$

számolásból indukcióval adódik, hogy

$$\hat{\mathbf{w}}_t^\top \hat{\mathbf{w}}_{\text{opt}} \geq t\eta\gamma. \quad (2.9)$$

Hasonlóan kapjuk (2.7) és (2.8) alapján (és $y_i^2 = 1$ -et figyelembe véve), hogy

$$\begin{aligned} \|\hat{\mathbf{w}}_t\|^2 &= \|\hat{\mathbf{w}}_{t-1}\|^2 + 2\eta y_i \hat{\mathbf{w}}_{t-1}^\top \hat{\mathbf{x}}_i + \eta^2 \|\hat{\mathbf{x}}_i\|^2 \\ &\leq \|\hat{\mathbf{w}}_{t-1}\|^2 + \eta^2 \|\hat{\mathbf{x}}_i\|^2 \\ &\leq \|\hat{\mathbf{w}}_{t-1}\|^2 + \eta^2 (\|\mathbf{x}_i\|^2 + R^2) \\ &\leq \|\hat{\mathbf{w}}_{t-1}\|^2 + 2\eta^2 R^2. \end{aligned}$$

Ebből indukcióval következik, hogy

$$\|\hat{\mathbf{w}}_t\|^2 \leq 2t\eta^2 R^2. \quad (2.10)$$

A fenti (2.9) és (2.10) egyenlőtlenségből és a Cauchy-egyenlőtlenségből adódik, hogy

$$\|\hat{\mathbf{w}}_{\text{opt}}\| \sqrt{2t\eta} R \geq \|\hat{\mathbf{w}}_{\text{opt}}\| \|\hat{\mathbf{w}}_t\| \geq \hat{\mathbf{w}}_t^\top \hat{\mathbf{w}}_{\text{opt}} \geq t\eta\gamma,$$

amiből

$$t \leq 2 \left(\frac{R}{\gamma} \right)^2 \|\hat{\mathbf{w}}_{\text{opt}}\|^2 \leq \left(\frac{2R}{\gamma} \right)^2.$$

Itt kihasználtuk, hogy $|b_{\text{opt}}| \leq R$ a tanító pontok nem-triviális szeparálása esetén, amiből

$$\|\hat{\mathbf{w}}_{\text{opt}}\|^2 \leq \|\mathbf{w}_{\text{opt}}\|^2 + 1 = 2. \quad \square$$

2.5. A perceptron algoritmusának duális alakja

Ebben a részben [10] alapján ismertetjük a perceptron algoritmus duális alakját. Látható, hogy a perceptron esetén a kapott elválasztó hipersík normálvektora

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$$

alakú lesz (legalábbis nulla kezdőérték esetén). Tehát az elválasztó hipersík egyenlete

$$\mathbf{w}^\top \mathbf{x} + b = \left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \right)^\top \mathbf{x} + b = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b$$

alakú lesz. Ez az alak két szempontból is előremutató. Egyrészt láthatjuk, hogy itt (és az alábbi algoritmusban is) nem maguk a vektorok, hanem csak azok belső szorzatai szerepelnek. A belső szorzatokat pedig majd az SVM esetén magfüggvényekkel fogjuk helyettesíteni abból a célból, hogy a nem lineáris szeparálás feladatát is meg tudjuk oldani. Másrészt az SVM esetén fogjuk látni, hogy a duális alak vezet végrehajtható számolási eljáráshoz.

Tekintsük most a perceptron duális alakjának algoritmusát. Jelölje $\boldsymbol{\alpha}$ az α_i -kből álló vektort. Legyen az η tanulási paraméter 1.

```

 $\boldsymbol{\alpha} := \mathbf{0}; b := 0$ 
 $R := \max_{1 \leq i \leq N} \|\mathbf{x}_i\|$ 
repeat
  for  $i = 1$  to  $N$ 
    if  $y_i (\sum_{j=1}^N \alpha_j y_j \mathbf{x}_j^\top \mathbf{x}_i + b) \leq 0$  then
       $\alpha_i := \alpha_i + 1$ 
       $b := b + y_i R^2$ 
    end if
  end for
until egy teljes for ciklus alatt nincs korrekció
return  $\boldsymbol{\alpha}, b$ 

```

2.6. Lineáris regresszió

Illesszünk $\mathbf{x}^\top \mathbf{w} + b$ alakú (lineáris) függvényt mért adatokra. A jelölések egyszerűsítése érdekében változtassuk meg a jelöléseinket a szokásos módon. Tehát legyen az új \mathbf{x} vektor az eredetitől eggyel magasabb dimenziós úgy,

hogy a meglévő koordináták elé mindig egy 1-est írunk. Hasonlóan egészítjük ki \mathbf{w} -t úgy, hogy a meglévő koordináták elé b -t írunk. Ezzel a keresett függvény alakja:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}.$$

Tegyük fel, hogy $(\mathbf{x}(1), d(1)), \dots, (\mathbf{x}(N), d(N))$ a megfigyeléseink, ahol $d(i) = f(\mathbf{x}(i))$. Rendezzük az $\mathbf{x}(i)^\top$ sorvektorokat egymás alá, így kapjuk az X mátrixot. A $d(i)$ számokból pedig alkossuk meg a \mathbf{d} vektort. A szokásos legkisebb négyzetes módszer szerint az ismeretlen \mathbf{w} együttható vektor becslését a

$$\sum_{i=1}^N (\mathbf{w}^\top \mathbf{x}(i) - d(i))^2 = \|X\mathbf{w} - \mathbf{d}\|^2$$

kifejezés \mathbf{w} szerinti minimumhelyeként kapjuk meg. Ez tehát a lineáris modell (lásd Appendix), ebben a $\hat{\mathbf{w}}$ becslést az

$$X^\top X \mathbf{w} = X^\top \mathbf{d} \quad (2.11)$$

normálegyenlet megoldásaként nyerjük. Invertálható esetben ebből $\hat{\mathbf{w}}$ a

$$\hat{\mathbf{w}} = (X^\top X)^{-1} X^\top \mathbf{d}$$

képlet alapján adódik. Az invertálási problémák elkerülésére a

$$\hat{\mathbf{w}} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{d}$$

képletet alkalmazhatjuk, ahol I egységmátrix, λ pedig pozitív szám. Ezt nevezik *ridge regresszió*nak

A fenti feladat megoldására Widrow és Hoff 1960-ban (lásd [50]) rekurzív eljárást javasolt. Ez a *Least-Mean-Square* (LMS) módszer, más szóval a *Widrow-Hoff-algoritmus*. Ez valójában a nevezetes Adaline, azaz az adaptive linear element, amelyet most ismertetünk. Az előző (2.11) módszer (azaz a lineáris legkisebb négyzetes módszere) esetén az első n megfigyelés (adat, tanító pont) által adott információt használtuk. Most viszont nem az n -edik megfigyelésig terjedő összes (azaz n db) eltérés alapján, hanem csupán az n -edik (azaz 1 db) eltérés alapján aktualizáljuk az előző becslést.

Tehát az n -edik eltérés:

$$e(n) = d(n) - \mathbf{x}(n)^\top \mathbf{w}(n),$$

a legkisebb négyzetes hiba pedig:

$$\mathcal{E}(\mathbf{w}(n)) = \frac{1}{2} e(n)^2.$$

Innen

$$\frac{\partial e(n)}{\partial \mathbf{w}(n)} = -\mathbf{x}(n),$$

tehát

$$\frac{\partial \mathcal{E}(\mathbf{w}(n))}{\partial \mathbf{w}(n)} = -e(n)\mathbf{x}(n).$$

Tehát a gradiens vektor becslése:

$$\mathbf{g}(n) = -e(n)\mathbf{x}(n).$$

Mivel a gradiens módszer szerint a negatív gradiens irányába kell lépniünk, kapjuk a következőt:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta e(n)\mathbf{x}(n). \quad (2.12)$$

Ez tehát az LMS (Least-Mean-Square) algoritmus szerinti aktualizálása a \mathbf{w} együttható vektor $\hat{\mathbf{w}}(n)$ közelítésének. Itt $\eta > 0$ tanulási paraméter.

2.7. Feladatok

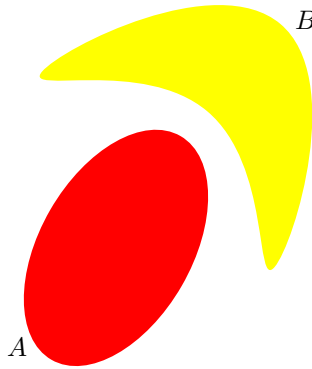
1. Vizsgáljuk a megadott aktivációs függvények menetét! Adjunk meg további aktivációs függvényeket! Adjuk meg a szereplő φ aktivációs függvény $\lim_{x \rightarrow -\infty} \varphi(x) = 0$, valamint $\lim_{x \rightarrow -\infty} \varphi(x) = -1$ feltételt teljesítő változatát is!
2. Írjunk programot a perceptron tanítására!
3. Írjunk pszeudokódot a Least-Mean-Square algoritmusra!
4. Legyen x értéke 1, ill. 0 aszerint, hogy x igaz, vagy hamis. Hasonlóan y -ra. Adjuk meg ezekkel a változókkal a logikai ÉS, VAGY, továbbá a KIZÁRÓ VAGY (azaz XOR) művelet tábláját. Ábrázoljuk a lehetséges (x, y) pontokat a koordináta rendszerben. Lássuk be, hogy az ÉS, valamint a VAGY esetén lineárisan szeparálható két halmazt kapunk, de XOR esetén nem.
5. A 2.2. Tétel bizonyításának befejező lépéséhez lássuk be, hogy $|b_{\text{opt}}| \leq R$.

3. fejezet

A többrétegű perceptron

3.1. A többrétegű perceptron felépítése

Ebben a fejezetben elsősorban Haykin [18] könyvére támaszkodva ismertetjük a többrétegű perceptront. A lineáris szeparálás feladata elvégezhető egyetlen perceptronnal. Viszont a gyakorlatban legtöbbször nem lineárisan szeparálható halmazokkal találkozunk. A 3.1. ábrán látható két olyan halmaz, amelyek nem szeparálhatóak lineárisan.

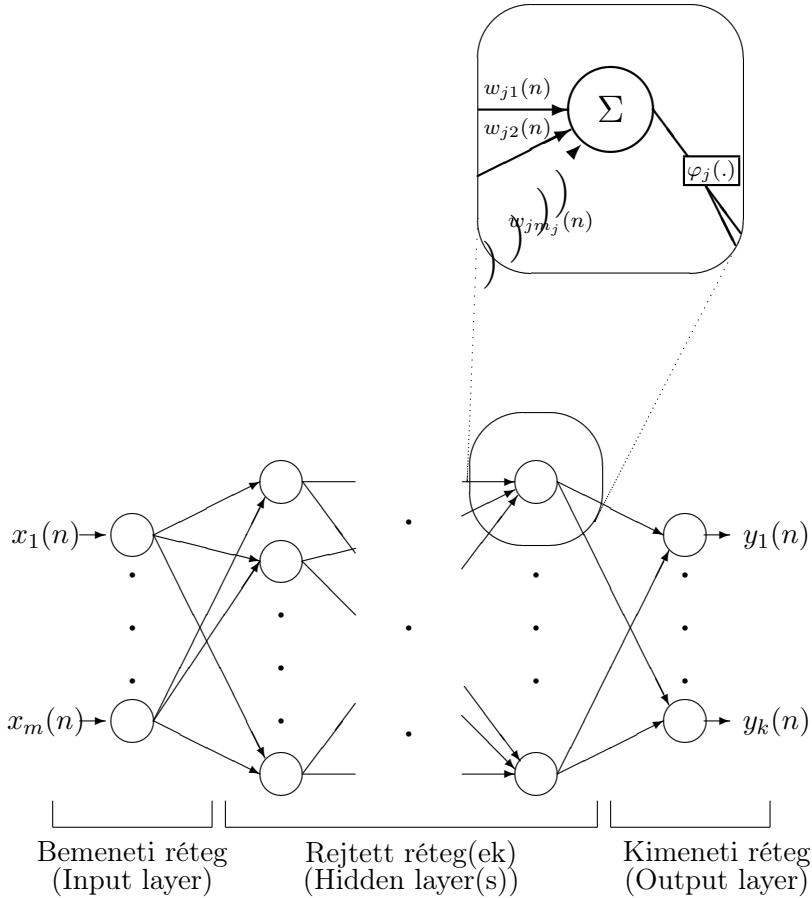


3.1. ábra. Lineárisan nem szeparálható halmazok

A lineárisan nem szeparálható halmazok szétválasztása megoldható *többrétegű perceptronnal* (Multi Layer Perceptron, MLP).

A többrétegű perceptron a legismertebb, mondhatni a legtipikusabb neurális hálózat. Ebben nem egyszerűen neuronokat (perceptronokat) kapcsolunk össze egy hálózatban, hanem a neuronokat rétegekbe is szervezzük.

A hálózat háromféle rétegből (layer) épül fel: bemeneti, rejtett, valamint kimeneti rétegből. A rétegek angol nevei: input layer, hidden layer, output layer. Rejtett rétegből tetszőleges számú lehet, viszont bemenetiből és kimenetiből csak egy-egy. Ezt a 3.2. ábra szemlélteti. Bal oldalon van a bemeneti



3.2. ábra. A többrétegű perceptron felépítése

réteg, jobb oldalon a kimeneti, közöttük pedig egy vagy több rejtett réteg. A jel balról jobbra áramlik, azaz egy adott rétegbeli neuron bemenete (inputja) a tőle balra lévő rétegbeli neuronok kimenete (outputja). Az általunk tárgyalt modell esetén nincs kapcsolat rétegen belül és távolabbi rétegek között sem. Viszont minden neuron kapcsolatban van a vele közvetlenül szomszédos rétegek minden neuronjával. A többrétegű perceptron fontos tulajdonsága, hogy minden neuronjának saját aktivációs függvénye és saját súlyai vannak.

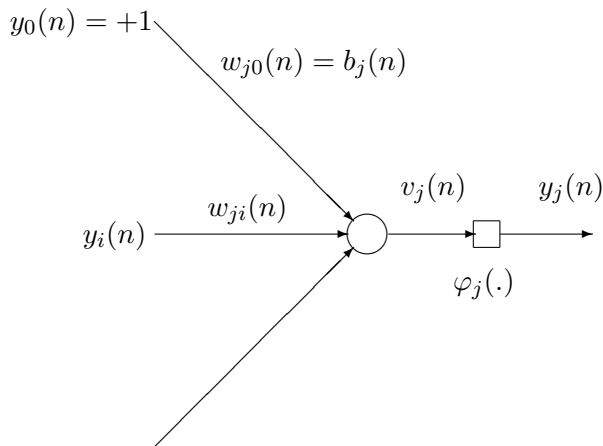
3.2. A többrétegű perceptron tanítása

A többrétegű perceptron egy előrecsatolt hálózat (feedforward network). Azaz az input jel rétegről rétegre halad előre (a 3.2. ábrán balról jobbra), az output réteg pedig megadja a kimenő jelet.

A tanítás fő lépései:

- Megadjuk a kezdeti súlyokat.
- A bemeneti jelet (azaz a tanító pontot) végigáramoltatjuk a hálózaton, de a súlyokat nem változtatjuk meg.
- Az így kapott kimeneti jelet összevetjük a tényleges kimeneti jellel.
- A hibát visszaáramoltatjuk a hálózaton, súlyokat pedig megváltoztatjuk a hiba csökkentése érdekében.

Hogyan kell megváltoztatni a súlyokat ahhoz, hogy a hiba minimális legyen? A többrétegű perceptron tanítása a *hiba visszaáramoltatása* módszerrel (hiba visszaterjesztése, error back-propagation algorithm) történik. A jelölések megértéséhez először vizsgáljunk egyetlen neuront a hálózathoz (lásd 3.3. ábra).



3.3. ábra. A többrétegű perceptron egy neuronja

A jelölések az alábbiak:

i, j, k : i -edik, j -edik, illetve k -adik neuront jelöl (az i, j, k sorrend mindig balról jobbra haladást jelöl);

n : a betanítás n -edik lépése;

$y_i(n)$: az i -edik neuron kimenete (egyben a j -edik neuron bemenete, ha j rétege közvetlenül i rétege után áll);

$y_0(n) \equiv 1$;

$w_{j0}(n) = b_j(n)$: a j -edik neuron torzítása;

$w_{ji}(n)$: az i -edik neuronból a j -edik neuronba mutató élen lévő súly;

$v_j(n)$: a j -edik neuron által produkált összegzett érték, azaz

$$v_j(n) = \sum_{i \in \{j \text{ előtti réteg}\}} w_{ji}(n) y_i(n);$$

$\varphi_j(\cdot)$: a j -edik neuronhoz tartozó transzfer függvény;

$y_j(n)$: a j -edik neuron kimenete, azaz $y_j(n) = \varphi_j(v_j(n))$;

$d_j(n)$: a kimenet igazi értéke, azaz amihez viszonyítjuk $y_j(n)$ -et (megjegyezzük, hogy $d_j(n)$ csak a kimeneti rétegben ismert, hiszen ekkor ez az $\mathbf{x}(n)$ tanító pont által meghatározott output).

Jelölje C a kimeneti réteg neuronjainak halmazát. Ekkor az n -edik lépés (négyzetes) hibája az alábbi

$$\mathcal{E}(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) = \frac{1}{2} \sum_{j \in C} (d_j(n) - y_j(n))^2.$$

Az ebből képzett átlagos hiba pedig

$$\bar{\mathcal{E}} = \frac{1}{N} \sum_{n=1}^N \mathcal{E}(n),$$

ahol N a tanító pontok számát jelöli.

A súlyok korrekcióját a *gradiens módszer* (delta rule) segítségével kapjuk, azaz

$$w_{ji}(n+1) - w_{ji}(n) = \Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}, \quad (3.1)$$

ahol $\eta > 0$ a tanulási paraméter. Viszont az összes w_{ji} súly szerint kellene képezni a (bonyolult) \mathcal{E} függvény parciális deriváltjait (azaz a gradienseit). Ez viszont numerikusan nehezen kivitelezhető. Az error back-propagation előnye az, hogy rekurzíve számítja ki a deriváltakat. Az error back-propagation eljárás kialakítása és bevezetése tette alkalmazhatóvá a többrétegű perceptront. A gradiens rétegenként visszafelé haladva határozzuk meg. A j neuron lokális gradiense megkapható a j utáni réteg lokális gradiensei segítségével. Mivel az utolsó (kimeneti) réteg lokális gradiensei közvetlenül számolhatók, így rétegenként visszafelé haladva az összes lokális gradiens kiszámolható. Ennek részletes magyarázata az alábbi.

Tekintsük a (3.1) összefüggést. Alkalmazzuk az összetett függvény differenciálási szabályát:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}.$$

Az itt szereplő függvények alakjából kapjuk, hogy:

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} &= e_j(n), & \frac{\partial e_j(n)}{\partial y_j(n)} &= -1, \\ \frac{\partial y_j(n)}{\partial v_j(n)} &= \varphi'_j(v_j(n)), & \frac{\partial v_j(n)}{\partial w_{ji}(n)} &= y_i(n). \end{aligned}$$

Ezek alapján a (3.1) képletből:

$$\Delta w_{ji}(n) = \eta e_j(n) \varphi'_j(v_j(n)) y_i(n) = \eta \delta_j(n) y_i(n),$$

ahol $\delta_j(n)$ a *lokális gradiens*, azaz

$$\boxed{\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n) \varphi'_j(v_j(n))}. \quad (3.2)$$

Most meg kell különböztetnünk két esetet: a kimeneti réteget másképpen kell kezelni, mint a rejtett rétegeket.

Amikor j kimeneti rétegbeni neuron, akkor az előbbi képlet közvetlenül alkalmazható, hiszen ekkor $e_j(n) = d_j(n) - y_j(n)$ ismert.

Ha viszont j rejtett rétegbeni neuron, akkor a $d_j(n)$ várt kimenet nem ismert. De a lokális gradiens rekurzíve mégis kiszámolható. Ugyanis az összetett függvény deriválási szabálya miatt:

$$\begin{aligned} \delta_j(n) &= -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = - \sum_{k \in \{j \text{ utáni réteg}\}} \frac{\partial \mathcal{E}(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial v_j(n)} = \\ &= \sum_{k \in \{j \text{ utáni réteg}\}} \delta_k(n) w_{kj}(n) \varphi'_j(v_j(n)). \end{aligned}$$

Hiszen ha k a j utáni rétegben van, akkor

$$v_k(n) = \sum_{l \in \{j \text{ rétege}\}} w_{kl}(n) \varphi_l(v_l(n))$$

az MLP szerkezete miatt. Ezt deriválva $v_j(n)$ szerint, $w_{kj}(n) \varphi'_j(v_j(n))$ adódik. Így igazoltuk, hogy

$$\boxed{\delta_j(n) = \varphi'_j(v_j(n)) \sum_{k \in \{j \text{ utáni réteg}\}} \delta_k(n) w_{kj}(n)}. \quad (3.3)$$

Ez az összefüggés a hiba visszaáramoltatási algoritmus legfontosabb képlete. Tehát a j neuron lokális gradiense megkapható a j utáni réteg lokális gradiensei segítségével. Mivel az utolsó (kimeneti) réteg lokális gradiensei közvetlenül számolhatóak, így rétegről-rétegre visszafelé haladva az összes lokális gradiens meghatározható. Összefoglalva az előzőeket:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \eta \left(-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} \right) \frac{\partial v_j(n)}{\partial w_{ji}(n)} = \eta \delta_j(n) y_i(n).$$

Tehát a súlyok aktualizálási mechanizmusa

$$\boxed{\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)},$$

ahol $y_i(n)$ a j réteget közvetlenül megelőző rétegbeli i neuron kimenete, $\delta_j(n)$ pedig (3.2) és (3.3) alapján kapható meg (a kimeneti, ill. a rejtett rétegekben).

Az algoritmus megállítása. A hiba nullára redukálása általában nem érhető el, sőt ez nem is helyes cél az esetek többségében. A másik probléma a hosszú futási idő. Csupán gyakorlati tanácsok adhatóak a megállítási szabályra.

- Első lehetséges megállási szabály. Ha a gradiens vektor kicsi, akkor megállunk, mivel ilyenkor már jelentősen nem fognak változni a súlyok.
- Második lehetséges megállási szabály. Ha egy epoch alatt az átlagos négyzetes hiba csökkenése kicsi.
- Harmadik lehetséges megállási szabály. Ha az epochok száma vagy a futási idő túl nagy.

Az error back-propagation algoritmus, mivel a gradiens módszeren alapul, érzékeny a kezdeti értékekre. Az is előfordul, hogy nem globális, hanem csupán egy lokális minimumot talál meg. Tehát a módszer inicializálása (azaz a kezdeti súlyok megválasztása) szintén fontos feladat.

A többrétegű perceptron használatának lépései:

1. Megszerkesztjük a hálózatot (elméleti megfontolások vagy korábbi tapasztalat alapján). Azaz megadjuk a hálózat típusát, a rétegek számát, a rétegekben a neuronok számát, valamint a transzfer függvényeket.
2. Meghatározzuk a tanítás módszerét. Betanítjuk a hálózatot az adott tanító pontokkal.

3. Teszteljük a hálózatot (tesztpontokkal, esetleg magukkal a tanító pontokkal). Ekkor vagy elfogadjuk, vagy új hálózatot alakítunk ki. Ha új hálózat szükséges, akkor visszatérünk az 1. pontra.
4. Az elfogadott hálózatot használjuk (új adatokra, azaz olyanokra, melyek se nem tanító pontok, se nem tesztpontok). Ekkor természetesen az adott inputhoz tartozó outputot nem ismerjük. A hálózat által nyújtott outputot fogadjuk el.

3.3. A back-propagation eljárás variánsai és tulajdonságai

Az MLP tanításának számos változatát dolgozták ki, az egyes programcsomagok több tanítási módszert is felkínálnak a felhasználónak. Ezek közül kell a konkrét feladatnak megfelelőt kiválasztani. Itt a teljesség igénye nélkül felsorolunk néhány módszert, amelyek közül többet ki is fejtünk ebben a szakaszban és az Appendixben.

Talán a legfontosabb megjegyzés: a numerikus matematikában számos általános eljárás létezik az \mathcal{E} célfüggvény \mathbf{w} szerinti minimumhelyének meghatározására. Ezeket az MLP esetén úgy alkalmazzák, hogy a gradienst az error back-propagation algoritmussal kiszámítják, majd a konkrét eljárásban felhasználják.

Erre a leginkább szemléletes példa a konjugált gradiens módszer, azaz amikor a gradienst az error back-propagation algoritmussal kiszámoljuk, de a lépés nem a negatív gradiens irányába történik, hanem bizonyos ortogonalizációs eljárással meghatározott, módosított irányba. A konjugált gradiens módszernek több variánsa is létezik: a Fletcher–Reeves-formula (ennek MLP-beli megvalósítása a conjugate gradient back-propagation with Fletcher–Reeves updates), a Polak–Ribière-formula (conjugate gradient back-propagation with Polak–Ribière updates), a Powell–Beale-formula (conjugate gradient back-propagation with Powell–Beale restarts).

A kvázi Newton-módszerek közül leggyakrabban a Broyden–Fletcher–Goldfarb–Shanno-formulát használják (BFGS quasi-Newton back-propagation). Ehhez természetesen a gradienst az error back-propagation algoritmussal számítják ki.

Szintén használatos a Levenberg–Marquardt-eljárás (Levenberg–Marquardt back-propagation).

A gradiens módszert finomíthatjuk a korábbi lépések bevonásával is, ennek neve általánosított delta szabály, vagy momentum módszer (gradient descent with momentum back-propagation).

A tanító pontok feldolgozása történhet egyenként (soros feldolgozás), vagy epochonként (kötegetelt feldolgozás).

Megjegyezzük még, hogy az egyes módszereknek további variánsai is léteznek, bizonyos módszerek pedig egymással is kombinálhatóak.

3.3.1. Az aktivációs függvény deriváltjáról

Az aktivációs függvényeket már a 2.2. szakaszban tárgyaltuk. Az MLP-nél nyilván csak differenciálható transzfer függvényt alkalmazhatunk. A δ lokális gradiens kiszámításához ismernünk kell φ deriváltját.

Tekintsük a gyakran használt *logisztikus függvényt*:

$$\varphi(x) = \frac{1}{1 + \exp(-ax)}, \quad \text{ahol } a > 0 \text{ konstans.}$$

$$\varphi'(x) = a \frac{\exp(-ax)}{(1 + \exp(-ax))^2} = a\varphi(x)(1 - \varphi(x)).$$

Innen

$$\varphi'_j(v_j(n)) = a\varphi_j(v_j(n))(1 - \varphi_j(v_j(n))) = ay_j(n)(1 - y_j(n)).$$

Azaz ebben az esetben a lokális gradiens számolásánál a (3.2) és (3.3) képletekben a j neuron y_j outputja kell csupán.

Most vizsgáljuk a *tangens hiperbolikus függvényt*:

$$\varphi(x) = \frac{2}{1 + \exp(-2x)} - 1 = \tanh(x).$$

Ezért

$$\varphi'(x) = \frac{4\exp(-2x)}{(1 + \exp(-2x))^2} = (1 + \varphi(x))(1 - \varphi(x)).$$

Innen

$$\varphi'_j(v_j(n)) = (1 + \varphi_j(v_j(n)))(1 - \varphi_j(v_j(n))) = (1 + y_j(n))(1 - y_j(n)).$$

Tehát ebben az esetben is a lokális gradiens számolásánál a (3.2) és (3.3) képletekben a j neuron y_j outputja kell csupán.

3.3.2. Az η tanulási paraméter és a momentum konstans

A tanulás η paramétere (rate of learning) nagy mértékben befolyásolja a tanítási algoritmus sebességét, stabilitását. Kicsi η esetén kis mértékben változnak a súlyok, sima lesz a súlyok terében a trajektória, de lassú a konvergencia. Nagy η esetén gyorsabb a konvergencia, de az algoritmus instabillá válhat, oszcillálhat. Lehetséges az is, hogy a tanulási paramétert menet közben változtatjuk (gradient descent with adaptive learning rate back-propagation).

Az oszcillálást elkerülhetjük a delta szabály alábbi módosításával:

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n), \quad n = 1, 2, \dots,$$

ahol α általában egy pozitív szám, ami a korábbi lépés bevonásának a mértékét szabályozza (a képletben $\Delta w_{ji}(0) = 0$). α -t *momentum konstansnak* nevezzük. A módszer neve: általánosított delta szabály (vagy gradient descent with momentum back-propagation). A fenti differencia egyenlet megoldása:

$$\Delta w_{ji}(n) = \eta \sum_{t=1}^n \alpha^{n-t} \delta_j(t) y_i(t) = -\eta \sum_{t=1}^n \alpha^{n-t} \frac{\partial \mathcal{E}(t)}{\partial w_{ji}(t)}.$$

Innen látszik, hogy α -t 1-nél kisebb pozitív számnak érdemes választani. Továbbá az is, hogy az általánosított delta szabály gyorsítja a konvergenciát, de véd az oszcillációtól.

3.3.3. Soros és kötegelt tanítás

Szakasznak (epochnak) nevezzük a teljes tanító halmaz egyszeri végigáramoltatását a hálózaton.

Soros (szekvenciális) tanításnak nevezzük, ha minden egyes tanító pont után összevetjük a kapott outputot a várt outputtal, és módosítjuk a súlyokat. Előnyei: kevesebb tárolási helyre van szükség, könnyű implementálni.

Kötegelt (batch) tanításnak nevezzük, ha csak egy-egy teljes epoch után végezzük el a súlyok korrekcióját. Ehhez az átlagos négyzetes hibát kell minimalizálni:

$$\bar{\mathcal{E}} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n).$$

Ebből a képletből ki lehet számítani a gradienst az előzőekhez hasonló módon. A képletből az is látszik, hogy az eljárás elvileg stabilabb. Azonban a nagy számolási igény miatt numerikus problémák adódhatnak.

Nagy és bonyolult problémákra szívesebben alkalmazzák a szekvenciális tanítást. Tanácsos epochonként más-más (véletlen) sorrendben feldolgozni a tanító pontokat.

3.3.4. Heurisztikus megjegyzések a back-propagation algoritmusról

- Nagy és redundáns tanító halmaz esetén a soros tanítás gyorsabb, mint a kötegelt.
- A tanító pontokat úgy válasszuk meg, hogy a lehető legtöbb információt tartalmazzák. Ez elérhető, ha olyan pontot veszünk, melynél a hiba maximális. Vagy pedig olyat, amelyik „nagyon eltér” a korábbiaktól.
- Általában gyorsabb a tanulás, ha az aktivációs függvény antiszimmetrikus (azaz páratlan: $\varphi(-x) = -\varphi(x)$). Antiszimmetrikus például a tangens hiperbolikus függvény.
- Az az előnyös, ha a súlyok közel azonos mértékben tanulnak. Ahhoz, hogy ezt elérjük, a tanító pontokat „előfeldolgozni” (preprocessing) kell: centrálni, normalizálni,...

3.3.5. Osztályozás több halmaz esetén

A többrétegű perceptronnal elvégezhető az M osztályba sorolás is. Legyen tehát most m -dimenziós a bemenet, és M -dimenziós a kimenet. Az MLP által előállított függvény pedig az egyszerűség kedvéért legyen

$$\mathbf{F}: \mathbb{R}^m \longrightarrow \mathbb{R}^M.$$

Az MLP az $\mathbf{x}_j \in \mathbb{R}^m$, $j = 1, \dots, N$, tanító pontokra meghatározza az $\mathbf{y}_j = \mathbf{F}(\mathbf{x}_j)$ értékeket. De az igazi függvénynek az \mathbf{x}_j tanító pontokra ismerjük az igazi értékét: \mathbf{d}_j . Mivel osztályozásról van szó, így ha \mathbf{x}_j a k -adik osztályba tartozik, akkor a \mathbf{d}_j vektor k -adik komponense legyen 1, többi pedig 0:

$$\mathbf{d}_j = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow k\text{-adik elem.}$$

Az

$$R = \frac{1}{2N} \sum_{j=1}^N \|d_j - F(x_j)\|^2$$

hibát kell minimalizálni. A fenti esetben tehát az output rétegben M neuron van.

Azonban az osztályokba sorolás megoldható olyan hálózattal is, amelyben egyetlen output neuron van. Ekkor ezen egydimenziós érték különböző (diszjunkt) intervallumokba esése jelentheti a különböző osztályokat.

3.3.6. Az MLP mint univerzális approximátor

Egy rejtett rétegű perceptronnal tetszőleges folytonos függvény approximálható. Ez az alábbi tétel lényege.

3.1. Tétel. Legyen $\varphi: \mathbb{R} \rightarrow \mathbb{R}$ monoton növekvő függvény, melyre $\varphi(-\infty) = -1$, $\varphi(\infty) = 1$. Legyen $f: [a, b]^{m_0} \rightarrow \mathbb{R}$ folytonos, és legyen $\varepsilon > 0$. Ekkor léteznek m_1 , a_0 , valamint a_i, b_i, w_{ij} konstansok ($i = 1, 2, \dots, m_1$ és $j = 1, 2, \dots, m_0$) úgy, hogy

$$\left| a_0 + \sum_{i=1}^{m_1} a_i \varphi \left(\sum_{j=1}^{m_0} w_{ij} x_j + b_i \right) - f(x_1, \dots, x_{m_0}) \right| < \varepsilon$$

minden $(x_1, \dots, x_{m_0}) \in [a, b]^{m_0}$ esetén.

A tétel bizonyítását és további irodalmi utalásokat megtalálhatjuk a [28] monográfiában, lásd még [18].

3.3.7. Általánosítás

A hálózatot akkor nevezzük betanítottnak, ha a tanító pontokra helyes eredményt ad. Ez függvényillesztésnél azt jelenti, hogy a kapott függvény a tanító pontokra jól illeszkedik. Osztályozás esetén pedig azt, hogy a tanító pontokat (vagy azok többségét) helyesen osztályozza. De sajnos még megtörténhet, hogy a többi adatra (pl. teszt adatokra) nem helyesen viselkedik a hálózat. Ez természetesen problémát jelent a használatban.

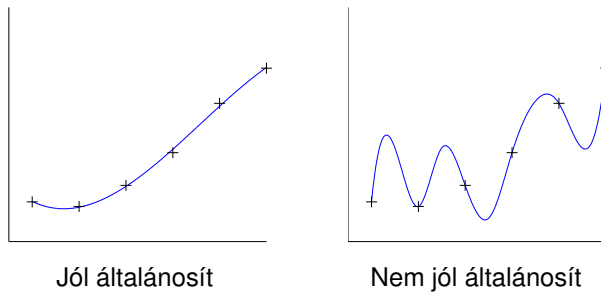
Azt mondjuk, hogy a betanított hálózat *jól általánosít*, ha helyes eredményt szolgáltat minden adatra, nem csak a tanító pontokra. Ezt természetesen csak a teszt adatokra tudjuk ellenőrizni.

Amikor a hálózat csak a tanító pontokra működik jól (túlságosan is jól), de nem jól általánosít, azaz a többi adatra hibás eredményt ad, akkor tipikusan az alábbi láthatjuk:

- függvényillesztésnél a tanító pontokra jól illeszkedő, de nagyon „hullámzó” függvényt;
- osztályozásnál a tanító pontokat jól osztályozó, de nagyon „szabálytalan” alakú tartományokat.

Ekkor mondjuk a hálózatot túlillesztettnek (túltanítottnak). Ezen hiba ellen egyrészt úgy védekezhetünk, hogy nem szerkesztünk túl bonyolult hálózatot, másrészt pedig úgy, hogy az elérendő legkisebb négyzetes hibát nem állítjuk túlságosan kicsire.

A 3.4. ábra bal oldalán egy jó függvény illesztés látható (azaz ez a modell jól általánosít), míg a jobb oldalán a tanító pontokra jól illeszkedő, de nagyon „hullámzó” függvény (azaz ez a modell nem jól általánosít).



3.4. ábra. Ugyanazon tanító pontokra két függvény illesztése

3.3.8. A konjugált gradiens módszer

A konjugált gradiens módszer általánosan használt szélsőérték meghatározására. Az átlagos hiba $\mathbf{w}(n)$ körüli Taylor-sora:

$$\begin{aligned}\bar{\mathcal{E}}(\mathbf{w}(n) + \Delta\mathbf{w}(n)) &= \bar{\mathcal{E}}(\mathbf{w}(n)) + \mathbf{g}(\mathbf{w}(n))^\top \Delta\mathbf{w}(n) + \\ &+ \frac{1}{2} \Delta\mathbf{w}(n)^\top \mathbf{H}(\mathbf{w}(n)) \Delta\mathbf{w}(n) + \dots,\end{aligned}$$

ahol $\mathbf{g}(\mathbf{w})$ a gradiens vektor, $\mathbf{H}(\mathbf{w})$ pedig a Hesse-mátrix. A gradiens módszer esetén a súly korrekciója a negatív gradiens irányába történik:

$$\Delta\mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n) = -\eta \mathbf{g}(\mathbf{w}(n)).$$

Viszont a konjugált gradiens módszer is alkalmazható, hiszen a fenti Taylor-sor alapján közelítőleg egy

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{b}^\top \mathbf{x} + c$$

alakú kvadratikus formát kell minimalizálni. Tehát úgy fogunk eljárni, hogy a gradienst a korábban megismert error back-propagation eljárással kiszámítjuk, de azt a konjugált gradiens módszer alapján úgy módosítjuk, mintha a közelítő kvadratikus forma minimumát keresnénk.

A formulák levezetése az Appendixben található, itt csak a végeredményként kapott algoritmust ismertetjük.

Jelölje tehát $\mathbf{s}(0), \mathbf{s}(1), \dots, \mathbf{s}(n-1)$ azokat az irányokat, amelyek felé már léptünk. A rekurzív eljárás az alábbi.

1. Inicializálás. Legyen $\mathbf{w}(0)$ a kezdeti súlyvektor. Alkalmazzuk ebben a pontban a back-propagation eljárást a $\mathbf{g}(0)$ gradiens kiszámítására ugyanúgy, ahogyan korábban megismertük. Legyen

$$\mathbf{s}(0) = \mathbf{r}(0) = -\mathbf{g}(0)$$

a negatív gradiens, tehát az az irány, amerre $\mathbf{w}(0)$ -at változtatjuk.

2. Lépés $n-1$ -ről n -re. Ha az $\mathbf{s}(0), \dots, \mathbf{s}(n-1)$ irányok már megvannak, akkor megvan a $\mathbf{w}(n-1)$ súlyvektor is. Keressük az új súlyvektort

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \eta(n-1)\mathbf{s}(n-1)$$

alakban. Úgy határozzuk meg $\eta(n-1)$ -et, hogy $\bar{\mathcal{E}}(\mathbf{w}(n))$ minimális legyen. (Ez ún. direkt keresés, azaz egy egyenes mentén való keresés, ami numerikusan megoldható.)

Ezután a $\mathbf{w}(n)$ pontban back-propagation algoritmussal keressük meg $\bar{\mathcal{E}}$ gradienstét, azaz $\mathbf{g}(n)$ -et. Legyen $\mathbf{r}(n) = -\mathbf{g}(n)$. Ha $\|\mathbf{r}(n)\| \leq \delta\|\mathbf{r}(0)\|$, ahol $\delta > 0$ előre adott kis szám, akkor megállunk. Ellenkező esetben folytatjuk. Viszont a következő lépés nem a $-\mathbf{g}(n)$ negatív gradiens irányba történik, hanem a konjugált gradiens módszer szerint az

$$\mathbf{s}(n) = \mathbf{r}(n) + \beta(n)\mathbf{s}(n-1)$$

irányba, ahol (a *Fletcher-Reeves-formulaszerint*)

$$\beta(n) = \frac{\mathbf{r}(n)^\top \mathbf{r}(n)}{\mathbf{r}(n-1)^\top \mathbf{r}(n-1)}.$$

Ezután n -et 1-gyel megnöveljük, és visszatérünk a **2.** lépés elejéhez.

A Fletcher-Reeves-formula levezetése az Appendixben található. Megjegyezzük, hogy a Fletcher-Reeves-formula helyett használható a *Polak-Ribière-formula* is, ami szerint

$$\beta(n) = \frac{\mathbf{r}(n)^\top (\mathbf{r}(n) - \mathbf{r}(n-1))}{\mathbf{r}(n-1)^\top \mathbf{r}(n-1)}.$$

3.3.9. Kvázi Newton-módszerek

A kvázi Newton-módszerek közül a leggyakrabban használt Broyden–Fletcher–Goldfarb–Shanno-formula algoritmusát ismertetjük. (Részletesebb kifejtését pedig az Appendixben adjuk.) Tehát az \mathcal{E} célfüggvény minimumát keressük. Jelölje \mathbf{g}_k a gradiens k -adik értékét, B_k a közelítő Hesse-mátrix k -adik értékét, \mathbf{w}_k a minimumhely k -adik közelítését. Legyen \mathbf{w}_0 a kezdeti közelítés, B_0 a közelítő Hesse-mátrix kezdeti értéke (általában az egységmátrix). Határozzuk meg az \mathcal{E} célfüggvény \mathbf{g}_0 gradiensét a \mathbf{w}_0 pontban.

Az iterációs lépések:

- Oldjuk meg a $B_k \mathbf{s} = -\mathbf{g}_k$ egyenletet, a megoldást jelölje \mathbf{s}_k .
- Egyenes mentén történő keresést alkalmazunk. Az aktuális \mathbf{w}_k pontból \mathbf{s}_k irányában keressük meg az \mathcal{E} célfüggvény minimumhelyét. Ez a pont lesz a minimumhely következő közelítése: \mathbf{w}_{k+1} .
- Számítsuk ki az \mathcal{E} célfüggvény \mathbf{g}_{k+1} gradiensét a \mathbf{w}_{k+1} pontban.
- Legyen $\boldsymbol{\delta}_k = \mathbf{w}_{k+1} - \mathbf{w}_k$ a változó differenciája, $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ pedig a gradiens differenciája.
- Legyen

$$B_{k+1} = B_k + \frac{\boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^\top}{\boldsymbol{\gamma}_k^\top \boldsymbol{\delta}_k} - \frac{B_k \boldsymbol{\delta}_k \boldsymbol{\delta}_k^\top B_k}{\boldsymbol{\delta}_k^\top B_k \boldsymbol{\delta}_k}$$

a közelítő Hesse-mátrix aktualizálása.

Az MLP esetén természetesen a gradienst az error-back-propagation eljárással számítjuk ki.

3.3.10. A Levenberg–Marquardt-eljárás

A Levenberg–Marquardt-módszereket szokták a korlátozott lépésű módszerek, illetve a megbízhatósági tartomány módszerek között tárgyalni. Azaz úgy tekintjük, hogy egy bizonyos tartományon a másodfokú Taylor-közelítés elég jó, és ezen a tartományon belül keressük a közelítő polinom minimumát. Két szempont között kell egyensúlyoznunk: a környezet legyen minél tágabb, de a választott környezetben legyen megfelelő a közelítés.

Tekintsük a négyzetes hibafüggvényt:

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2} \|\mathbf{e}(\mathbf{w})\|^2 = \frac{1}{2} \sum_{i=1}^p e_i^2(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^p (d_i - y_i(\mathbf{w}))^2.$$

Az \mathcal{E} többváltozós célfüggvény minimumát keressük. Itt $\mathbf{w} = (w_1, \dots, w_m)^\top$ m -dimenziós változó és $\mathbf{e}(\mathbf{w}) = (e_1(\mathbf{w}), \dots, e_p(\mathbf{w}))^\top$ a p -dimenziós hiba vektor. $J(\mathbf{w})$ jelölje a $p \times m$ -es Jacobi-mátrixot:

$$J(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_1(\mathbf{w})}{\partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_p(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_p(\mathbf{w})}{\partial w_m} \end{bmatrix}.$$

A rövidség kedvéért jelölje \mathbf{w}_k a minimumhely k -adik közelítését. Legyen $J_k = J(\mathbf{w}_k)$, $\mathbf{e}_k = \mathbf{e}(\mathbf{w}_k)$ a Jacobi-mátrix, illetve a hiba vektor értéke a k -adik közelítés esetén. A közelítő Taylor-polinom alábbi módosítását használjuk:

$$q_k(\boldsymbol{\delta}) = \mathcal{E}(\mathbf{w}_k) + \mathbf{e}_k^\top J_k \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top J_k^\top J_k \boldsymbol{\delta}.$$

A Levenberg–Marquardt-algoritmus egy lehetséges változata az alábbi:

(i) Adott \mathbf{w}_k és $\nu_k > 0$, számítsuk ki az \mathbf{e} hiba vektor \mathbf{w}_k -beli Jacobi-mátrixát: J_k ;

(ii) oldjuk meg a

$$(J_k^\top J_k + \nu_k I) \boldsymbol{\delta} = -J_k^\top \mathbf{e}_k$$

egyenletet, a megoldás legyen $\boldsymbol{\delta}_k$ (itt I az egységmátrix);

(iii) számítsuk ki az $\mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}_k)$ függvényértéket és a közelítés jóságát mérő

$$r_k = \frac{\mathcal{E}(\mathbf{w}_k) - \mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}_k)}{q_k(\mathbf{0}) - q_k(\boldsymbol{\delta}_k)}$$

hányadost;

(v) ha $r_k < 0.25$, akkor legyen $\nu_{k+1} = 4\nu_k$,
 ha $r_k > 0.75$, akkor legyen $\nu_{k+1} = \nu_k/2$,
 egyébként legyen $\nu_{k+1} = \nu_k$;

(vi) ha $r_k \leq 0$, akkor legyen $\mathbf{w}_{k+1} = \mathbf{w}_k$, egyébként legyen $\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\delta}_k$.

A fenti algoritmusnak számos változata van, azoknak pedig számos implementációja. Az Appendixben alaposabban magyarázzuk az algoritmust. További részletekről, valamint a Levenberg–Marquardt-algoritmus jó tulajdonságairól lásd a Fletcher [14] és a Nocedal és Wright [34] műveket.

3.4. A hálózat metszése

A hálózat felépítésére két út kínálkozik.

- A hálózat növelése. Ekkor kiindulunk egy kicsiny hálózattól. Ha azt nem vagyunk képesek jól betanítani, akkor bővítjük újabb neuronnal vagy réteggel. Ezt addig folytatjuk, amíg nem kapunk olyan hálózatot, amely már jól betanítható.
- *A hálózat metszése* (network pruning). Ez pont az ellenkezője az előzőnek. Kiindulunk egy nagy és jól betanított hálózattól. Ezután csökkentjük a neuronok és súlyok számát addig, hogy még jó eredményt adjon.

Ebben a szakaszban a hálózat metszésével foglalkozunk. A metszés fő indoka az alábbi. Egy nagy hálózatot általában sikerül betanítani, azaz a minta pontokhoz illeszteni. De lehet, hogy túl bonyolult lett a hálózat. Ennek hátránya egyrészt a lassúság lehet, másrészt a túlillesztés. Azaz nagyon pontosan illeszkedik a tanító pontokhoz, de nem jól általánosít.

Felsorolunk néhány hálózat metszési technikát.

3.4.1. Regularizáció

Ez a nagyon általános elv Tyihonovtól származik. Az alábbi rizikó függvényt kell minimalizálni.

$$R(\mathbf{w}) = \mathcal{E}_s(\mathbf{w}) + \lambda \mathcal{E}_c(\mathbf{w}),$$

ahol \mathbf{w} a súlyvektor. \mathcal{E}_s a már korábban megismert standard hiba, azaz a rendszer működésének hibája. Például

$$\mathcal{E}_s(\mathbf{w}) = \frac{1}{2} \sum_i (d_i - y_i(\mathbf{w}, \mathbf{x}))^2$$

egy olyan többrétegű perceptronnál, amelynél az i -edik output neuron kimenete $y_i(\mathbf{w}, \mathbf{x})$, míg az igazi output d_i . A fenti képletben $0 \leq \lambda < \infty$ a regularizációs paraméter, \mathcal{E}_c pedig az ún. *büntető függvény* (penalty function), ami csak magától a modelltől függ. Ez a modellre vonatkozó korábbi információkat tartalmazhat, illetve a bonyolult modellt bünteti. Ennek általános alakja:

$$\mathcal{E}_c(\mathbf{w}) = \frac{1}{2} \int \left\| \frac{\partial^k}{\partial \mathbf{x}^k} F(\mathbf{x}, \mathbf{w}) \right\|^2 \mu(\mathbf{x}) d\mathbf{x}.$$

Ezen általános alakot az RBF tárgyalásánál vizsgáljuk majd. Most csupán felsorolunk néhány, a gyakorlatban használt büntető függvényt.

A *súly puszttítása* (weight decay). Ekkor a büntető függvény:

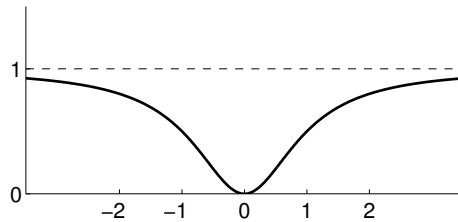
$$\mathcal{E}_c(\mathbf{w}) = \|\mathbf{w}\|^2 = \sum_i w_i^2.$$

Ezzel a súlyok egy részét 0 közelébe kényszeríthetjük. Két osztályba sorolhatjuk a súlyokat: vannak nagy és vannak kicsi hatású súlyok. Az utóbbiak felesleges súlyok, ezek törlésével a hálózat jobban fog általánosítani.

A *súly eliminálása* (weight elimination). Ekkor a büntető függvény:

$$\mathcal{E}_c(\mathbf{w}) = \sum_i \frac{(w_i/w_0)^2}{1 + (w_i/w_0)^2},$$

ahol $w_0 > 0$ rögzített paraméter. Itt a w_i/w_0 függvényében a hiba szimmetrikus. A 3.5. ábrán látható az $\frac{x^2}{1+x^2}$ függvény grafikonja. Ebből látszik, hogy



3.5. ábra. Az $\frac{x^2}{1+x^2}$ függvény

ez a büntető függvény néhány nagy súlyt is meghagyhat.

Közelítő simítás (approximate smoother). Ha egy rejtett réteg van és egyetlen kimeneti neuron, akkor a büntető függvényt definiáljuk az

$$\mathcal{E}_c(\mathbf{w}) = \sum_{j=1}^M w_{0j}^2 \|\mathbf{w}_j\|^p$$

képlettel, ahol w_{0j} a kimeneti neuron j -edik súlya, \mathbf{w}_j pedig a j -edik rejtett neuron súlyvektora.

3.4.2. A Hesse-mátrixon alapuló metszés

Fejtsük Taylor-sorba az átlagos hibát a \mathbf{w} súlyvektor körül.

$$\bar{\mathcal{E}}(\mathbf{w} + \Delta\mathbf{w}) = \bar{\mathcal{E}}(\mathbf{w}) + \mathbf{g}^\top(\mathbf{w})\Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^\top H \Delta\mathbf{w} + O(\|\Delta\mathbf{w}\|^3), \quad (3.4)$$

ahol \mathbf{w} az éppen kiszámított súly, $\Delta\mathbf{w}$ a keresett változása a súlynak, $\mathbf{g}(\mathbf{w})$ a gradiens vektor, $H = H(\mathbf{w})$ a Hesse-mátrix, $O(\|\Delta\mathbf{w}\|^3)$ pedig a közelítés hibája. Keressük meg a súlyok azon $\Delta\mathbf{w}$ változtatását, amely $\bar{\mathcal{E}}$ -nak a lehető legkisebb növekedését idézi elő. Itt a változtatás pl. törlés lehet.

Tegyük fel, hogy a tanítás befejeződött, így azt is feltehetjük, hogy a gradiens vektor értéke (közelítőleg) nulla. Ekkor a fenti (3.4) összefüggésből $\mathbf{g}^\top(\mathbf{w})\Delta\mathbf{w}$ elhagyható, azaz

$$\Delta\bar{\mathcal{E}} = \bar{\mathcal{E}}(\mathbf{w} + \Delta\mathbf{w}) - \bar{\mathcal{E}}(\mathbf{w}) \simeq \frac{1}{2}\Delta\mathbf{w}^\top H \Delta\mathbf{w}. \quad (3.5)$$

Az *OBS* (Optimal Brain Surgeon, optimális agysebész) eljárást ismertetjük. Ennek speciális esete *OBD* (Optimal Brain Damage, optimális agyrongálás), amikor H diagonális voltát tételezzük fel. Az *OBS* célja egy súlyt nullává tenni úgy, hogy ugyanakkor a (3.5) kifejezést minimalizáljuk (azaz a hibát a lehető legkevésbé növeljük). A w_i súly nullává tétele megegyezik azzal a feltétellel, hogy $\Delta w_i + w_i = 0$, azaz

$$\mathbf{1}_i^\top \Delta\mathbf{w} + w_i = 0, \quad (3.6)$$

ahol $\mathbf{1}_i$ az i -edik egységvektor, azaz olyan vektor, amelynek az i -edik komponense 1, a többi pedig 0. Tehát egy feltételes minimum problémával állunk szemben: keressük $\frac{1}{2}\Delta\mathbf{w}^\top H \Delta\mathbf{w}$ minimumát a (3.6) feltétel mellett. A λ Lagrange-multiplikátorral megkonstruáljuk a Lagrange-függvényt:

$$S = \frac{1}{2}\Delta\mathbf{w}^\top H \Delta\mathbf{w} - \lambda(\mathbf{1}_i^\top \Delta\mathbf{w} + w_i).$$

A minimum meghatározásához deriváljuk S -et $\Delta\mathbf{w}$ szerint. Ekkor kapjuk, hogy

$$H\Delta\mathbf{w} - \lambda\mathbf{1}_i = \mathbf{0},$$

ahonnan

$$\begin{aligned} \Delta\mathbf{w} &= \lambda H^{-1}\mathbf{1}_i, \\ \mathbf{1}_i^\top \Delta\mathbf{w} &= \lambda \mathbf{1}_i^\top H^{-1}\mathbf{1}_i. \end{aligned} \quad (3.7)$$

Felhasználva a (3.6) feltételt, azt kapjuk, hogy

$$-w_i = \lambda [H^{-1}]_{i,i},$$

ahol $[H^{-1}]_{i,i}$ a H inverzének (i, i) -edik eleme. Innen λ -t kifejezve és visszahelyettesítve a (3.7) egyenlőségbe, kapjuk végül \mathbf{w} optimális változását:

$$\Delta\mathbf{w} = -\frac{w_i}{[H^{-1}]_{i,i}} H^{-1}\mathbf{1}_i.$$

Ezen pontban S értéke

$$S_i = \frac{1}{2} \left(\frac{w_i}{[H^{-1}]_{i,i}} \right)^2 \mathbf{1}_i^\top H^{-1} H H^{-1} \mathbf{1}_i = \frac{w_i^2}{2[H^{-1}]_{i,i}}.$$

S_i mutatja a w_i törléséből adódó hiba növekedést, ezt a w_i súly kidomborodásának (saliency) nevezik. Azaz a felületen a w_i súly 0-ba történő mozgatása esetén bekövetkező kidomborodás. Kiszámítjuk az összes S_i kidomborodást. Végül az OBS eljárásban a legkisebb S_i -hez tartozó w_i súlyt tesszük nullává.

A Hesse-mátrix inverzének kiszámítása. Az előző eljáráshoz meg kell határozni a Hesse-mátrix inverzét. Azonban a mátrix invertálás elvégzése numerikus problémákhoz vezethet. Az egyszerűség kedvéért tegyük fel, hogy az MLP-nek egyetlen kimeneti neuronja van. Ez pl. függvényközelítésnél realisztikus. Deriváljuk kétszer az

$$\bar{\mathcal{E}} = \frac{1}{2N} \sum_{n=1}^N (d(n) - y(n))^2$$

átlagos hibafüggvényt. Jelölje $F(\mathbf{w}, \mathbf{x})$ a hálózat által megvalósított leképezést. A két változó a \mathbf{w} súlyvektor és az \mathbf{x} input vektor, azaz $y(n) = F(\mathbf{w}(n), \mathbf{x}(n))$ az output. Az első derivált (azaz a gradiens vektor):

$$\frac{\partial \bar{\mathcal{E}}}{\partial \mathbf{w}} = -\frac{1}{N} \sum_{n=1}^N (d(n) - y(n)) \frac{\partial F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}}.$$

A második derivált (azaz maga a Hesse-mátrix):

$$\begin{aligned} \frac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{w}^2} = \frac{1}{N} \sum_{n=1}^N \left\{ \left(\frac{\partial F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}} \right) \left(\frac{\partial F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}} \right)^\top - \right. \\ \left. - (d(n) - y(n)) \frac{\partial^2 F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}^2} \right\}. \end{aligned}$$

Feltesszük, hogy a hálózat jól illeszkedik a tanító pontokra, azaz $d(n)$ és $y(n)$ közel van egymáshoz, így a fenti összegben a második tagot elhanyagolhatjuk. Így kapjuk a Hesse-mátrix következő közelítését:

$$H(N) = \frac{1}{N} \sum_{n=1}^N \left(\frac{\partial F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}} \right) \left(\frac{\partial F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}} \right)^\top. \quad (3.8)$$

Vezessük be az alábbi jelölést:

$$\boldsymbol{\xi}(n) = \frac{1}{\sqrt{N}} \frac{\partial F(\mathbf{w}, \mathbf{x}(n))}{\partial \mathbf{w}}.$$

Legyen

$$H(n) = \sum_{k=1}^n \boldsymbol{\xi}(k) \boldsymbol{\xi}(k)^\top = H(n-1) + \boldsymbol{\xi}(n) \boldsymbol{\xi}(n)^\top,$$

ahol $n = 1, 2, \dots, N$. Ez összhangban van $H(N)$ fenti (3.8) képletével. Ez a rekurzió alkalmas az alábbi mátrix inverziós lemma (*Woodbury-egyenlőség*, melynek speciális esete a Sherman–Morrison-formula $D = 1$ esetén) alkalmazásához.

3.2. Lemma. *Legyenek A és B pozitív definit mátrixok, C és D pedig tetszőleges (alkalmas méretű) mátrixok. Ha*

$$A = B + CDC^\top,$$

akkor

$$A^{-1} = B^{-1} - B^{-1}C(D^{-1} + C^\top B^{-1}C)^{-1}C^\top B^{-1}.$$

Bizonyítás. Szorozzuk össze a fenti két egyenlőség jobb oldalait (jelöljük a szorzatot X -szel):

$$\begin{aligned} X &= \left[B^{-1} - B^{-1}C(D^{-1} + C^\top B^{-1}C)^{-1}C^\top B^{-1} \right] \left[B + CDC^\top \right] = \\ &= I - \underbrace{B^{-1}C(D^{-1} + C^\top B^{-1}C)^{-1}C^\top}_{B^{-1}C(D^{-1} + C^\top B^{-1}C)^{-1}D^{-1}DC^\top} + B^{-1}CDC^\top - \\ &\quad - B^{-1}C(D^{-1} + C^\top B^{-1}C)^{-1}C^\top B^{-1}CDC^\top. \end{aligned}$$

A második tagot azért bővítettük $D^{-1}D$ -vel, hogy össze tudjuk vonni a negyedik taggal:

$$X = I - B^{-1}C(D^{-1} + C^\top B^{-1}C)^{-1}(D^{-1} + C^\top B^{-1}C)DC^\top + B^{-1}CDC^\top.$$

A második tagban két tényező egymás inverze, így

$$X = I - B^{-1}CDC^\top + B^{-1}CDC^\top = I. \quad \square$$

Alkalmazzuk most a fenti 3.2. lemmát

$$H(n) = H(n-1) + \boldsymbol{\xi}(n) \boldsymbol{\xi}(n)^\top$$

esetén. Legyen

$$A = H(n), \quad B = H(n-1), \quad C = \boldsymbol{\xi}(n), \quad D = 1.$$

A lemma alapján

$$\begin{aligned} H(n)^{-1} &= H(n-1)^{-1} - H(n-1)^{-1} \boldsymbol{\xi}(n) \times \\ &\quad \times \left(1 + \boldsymbol{\xi}(n)^\top H(n-1)^{-1} \boldsymbol{\xi}(n) \right)^{-1} \boldsymbol{\xi}(n)^\top H(n-1)^{-1}. \end{aligned}$$

Innen

$$H(n)^{-1} = H(n-1)^{-1} - \frac{H(n-1)^{-1} \boldsymbol{\xi}(n) \boldsymbol{\xi}(n)^\top H(n-1)^{-1}}{1 + \boldsymbol{\xi}(n)^\top H(n-1)^{-1} \boldsymbol{\xi}(n)},$$

mivel 1×1 -es mátrixot kell invertálni. Ezzel a rekurzióval lépésenként kiszámolható $H(N)^{-1}$.

Viszont kezdőértéket is meg kell adnunk. Mivel $H(0)$ igazi értéke 0 lenne, ami nem invertálható, így legyen $H(0) = \delta I$, ahol δ egy kicsi pozitív szám, I pedig az egységmátrix. Ezzel adódik a

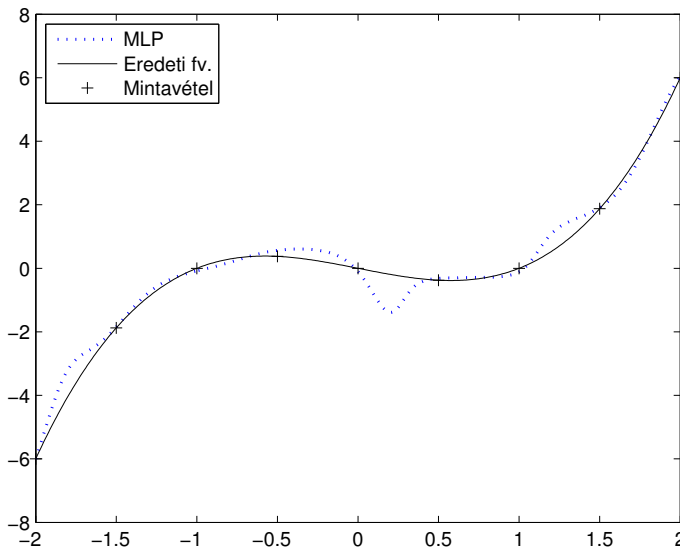
$$H(0)^{-1} = \frac{1}{\delta} I$$

kezdőérték választás a rekurzióhoz.

3.5. Numerikus példák

3.3. Példa. Közelítsük az $f(x) = x^3 - x$ függvényt a $[-2, 2]$ intervallumon, ha a megfigyelések a -2 és 2 közötti, egymástól egyenlő távolságban lévő (azaz ekvidisztáns) alappontokban a hiba nélkül mért függvényértékek. Konkrétan 9 tanító pontot választottunk. A megoldás két rejtett réteget tartalmazó hálózattal a 3.6. ábrán látható. Az első rejtett rétegben 6, a másodikban 3 neuron van és mindegyik neuronban tangens hiperbolikus transzfer függvényt használtunk. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvényvel. A tanítás konjugált gradiens módszerrel történt (pontosabban szólva konjugált gradiens back-propagation eljárással, Fletcher–Reeves-formulával). Itt a közelítő függvény viszonylag jól illeszkedik a becsülendőhöz.

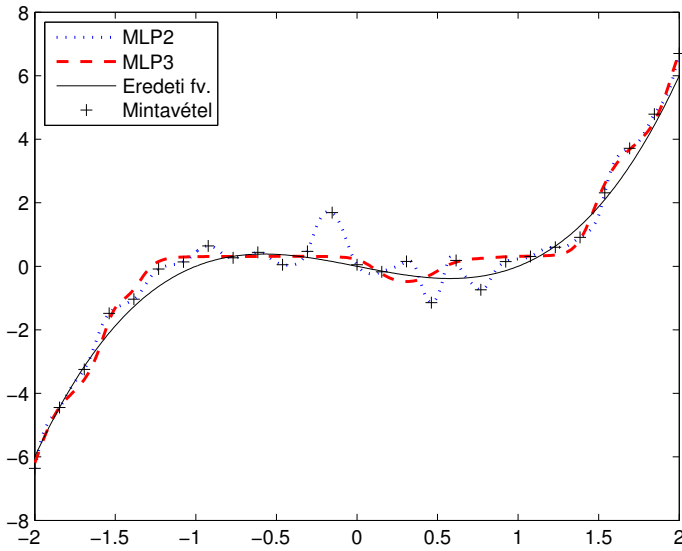
A példa következő részében a tényleges mérések egy realiztikusabb megvalósítási lehetőségét, a mérési hibával terhelt esetet tekintjük. Most az $f(x) = x^3 - x$ függvényt szintén a $[-2, 2]$ intervallumon figyeljük meg, a



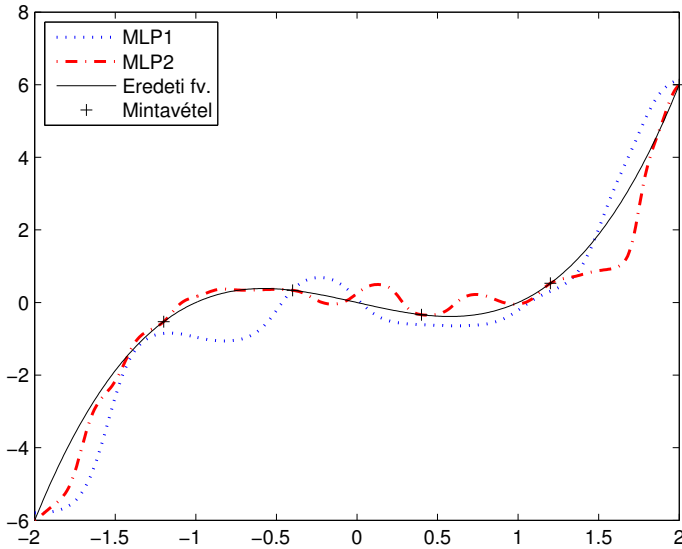
3.6. ábra. Az $f(x) = x^3 - x$ függvény megfelelő közelítése MLP-vel

megfigyelések -2 és 2 között 27 db ekvidisztáns alappontban történnek, de a függvényértékeket csak hibával terhelten tudjuk mérni. A mérési hiba 0 várható értékű, 0.5 szórású (független) normális eloszlású. Azonban közelíteni a mérési hibák nélküli függvényt kell!

A 3.7. ábrán látható két próbálkozás közül az MLP2 kevésbé sikeres: „túlillesztett”. Itt két rejtett réteg van, mindkettőben 18-18 neuron. Az első rétegben logisztikus, a másodikban tangens hiperbolikus transzfer függvényt használunk. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvényvel. A tanítás konjugált gradiens módszerrel történt. Habár a hálózat nagy méretű, és a tanító pontokban a mért (de hibás) adatokat jól közelíti, azonban rosszul általánosít. Próbálkoztunk kisebb hálózattal és nagyobb (legkisebb négyzetes \mathcal{E}) hibát engedve. Az eredmény az MLP3 (3.7. ábra), amely a tanító pontokban kevésbé pontosan illeszkedik (az egyébként is hibás) mérési eredményekre, azonban jobban általánosít. MLP3-ban két rejtett réteg van, 9 illetve 3 neuronnal. Mindkettőben tangens hiperbolikus transzfer függvényt használunk. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvényvel. A tanítás konjugált gradiens módszerrel történt. A fenti eset azt is mutatja, hogy mérési hibák esetén több tanító pont kell, mint pontos mérések esetén.



3.7. ábra. Az $f(x) = x^3 - x$ függvény közelítése mérési hibák esetén. MLP2 túl bonyolult és túlillesztett, MLP3 egyszerűbb és jobban általánosít



3.8. ábra. Az $f(x) = x^3 - x$ függvény közelítése kevés tanító pont esetén. MLP1 kevés neuron, MLP2 sok neuron

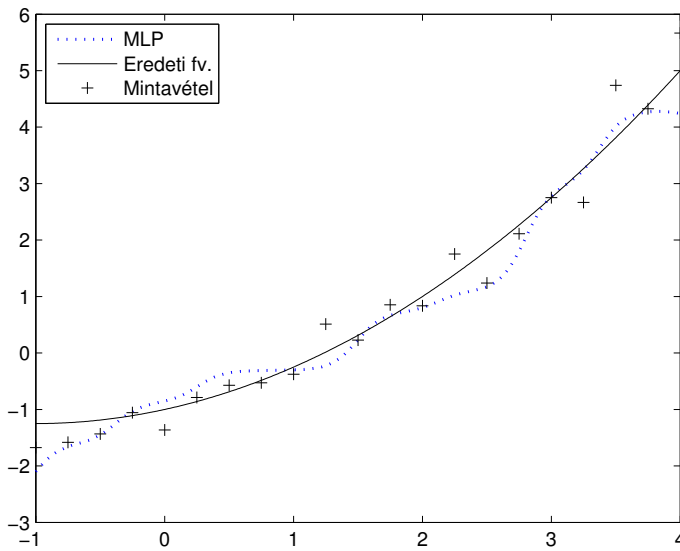
Vizsgáljuk meg, hogy mi történik kevesebb tanító pont esetén! Ha csak 6 tanító pontunk van, akkor már kevésbé jó közelítéseket kapunk (annak ellenére, hogy maga a függvény nem „bonyolult”). Az első megoldás a 3.8. ábrán látható MLP1. Itt egy rejtett réteg van, 9 neuronnal, tangens hiperbolikus transzfer függvénnyel. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvénnyel. A legkisebb négyzetes hibát nem szorítottuk túlságosan le. A neurális hálózat egyszerűsége ellenére a közelítő függvény elég jó, de sajnos nem tökéletesen illeszkedik a becslendőhöz. A 3.8. ábrán látható MLP2 esetén viszont bonyolult hálózatot alkalmaztunk. Az első rejtett rétegben 15 neuron van logisztikus transzfer függvényekkel, a másodikban 9 neuron van tangens hiperbolikus transzfer függvényekkel, a kimenetiben pedig egy neuron lineáris transzfer függvénnyel. A tanítás Levenberg–Marquardt-féle back-propagation eljárással történt. Ezzel a közelítő függvény ismét „ugrál”.

3.4. Példa. Közelítsük az

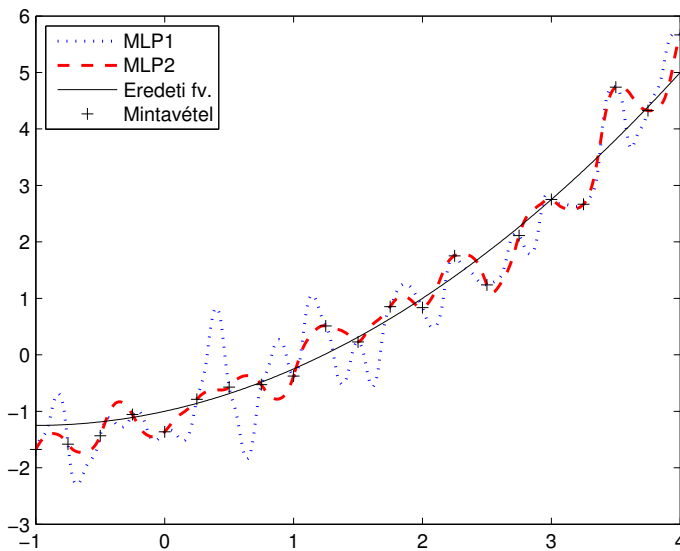
$$f(x) = \frac{1}{4}x^2 + \frac{1}{2}x - 1$$

függvényt a $[-1, 4]$ intervallumon, ha a megfigyelések a -1 és 4 közötti, egymástól egyenlő távolságban lévő (azaz ekvidisztáns) alappontokban a hibával terhelt mért függvényértékek. Konkrétan 21 tanító pontot választottunk. A mérési hiba pedig minden tanító pont esetén 0 várható értékű és 0.3 szórású normális eloszlású véletlen szám volt. Az első megoldás a 3.9. ábrán látható. Itt egy rejtett réteg van, 9 neuronnal, tangens hiperbolikus transzfer függvénnyel. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvénnyel. A tanítás konjugált gradiens módszerrel történt (konjugált gradiens back-propagation eljárással, Fletcher–Reeves-formulával). Az \mathcal{E} legkisebb négyzetes hibát nem szorítottuk túlságosan le. A neurális hálózat egyszerűsége ellenére a közelítő függvény viszonylag jó, de sajnos nem tökéletesen illeszkedik a becslendőhöz.

A 3.10. ábrán látható két „próbálkozás” azonban kevésbé sikeres: a közelítő függvények a tanító pontokban jól illeszkednek a hibával terhelt értékekre, de a közelítendő függvényt rosszul approximálják, „ugrálnak”. Az alkalmazott modellek nem megfelelőek, bonyolultak, túlillesztettek. MLP1-ben egy rejtett réteg van, 42 neuronnal, tangens hiperbolikus transzfer függvénnyel. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvénnyel. A tanítás konjugált gradiens back-propagation eljárással, Fletcher–Reeves-formulával történt. MLP2-ben két rejtett réteg van, mindkettőben 21-21 neuron. Az első rétegben logisztikus, a másodikban tangens hiperbolikus



3.9. ábra. Az $f(x) = \frac{1}{4}x^2 + \frac{1}{2}x - 1$ függvény megfelelő közelítése MLP-vel (mérési hibák esetén)



3.10. ábra. Az $f(x) = \frac{1}{4}x^2 + \frac{1}{2}x - 1$ függvény közelítése (mérési hibák esetén). MLP1 és MLP2 túl sok neuron, túlillesztett

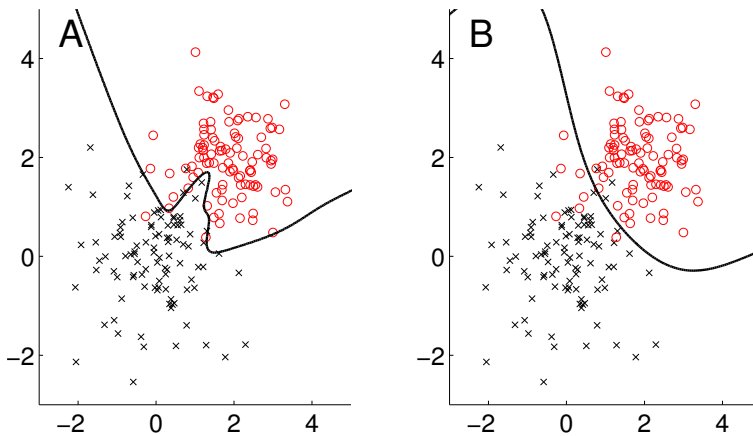
transzfer függvényt használunk. A kimeneti rétegben egyetlen neuron talál-

ható lineáris transzfer függvényvel. A tanítás Levenberg–Marquardt back-propagation eljárással történt. Mindkét utóbbi hálózaton a legkisebb négyzetes hibát leszorítottuk.

3.5. Példa. Két halmaz szétválasztása MLP segítségével megvalósítható. Legyen az egyik halmaz 100 elemű minta kétdimenziós, $(0, 0)^\top$ várható érték vektorú és

$$\begin{pmatrix} 0.8 & 0 \\ 0 & 0.8 \end{pmatrix}$$

szórásmátrixú normális eloszlásból. A másik halmaz pedig ugyanilyen minta, azzal a különbséggel, hogy ott a várható érték vektor $(2, 2)^\top$. A 3.11. ábra



3.11. ábra. Két normális eloszlás szeparálása MLP-vel.

A rész: túl sok neuron, rosszul általánosít. B rész: megfelelő számú neuron, jól általánosít

mind A, mind B részén \times jelöli az első, \circ pedig a második halmaz elemeit. Azonban a megkonstruált elválasztó görbék különböznek. Az A részen 3 rejtett rétegű MLP szeparálása látható. Az első rejtett rétegben 6 neuron van logisztikus, a másodikban 20 neuron szintén logisztikus, a harmadikban pedig 2 neuron tangens hiperbolikus transzfer függvényvel. A kimeneti rétegben 1 neuron van tangens hiperbolikus transzfer függvényvel. A tanítás konjugált gradiens back-propagation eljárással, Fletcher–Reeves-formulával történt. Az elválasztó görbe a tanító pontokat helyesen szeparálja, azonban rosszul általánosít. Hiszen tudjuk, hogy esetünkben a két normális eloszlást optimális módon egy egyenes osztaná ketté. Az ábra B részén lévő szeparálás azért sokkal jobb, mert ugyan az aktuális tanító pontokat nem tökéletesen osztja ketté, azonban jól általánosít. Ezt az eredményt két rejtett réteggel,

azokban 5, ill. 3 neuronnal értük el. Az első rétegben logisztikus, a másodikban pedig tangens hiperbolikus transzfer függvényt használunk. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvénnyel. A tanítás Levenberg–Marquardt back-propagation eljárással történt.

3.6. Feladatok

1. Határozzuk meg a tangens hiperbolikus aktivációs függvény deriváltját! Ez alapján számítsuk ki az MLP lokális gradiensét mind az output rétegben, mind a rejtett rétegekben!
2. Rajzoljuk fel a 3.1. Tételbeli egy rejtett rétegű perceptront!
3. Igazoljuk a 3.1. Tételt a Stone–Weierstrass-tételre visszavezetve!
4. Oldjuk meg MLP-vel a 2. fejezet feladatai között definiált XOR-problémát!
5. Közelítsük az alábbi függvényeket MLP segítségével!

$$f(x) = x^2, \quad f(x) = \log x, \quad f(x) = \sin x, \quad f(x) = (\sin x)/x,$$

$$f(x, y) = x^2 + y^2, \quad f(x, y) = x^2/y.$$

Válasszunk alkalmas tartományt, ahol a fenti függvényeket közelíteni akarjuk! Generáljunk különböző számú és sűrűségű tanító pontokat! Vizsgáljuk külön azt az esetet is, amikor a függvényértéket csak valamilyen véletlen hibával tudjuk megfigyelni (azaz a tanító pontok ilyenek, de a közelítendő függvény a hiba nélküli)! Alkalmazzunk különböző rejtett réteg számú, ill. különböző neuron számú és transzfer függvényű MLP-eket! Ábrázoljuk az eredményt!

6. Vizsgáljuk az MLP-vel való szeparálást! Állítsuk valamilyen módon elő a tanító pontok A és B halmazát, és szeparáljuk MLP-vel! A két halmaz lehet 2- vagy 3-dimenziós, véletlen vagy determinisztikus módon megadott, akár egymásba átnyúló is. Alkalmazzunk különböző rejtett réteg számú, ill. különböző neuron számú és transzfer függvényű MLP-eket! Ábrázoljuk az eredményt!
7. Vizsgáljuk az MLP-vel való szeparálást 3 halmaz esetén! Szeparáljunk el egymástól 3 síkbeli halmazt! Ábrázoljuk az eredményt!

4. fejezet

A radiális bázis függvények

A radiális bázis függvények hálózata (Radial Basis Function Network, RBF) rokonságban áll a matematika számos konstrukciójával. A feladat minden esetben az, hogy bizonyos függvényekből kiindulva adjuk meg egy olyan függvény közelítését, amelynek alakjáról (képletéről) nem tudunk előzetesen semmit, de vannak rá méréseink. A kiinduló függvényeket alap függvényeknek (más területen magfüggvényeknek) szokás nevezni. A radiális bázis függvény olyan alap függvény, melynek értéke bizonyos középponttól való távolságtól függ csupán.

Ebben a fejezetben bevezetésképp szó lesz a radiális bázis függvényekkel történő interpolációról (ez a numerikus matematika fontos feladata). A radiális bázis függvények hálózatát viszont (büntető függvénnyel kiegészített) legkisebb négyzetes módszerrel fogjuk megkapni. Tárgyalni fogjuk a statisztika egyik alapvető eszközét, (az RBF-fel rokon) magfüggvényes becslést is. Megjegyezzük, hogy az RBF nem csak függvény közelítésre, hanem osztályozásra is használható. Ebben a fejezetben is főként Haykin [18] könyvére támaszkodunk.

4.1. A szeparálás Cover-féle elmélete

Számos hálózat (bizonyos típusú MLP, RBF, SVM) a következő séma szerint épül fel. Az adatokat nem-lineáris transzformációnak vetjük alá a rejtett rétegben (ahol sok neuron található) viszont az output rétegben csak lineáris transzformáció van. Ennek elvi hátterét világítja meg Cover szeparálási elmélete.

Tekintsünk egy osztályozási problémát. A nem-lineáris szeparáláshoz bonyolult hálózatot kell építenünk (MLP), viszont a lineáris szeparálás egyet-

len neuronnal (perceptronnal) is elvégezhető. Vissza lehet-e vezetni a nem-lineáris szeparálást lineárisra? Látni fogjuk, hogy ez alkalmas transzformációval megtehető.

A linearizálás valójában általános eszköz a feladatok numerikus megoldására. A számítógépek elterjedése előtt a szorzás és osztás elvégzésére használt „logarléc” is ilyen elven működött. Szintén a logaritmus függvény tulajdonságain alapuló példa az alábbi.

4.1. Példa. A Newton-féle gravitációs törvény alapján két, egymástól r távolságra lévő, m_1 és m_2 tömegű test közötti gravitációs erő:

$$f(m_1, m_2, r) = C \frac{m_1 m_2}{r^2},$$

ahol C pozitív konstans. Mindkét oldal logaritmusát véve

$$\log f(m_1, m_2, r) = \log C + \log m_1 + \log m_2 - 2 \log r.$$

Tehát a szereplő mennyiségek logaritmusai között már lineáris a függvénykapcsolat.

Cover [7] cikkében javasolta, hogy nem-lineáris transzformációval vigyük át a feladatot magasabb dimenzióba. Cover eredményét az alábbi népszerű formában szokták emlegetni (lásd Haykin [18]). „Egy osztályozási problémát nem-lineáris módon magas dimenziós térbe átvíve, nagyobb a valószínűsége, hogy lineárisan szeparálható lesz, mint alacsony dimenzióban.” Tehát a bemeneti vektorokat át fogjuk transzformálni magasabb dimenziós térbe, ott pedig egy hipersíkkal választjuk el az osztályokat. Ezt az eljárást fogjuk később az SVM esetén is alkalmazni.

Legyen H a tanító pontok N -elemű halmaza: $H = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. Ezt a halmazt kell a diszjunkt A és B részhalmazokra felbontani (két részre osztás, $\{A, B\}$ -dichotómia). Legyenek a tanító pontok m_0 -dimenziósak. Defináljuk a $\varphi(\mathbf{x})$ vektor értékű függvényt (aminek tehát a komponensei valós értékű függvények) az alábbi módon:

$$\varphi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \dots, \varphi_{m_1}(\mathbf{x}))^\top, \quad \mathbf{x} \in \mathbb{R}^{m_0}.$$

Azaz φ leképezés az m_0 -dimenziós és az m_1 -dimenziós euklideszi terek között. Az előzőek értelmében általában $m_1 > m_0$. Azt mondjuk, hogy a φ függvény a tulajdonságok terébe (feature space) transzformálja a problémát.

4.2. Definíció. Azt mondjuk, hogy a H halmaz $\{A, B\}$ -dichotómiája φ -szeparálható, ha létezik egy m_1 -dimenziós \mathbf{w} vektor úgy, hogy

$$\mathbf{w}^\top \varphi(\mathbf{x}) > 0, \quad \text{ha } \mathbf{x} \in A,$$

$$\mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}) < 0, \quad \text{ha } \mathbf{x} \in B.$$

Ekkor a szeparáló felület egyenlete

$$\mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}) = 0$$

alakú.

Cover fenti állítása szerint a dimenzió növelésével növekszik a lineáris szeparálhatóság valószínűsége. Határozzuk meg, hogy mennyi a valószínűsége annak, hogy az m_1 -dimenziós térben egy N elemű halmazt lineárisan ketté tudunk választani.

Tegyük fel, hogy a $\boldsymbol{\varphi}(\mathbf{x}_1), \boldsymbol{\varphi}(\mathbf{x}_2), \dots, \boldsymbol{\varphi}(\mathbf{x}_N)$ m_1 -dimenziós vektorok általános helyzetűek, azaz közülük bármely m_1 számú (vagy m_1 -nél kevesebb) lineárisan független. Ekkor a $\boldsymbol{\varphi}(\mathbf{x}_1), \boldsymbol{\varphi}(\mathbf{x}_2), \dots, \boldsymbol{\varphi}(\mathbf{x}_N)$ pontoknak

$$2 \sum_{i=0}^{m_1-1} \binom{N-1}{i} \quad (4.1)$$

számú olyan kettéosztása van, amely (origón átmenő) síkkal valósítható meg. Ez a tétel Cover [7] cikkében található (de a szerző felsorolja, hogy korábban mely művekben szerepelt ugyanez az állítás). Lásd még [28].

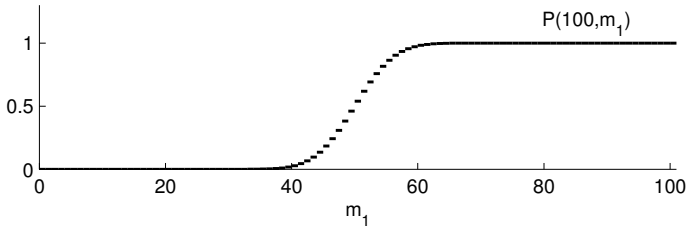
Tegyük fel most, hogy az $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ pontok 2^N számú lehetséges kettéosztásának (dichotómiájának) mindegyike egyformán valószínű. Jelölje $P(N, m_1)$ annak a valószínűségét, hogy az N -elemű H halmaz egy véletlenszerűen kiválasztott dichotómiája $\boldsymbol{\varphi}$ -szeparálható (azaz $\{\boldsymbol{\varphi}(\mathbf{x}_1), \boldsymbol{\varphi}(\mathbf{x}_2), \dots, \boldsymbol{\varphi}(\mathbf{x}_N)\}$ az m_1 -dimenziós térben lineárisan szeparálható). Tegyük fel szintén, hogy a $\boldsymbol{\varphi}(\mathbf{x}_1), \boldsymbol{\varphi}(\mathbf{x}_2), \dots, \boldsymbol{\varphi}(\mathbf{x}_N)$ pontok általános helyzetűek.

A (4.1) képlet alapján a valószínűség klasszikus kiszámítási módja szerint

$$P(N, m_1) = \frac{\sum_{i=0}^{m_1-1} \binom{N-1}{i}}{2^{N-1}}. \quad (4.2)$$

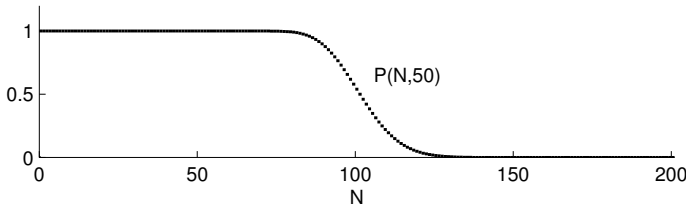
A felület szeparáló képessége

$P(N, m_1)$ fenti (4.2) képletét fix pontszám (azaz N) esetén a dimenzió (azaz m_1) függvényében tekinthetjük mint eloszlásfüggvényt. Ez természetesen a binomiális eloszlás eloszlásfüggvénye. Mint ilyen, monoton növekvő: 0-ról felnövekszik 1-re. Azaz a lineáris szeparálhatóság valószínűsége növekszik a dimenzió növelésével. $m_1 \geq N$ esetén $P(N, m_1)$ értéke 1. Ezt ábrázoltuk a 4.1. ábrán, azaz a valószínűségek változását a tanító pontok rögzített száma esetén.



4.1. ábra. A lineáris szeparálhatóság a dimenzió függvényében: $P(N, m_1)$, amikor $N = 100$

Sokkal tanulságosabb azonban a 4.2. ábra. Ebben az esetben a dimenziószámot rögzítettük, és a tanító pontok számát növeltük. A 4.2. ábrán látható, hogy a lineáris szeparálhatóság valószínűsége csökken a tanító pontok számának növelésével, sőt 0-hoz tart.



4.2. ábra. A lineáris szeparálhatóság a tanító pontok számának függvényében: $P(N, m_1)$, amikor $m_1 = 50$

Most tehát rögzítsük a tér (pontosabban szólva a feature space) m_1 dimenzióját. Legyen $\mathbf{x}_1, \mathbf{x}_2, \dots$ egy olyan sorozat, melyre minden rögzített n esetén $\varphi(\mathbf{x}_1), \dots, \varphi(\mathbf{x}_n)$ általános helyzetű, továbbá $\varphi(\mathbf{x}_1), \dots, \varphi(\mathbf{x}_n)$ dichotómiái legyenek egyforma valószínűségűek. Jelölje ξ a legnagyobb n egész számot, melyre $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ egy (véletlenszerűen választott) kettéosztása φ -szeparálható. Ekkor

$$\mathbb{P}(\xi = n) = P(n, m_1) - P(n+1, m_1) = \left(\frac{1}{2}\right)^n \binom{n-1}{m_1-1}, \quad (4.3)$$

$n = m_1, m_1 + 1, \dots$ Ennek levezetése az alábbi:

$$\begin{aligned} P(n, m_1) - P(n+1, m_1) &= \frac{1}{2^n} \left[2 \sum_{i=0}^{m_1-1} \binom{n-1}{i} - \sum_{i=0}^{m_1-1} \binom{n}{i} \right] = \\ &= \frac{1}{2^n} \left[\binom{n-1}{m_1-1} + \left\{ \binom{n-1}{m_1-1} + \binom{n-1}{m_1-2} - \binom{n}{m_1-1} \right\} + \right. \end{aligned}$$

$$+ \left\{ \binom{n-1}{m_1-2} + \binom{n-1}{m_1-3} - \binom{n}{m_1-2} \right\} + \cdots + \left\{ \binom{n-1}{0} - \binom{n}{0} \right\} \Big].$$

Itt minden kapcsos zárójelben lévő kifejezés 0, tehát az eredmény $\frac{1}{2^n} \binom{n-1}{m_1-1}$. A (4.3) képlet pedig az m_1 és $1/2$ paraméterű negatív binomiális eloszlás. Ennek a várható értéke

$$\mathbb{E}(\xi) = 2m_1,$$

a szórásnégyzete

$$\mathbb{D}^2(\xi) = 2m_1.$$

Így megkaptuk Cover [7] eredményét: „Az m_1 -dimenziós térben a lineárisan szeparálható véletlen minta maximális elemszámának várható értéke $2m_1$ ”. Ez az eredmény összhangban van a Vapnik–Chervonenkis-dimenzióval.

4.2. Interpoláció radiális bázis függvényekkel

Legyen adott az N számú, egymástól különböző elemből (alappontból) álló

$$\{\mathbf{x}_i \in \mathbb{R}^{m_0} : i = 1, 2, \dots, N\}$$

halmaz, valamint a szintén N elemű

$$\{d_i \in \mathbb{R} : i = 1, 2, \dots, N\}$$

halmaz. Keressünk egy olyan $F: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ függvényt, ami eleget tesz az

$$F(\mathbf{x}_i) = d_i, \quad i = 1, 2, \dots, N, \quad (4.4)$$

feltételeknek! Ez az *interpoláció* általános feladata.

A jól ismert Lagrange-féle interpoláció esetén az $x_i = \mathbf{x}_i$ *alappontok* valóságok, és F -et a polinomok közül választjuk. Nevezetesen, mindig találhatók olyan w_k együtthatók, hogy $F(x) = \sum_{k=0}^{N-1} w_k x^k$ kielégíti a (4.4) feltételt. Még pontosabban szólva: N alappont esetén található pontosan egy, legfeljebb $(N-1)$ -edfokú polinom, mely az alappontokon az előírt értékeket veszi fel.

Hatványfüggvények helyett más függvények lineáris kombinációival is interpolálhatunk, pl. trigonometrikus függvényekkel, vagy az alább bevezetendő radiális bázis függvényekkel.

A $\{\varphi(\|\mathbf{x} - \mathbf{x}_i\|) : i = 1, 2, \dots, N\}$ alakú függvényeket *radiális bázis függvényeknek* (radial basis function, RBF) nevezzük. Itt $\|\cdot\|$ az euklideszi norma. Az elnevezés arra utal, hogy a $\varphi(\|\mathbf{x} - \mathbf{x}_i\|)$ alap függvény az \mathbf{x}_i középpontból

kiindulva sugár irányban változik. Itt φ -t egyelőre tetszőleges függvénynek tekinthetjük, természetesen jó tulajdonságokat bizonyos függvényosztályoktól várhatunk. A radiális bázis függvények vizsgálata a matematikai analízis klasszikus területe, egy összefoglalása Buhmann [4] monográfiájában található.

Az RBF-interpoláció tehát a következő alakú F függvény választásából áll (lásd pl. Powell [38])

$$F(\mathbf{x}) = \sum_{j=1}^N w_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|), \quad (4.5)$$

ahol $\|\cdot\|$ az euklideszi norma, w_j -k az ismeretlen súlyok, φ pedig általunk választott (általában nem lineáris) függvény. A (4.4) feltételek teljesülését egyszerű behelyettesítéssel ellenőrizhetjük a (4.5) alakú függvényekre. Így kapjuk a

$$\begin{pmatrix} \varphi_{11} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{NN} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_N \end{pmatrix} \quad (4.6)$$

lineáris egyenletrendszert, ahol

$$\varphi_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i, j = 1, 2, \dots, N.$$

Legyenek

$$\begin{aligned} \mathbf{d} &= (d_1, d_2, \dots, d_N)^\top, \\ \mathbf{w} &= (w_1, w_2, \dots, w_N)^\top \end{aligned}$$

N -dimenziós vektorok, és

$$\Phi = \{\varphi_{ij} : i, j = 1, \dots, N\}$$

$N \times N$ -es mátrix. Φ -t *interpolációs mátrixnak* nevezzük. Ezekkel a jelölésekkel a (4.6) egyenletrendszer:

$$\Phi \mathbf{w} = \mathbf{d}.$$

Az ismeretlen \mathbf{w} meghatározása ebből

$$\mathbf{w} = \Phi^{-1} \mathbf{d}.$$

A (4.6) egyenletrendszer akkor és csakis akkor oldható meg minden \mathbf{d} konstans vektor esetén, ha Φ invertálható.

Számos függvény, illetve függvényosztály esetén bebizonyították, hogy Φ invertálható (lásd pl. Micchelli [31], Buhmann [4]).

4.3. Definíció. Legyen g a pozitív valós számokon értelmezett valós értékű, tetszőleges sokszor differenciálható függvény. g -t teljesen monotonnak nevezzük, ha deriváltjaira $(-1)^l g^{(l)}(x) \geq 0$ teljesül minden pozitív x és $l = 0, 1, 2, \dots$ esetén.

4.4. Tétel. Legyen g teljesen monoton és nem konstans, $\varphi(x) = g(x^2)$. Legyenek az $\{\mathbf{x}_i \in \mathbb{R}^{m_0} : i = 1, 2, \dots, N\}$ alappontok páronként különbözőek. Ekkor a Φ interpolációs mátrix pozitív definit.

4.5. Tétel. Legyen g a pozitív valós számokon értelmezett valós értékű, tetszőleges sokszor differenciálható olyan függvény, melyre g' teljesen monoton, de nem konstans, $\varphi(x) = g(x^2)$. Legyen $g(0) \geq 0$. Legyenek az $\{\mathbf{x}_i \in \mathbb{R}^{m_0} : i = 1, 2, \dots, N\}$ alappontok páronként különbözőek. Ekkor a Φ interpolációs mátrix nem szinguláris.

Például az alábbi φ függvények esetén az interpolációs mátrix mindig invertálható, amennyiben az \mathbf{x}_i tanító pontok páronként különbözőek.

(i) *Multikvadratikus:*

$$\varphi(x) = (x^2 + c^2)^{\frac{1}{2}}, \quad x \in \mathbb{R},$$

ahol $c > 0$ rögzített paraméter.

(ii) *Inverz multikvadratikus:*

$$\varphi(x) = \frac{1}{(x^2 + c^2)^{\frac{1}{2}}}, \quad x \in \mathbb{R},$$

ahol $c > 0$ rögzített paraméter.

(iii) *Gauss-függvény:*

$$\varphi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \quad x \in \mathbb{R},$$

ahol $\sigma > 0$ rögzített paraméter.

Az RBF-interpolációt neurális hálózattal is megvalósíthatjuk. Tekintsünk egy olyan hálózatot, aminek a bemenete m_0 -dimenziós, és legyen csak egyetlen kimeneti neuron. A bemeneti és a kimeneti réteg között pedig vegyünk egyetlen rejtett réteget. A bemeneti és a rejtett réteg között nem lineáris, viszont a rejtett és kimeneti réteg között lineáris kapcsolatot alakítunk ki. Pontosabban szólva, az \mathbf{x} input jelre a rejtett rétegbeli i -edik

neuron adja a $\varphi(\|\mathbf{x} - \mathbf{x}_i\|)$ outputot, a kimeneti rétegbeli neuron pedig ezek $F(\mathbf{x}) = \sum_{i=1}^N w_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|)$ súlyozott összegét. Ekkor a hálózat tulajdonképpen egy olyan függvényt valósít meg, amely az m_0 -dimenziós térből az 1-dimenziós térbe képez. Az RBF hálózatok azonban nem közvetlenül a fenti interpolációs sémát követik, hanem a legkisebb négyzetek elvét, amelyet ráadásul büntető függvények alkalmazásával teszünk a modell iránt érzékenyebbé.

4.3. A Tyihonov-féle regularizáció

Az RBF hálózat kialakításához a regularizáció elmélete nyújt elméleti alapot. A matematikai levezetés azonban csupán egy sémát szolgáltat, amelyet a neurális hálózat tényleges alkalmazásához számítógépes (kísérleti) úton konkretizálhatunk.

Haykin [18] szóhasználatát követve, direkt problémának tekintjük azon fizikai, kémiai, biológiai jelenséget, amely az adatokat produkálja. Inverz problémának pedig az adatokból a jelenség rekonstruálását. Ismeretes, hogy az inverz problémák gyakran rosszul felállítottak.

A *rosszul felállított problémák* (ill-posed problem) megoldására Tyihonov a regularizációt javasolta, lásd Tyihonov [46], Tyihonov–Arszenyin [47]. Az alábbi definíció Kabanikhin [22] cikkéből származik.

4.6. Definíció. Legyenek Q és S topologikus terek, $A: Q \rightarrow S$ pedig egy adott operátor. Az $Aq = f$ problémát (Hadamard-féle értelemben) *jól felállított inverz problémának* (well-posed) nevezzük, ha az alábbi három feltétel teljesül.

- (i) *Egzisztencia.* Bármely $f \in S$ esetén létezik $Aq = f$ -nek q_e megoldása.
- (ii) *Unicitás.* A fenti q_e egyértelmű megoldása $Aq = f$ -nek (azaz az A^{-1} inverz létezik).
- (iii) *Folytonosság (stabilitás).* Az A^{-1} inverz folytonos. Ez utóbbi metrikus terek esetén azt jelenti, hogy bármely $\varepsilon > 0$ -hoz létezik $\delta > 0$ úgy, hogy ha $d_1(f_1, f) < \delta$, akkor $d_2(A^{-1}f_1, A^{-1}f) < \varepsilon$. Itt d_1 és d_2 a két térben a távolság.

Azt a problémát, amely nem teljesíti mindhárom fenti feltételt, *rosszul felállított inverz problémának* nevezzük.

Rosszul felállított inverz probléma például egy nem invertálható mátrixú lineáris egyenletrendszer megoldásának keresése.

4.7. Példa. Rosszul felállított direkt probléma például a differenciálás. Legyen $q(x) = f'(x)$ és tegyük fel, hogy $f(x)$ -et csak egy kis perturbációval ismerjük: $f_n(x) = f(x) + \sin(nx)/\sqrt{n}$. Ekkor $f_n \rightarrow f$ teljesül, azonban az f_n deriváltja $q_n(x) = q(x) + \sqrt{n} \cos(nx)$, aminek a távolsága q -tól a ∞ -hez tart.

4.8. Példa. Rosszul felállított inverz probléma a Laplace-féle differenciálegyenletre vonatkozó Cauchy-probléma. Legyen $u(x, y)$ a következő probléma megoldása.

$$\begin{aligned}\nabla^2 u &= 0, \\ u(0, y) &= f(y), \\ \frac{\partial u(0, y)}{\partial x} &= 0.\end{aligned}$$

Itt

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

a Laplace-operátor. Legyen

$$f(y) = \frac{1}{n} \sin(ny).$$

Ekkor a fenti probléma megoldása (lásd a Feladatok között)

$$u(x, y) = \frac{1}{n} \sin(ny) \frac{e^{nx} + e^{-nx}}{2}.$$

Minden $x > 0$ esetén, n -et elég nagynak választhatjuk ahhoz, hogy f tetszőlegesen kicsi, míg u tetszőlegesen nagy legyen.

A rosszul felállítottság a mi jelenlegi szempontunkból azt jelenti, hogy egy nagy adathalmaz meglepően kevés információt hordoz az azt generáló jelenségről. Azok a fizikai folyamatok, amelyek a tanító pontokat (pl. beszédet, képet, radar jelet, stb.) generálják, jól felállított direkt problémák. Tehát folytonos leképezést valósítanak meg. Azonban ezen folyamatok eredményeiből vett minták alapján tanulni már rosszul felállított inverz problémákat jelentenek. Gondoljunk csak arra, hogy egy zajos felvételből vagy egy csúnya kézírásból milyen nehéz az eredeti közleményt helyreállítani. 1963-ban Tyihonov javasolta a *regularizációs eljárást* a rosszul felállított problémák kezelésére.

Legyenek

$$\mathbf{x}_i \in \mathbb{R}^{m_0}, \quad i = 1, 2, \dots, N,$$

az input vektorok,

$$d_i \in \mathbb{R}, \quad i = 1, 2, \dots, N,$$

pedig a megfelelő output értékek. Keressük azt az $F: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ függvényt, mely az \mathbf{x}_i pontokban jól illeszkedik a d_i értékekre.

A Tyihonov-féle regularizáció elmélete szerint az illeszkedés jóságát az alábbi hibatagokkal mérjük.

(i) *Standard hiba*:

$$\mathcal{E}_s(F) = \frac{1}{2} \sum_{i=1}^N (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^N [d_i - F(\mathbf{x}_i)]^2.$$

(ii) *Regularizációs tag*:

$$\mathcal{E}_c(F) = \frac{1}{2} \|DF\|_{\mathcal{H}}^2.$$

Itt D egy lineáris differenciál-operátor. Nevezetesen D a négyzetesen integrálható $F: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ függvények egy \mathcal{H} Hilbert-terén értelmezett, azaz $D: \mathcal{H} \rightarrow \mathcal{H}$ operátor. Továbbá $\|DF\|_{\mathcal{H}}^2$ ezen a \mathcal{H} téren a DF függvény norma-négyzete. Az F függvény a $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ függvénytérnek (számunkra ismeretlen) eleme. D tartalmazza a problémáról az előzetes információt. $\mathcal{E}_c(F)$ -t szokták *büntető függvénynek* (penalty function) nevezni. Ez stabilizálja a megoldást, ill. bünteti a túl bonyolult modellt.

Tehát minimalizálni kell az alábbi kifejezést:

$$\mathcal{E}(F) = \mathcal{E}_s(F) + \lambda \mathcal{E}_c(F) = \frac{1}{2} \sum_{i=1}^N [d_i - F(\mathbf{x}_i)]^2 + \frac{1}{2} \lambda \|DF\|_{\mathcal{H}}^2, \quad (4.7)$$

ahol $\lambda > 0$ a *regularizációs paraméter*. Ha λ -t kicsinek választjuk, akkor főleg a tanító pontok határozzák meg az F_λ megoldást. Ha viszont λ -t nagynak választjuk, akkor a modellről szóló előzetes információk lesznek döntőek. $\mathcal{E}(F)$ -et pedig *Tyihonov-funkciónak* nevezzük. Jelöljük $\mathcal{E}(F)$ minimumhelyét F_λ -val.

Az F_λ helyen az $\mathcal{E}(F)$ -nek akkor van minimuma, ha

$$\mathcal{E}(F_\lambda) \leq \mathcal{E}(F_\lambda + \beta h)$$

tetszőleges h függvény és β skalár esetén. Azaz bármely $h \in \mathcal{H}$ irány menti deriválnak nullának kell lennie az F_λ pontban. Tehát

$$\left[\frac{d}{d\beta} \mathcal{E}(F_\lambda + \beta h) \right]_{\beta=0} = 0,$$

ahol h tetszőleges, rögzített, nem nulla függvény.

Ebből (4.7) alapján:

$$\begin{aligned} 0 &= \left[\frac{d}{d\beta} \mathcal{E}(F_\lambda + \beta h) \right]_{\beta=0} = \\ &= \left[\frac{d}{d\beta} \mathcal{E}_s(F_\lambda + \beta h) \right]_{\beta=0} + \lambda \left[\frac{d}{d\beta} \mathcal{E}_c(F_\lambda + \beta h) \right]_{\beta=0}. \end{aligned} \quad (4.8)$$

Határozzuk meg tagonként a deriváltakat.

$$\begin{aligned} \left[\frac{d}{d\beta} \mathcal{E}_s(F + \beta h) \right]_{\beta=0} &= \frac{1}{2} \left[\frac{d}{d\beta} \sum_{i=1}^N [d_i - F(\mathbf{x}_i) - \beta h(\mathbf{x}_i)]^2 \right]_{\beta=0} = \\ &= - \sum_{i=1}^N [d_i - F(\mathbf{x}_i) - \beta h(\mathbf{x}_i)] h(\mathbf{x}_i) \Big|_{\beta=0} = - \sum_{i=1}^N [d_i - F(\mathbf{x}_i)] h(\mathbf{x}_i) = \\ &= - \int_{\mathbb{R}^{m_0}} \sum_{i=1}^N (d_i - F(\mathbf{x})) h(\mathbf{x}) \delta_{\mathbf{x}_i}(\mathbf{x}) d\mathbf{x} = - \left\langle h, \sum_{i=1}^N (d_i - F) \delta_{\mathbf{x}_i} \right\rangle_{\mathcal{H}}, \end{aligned} \quad (4.9)$$

ahol $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ a \mathcal{H} Hilbert-térbeli belső szorzat,

$$\delta_{\mathbf{x}_i}(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i)$$

és δ a *Dirac-delta*. A regularizációs tag deriváltja:

$$\begin{aligned} \left[\frac{d}{d\beta} \mathcal{E}_c(F + \beta h) \right]_{\beta=0} &= \frac{1}{2} \left[\frac{d}{d\beta} \int_{\mathbb{R}^{m_0}} (D[F(\mathbf{x}) + \beta h(\mathbf{x})])^2 d\mathbf{x} \right]_{\beta=0} = \\ &= \left[\int_{\mathbb{R}^{m_0}} D[F(\mathbf{x}) + \beta h(\mathbf{x})] Dh(\mathbf{x}) d\mathbf{x} \right]_{\beta=0} = \\ &= \int_{\mathbb{R}^{m_0}} DF(\mathbf{x}) Dh(\mathbf{x}) d\mathbf{x} = \langle Dh, DF \rangle_{\mathcal{H}}. \end{aligned} \quad (4.10)$$

A második lépésben felcseréltük a deriválást és az integrálást. Jelölje \tilde{D} a D operátor adjungáltját. Így a (4.10) egyenlőségből

$$\left[\frac{d}{d\beta} \mathcal{E}_c(F + \beta h) \right]_{\beta=0} = \langle Dh, DF \rangle_{\mathcal{H}} = \langle h, \tilde{D}DF \rangle_{\mathcal{H}}. \quad (4.11)$$

A (4.9) és a (4.11) eredményeket visszahelyettesítve a (4.8) egyenletbe:

$$0 = \lambda \langle h, \tilde{D}DF \rangle_{\mathcal{H}} - \left\langle h, \sum_{i=1}^N (d_i - F) \delta_{\mathbf{x}_i} \right\rangle_{\mathcal{H}},$$

azaz

$$\left\langle h, \left[\tilde{D}DF - \frac{1}{\lambda} \sum_{i=1}^N (d_i - F) \delta_{\mathbf{x}_i} \right] \right\rangle_{\mathcal{H}} = 0.$$

Mivel ennek minden h -ra teljesülnie kell, így ez akkor lesz nulla, ha a belső szorzat második tagja nulla, azaz

$$\tilde{D}DF - \frac{1}{\lambda} \sum_{i=1}^N (d_i - F) \delta_{\mathbf{x}_i} = 0,$$

tehát

$$\tilde{D}DF(\mathbf{x}) = \frac{1}{\lambda} \sum_{i=1}^N (d_i - F(\mathbf{x}_i)) \delta(\mathbf{x} - \mathbf{x}_i). \quad (4.12)$$

Tehát ezen egyenlőség teljesülése esetén lesz a derivált nulla. A (4.12) egyenletet nevezzük a Tyihonov-funkcionálra vonatkozó *Euler–Lagrange-egyenlet-nek*. Ez az egyenlet tulajdonképpen egy parciális differenciálegyenlet F -re, lévén $\tilde{D}D$ egy differenciáloperátor.

A differenciálegyenletek elméletéből szükségesek a következők. Legyen adott egy L lineáris differenciáloperátor, és tekintsük az

$$LF(\mathbf{x}) = \varphi(\mathbf{x}) \quad (4.13)$$

differenciálegyenletet. Legyen $G(\mathbf{x}, \boldsymbol{\xi})$ olyan függvény, amelyben \mathbf{x} paraméter, $\boldsymbol{\xi}$ pedig az argumentum (\mathbf{x} és $\boldsymbol{\xi}$ is vektor), és amely eleget tesz az alábbi feltételeknek.

- (i) Rögzített $\boldsymbol{\xi}$ esetén, \mathbf{x} függvényeként G kielégíti az előírt határfeltételeket.
- (ii) Az $\mathbf{x} = \boldsymbol{\xi}$ pont kivételével a $G(\mathbf{x}, \boldsymbol{\xi})$ függvény \mathbf{x} szerinti deriváltjai folytonosak (a deriváltak rendje az L differenciáloperátor által meghatározott).
- (iii) $G(\mathbf{x}, \boldsymbol{\xi})$ az \mathbf{x} függvényeként kielégíti az

$$LG(\mathbf{x}, \boldsymbol{\xi}) = 0$$

parciális differenciálegyenletet az $\mathbf{x} = \boldsymbol{\xi}$ pont kivételével, ahol szingularitása van. Azaz kielégíti az

$$LG(\mathbf{x}, \boldsymbol{\xi}) = \delta(\mathbf{x} - \boldsymbol{\xi})$$

egyenletet, ahol $\delta(\mathbf{x} - \boldsymbol{\xi})$ a $\boldsymbol{\xi}$ középpontú Dirac-féle delta-függvény.

Ezt a $G(x, \xi)$ függvényt nevezzük az L differenciáloperátor *Green-függvényének*.

Ez alapján a (4.13) egyenlet megoldása

$$F(\mathbf{x}) = \int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi}) \varphi(\boldsymbol{\xi}) \, d\boldsymbol{\xi}. \quad (4.14)$$

Valóban, L -et és az integrálást felcserélve

$$LF(\mathbf{x}) = L \int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi}) \varphi(\boldsymbol{\xi}) \, d\boldsymbol{\xi} = \int_{\mathbb{R}^{m_0}} LG(\mathbf{x}, \boldsymbol{\xi}) \varphi(\boldsymbol{\xi}) \, d\boldsymbol{\xi}.$$

A Green-függvény harmadik tulajdonsága alapján

$$LF(\mathbf{x}) = \int_{\mathbb{R}^{m_0}} \delta(\mathbf{x} - \boldsymbol{\xi}) \varphi(\boldsymbol{\xi}) \, d\boldsymbol{\xi}$$

adódik. Végül a Dirac-delta tulajdonsága alapján kapjuk, hogy

$$LF(\mathbf{x}) = \varphi(\mathbf{x}).$$

Tehát igazoltuk a (4.14) egyenlőséget.

Most térjünk vissza a (4.12) Euler–Lagrange-egyenlethez. Az

$$L = \tilde{D}D, \quad \varphi(\mathbf{x}) = \frac{1}{\lambda} \sum_{i=1}^N (d_i - F(\mathbf{x}_i)) \delta(\mathbf{x} - \mathbf{x}_i)$$

választással a (4.14) egyenlőségből:

$$\begin{aligned} F(\mathbf{x}) &= \int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi}) \left\{ \frac{1}{\lambda} \sum_{i=1}^N (d_i - F(\mathbf{x}_i)) \delta(\boldsymbol{\xi} - \mathbf{x}_i) \right\} d\boldsymbol{\xi} = \\ &= \frac{1}{\lambda} \sum_{i=1}^N (d_i - F(\mathbf{x}_i)) \int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi}) \delta(\boldsymbol{\xi} - \mathbf{x}_i) d\boldsymbol{\xi}. \end{aligned}$$

Itt G az $L = \tilde{D}D$ -hez tartozó Green-függvény.

Végül az Euler–Lagrange-egyenlet megoldását a Dirac-delta függvény tulajdonságát felhasználva kapjuk:

$$\boxed{F_\lambda(\mathbf{x}) = \frac{1}{\lambda} \sum_{i=1}^N [d_i - F_\lambda(\mathbf{x}_i)] G(\mathbf{x}, \mathbf{x}_i)}. \quad (4.15)$$

Tehát ha adottak az (\mathbf{x}_i, d_i) , $i = 1, \dots, N$, tanító pontok, a G Green-függvény, a λ regularizációs paraméter, akkor az \mathcal{E} Tyihonov-funkcionál minimumát adó F_λ függvény kielégíti a (4.15) egyenletet.

Most ennek az egyenletnek a megoldását keressük. Legyen

$$w_i = \frac{1}{\lambda} [d_i - F_\lambda(\mathbf{x}_i)], \quad i = 1, 2, \dots, N, \quad (4.16)$$

és

$$\mathbf{w} = (w_1, w_2, \dots, w_N)^\top.$$

Tehát minden w_i súlyhoz tartozik egy $(F_\lambda(\mathbf{x}_i), d_i)$ pár. A w_i súlyok meghatározása az alábbi. A (4.15)-be \mathbf{x}_j -t helyettesítve, és figyelembe véve a (4.16) egyenlőséget:

$$F_\lambda(\mathbf{x}_j) = \sum_{i=1}^N w_i G(\mathbf{x}_j, \mathbf{x}_i), \quad j = 1, 2, \dots, N. \quad (4.17)$$

Vezessük be az alábbi N -dimenziós vektorokat

$$\begin{aligned} \mathbf{F}_\lambda &= (F_\lambda(\mathbf{x}_1), F_\lambda(\mathbf{x}_2), \dots, F_\lambda(\mathbf{x}_N))^\top, \\ \mathbf{d} &= (d_1, d_2, \dots, d_N)^\top, \end{aligned}$$

valamint az alábbi $N \times N$ -es mátrixot

$$G = \begin{pmatrix} G(\mathbf{x}_1, \mathbf{x}_1) & \dots & G(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ G(\mathbf{x}_N, \mathbf{x}_1) & \dots & G(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}.$$

G az úgynevezett *Green-mátrix*. Tehát a fenti (4.16) és (4.17) egyenletek vektoros-mátrixos alakja

$$\mathbf{w} = \frac{1}{\lambda} (\mathbf{d} - \mathbf{F}_\lambda), \quad \mathbf{F}_\lambda = G\mathbf{w}.$$

Ezekből

$$(G + \lambda I)\mathbf{w} = \mathbf{d}, \quad (4.18)$$

ahol I az $N \times N$ -es egységmátrix.

A Green-függvények szimmetrikusak. (Ezt az alábbi módon igazolhatjuk. Az $L = \tilde{D}D$ operátor önadjungált. Így a hozzá tartozó G Green-függvény

szimmetrikus. Valóban, a Green-függvény tulajdonságai miatt $G(\boldsymbol{\xi}, \boldsymbol{y}) = \int \delta(\boldsymbol{x} - \boldsymbol{\xi})G(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} = \int LG(\boldsymbol{x}, \boldsymbol{\xi})G(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} = \int G(\boldsymbol{x}, \boldsymbol{\xi})LG(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} = \int G(\boldsymbol{x}, \boldsymbol{\xi})\delta(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{x} = G(\boldsymbol{y}, \boldsymbol{\xi})$.) Tehát a G mátrix is szimmetrikus lesz:

$$G(\boldsymbol{x}_i, \boldsymbol{x}_j) = G(\boldsymbol{x}_j, \boldsymbol{x}_i) \quad \forall i, j,$$

azaz $G = G^\top$.

A 4.2. szakasz alapján láthatjuk, hogy a G Green-mátrix hasonló funkciót tölt be, mint a Φ interpolációs mátrix ($\lambda = 0$ esetén pontosan olyat). Az inverz multikvadratikus és a Gauss-féle radiális bázis függvények a Micchellitétel hatáskörébe eső Green-függvények, de a multikvadratikus nem ilyen (lásd [18]).

Elegendően nagy λ esetén $G + \lambda I$ pozitív definit, tehát invertálható. Ekkor a fenti (4.18) egyenletrendszer megoldása:

$$\boldsymbol{w} = (G + \lambda I)^{-1} \boldsymbol{d}. \quad (4.19)$$

Ezen \boldsymbol{w} -t beírva a (4.15) egyenletbe, kapjuk a regularizációs probléma megoldását:

$$F_\lambda(\boldsymbol{x}) = \sum_{i=1}^N w_i G(\boldsymbol{x}, \boldsymbol{x}_i). \quad (4.20)$$

Ez a megoldás valójában csak akkor alkalmazható, ha nemcsak az \boldsymbol{x}_i bemeneti és d_i kimeneti adatok, hanem a G Green-függvény és a λ paraméter is adott. Valójában magát a Green-függvényt nekünk kell megválasztani. Hiszen nem áll előzetesen rendelkezésre a probléma leírásához a D differenciáloperátor. Így bizonyos probléma típusokhoz a tapasztalatok alapján kialakult Green-függvény típusok közül kell választanunk. Erre utaltunk korábban, amikor a matematikai elmélettől csak a megoldás sémáját reméltük, a konkrét megoldást azonban számítógépes kísérletektől vártuk.

A G *eltolásinvariáns* (transzláció invariáns) volta azt jelenti, hogy G csak az adott pont és a középpont különbségétől függ:

$$G(\boldsymbol{x}, \boldsymbol{x}_i) = G(\boldsymbol{x} - \boldsymbol{x}_i).$$

Ha G *eltolásinvariáns* és *forgatásinvariáns* (rotáció invariáns) is, akkor G csak \boldsymbol{x} és \boldsymbol{x}_i távolságától függ:

$$G(\boldsymbol{x}, \boldsymbol{x}_i) = G(\|\boldsymbol{x} - \boldsymbol{x}_i\|).$$

Ezen feltétel teljesülése esetén nevezzük G -t *radiális bázis függvénynek* (radial basis function). Ekkor a regularizációs probléma megoldása az alábbi alakú:

$$F_\lambda(\mathbf{x}) = \sum_{i=1}^N w_i G(\|\mathbf{x} - \mathbf{x}_i\|). \quad (4.21)$$

A (4.21) és a (4.5) képletek közötti analógia világos. A különbség az, hogy a (4.21) képlet a büntető tag által regularizálva van.

4.9. Példa. A statisztikából ismert többdimenziós normális eloszlás sűrűségfüggvénye rendelkezik a magfüggvények definiáló tulajdonságaival. Amikor pedig a szórásmatrix az egységmatrix skalárszorosa, akkor transláció és rotáció invariáns Green-függvényhez jutunk:

$$G(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma_i^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right),$$

ahol $\mathbf{x}, \mathbf{x}_i \in \mathbb{R}^{m_0}$, $\sigma_i > 0$. Ez valójában az $\mathcal{N}_{m_0}(\mathbf{x}_i, \sigma_i^2 I)$ eloszlás sűrűségfüggvényének a skalárszorosa. Ez az

$$L = \sum_{n=0}^{\infty} (-1)^n \alpha_n \nabla^{2n}$$

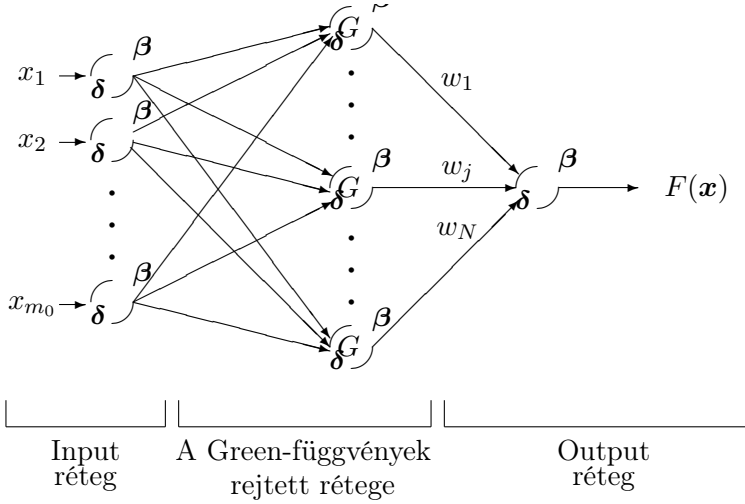
differenciál operátorhoz tartozó Green-függvény, ahol $\alpha_n = \sigma_i^{2n} / (n! 2^n)$, ∇^{2n} pedig a

$$\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_{m_0}^2}$$

Laplace-operátor n -szeres iteráltja (lásd [18]).

4.3.1. A regularizációs hálózat

A (4.20) képlet alapján megadhatjuk a regularizációs probléma megoldásának neurális hálózatát (4.3. ábra). Itt három réteg van. A rejtett réteg nem lineáris. A rejtett rétegbeli neuronok száma pontosan annyi, amennyi tanító pontunk van, azaz N . Ezek a csomópontok tartalmazzák az \mathbf{x}_i középpontú Green-függvényeket. A kimeneti réteg ábránkon egyetlen lineáris neuron. Általában tetszőleges véges számú neuron lehet benne a konkrét problémától függően (az osztályozás esetén az osztályok száma, függvényközelítésnél pedig a képtér dimenziója).



4.3. ábra. A regularizációs hálózat sémája

4.4. Az általánosított RBF hálózat

A regularizációs hálózatnál pontosan annyi neuron van a rejtett rétegben, ahány tanító pont. N számú tanító pont esetén viszont a Green-mátrix $N \times N$ -es mátrix. Így az inverzének meghatározása N^3 nagyságrendű műveletet igényel.

A számolások egyszerűsítése érdekében redukáljuk a hálózatot. Legyen az új dimenzió m_1 . Ezen neurális hálózat sémáját mutatja a 4.4. ábra. Ez hasonló a regularizációs hálózathoz, azonban a rejtett réteg neuronjainak száma csak m_1 (illetve az ábrán $m_1 + 1$, mivel a nulladik a többrétegű perceptronnál már megismert torzítás, amit $\varphi \equiv 1$ bázis függvény választással kezelhetünk).

Az

$$F^*(\mathbf{x}) = \sum_{i=1}^{m_1} w_i \varphi_i(\mathbf{x}) \quad (4.22)$$

alakú függvények között keressük az optimálisat. Itt $m_1 \leq N$, továbbá $\{\varphi_i(\mathbf{x}) : i = 1, 2, \dots, m_1\}$ a bázis függvények új halmaza. φ_i függvényeknek válasszunk radiális bázis függvényeket:

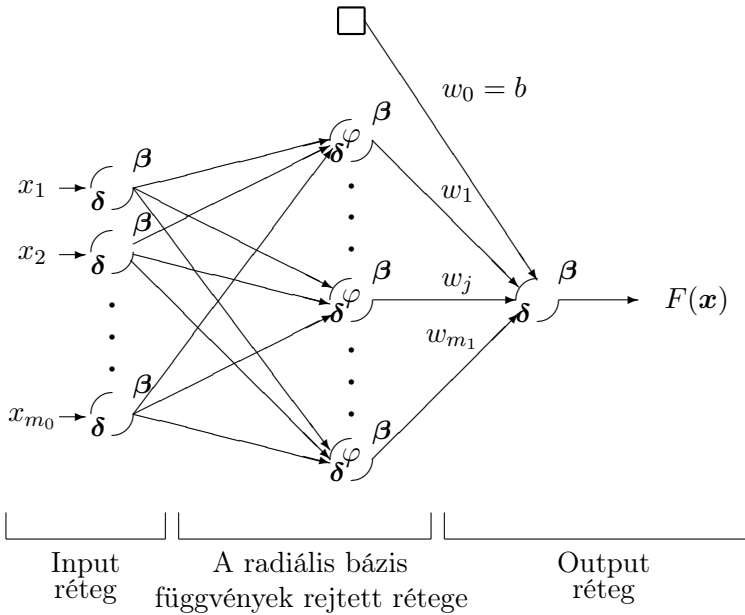
$$\varphi_i(\mathbf{x}) = G(\mathbf{x}, \mathbf{t}_i) = G(\|\mathbf{x} - \mathbf{t}_i\|), \quad i = 1, 2, \dots, m_1,$$

ahol $\{t_i : i = 1, 2, \dots, m_1\}$ az új középpontok halmaza. Ezen esetben a (4.22) képlet:

$$F^*(\mathbf{x}) = \sum_{i=1}^{m_1} w_i G(\|\mathbf{x} - \mathbf{t}_i\|). \quad (4.23)$$

Az új $\{w_i : i = 1, 2, \dots, m_1\}$ súlyokat úgy kell meghatározni, hogy minimalizáljuk az $\mathcal{E}(F^*)$ Tyihonov-funkcionált:

$$\mathcal{E}(F^*) = \frac{1}{2} \sum_{j=1}^N \left(d_j - \sum_{i=1}^{m_1} w_i G(\|\mathbf{x}_j - \mathbf{t}_i\|) \right)^2 + \frac{\lambda}{2} \|DF^*\|_{\mathcal{H}}^2. \quad (4.24)$$



4.4. ábra. Az RBF hálózat sémája

A jobb oldal első tagjának kétszerese

$$\|\mathbf{d} - G\mathbf{w}\|^2,$$

ahol

$$\mathbf{d} = (d_1, d_2, \dots, d_N)^\top, \quad \mathbf{w} = (w_1, w_2, \dots, w_{m_1})^\top$$

és

$$G = \begin{pmatrix} G(\mathbf{x}_1, \mathbf{t}_1) & \dots & G(\mathbf{x}_1, \mathbf{t}_{m_1}) \\ \vdots & \ddots & \vdots \\ G(\mathbf{x}_N, \mathbf{t}_1) & \dots & G(\mathbf{x}_N, \mathbf{t}_{m_1}) \end{pmatrix}.$$

A második tag (a $\lambda/2$ elhagyásával)

$$\begin{aligned}\|DF^*\|_{\mathcal{H}}^2 &= \langle DF^*, DF^* \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{m_1} w_i G(\mathbf{x}, \mathbf{t}_i), \tilde{D}D \sum_{i=1}^{m_1} w_i G(\mathbf{x}, \mathbf{t}_i) \right\rangle_{\mathcal{H}} = \\ &= \sum_{i=1}^{m_1} \sum_{j=1}^{m_1} w_i w_j \left\langle G(\mathbf{x}, \mathbf{t}_i), \tilde{D}DG(\mathbf{x}, \mathbf{t}_j) \right\rangle_{\mathcal{H}}.\end{aligned}$$

Mivel G éppen a $\tilde{D}D$ operátor Green-függvénye, ezért $\tilde{D}DG(\mathbf{x}, \boldsymbol{\xi}) = \delta(\mathbf{x} - \boldsymbol{\xi})$. Így

$$\left\langle G(\mathbf{x}, \mathbf{t}_i), \tilde{D}DG(\mathbf{x}, \mathbf{t}_j) \right\rangle_{\mathcal{H}} = \int G(\mathbf{x}, \mathbf{t}_i) \delta(\mathbf{x} - \mathbf{t}_j) d\mathbf{x} = G(\mathbf{t}_j, \mathbf{t}_i).$$

A fentiek alapján

$$\|DF^*\|_{\mathcal{H}}^2 = \sum_{i=1}^{m_1} \sum_{j=1}^{m_1} w_i w_j G(\mathbf{t}_i, \mathbf{t}_j) = \mathbf{w}^\top G_0 \mathbf{w},$$

ahol

$$G_0 = \begin{pmatrix} G(\mathbf{t}_1, \mathbf{t}_1) & \dots & G(\mathbf{t}_1, \mathbf{t}_{m_1}) \\ \vdots & \ddots & \vdots \\ G(\mathbf{t}_{m_1}, \mathbf{t}_1) & \dots & G(\mathbf{t}_{m_1}, \mathbf{t}_{m_1}) \end{pmatrix}.$$

Tehát a (4.24) Tyihonov-funkcionál az alábbi:

$$\begin{aligned}\mathcal{E}(F^*) &= \frac{1}{2} \|\mathbf{d} - G\mathbf{w}\|^2 + \frac{\lambda}{2} \mathbf{w}^\top G_0 \mathbf{w} = \\ &= \frac{1}{2} \mathbf{w}^\top G^\top G \mathbf{w} - \mathbf{d}^\top G \mathbf{w} + \frac{1}{2} \mathbf{d}^\top \mathbf{d} + \frac{\lambda}{2} \mathbf{w}^\top G_0 \mathbf{w} = \\ &= \frac{1}{2} \mathbf{w}^\top \underbrace{(G^\top G + \lambda G_0)}_A \mathbf{w} - \mathbf{w}^\top \underbrace{G^\top \mathbf{d}}_b + \frac{1}{2} \underbrace{\mathbf{d}^\top \mathbf{d}}_c.\end{aligned}$$

Ismeretes, hogy $\mathbf{w}^\top A \mathbf{w} - 2\mathbf{w}^\top \mathbf{b} + c$ minimumhelye kielégíti az $A\mathbf{w} = \mathbf{b}$ egyenletet. (Ennek pontosabb elemzését lásd a 6. fejezetben.) Tehát \mathbf{w} az alábbi egyenlet megoldása

$$(G^\top G + \lambda G_0) \mathbf{w} = G^\top \mathbf{d}. \quad (4.25)$$

Amennyiben λ -val tartunk nullához, akkor (4.25) \mathbf{w} megoldása a $\|\mathbf{d} - G\mathbf{w}\|^2$ minimumhelyéhez, azaz $(G^\top G)^- G^\top \mathbf{d}$ -hez konvergál. Itt M^- az M mátrix általánosított inverze, lásd Appendix.

Az RBF univerzális approximátor

A radiális bázis függvények approximáló tulajdonságairól számos eredmény ismeretes (lásd pl. Buhmann [4]). Mi itt Park és Sandberg [36] cikkéből idézzük a következő eredményt. $f: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ függvényeket kívánunk közelíteni

$$F(\mathbf{x}) = \sum_{i=1}^{m_1} w_i G\left(\frac{\mathbf{x} - \mathbf{t}_i}{\sigma}\right), \quad \mathbf{x} \in \mathbb{R}^{m_0} \quad (4.26)$$

alakú $F: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ függvénnyel, ahol $\sigma > 0$, $\mathbf{t}_i \in \mathbb{R}^{m_0}$, $G: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ pedig előre rögzített magfüggvény. Az összes lehetséges (4.26) alakú függvény halmazát jelölje S_G .

4.10. Tétel. *Legyen $G: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ előre rögzített, integrálható, korlátos, majdnem mindenütt folytonos, és $\int_{\mathbb{R}^{m_0}} G(\mathbf{x}) d\mathbf{x} \neq 0$. Ekkor az S_G függvény-család mindenütt sűrű $L^p(\mathbb{R}^{m_0})$ -ban tetszőleges $p \geq 1$ esetén.*

Ez azt jelenti, hogy tetszőleges (de bizonyos integrálhatósági feltételt teljesítő) $f: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ függvényt meg tudunk közelíteni (4.26)-ben adott alakú F függvénnyel bármilyen pontossággal.

4.4.1. A regularizációs paraméter meghatározása

A λ regularizációs paraméter megválasztásától jelentősen függ az optimális megoldás. Nem-lineáris regresszió esetén adjuk meg λ becslését. Modellünk az alábbi. Az adott (mért) \mathbf{x}_i pontok és a szintén megfigyelhető y_i értékek közötti kapcsolat:

$$y_i = f(\mathbf{x}_i) + \zeta_i, \quad i = 1, 2, \dots, N, \quad (4.27)$$

ahol $f(\cdot)$ egy valós értékű függvény, és ζ_i úgynevezett diszkrét idejű fehérzaj. Nevezetesen ζ_i kielégíti az alábbi feltételeket

$$\begin{aligned} \mathbb{E}[\zeta_i] &= 0, & \text{minden } i\text{-re,} \\ \mathbb{E}[\zeta_i \zeta_k] &= \begin{cases} \sigma^2, & \text{ha } k = i, \\ 0, & \text{egyébként.} \end{cases} \end{aligned}$$

Az ismeretlen $f(\cdot)$ függvényt kell meghatározni. Legyen $F_\lambda(\mathbf{x})$ a regularizációs hálózat általi becslése az $f(\mathbf{x})$ -nek. A (4.7) képlet alapján a regressziós problémánk Tyihonov-funkcionálja:

$$\mathcal{E}(F) = \frac{1}{2} \sum_{i=1}^N [y_i - F(\mathbf{x}_i)]^2 + \frac{\lambda}{2} \|DF\|_{\mathcal{H}}^2.$$

Ebből kiindulva két konkrét becslési eljárást adunk meg λ -ra.

Becslés az átlagos négyzetes hiba alapján

Jelölje $R(\lambda)$ az $f(\mathbf{x})$ regressziós függvény és az $F_\lambda(\mathbf{x})$ approximáló függvény közötti átlagos négyzetes eltérést,

$$R(\lambda) = \frac{1}{N} \sum_{i=1}^N [f(\mathbf{x}_i) - F_\lambda(\mathbf{x}_i)]^2. \quad (4.28)$$

$R(\lambda)$ -nak a minimumhelye szolgáltatja az optimális λ -t. Sajnos $R(\lambda)$ nem számolható ki az adott mennyiségekből, hiszen f nem ismert. Tegyük fel, hogy $F_\lambda(\mathbf{x}_k)$ az y_i -k lineáris függvénye (ami a (4.23) és a (4.25) alapján realizisztikus), azaz

$$F_\lambda(\mathbf{x}_k) = \sum_{i=1}^N a_{ki}(\lambda) y_i.$$

Ez az alábbi mátrixos alakba írható:

$$\mathbf{F}_\lambda = A(\lambda) \mathbf{y},$$

ahol

$$\begin{aligned} \mathbf{F}_\lambda &= (F_\lambda(\mathbf{x}_1), \dots, F_\lambda(\mathbf{x}_N))^\top, \\ \mathbf{y} &= (y_1, \dots, y_N)^\top, \\ A(\lambda) &= \begin{pmatrix} a_{11}(\lambda) & \dots & a_{1N}(\lambda) \\ \vdots & \ddots & \vdots \\ a_{N1}(\lambda) & \dots & a_{NN}(\lambda) \end{pmatrix}. \end{aligned}$$

Ezen jelölésekkel a (4.28) kifejezés:

$$R(\lambda) = \frac{1}{N} \|\mathbf{f} - \mathbf{F}_\lambda\|^2 = \frac{1}{N} \|\mathbf{f} - A(\lambda) \mathbf{y}\|^2, \quad (4.29)$$

ahol $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^\top$. A kiinduló regressziós feladatunk pedig vektoros alakban:

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\zeta},$$

ahol $\boldsymbol{\zeta} = [\zeta_1, \dots, \zeta_N]^\top$. Ezekkel a jelölésekkel a (4.29) képlet az alábbi:

$$\begin{aligned} R(\lambda) &= \frac{1}{N} \|\mathbf{f} - A(\lambda)(\mathbf{f} + \boldsymbol{\zeta})\|^2 = \frac{1}{N} \|(I - A(\lambda))\mathbf{f} - A(\lambda)\boldsymbol{\zeta}\|^2 = \\ &= \frac{1}{N} \|(I - A(\lambda))\mathbf{f}\|^2 - \frac{2}{N} \boldsymbol{\zeta}^\top A(\lambda)^\top (I - A(\lambda))\mathbf{f} + \frac{1}{N} \|A(\lambda)\boldsymbol{\zeta}\|^2. \end{aligned}$$

Határozzuk meg ennek a várható értékét. A jobb oldal első tagja nem véletlen, így várható értéke önmaga. Mivel a ζ vektor várható értéke nulla, így a második tag várható értéke nulla lesz. A harmadik tag N -szeresének várható értéke pedig az alábbi.

$$\begin{aligned}\mathbb{E}[\|A(\lambda)\zeta\|^2] &= \mathbb{E}[\zeta^\top A(\lambda)^\top A(\lambda)\zeta] = \\ &= \text{tr}\{\mathbb{E}[\zeta^\top A(\lambda)^\top A(\lambda)\zeta]\} = \mathbb{E}[\text{tr}\{\zeta^\top A(\lambda)^\top A(\lambda)\zeta\}].\end{aligned}$$

Ismert, hogy adott A és B (összeszorozható) mátrixok estén $\text{tr}\{AB\} = \text{tr}\{BA\}$. Ezért

$$\mathbb{E}[\text{tr}\{\zeta^\top A(\lambda)^\top A(\lambda)\zeta\}] = \mathbb{E}[\text{tr}\{A(\lambda)^\top A(\lambda)\zeta\zeta^\top\}] = \sigma^2 \text{tr}\{A(\lambda)^\top A(\lambda)\}.$$

Összeadva a fentieket

$$\mathbb{E}[R(\lambda)] = \frac{1}{N}\|(I - A(\lambda))\mathbf{f}\|^2 + \frac{\sigma^2}{N}\text{tr}\{A(\lambda)^\top A(\lambda)\}.$$

Mivel maga \mathbf{f} nem ismert, ezért sem $R(\lambda)$, sem $\mathbb{E}R(\lambda)$ nem számolható ki az adatokból.

Ezen probléma megoldására Craven és Wahba [8] az alábbi mennyiség használatát javasolta:

$$\hat{R}(\lambda) = \frac{1}{N}\|(I - A(\lambda))\mathbf{y}\|^2 + \frac{\sigma^2}{N}\text{tr}\{A(\lambda)^2\} - \frac{\sigma^2}{N}\text{tr}\{(I - A(\lambda))^2\}. \quad (4.30)$$

Egyrészt $\hat{R}(\lambda)$ már kiszámolható, másrészt „torzítatlan becslése” $R(\lambda)$ -nak abban az értelemben, hogy

$$\mathbb{E}[\hat{R}(\lambda)] = \mathbb{E}[R(\lambda)]. \quad (4.31)$$

Ezért $\hat{R}(\lambda)$ minimumhelyét fogadjuk el λ becsléseként.

A (4.31) egyenlőség bizonyításához elegendő $\hat{R}(\lambda)$ első tagját tekinteni:

$$\begin{aligned}\frac{1}{N}\|(I - A(\lambda))\mathbf{y}\|^2 &= \frac{1}{N}\|\mathbf{f} + \zeta - A(\lambda)\mathbf{y}\|^2 = \\ &= \frac{1}{N}\|\mathbf{f} - A(\lambda)\mathbf{y}\|^2 + \frac{2}{N}\zeta^\top(\mathbf{f} - A(\lambda)\mathbf{y}) + \frac{1}{N}\zeta^\top\zeta = \\ &= R(\lambda) + \frac{2}{N}\text{tr}\{\zeta^\top(\mathbf{f} - A(\lambda)\mathbf{f} - A(\lambda)\zeta)\} + \frac{1}{N}\text{tr}\{\zeta^\top\zeta\}.\end{aligned}$$

Ennek várható értéke pedig

$$\mathbb{E}R(\lambda) - \frac{2\sigma^2}{N}\text{tr}\{A(\lambda)\} + \frac{\sigma^2}{N}\text{tr}\{I\} =$$

$$= \mathbb{E}R(\lambda) - \frac{\sigma^2}{N} \text{tr}\{A(\lambda)^2\} + \frac{\sigma^2}{N} \text{tr}\{(I - A(\lambda))^2\}.$$

Azaz a (4.30) jobb oldalának második és harmadik tagja eliminálja az $\mathbb{E}R(\lambda)$ -tól való eltérést.

A cross-validation eljárás

Az előbbi becslés hátránya, hogy ismernünk kell a ζ_i sorozat σ^2 szórásnégyzetét. Ez a hátrány nem jelentkezik a *cross-validation* eljárásnál. A *cross-validation* (más szóval *leave one out*, magyarul „hagyj ki egy pontot”) eljárás a statisztikában gyakran alkalmazott. Hagyjunk tehát el egy elemet a mintából. Határozzuk meg a maradék mintaelemek alapján az F ismeretlen függvényt. Jelölje $F_\lambda^{[k]}$ az

$$\mathcal{E}(F) = \frac{1}{2} \sum_{i=1, i \neq k}^N [y_i - F(\mathbf{x}_i)]^2 + \frac{\lambda}{2} \|DF\|^2$$

Tyihonov-funkcionál minimumhelyét. Most pedig ellenőrizzük, hogy $F_\lambda^{[k]}(\mathbf{x})$ milyen jól képes a kihagyott k -adik értéket megjósolni. Tehát a λ jóságának mértéke az alábbi:

$$V_0(\lambda) = \frac{1}{N} \sum_{k=1}^N [y_k - F_\lambda^{[k]}(\mathbf{x}_k)]^2. \quad (4.32)$$

Ez a mennyiség az adatokból számolható. Az optimális λ -át pedig $V_0(\lambda)$ minimumhelye szolgáltatja.

4.4.2. Tanítási stratégiák

A regularizációs hálózat esetén a középpontok az \mathbf{x}_i tanító pontok. Ezzel szemben a gyakorlatban ténylegesen alkalmazott RBF hálózatban a középpontok nem adóttak előre. A radiális bázis függvények középpontjának megválasztása alapvető feladat. A 4.4. fejezet elején láttuk, hogy a RBF-ek számát (m_1 -et) kevesebbre érdemes választani, mint a tanító pontok száma (azaz mint N).

Véletlenszerűen választott, de azután rögzített középpontok

Az adott $\mathbf{x}_1, \dots, \mathbf{x}_N$ alappontok közül válasszunk véletlenszerűen m_1 darabot. Ezek lesznek a $\mathbf{t}_1, \dots, \mathbf{t}_{m_1}$ középpontok. Jelölje d_{\max} a \mathbf{t}_i pontok közötti

legnagyobb távolságot. Legyen

$$\sigma = \frac{d_{\max}}{\sqrt{2m_1}}.$$

A radiális bázis függvények pedig legyenek \mathbf{t}_i középpontú és $\sigma^2 I$ szórás-mátrixú Gauss-féle sűrűségfüggvények, azaz

$$G(\|\mathbf{x} - \mathbf{t}_i\|^2) = \exp\left(-\frac{m_1}{d_{\max}^2}\|\mathbf{x} - \mathbf{t}_i\|^2\right), \quad i = 1, 2, \dots, m_1.$$

(Ez ténylegesen az $\mathcal{N}_{m_0}(\mathbf{t}_i, \sigma^2 I)$ sűrűségfüggvényének a konstansszorosa.) Az így definiált Gauss-függvény biztosítja, hogy az RBF ne legyen sem túlságosan lapos, sem túlságosan csúcsos.

Most csak a $\lambda = 0$ esetet tekintjük. Ekkor a 4.4. szakaszban levezetett módon a súlyvektort a $\|\mathbf{d} - G\mathbf{w}\|^2$ minimumhelyeként kell meghatározni. Itt a G mátrix $N \times m_1$ méretű, (i, j) -edik eleme pedig

$$g_{ij} = \exp\left(-\frac{m_1}{d_{\max}^2}\|\mathbf{x}_i - \mathbf{t}_j\|^2\right), \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, m_1.$$

A $\|\mathbf{d} - G\mathbf{w}\|^2$ minimumhelye $\mathbf{w} = G^+ \mathbf{d}$ lesz. Itt G^+ a G mátrix pszeudoinverze. G^+ -ról lásd a 6.2. szakaszt.

A középpontok önszervező kiválasztása

Ez kétlépéses tanítás az alábbi fázisokkal.

- (i) A középpontok meghatározása valamely statisztikai önszerveződő eljárás alapján.
- (ii) Az output rétegben \mathbf{w} súlyvektor kiszámítása.

Most leírjuk a középpontok *k-közép módszerrel* történő kiválasztását. Ez a klaszterezés standard eljárása. A klaszterezés fő ismérve, hogy nem csak az egyedek osztályba sorolása a feladat, hanem a megfigyelések alapján az osztályokat is nekünk magunknak kell kialakítanunk.

Tehát a $\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{m_1}$ középpontokat akarjuk meghatározni. Az n -edik lépésben jelölje $\mathbf{t}_1(n), \mathbf{t}_2(n), \dots, \mathbf{t}_{m_1}(n)$ az aktuális középpontokat.

- (1) A kezdeti $\mathbf{t}_1(0), \mathbf{t}_2(0), \dots, \mathbf{t}_{m_1}(0)$ középpontokat válasszuk tetszőlegesen.

(2) Az n -edik lépésben az input mintapont $\mathbf{x}(n)$. Jelölje $k(n)$ az új mintapont *legközelebbi társának* indexét. Azaz legyen $k(n)$ azon $\mathbf{t}_i(n)$ indexe, amelyik $\mathbf{x}(n)$ -hez a legközelebb van.

(3) Legyen az új klaszter középpont

$$\mathbf{t}_{k(n)}(n+1) = \mathbf{t}_{k(n)}(n) + \eta[\mathbf{x}(n) - \mathbf{t}_{k(n)}(n)],$$

ahol $\eta \in (0, 1)$ előre adott tanulási paraméter.

A többi klaszter középpontja változatlan marad, azaz $\mathbf{t}_i(n+1) = \mathbf{t}_i(n)$, ha $i \neq k(n)$.

(4) Döntés a leállításról. Vagy elfogadjuk a kialakított középpontokat, vagy n értékét növeljük 1-gyel, és visszatérünk (2)-re. (Azaz ez utóbbi esetben újabb mintaponttal módosítjuk a középpontokat.)

A középpontok felügyelt kiválasztása

Tekintsük az

$$F(\mathbf{x}) = \sum_{i=1}^{m_1} w_i G(\|\mathbf{x} - \mathbf{t}_i\|_{C_i}^2)$$

alakú közelítéseket. Itt

$$\|\mathbf{x} - \mathbf{t}_i\|_{C_i}^2 = \|C_i(\mathbf{x} - \mathbf{t}_i)\|^2 = (\mathbf{x} - \mathbf{t}_i)^\top \Sigma_i^{-1}(\mathbf{x} - \mathbf{t}_i),$$

ahol $\Sigma_i^{-1} = C_i^\top C_i$. A normális eloszlás szórásmátrixának szokásos jelölése miatt használjuk a Σ_i^{-1} jelölést. Most ez maga a meghatározandó paraméter, nem pedig Σ_i .

Tehát keressük w_i , \mathbf{t}_i és Σ_i^{-1} optimális értékét. A tanító pontok: (\mathbf{x}_j, d_j) , $j = 1, \dots, N$. A minimalizálandó hiba:

$$\mathcal{E} = \frac{1}{2} \sum_{j=1}^N [d_j - F(\mathbf{x}_j)]^2.$$

A minimalizálást a gradiens módszerrel hajthatjuk végre. Most az n -edik lépésben a változtatásokat és a gradienseket határozzuk meg.

A w_i súlyok változtatása.

$$w_i(n+1) = w_i(n) - \eta_1 \frac{\partial \mathcal{E}(n)}{\partial w_i(n)},$$

ahol

$$\frac{\partial \mathcal{E}(n)}{\partial w_i(n)} = - \sum_{j=1}^N [d_j - F(\mathbf{x}_j)] G \left[(\mathbf{x}_j - \mathbf{t}_i(n))^\top \Sigma_i^{-1}(n) (\mathbf{x}_j - \mathbf{t}_i(n)) \right].$$

A \mathbf{t}_i középpontok változtatása.

$$\mathbf{t}_i(n+1) = \mathbf{t}_i(n) - \eta_2 \frac{\partial \mathcal{E}(n)}{\partial \mathbf{t}_i(n)},$$

ahol

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial \mathbf{t}_i(n)} &= 2w_i(n) \sum_{j=1}^N [d_j - F(\mathbf{x}_j)] G' \left((\mathbf{x}_j - \mathbf{t}_i(n))^\top \Sigma_i^{-1}(n) (\mathbf{x}_j - \mathbf{t}_i(n)) \right) \times \\ &\quad \times \Sigma_i^{-1}(n) (\mathbf{x}_j - \mathbf{t}_i(n)). \end{aligned}$$

A Σ_i^{-1} szórások változtatása.

$$\Sigma_i^{-1}(n+1) = \Sigma_i^{-1}(n) - \eta_3 \frac{\partial \mathcal{E}(n)}{\partial \Sigma_i^{-1}(n)},$$

ahol

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial \Sigma_i^{-1}(n)} &= -w_i(n) \sum_{j=1}^N [d_j - F(\mathbf{x}_j)] G' \left((\mathbf{x}_j - \mathbf{t}_i(n))^\top \Sigma_i^{-1}(n) (\mathbf{x}_j - \mathbf{t}_i(n)) \right) \times \\ &\quad \times (\mathbf{x}_j - \mathbf{t}_i(n)) (\mathbf{x}_j - \mathbf{t}_i(n))^\top. \end{aligned}$$

A fenti képletekben η_1, η_2, η_3 pozitív tanulási paraméterek.

4.5. Magfüggvényes becslések

A statisztikában gyakran alkalmazott nem-paraméteres függvény becslések közel állnak az RBF hálózattal adható függvény közelítésekhez. A nem-paraméteres jelző arra utal, hogy nem csak a függvény paraméterei ismeretlenek, hanem annak alakja is.

4.5.1. A sűrűségfüggvény magfüggvényes becslése

4.11. Definíció. Legyen $K: \mathbb{R}^{m_0} \rightarrow \mathbb{R}$ korlátos, folytonos, az origóra szimmetrikus függvény, melynek maximuma az origóban van. Tegyük fel, hogy $\int_{\mathbb{R}^{m_0}} K(\mathbf{x}) d\mathbf{x} = 1$. Ekkor K -t *magfüggvénynek* (kernel function) nevezzük.

Legyen $\mathbf{x}_1, \dots, \mathbf{x}_N$ minta egy ismeretlen f sűrűségfüggvényű sokaságból ($\mathbf{x}_i \in \mathbb{R}^{m_0}$). Az f magfüggvényes, más szóval *Parzen–Rosenblatt-féle becslése*:

$$\hat{f}_N(\mathbf{x}) = \frac{1}{Nh^{m_0}} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \quad \mathbf{x} \in \mathbb{R}^{m_0}. \quad (4.33)$$

Itt h -t *sávszélességnek* (bandwidth) nevezzük.

Tegyük fel, hogy $h = h(N) \rightarrow 0$, amint $N \rightarrow \infty$. Ekkor a sávszélességre és a magfüggvényre vonatkozó bizonyos feltételek esetén \hat{f}_N az f -nek aszimptotikusan torzítatlan és L^2 -konzisztens becslése (lásd Prakasa Rao [39]). Az aszimptotikus torzítatlanság azt jelenti, hogy $\lim_{N \rightarrow \infty} \mathbb{E}\hat{f}_N(\mathbf{x}) = f(\mathbf{x})$. Az L^2 -beli konzisztencia pedig $\lim_{N \rightarrow \infty} \mathbb{E}\left(\hat{f}_N(\mathbf{x}) - f(\mathbf{x})\right)^2 = 0$.

4.5.2. A regressziós függvény magfüggvényes becslése

Tegyük fel, hogy az Y egydimenziós véletlen mennyiséget akarjuk az X m_0 -dimenziós véletlen mennyiség valamely g függvényével közelíteni. Az optimális elméleti választás (amennyiben Y -nak van véges várható értéke) a

$$g(\mathbf{x}) = \mathbb{E}\{Y \mid X = \mathbf{x}\}$$

feltételes várható érték. Ennek elméleti kiszámítása (ha van együttes sűrűségfüggvény)

$$g(\mathbf{x}) = \int y f(y \mid \mathbf{x}) dy = \int y \frac{f(y, \mathbf{x})}{f(\mathbf{x})} dy,$$

ahol $f(y, \mathbf{x})$ az Y és X együttes sűrűségfüggvénye, $f(\mathbf{x})$ az X sűrűségfüggvénye, $f(y \mid \mathbf{x})$ pedig Y -nak X -re vonatkozó feltételes sűrűségfüggvénye.

Legyen most $(y_1, \mathbf{x}_1), \dots, (y_N, \mathbf{x}_N)$ minta (Y, X) -re. Az f sűrűségfüggvény magfüggvényes becslése (4.33) alapján történik, míg az együttes sűrűségfüggvény ennek a képletnek az alábbi, $(m_0 + 1)$ -dimenziós változatával:

$$\hat{f}(y, \mathbf{x}) = \frac{1}{Nh^{m_0+1}} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) K_0\left(\frac{y - y_i}{h}\right), \quad (4.34)$$

$y \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^{m_0}$. (Itt K_0 is magfüggvény.)

Építsük fel g becslését a sűrűségfüggvények becsléséből ugyanúgy, mint ahogyan g -t kapjuk az elméleti sűrűségfüggvényekből.

$$\hat{g}(\mathbf{x}) = \int y \frac{\hat{f}(y, \mathbf{x})}{\hat{f}(\mathbf{x})} dy =$$

$$= \frac{\frac{1}{Nh^{m_0+1}} \sum_{i=1}^N K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \int y K_0\left(\frac{y-y_i}{h}\right) dy}{\frac{1}{Nh^{m_0}} \sum_{i=1}^N K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} = \frac{\sum_{i=1}^N y_i K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^N K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}.$$

Az utolsó lépésben az

$$\int \frac{1}{h} y K_0\left(\frac{y-y_i}{h}\right) dy = \int (hz + y_i) K_0(z) dz = y_i$$

egyenlőséget használtuk ki. A végeredményként kapott

$$\hat{g}(\mathbf{x}) = \frac{\sum_{i=1}^N y_i K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\sum_{i=1}^N K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}$$

képlet a g függvény magfüggvényes, más szóval *Nadaraya–Watson-féle becslése*.

Bizonyos feltételek esetén $\hat{g}(\mathbf{x}) = \hat{g}_N(\mathbf{x})$ konzisztens becslés, azaz $\hat{g}_N(\mathbf{x})$ sztochasztikusan konvergál $g(\mathbf{x})$ -hez, ha $N \rightarrow \infty$ (lásd Prakasa Rao [39]).

Ha a K magfüggvényt transláció és rotáció invariánsnak választjuk, akkor egy normalizált RBF hálózatot kapunk:

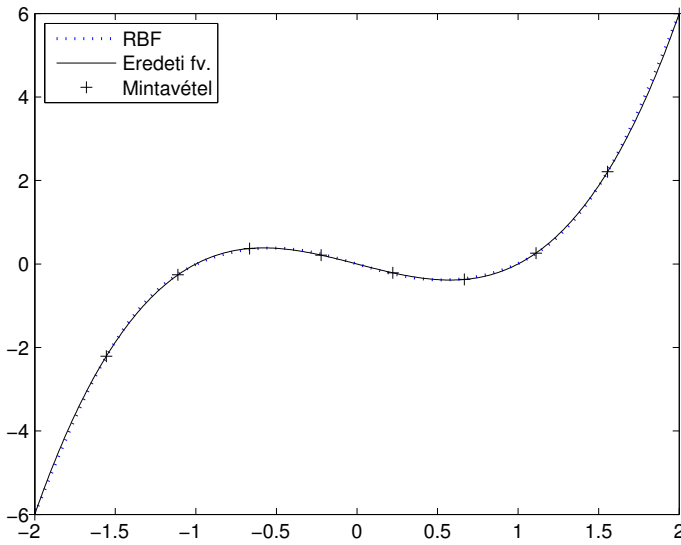
$$\hat{g}(\mathbf{x}) = \sum_{i=1}^N w_i \Psi(\mathbf{x}, \mathbf{x}_i),$$

ahol $w_i = y_i$ és $\Psi(\mathbf{x}, \mathbf{x}_i) = K\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|}{h}\right) / \sum_{i=1}^N K\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|}{h}\right)$.

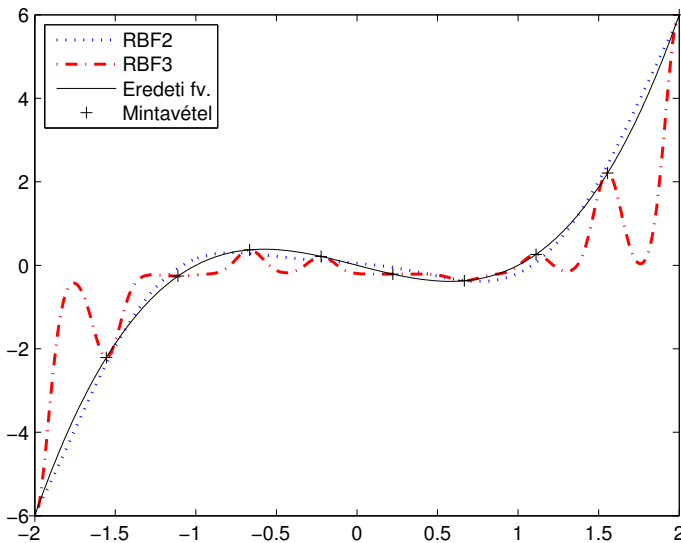
4.6. Numerikus példák

4.12. Példa. Közelítsük az $f(x) = x^3 - x$ függvényt a $[-2, 2]$ intervallumon, ha a megfigyelések a -2 és 2 közötti, egymástól egyenlő távolságban lévő (azaz ekvidisztáns) alappontokban a hiba nélkül mért függvényértékek. Konkrétan 10 tanító pontot választottunk. Az első megoldás: (maximum) 11 neuronnal és 1 sáv szélességgel a 4.5. ábrán látható. Itt a közelítő függvény gyakorlatilag tökéletesen illeszkedik a becslendőhöz. A 4.6. ábrán látható két „próbálkozás” azonban kevésbé sikeres: RBF2 esetén csak 5 neuront engedtünk meg, ez kevésnek bizonyult, az illeszkedés nem tökéletes. RBF3 esetén viszont a helyzet még rosszabb, itt (maximum) 50 neuront engedélyeztünk és a sáv szélességet levittük 0.1-re. Ezzel a közelítő függvény „ugrál”.

Valójában a fenti esetben a tanító pontok száma éppen elegendő a függvény közelítéséhez (hiszen maga a függvény nem „bonyolult”). Vizsgáljuk

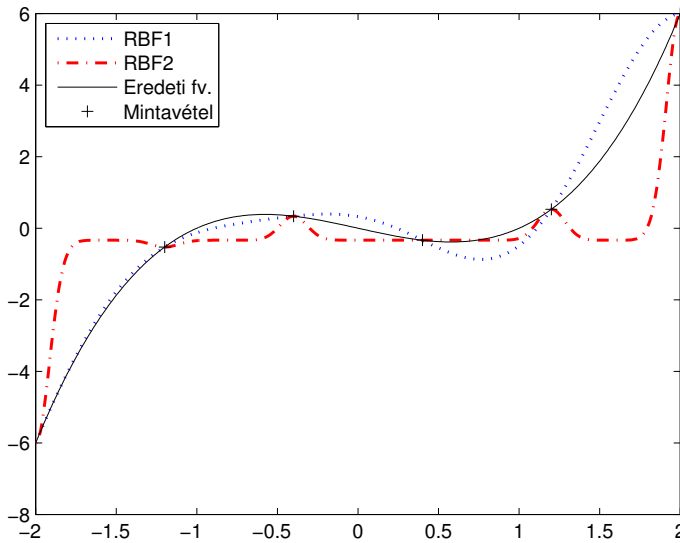


4.5. ábra. Az $f(x) = x^3 - x$ függvény jó közelítése RBF-fel



4.6. ábra. Az $f(x) = x^3 - x$ függvény közelítése. RBF2 túl kevés neuron, RBF3 túl sok neuron

meg, hogy mi történik kevesebb tanító pont esetén! Ha csak 6 tanító pontunk van, akkor már kevésbé jó közelítéseket kapunk. Az első megoldás: (maximum) 11 neuronnal és 1 sávszélességgel a 4.7. ábrán látható RBF1. Itt



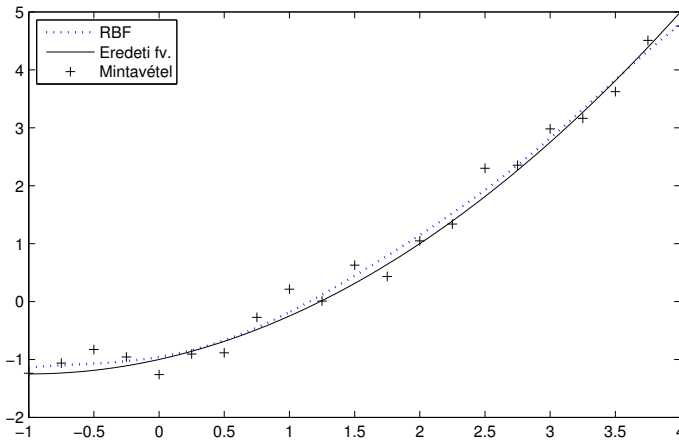
4.7. ábra. Az $f(x) = x^3 - x$ függvény közelítése kevés tanító pont esetén. RBF1 elegendő számú neuron, RBF2 túl sok neuron

a közelítő függvény elég jól, de nem tökéletesen illeszkedik az becsülendőhöz. RBF2 esetén viszont (maximum) 25 neuront engedélyeztünk és a sáv szélességet levittük 0.1-re. Ezzel a közelítő függvény ismét „ugrál”.

4.13. Példa. Közelítsük az

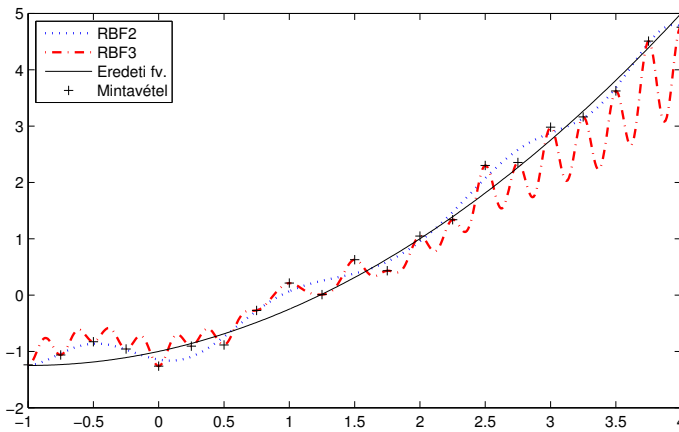
$$f(x) = \frac{1}{4}x^2 + \frac{1}{2}x - 1$$

függvényt a $[-1, 4]$ intervallumon, ha a megfigyelések a -1 és 4 közötti, egymástól egyenlő távolságban lévő (azaz ekvidisztáns) alappontokban a hibával terhelt mért függvényértékek. Konkrétan 21 tanító pontot választottunk. A mérési hiba pedig minden tanító pont esetén 0 várható értékű és 0.3 szórási normális eloszlású véletlen szám volt. Az első megoldás: (maximum) 5 neuronnal és 2 sáv szélességgel a 4.8. ábrán látható. Itt a közelítő függvény nagyon jól illeszkedik a becsülendőhöz. Ennek az az oka, hogy a közelítendő függvény alakja „egyszerű”, a nagy sáv szélesség és a kevés neuron pedig kismítja a mérési hibák hatását. A 4.9. ábrán látható két „próbálkozás” azonban kevésbé sikeres: RBF2 esetén 11 neuront engedtünk és 1 sáv szélességet. Itt az illeszkedés nem tökéletes. RBF3 esetén a helyzet még rosszabb, itt (maximum) 50 neuront engedélyeztünk és a sáv szélességet levittük 0.1-re. Ezzel a



4.8. ábra. Az $f(x) = \frac{1}{4}x^2 + \frac{1}{2}x - 1$ függvény jó közelítése RBF-fel

közelítő függvény a tanító pontokban jól illeszkedik a hibával terhelt értékekre, de a közelítendő függvényt rosszul approximálja, „ugrál”. Az alkalmazott modell nem megfelelő, túlillesztett.



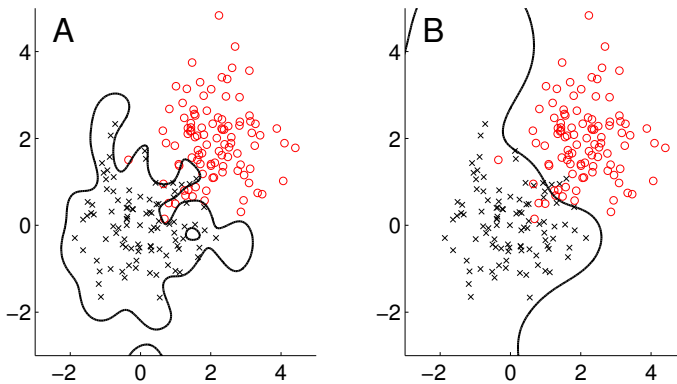
4.9. ábra. Az $f(x) = \frac{1}{4}x^2 + \frac{1}{2}x - 1$ függvény közelítése. RBF2 túl sok neuron, RBF3 túl sok neuron és túl kicsi sávszélesség

4.14. Példa. Két halmaz szétválasztása RBF segítségével megtehető. Legyen az egyik halmaz 100 elemű minta kétdimenziós, $(0, 0)^\top$ várható érték

vektorú és

$$\begin{pmatrix} 0.8 & 0 \\ 0 & 0.8 \end{pmatrix}$$

szórásmátrixú normális eloszlásból. A másik halmaz pedig ugyanilyen minta, azzal a különbséggel, hogy ott a várható érték vektor $(2, 2)^\top$. A 4.10. ábra mind A, mind B részén \times jelöli az első, \circ pedig a második halmaz elemeit. De a megkonstruált elválasztó görbék különböznek. Az A részen maximum 90 neuront engedélyeztünk, a sávszélességet 0.5-re állítottuk, az elérendő legkisebb négyzetes hibát pedig alacsonyra (0.0001-re) állítottuk. Az elválasztó görbe a tanító pontokat helyesen szeparálja, azonban rosszul általánosít. Hiszen tudjuk, hogy esetünkben a két normális eloszlást optimális módon egy egyenes osztaná ketté. Az ábra B részén lévő szeparálás azért sokkal jobb, mert ugyan az aktuális tanító pontokat nem tökéletesen osztja ketté, azonban jól általánosít. Ezt az eredményt maximum 30 neuron engedélyezésével és 2 sávszélességgel értük el, miközben az elérendő legkisebb négyzetes hibát is magasabbra (0.01-re) engedjük.



4.10. ábra. Két normális eloszlás szeparálása RBF-fel.
A rész: túl sok neuron, rosszul általánosít. B rész: megfelelő számú neuron, jól általánosít

4.7. Feladatok

1. Ábrázoljuk a 4.2 szakasz végén megadott RBF interpolációs hálózatot!
2. Igazoljuk, hogy a 4.7 és a 4.8 Példákban leírt esetek rosszul felállított problémák!
3. Igazoljuk, hogy $\text{tr } AB = \text{tr } BA$.

4. Ábrázoljuk a

$$\varphi(r) = \left(\frac{r}{\sigma}\right)^2 \log\left(\frac{r}{\sigma}\right)$$

függvényt, ahol σ rögzített pozitív paraméter, r pedig a pozitív számokon futó változó! Állapítsuk meg a φ függvény tulajdonságait! Ezt nevezzük thin-plate-spline magfüggvénynek. Az ezzel definiált $\sum_{i=1}^{m_1} w_i \varphi(\|\mathbf{x} - \mathbf{t}_i\|)$ RBF-eket kiterjedten alkalmazzák.

5. Oldjuk meg RBF-fel a 2. fejezet feladatai között definiált XOR-problémát!
6. Közelítsük az alábbi függvényeket RBF segítségével!

$$f(x) = x^2, \quad f(x) = \log x, \quad f(x) = \sin x, \quad f(x) = (\sin x)/x,$$

$$f(x, y) = x^2 + y^2, \quad f(x, y) = x^2/y.$$

Válasszunk alkalmas tartományt, ahol a fenti függvényeket közelíteni akarjuk. Generáljunk különböző számú és sűrűségű tanító pontokat. Vizsgáljuk külön azt az esetet is, amikor a függvényértéket csak valamilyen véletlen hibával tudjuk megfigyelni (azaz a tanító pontok ilyenek, de a közelítendő függvény a hiba nélküli). Alkalmazzunk különböző neuron számú RBF-eket! Ábrázoljuk az eredményt!

7. Vizsgáljuk az RBF-fel való szeparálást! Állítsuk valamilyen módon elő a tanító pontok A és B halmazát, és szeparáljuk RBF-fel! A két halmaz lehet 2- vagy 3-dimenziós, véletlen vagy determinisztikus módon megadott, akár egymásba átnyúló is. Alkalmazzunk különböző neuron számú RBF-eket! Ábrázoljuk az eredményt!
8. Az előző két feladatot oldjuk meg MLP-vel is! Hasonlítsuk össze a legjobb MLP-t a legjobb RBF-fel!
9. Vizsgáljuk az RBF-fel való szeparálást 3 halmaz esetén! Szeparáljunk el egymástól 3 síkbeli halmazt! Ábrázoljuk az eredményt!

5. fejezet

A tartó vektor gépek

5.1. A tartó vektor gépek kialakulása

A *Support Vector Machine* (SVM) tekinthető mint egy speciális neurális hálózat. Több szakember azonban szereti a statisztikus tanulási elmélet (statistical learning theory) keretébe sorolni. Valójában a statisztikus tanulási elméletet érdemes egy tágabb, elméleti tudománynak tekinteni, míg a neurális hálózatokat egy speciálisabb, gyakorlati megvalósításra fókuszáló eszköztárnak. Függetlenül a besorolástól, az SVM az utóbbi évtizedekben fontos gyakorlati eszközzé és egyben mély elméleti kutatások tárgyává vált.

A Support Vector Machine magyar megfelelője a *tartó vektor gép* (lehetne esetleg „tartó vektor mechanizmus” is). Közismert magyar rövidítés híján az SVM angol rövidítést használjuk. Az SVM elmélete és alkalmazásai, párhuzamosan a statisztikus tanulási elmélettel, az utóbbi 30-40 évben alakultak ki. Leginkább V. N. Vapnik és munkatársai szerepe volt ebben a döntő (lásd Vapnik–Chervonenkis [48]). A témakör ma is intenzíven fejlődik.

Korábban láttuk, hogy az MLP és az RBF segítségével elvégezhető a függvényközelítés és az osztályozás feladata. Az SVM is ezekre alkalmas, de az SVM-nél a cél az optimális megoldás megtalálása. Tehát itt a jól működő matematikai módszer kialakítása a fontos, nem pedig a heurisztika és a számítógépes kísérletezés. SVM-et alkalmaznak karakterfelismerésre, képfeldolgozásra, a bioinformatikában, az adatbányászatban és számos más területen is.

Ebben a fejezetben elsősorban Vapnik [49] közismert monográfiájára támaszkodva ismertetjük az SVM-et, de Cristianini és Shawe–Taylor [10] könyvét, valamint Haykin [18] átfogó művét is használjuk. Az SMO leírásánál Platt [37] cikkéből is merítünk.

5.2. SVM osztályozásra

5.2.1. Az optimális hipersík a lineárisan szeparálható esetben

Az optimális hipersík geometriai meghatározása

Ebben a szakaszban Vapnik [49] könyvét követjük. Tegyük fel, hogy az

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N), \quad \mathbf{x}_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\} \quad (5.1)$$

minta, más szóval tanító halmaz elemei két osztályból származnak. Ha az $\mathbf{x}_i \in \mathbb{R}^d$ mintaelem az A_1 osztályból származik, akkor $y_i = 1$, amennyiben az A_2 osztályból, akkor pedig $y_i = -1$ jelöléssel élünk. Azt mondjuk, hogy a fenti tanító halmaz lineárisan szeparálható, ha létezik olyan hipersík, amelynek egyik oldalán vannak A_1 , a másik oldalán pedig A_2 elemei. Pontosabban szólva lineárisan szeparálható az

$$\langle \mathbf{x}, \boldsymbol{\varphi} \rangle = c \quad (5.2)$$

hipersíkkal, ha

$$\langle \mathbf{x}_i, \boldsymbol{\varphi} \rangle > c, \quad \text{ha } y_i = 1, \quad (5.3)$$

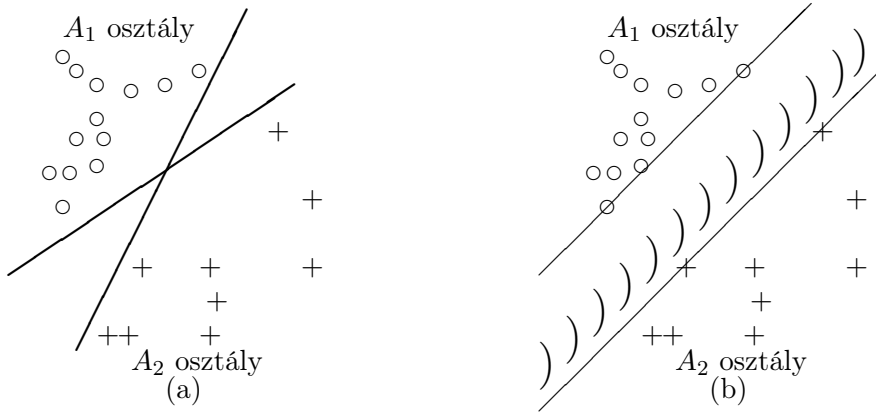
$$\langle \mathbf{x}_i, \boldsymbol{\varphi} \rangle < c, \quad \text{ha } y_i = -1, \quad (5.4)$$

ahol $\boldsymbol{\varphi} \in \mathbb{R}^d$ egységvektor, $c \in \mathbb{R}$ és $\langle \mathbf{a}, \mathbf{b} \rangle$ az \mathbf{a} és a \mathbf{b} vektorok közötti belső szorzatot jelöli.

Azt már láttuk, hogy perceptronnal ez a lineáris szeparálás megoldható, éppen ezt állítja a Rosenblatt–Novikoff-féle *perceptron konvergencia tétel*. Most azonban az optimális, azaz a legjobb elválasztó hipersíkot keressük. Szemléltessük a különböző lineáris szeparálásokat! Az 5.1. ábra (a) részén két nem optimális elválasztó egyenes látható, itt a „margók” keskenyek. A (b) részén pedig optimális a szeparálás, azaz az elválasztó hipersík a lehető legtávolabb van a minta pontoktól. Látható, hogy a vékony vonallal meghúzott margók a lehető legtávolabb vannak egymástól, a rájuk illeszkedő minta pontok az ún. tartó vektorok. A vastag vonallal húzott szeparáló egyenes pedig a „határsáv” közepén halad.

Feladatunk az (5.1) tanító halmaz ismeretében megkonstruálni az optimális hipersíkot. Egyszerű geometriai feltételt adunk az optimális hipersíkra, amelyet majd a Kuhn–Tucker-tétel alkalmazásához át fogunk fogalmazni. Tegyük fel tehát, hogy az (5.1) halmaz hipersíkkal szeparálható. Tetszőleges $\boldsymbol{\varphi}$ egységvektor esetén vezessük be az alábbi jelöléseket:

$$c_1(\boldsymbol{\varphi}) = \min_{y_i=1} \langle \mathbf{x}_i, \boldsymbol{\varphi} \rangle, \quad (5.5)$$



5.1. ábra. Elválasztás hipersíkkal (kétdimenzióban egyenesekkel): (a) két nem optimális elválasztás, (b) maximális margójú elválasztás

és

$$c_2(\varphi) = \max_{y_i=-1} \langle \mathbf{x}_i, \varphi \rangle. \quad (5.6)$$

Legyen a φ_0 egységvektor a

$$\varrho(\varphi) = \frac{c_1(\varphi) - c_2(\varphi)}{2} \quad (5.7)$$

kifejezés $\|\varphi\| = 1$ feltétel melletti maximumhelye.

5.1. Állítás. *A fentiekben definiált φ_0 vektor és a*

$$c_0 = \frac{c_1(\varphi_0) + c_2(\varphi_0)}{2} \quad (5.8)$$

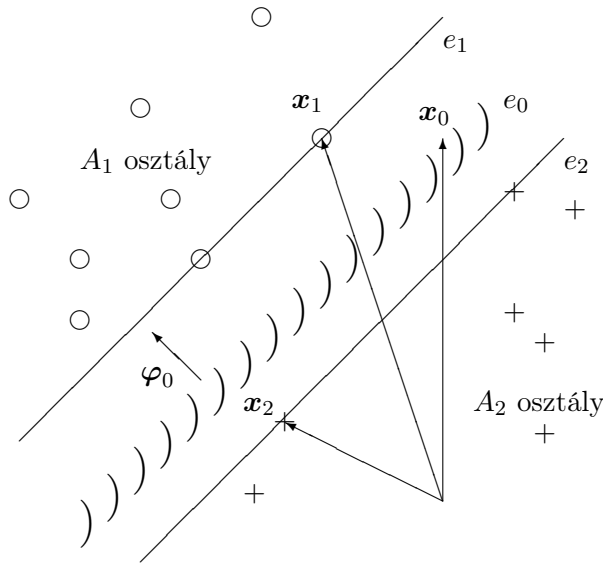
konstans megadja az $\langle \mathbf{x}, \varphi_0 \rangle = c_0$ optimális, vagyis maximális margójú szeparáló hipersíkot.

Bizonyítás. Legyen e_0 az optimális elválasztó hipersík (kétdimenzióban egyenes), ennek egységnyi hosszúságú normálvektora φ_0 . Legyen e_0 egy pontja \mathbf{x}_0 , az A_1 halmaz egy eleme pedig \mathbf{x}_1 . Kétdimenzióban az 5.2. ábra szemlélteti ezeket. Látható, hogy

$$\langle \varphi_0, \mathbf{x}_1 - \mathbf{x}_0 \rangle = m_1 > 0,$$

és \mathbf{x}_1 éppen m_1 távolságra van az e_0 hipersíktól. Az előbbi formulából

$$\langle \varphi_0, \mathbf{x}_1 \rangle = \langle \varphi_0, \mathbf{x}_0 \rangle + m_1, \quad (5.9)$$



5.2. ábra. Az elválasztás margójának szemléltetése

ezek minimuma (amint \mathbf{x}_1 végigfutja A_1 -et) pedig $c_1(\varphi_0)$. Másrészt $\mathbf{x}_2 \in A_2$ esetén a

$$\langle \varphi_0, \mathbf{x}_2 \rangle = \langle \varphi_0, \mathbf{x}_0 \rangle + m_2 \quad (5.10)$$

képletben m_2 éppen az $\mathbf{x}_2 \in A_2$ pont e_0 -tól való távolságának a negatívja. Ezek alapján

$$\varrho(\varphi) = \frac{c_1(\varphi) - c_2(\varphi)}{2}$$

maximalizálása pontosan a margó maximalizálását jelenti. Az 5.2 ábránk szerint az e_1 és e_2 egyenesek távolságának maximalizálásáról beszélünk. \square

Megjegyezzük, hogy ha a hipersík $\langle \varphi_0, \mathbf{x} \rangle = c_0$ egyenletében c_0 -t (5.8) szerint választjuk, akkor a hipersík az A_1 és A_2 halmazokat elválasztó (és lehető legszélesebb) sáv középvonalában halad. Valóban, \mathbf{x} akkor és csak akkor elégíti ki a $\langle \varphi_0, \mathbf{x} \rangle = c_0$ egyenletet, ha

$$\langle \varphi_0, \mathbf{x} \rangle = c_0 = \frac{c_1(\varphi_0) + c_2(\varphi_0)}{2}. \quad (5.11)$$

Jelölje m_1 és m_2 az e_0 -hoz legközelebbi elemek előjeles távolságát e_0 -tól. Ekkor a fenti egyenlőségből (5.9) és (5.10) alapján

$$\langle \varphi_0, \mathbf{x} \rangle = \frac{\langle \varphi_0, \mathbf{x}_0 \rangle + m_1 + \langle \varphi_0, \mathbf{x}_0 \rangle + m_2}{2} = \langle \varphi_0, \mathbf{x}_0 \rangle + \frac{m_1 + m_2}{2}.$$

Viszont tudjuk, hogy \mathbf{x} akkor és csak akkor van e_0 -on, ha $\langle \varphi_0, \mathbf{x} - \mathbf{x}_0 \rangle = 0$. Ez és a megelőző utóbbi egyenlőség pedig csupán $m_1 = -m_2$ fennállásakor teljesülhet egyidejűleg. Azaz e_0 az e_1 és e_2 egyenesektől egyenlő távol halad.

Felmerül a kérdés, hogy létezik-e több optimális hipersík. A józan szemlélet és néhány kétdimenziós próbálkozás mutatja, hogy nem.

5.2. Tétel. *Két lineárisan szeparálható halmaz esetén az optimális elválasztó hipersík egyértelmű.*

Bizonyítás. Mivel $\varrho(\varphi)$ folytonos, így létezik maximumhelye a $\|\varphi\| \leq 1$ zárt egységgömbben (azaz kompakt halmazon). Belátjuk, hogy a maximumhely a gömbfelületen van. Tegyük fel ellenkezőleg, hogy a maximum valamely φ^* belső pontban van. A szeparálhatóság miatt $\varphi^* \neq \mathbf{0}$. Ekkor viszont a

$$\varphi_0 = \frac{\varphi^*}{\|\varphi^*\|}$$

egységvektor nagyobb margót ad, mint φ^* :

$$\varrho(\varphi_0) = \frac{\varrho(\varphi^*)}{\|\varphi^*\|} > \varrho(\varphi^*),$$

mivel $\|\varphi^*\| < 1$ -et feltételeztük. Ezért a $\|\varphi^*\| = 1$, azaz a maximum csak a határon lehet. (Fent kihasználtuk azt, hogy a két részhalmaz hipersíkkal elválasztható, ami alapján valamely φ pontban $\varrho(\varphi)$ pozitív.)

Tegyük fel most, hogy a tétel állításával ellentétben két maximumhely is van. Az előzők szerint mindkettő a határon helyezkedik el. Mivel konkáv függvények minimuma is konkáv, így $c_1(\varphi)$ (lévén lineáris függvények minimuma) konkáv. Ebből következően a $\varrho(\varphi)$ függvény is konkáv. Ezért, ha $\varrho(\varphi)$ -nak két pontban is maximuma lenne, akkor a közöttük lévő szakasz pontjaiban is maximuma lenne. Ezek a pontok viszont az egységgömb belsejébe esnének, ami ellentmond az előzőekben bizonyítottaknak. \square

Most átfogalmazzuk a fenti feladatot konvex minimum problémává, mivel az ilyen módon kapott alakra már tudjuk a Kuhn–Tucker-tételt alkalmazni. Annak segítségével pedig a duális változók terében kapunk egy kvadratikusan programozási feladatot, ami numerikusan megoldható.

A feladat átfogalmazása a következő. Keressük azt a ψ_0 vektort és b_0 konstans, melyek teljesítik a

$$\langle \mathbf{x}_i, \psi \rangle + b \geq 1, \quad \text{ha } y_i = 1, \quad (5.12)$$

$$\langle \mathbf{x}_i, \psi \rangle + b \leq -1, \quad \text{ha } y_i = -1 \quad (5.13)$$

egyenlőtlenségeket, és minimalizálják a

$$\|\psi\|^2 = \langle \psi, \psi \rangle \quad (5.14)$$

norma négyzetet.

5.3. Tétel. Az (5.14) kvadratikus kifejezést az (5.12)–(5.13) lineáris feltételek mellett minimalizáló ψ_0 vektor normalizáltja éppen az optimális hipersíkot megadó φ_0 vektor, azaz

$$\varphi_0 = \frac{\psi_0}{\|\psi_0\|}. \quad (5.15)$$

Továbbá az optimális hipersík és a szeparálandó vektorok közötti margó nagysága:

$$\varrho(\varphi_0) = \sup_{\|\cdot\|=1} \frac{1}{2} \left(\min_{y_i=1} \langle x_i, \varphi \rangle - \max_{y_i=-1} \langle x_i, \varphi \rangle \right) = \frac{1}{\|\psi_0\|}. \quad (5.16)$$

Bizonyítás. Először is látható, hogy az (5.3)–(5.4) szeparálhatósági feltételekből következik, hogy van az (5.12)–(5.13) feltételeket kielégítő ψ vektor, és az is, hogy az ilyen vektorok nem lehetnek a nullához közel. Mivel ψ_0 egy négyzetes függvény minimumhelye lineáris feltételek mellett, így belátható, hogy egyértelmű, és az előzőek alapján nem nulla. Legyen

$$\varphi_1 = \frac{\psi_0}{\|\psi_0\|}$$

ψ_0 normált alakja. Ekkor ϱ , c_1 és c_2 definíciójából, valamint az (5.12)–(5.13) feltételekből következik, hogy

$$\begin{aligned} \varrho(\varphi_1) &= \varrho\left(\frac{\psi_0}{\|\psi_0\|}\right) = \frac{1}{2} \left(c_1\left(\frac{\psi_0}{\|\psi_0\|}\right) - c_2\left(\frac{\psi_0}{\|\psi_0\|}\right) \right) \geq \\ &\geq \frac{1}{2} \left(\frac{1-b_0}{\|\psi_0\|} - \frac{-1-b_0}{\|\psi_0\|} \right) = \frac{1}{\|\psi_0\|}. \end{aligned}$$

Azonban az eredeti φ_0 vektor az (5.7)-beli ϱ függvényt az egységgömbön maximalizáló egyértelmű vektor. Most ki fogjuk mutatni, hogy a ϱ függvény az egységgömbön nem vehet fel $1/\|\psi_0\|$ -nál nagyobb értéket. Ebből a fenti egyenlőtlenség miatt következik, hogy

$$\varrho(\varphi_1) = \frac{1}{\|\psi_0\|},$$

másrészt $\varphi_1 = \varphi_0$. Indirekt bizonyítunk. Tegyük fel ellenkezőleg, hogy létezik φ^* egységvektor, melyre

$$\varrho(\varphi^*) > \frac{1}{\|\psi_0\|} \quad (5.17)$$

teljesül. Tekintsük a

$$\psi^* = \frac{\varphi^*}{\varrho(\varphi^*)}$$

vektort. Erre (5.17) miatt

$$\|\psi^*\| = \frac{\|\varphi^*\|}{\varrho(\varphi^*)} < \|\varphi^*\| \cdot \|\psi_0\| = \|\psi_0\|,$$

azaz kisebb normájú, mint ψ_0 . Ellenőrizzük, hogy a ψ^* vektor a

$$b = -\frac{c_1(\varphi^*) + c_2(\varphi^*)}{2\varrho(\varphi^*)}$$

számmal teljesíti az (5.12)–(5.13) feltételeket. Valóban, $\mathbf{x}_i \in A_1$ esetén az (5.12) feltételt felírva:

$$\langle \mathbf{x}_i, \psi^* \rangle + b = \frac{\langle \mathbf{x}_i, \varphi^* \rangle}{\varrho(\varphi^*)} - \frac{c_1(\varphi^*) + c_2(\varphi^*)}{2\varrho(\varphi^*)} \geq \frac{c_1(\varphi^*) - c_2(\varphi^*)}{2\varrho(\varphi^*)} = 1.$$

Hasonló (5.13) esetén is. Tehát ψ^* kisebb normájú, mint ψ_0 , és teljesíti az (5.12)–(5.13) feltételeket. Ez ellentmond a ψ_0 definíciójának. \square

Az optimális hipersík numerikus meghatározása

Az optimális hipersík numerikusan kezelhető alakját keressük. Írjuk egységes alakba az (5.12)–(5.13) feltételeket:

$$y_i(\langle \mathbf{x}_i, \psi \rangle + b) \geq 1, \quad i = 1, \dots, N. \quad (5.18)$$

Ezen feltételek esetén kell megkeresnünk az

$$\frac{1}{2}\|\psi\|^2$$

függvény minimumhelyét. Az (5.18) feltételek ψ -ben és b -ben konvex (lévén lineáris) függvények, továbbá a célfüggvény is konvex. Tehát a feltételes minimum keresésére a Kuhn–Tucker-tételt (lásd Appendix) alkalmazhatjuk.

Mivel a lineáris szeperálhatóság miatt a Slater-feltétel teljesül, ezért létezik optimum.

Most tehát

$$L(\boldsymbol{\psi}, b, \boldsymbol{\alpha}) = \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle - \sum_{i=1}^N \alpha_i (y_i [\langle \mathbf{x}_i, \boldsymbol{\psi} \rangle + b] - 1) = \quad (5.19)$$

$$= \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle - \left\langle \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i, \boldsymbol{\psi} \right\rangle - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i \quad (5.20)$$

a Lagrange-függvény, ahol $\alpha_i \geq 0$, $i = 1, \dots, N$, a Lagrange-féle multiplikátorok. A minimumhely meghatározásához ezen L függvény nyeregponjtját kell megtalálni, azaz minimalizálni kell $\boldsymbol{\psi}$ és b szerint és maximalizálni $\boldsymbol{\alpha}$ szerint. Határozzuk meg az (5.20) Lagrange-függvény deriváltját $\boldsymbol{\psi}$ és b szerint.

$$0 = \frac{\partial L(\boldsymbol{\psi}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{\psi}} = \boldsymbol{\psi} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i,$$

$$0 = \frac{\partial L(\boldsymbol{\psi}, b, \boldsymbol{\alpha})}{\partial b} = - \sum_{i=1}^N \alpha_i y_i.$$

A fenti két egyenletből

$$\boldsymbol{\psi} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i, \quad (5.21)$$

$$\sum_{i=1}^N \alpha_i y_i = 0. \quad (5.22)$$

Ezeket visszahelyettesítjük az (5.20) egyenlőségbe:

$$W(\boldsymbol{\alpha}) = -\frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle + \sum_{i=1}^N \alpha_i,$$

azaz

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \quad (5.23)$$

A behelyettesítések nyomán $\boldsymbol{\psi}$ és b kiesett, az L -ből így kapott függvényt pedig $W(\boldsymbol{\alpha})$ jelöli az (5.23) formulában.

A Kuhn–Tucker-tétel értelmében meg kell találni azokat az α_i^0 konstansokat, amelyek esetén az (5.23) eléri a maximumát az $\alpha_i \geq 0$ és az (5.22)

feltételek mellett. Ez egy kvadratikus programozási feladat, amelynek megoldására léteznek módszerek. Ha ezt megoldjuk (a megoldást jelölje α_i^0), akkor megkapjuk az optimális hipersíkot megadó vektort:

$$\psi_0 = \sum_{i=1}^N y_i \alpha_i^0 \mathbf{x}_i.$$

A b_0 értéknek ki kell elégítenie a

$$\alpha_i^0 (y_i (\langle \mathbf{x}_i, \psi_0 \rangle + b_0) - 1) = 0, \quad i = 1, \dots, N,$$

Kuhn–Tucker-feltételeket (lásd (6.47)).

A zérustól különböző α_i^0 értékekhez tartozó \mathbf{x}_i vektorokra tehát teljesülnie kell az

$$y_i (\langle \mathbf{x}_i, \psi_0 \rangle + b_0) = 1 \quad (5.24)$$

feltételnek. Innen b_0 kifejezhető. Ezen feltételt kielégítő \mathbf{x}_i vektorokat nevezzük *tartó vektoroknak* (support vector). Az \mathbf{x}_i vektorok közül a tartó vektorok vannak a legközelebb az optimális hipersíkhhoz.

Végül az optimális hipersík egyenlete:

$$f(\mathbf{x}) = \sum_{i=1}^N y_i \alpha_i^0 \langle \mathbf{x}_i, \mathbf{x} \rangle + b_0 = 0.$$

Ezt a képletet az optimális hipersík (5.2) képletét, $c_1(\boldsymbol{\varphi})$ és $c_2(\boldsymbol{\varphi})$ definícióját, a c_0 -ra vonatkozó (5.11) képletet, az (5.12)–(5.13) feltételeket, továbbá $\boldsymbol{\varphi}_0$ és $\boldsymbol{\psi}_0$ (5.15) összefüggését, valamint az (5.24) egyenlőséget használva igazolhatjuk. Figyeljük meg, hogy mind $\boldsymbol{\psi}_0$ kifejezésében, mind az optimális hipersík képletében csak a tartó vektorok szerepelnek nem-nulla együtthatóval.

Az is látható, hogy mind az optimális hipersík képletében, mind a W célfüggvényben csak a vektorok belső szorzata szerepel, maguk a vektorok nem. Ez fontos lesz a lineárisan nem szeparálható esetben.

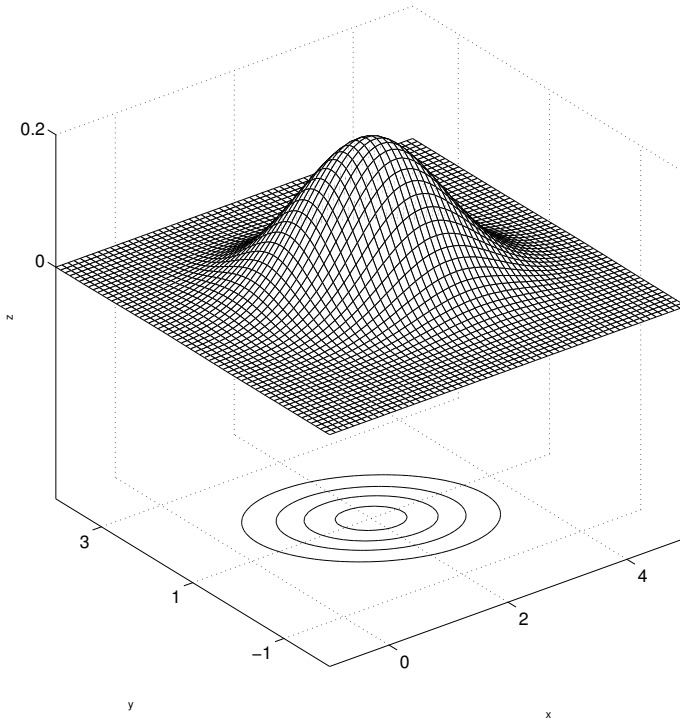
5.2.2. Az optimális hipersík a nem szeparálható esetben

A gyakorlatban előforduló esetek többsége olyan, hogy a halmazok nem szeparálhatóak lineárisan. De ilyenkor is előfordulhat, hogy az optimális szeparálás lineáris. Például közismert, hogy két, azonos szórás mátrixú, de különböző várható érték vektorú normális eloszlásból vett mintát hipersíkkal kell

szétválasztani. Az 5.3. ábrán a

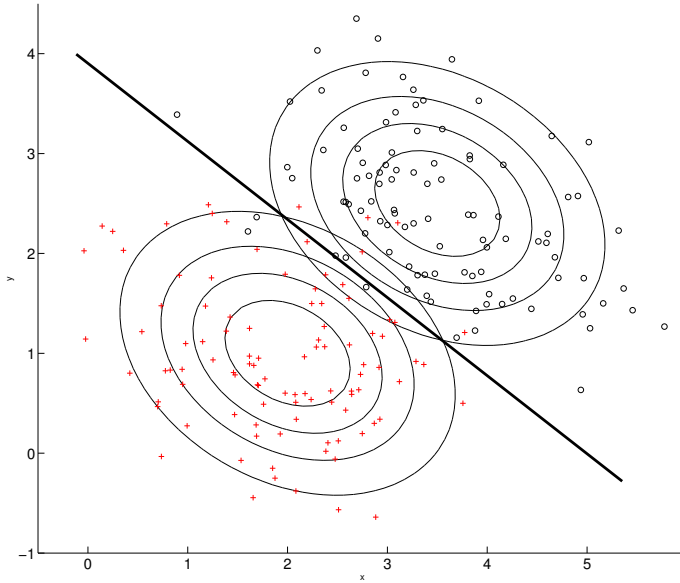
$$D = \begin{pmatrix} \frac{7}{8} & -\frac{\sqrt{3}}{8} \\ -\frac{\sqrt{3}}{8} & \frac{5}{8} \end{pmatrix}$$

szórásmátrixú, és $\mathbf{m} = (2, 1)^\top$ várható érték vektorú kétdimenziós normális eloszlás sűrűségfüggvénye látható a szintvonalakkal.



5.3. ábra. Kétdimenziós normális sűrűségfüggvény szintvonalakkal

Generáljunk 100 elemű mintát ebből a normális eloszlásból, valamint az ugyancsak D szórásmátrixú, de $\mathbf{n} = (3.5, 2.5)^\top$ várható érték vektorú kétdimenziós normális eloszlásból. Ábrázoljuk a két mintát a síkon! Hogyan lehet ezeket szeparálni? Az 5.4. ábrán látható az eredmény. A berajzolt (elméleti) szintvonalak, azaz a koncentráció ellipszisek jól mutatják, hogy az egyenes jól szeparál. De természetesen vannak rosszul szeparált minta pontok, hiszen a két eloszlás „átnyúlik” egymásba. Tehát ha egy hipersíkkal akarjuk elválasztani az adatokat, akkor valahogyan kezelniük kell a hibás szétválasztást.



5.4. ábra. Normális eloszlásból vett minták szeparálása

Két osztály közötti olyan elválasztást tekintünk, hogy néhány tanító pont megsérti a korábbi (5.18) feltételt, azaz az

$$y_i(\langle \psi, \mathbf{x}_i \rangle + b) \geq 1 \quad (5.25)$$

egyenlőtlenséget. Ez kétféle módon történhet: vagy

- az (\mathbf{x}_i, y_i) tanító pont a sík jó oldalán van, de a két margó köze esik, vagy
- az (\mathbf{x}_i, y_i) tanító pont a sík rossz oldalára esik.

A Vapnik által javasolt megoldás a következő. Vezessük be a nem-negatív $\{\xi_i\}_{i=1}^N$ segédváltozókat, és az (5.25) egyenlőtlenség helyett csak az

$$y_i(\langle \psi, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \dots, N, \quad (5.26)$$

feltétel teljesülését követeljük meg.

Ha $0 \leq \xi_i < 1$, akkor a tanító pont a sík jó oldalra esik, de a margók közé, ha viszont $1 < \xi_i$, akkor ez a pont a sík rossz oldalára esik. Az „átlógások” minimalizálása érdekében ezeket a hibákat korlátozni kell. Azaz a

$$\sum_{i=1}^N \xi_i$$

kifejezésnek a lehető legkisebbnek kell lennie.

A javasolt célfüggvény a következő:

$$\Phi(\boldsymbol{\psi}, \boldsymbol{\xi}) = \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle + C \sum_{i=1}^N \xi_i. \quad (5.27)$$

Minimalizálandó az (5.27) célfüggvény az (5.26) és a $\xi_i \geq 0$, $i = 1, 2, \dots, N$, feltételek mellett.

Két osztály közötti ilyen elválasztást Vapnik [49] *lágymargójúnak* (soft margin) nevezi. A fenti célfüggvény magyarázatául szakkönyvek a következőket adják. Haykin [18] könyve egyrészt a célfüggvény matematikailag kezelhető voltát emeli ki. Másrészt megjegyzi, hogy (5.27) jobb oldalának első tagja az illesztett modell bonyolultságát, második tagja pedig az „átlógási” hibát kontrollálja. A szabadon választható $C > 0$ paraméterrel e két tényező között egyensúlyozhatunk. Maga Vapnik [49] először olyan elválasztást tekint, ahol a margó Δ nagyságú (az eredeti esetben automatikusan $\Delta = 1/\|\boldsymbol{\psi}_0\|$ volt, lásd (5.16)). Annak az számolásait egyszerűsítendő ajánlja a lágymargót, azaz a fenti célfüggvényt. Cristianini és Shawe-Taylor [10] pedig a fenti célfüggvényt az osztályzás hibájára adott elméleti felső korlátból származtatja.

Most alkalmazzuk a Kuhn–Tucker-féle tételt az (5.27) konvex célfüggvénynek az (5.26) és a $\xi_i \geq 0$, $i = 1, 2, \dots, N$, lineáris feltételek melletti minimalizálására. Készítsük el az

$$\begin{aligned} L(\boldsymbol{\psi}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \\ &= \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i \end{aligned}$$

Lagrange-függvényt, ahol $\alpha_i \geq 0$, $\beta_i \geq 0$ a multiplikátorok. Megint nyereg-pontot keresünk. $\boldsymbol{\psi}$, b és ξ_i szerint minimumot, míg α_i , β_i szerint maximumot. A deriváltakra

$$\mathbf{0} = \frac{\partial L(\boldsymbol{\psi}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{\psi}} = \boldsymbol{\psi} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i, \quad (5.28)$$

$$0 = \frac{\partial L(\boldsymbol{\psi}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = - \sum_{i=1}^N \alpha_i y_i, \quad (5.29)$$

$$0 = \frac{\partial L(\boldsymbol{\psi}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = C - \alpha_i - \beta_i. \quad (5.30)$$

Az (5.28) egyenlőségből $\boldsymbol{\psi} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$. Ezt visszahelyettesítve L kifejezésébe, és alkalmazva az (5.29) és (5.30) egyenlőségekből adódó összevonásokat, látjuk, hogy a tagok többsége kiesik. Így behelyettesítés után a Lagrange-függvény a következő alakú lesz:

$$W(\boldsymbol{\alpha}) = -\frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle + \sum_{i=1}^N \alpha_i,$$

azaz

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \alpha_i \alpha_j. \quad (5.31)$$

Tehát az (5.31) kifejezést kell maximalizálni $\alpha_1, \dots, \alpha_N$ szerint a

$$\sum_{i=1}^N y_i \alpha_i = 0, \quad (5.32)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N, \quad (5.33)$$

feltételek mellett. Ez utóbbiak közül (5.33) az (5.30) és a $\beta_i \geq 0$ feltételekből adódik.

Ez a feltételes maximum keresés egy kvadratikus programozási feladat. Ha ezt megoldjuk (a megoldást $\alpha_1^0, \dots, \alpha_N^0$ jelöli), akkor megkapjuk az optimális hipersíkot megadó vektort:

$$\boldsymbol{\psi}_0 = \sum_{i=1}^N y_i \alpha_i^0 \mathbf{x}_i.$$

A b_0 értéknek ki kell elégítenie az

$$\alpha_i^0 [y_i (\langle \mathbf{x}_i, \boldsymbol{\psi}_0 \rangle + b_0) - 1 + \xi_i] = 0, \quad i = 1, \dots, N, \quad (5.34)$$

$$\beta_i \xi_i = 0, \quad i = 1, \dots, N, \quad (5.35)$$

Kuhn–Tucker-feltételeket. Elemezzük ezeknek a feltételeknek a következményét. Ha most $\alpha_i^0 < C$, akkor (5.30) miatt $\beta_i \neq 0$, de ekkor az előzőekből $\xi_i = 0$. Emiatt, ha $0 < \alpha_i^0 < C$, akkor

$$y_i (\langle \mathbf{x}_i, \boldsymbol{\psi}_0 \rangle + b_0) = 1. \quad (5.36)$$

Innen b_0 meghatározható.

Végül az optimális hipersík egyenlete:

$$f(\mathbf{x}) = \sum_{i=1}^N y_i \alpha_i^0 \langle \mathbf{x}_i, \mathbf{x} \rangle + b_0 = 0.$$

5.2.3. Az SVM használata nem-lineáris szeparálásra

Magfüggvények

A d -dimenziós térben akarunk szeparálni. Bizonyos esetekben az optimális elválasztó felület hipersík, még akkor is, ha a két halmaz hipersíkkal nem szeparálható hiba nélkül.

Azonban a valóságos problémákban az elválasztó felület gyakran nem hipersík. Amint azt már az RBF tárgyalásánál láttuk, megoldást jelenthet a magasabb dimenziós térbe történő transzformálás. A mintát a $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}^q$, $d < q$, nem-lineáris függvénnyel magasabb dimenziós térbe képezzük. Ezt a teret a *tulajdonságok terének* (sajátságtér, feature space) nevezzük. Ebben a térben illesztjük az optimális hipersíkot. Ismeretes, hogy ebben a térben $q + 1$ (általános helyzetű) vektort tudunk lineárisan szeparálni hiba nélkül.

Az előzőekben láttuk, hogy a feladat megoldásához nincs szükségünk explicit módon a tulajdonságok terében lévő vektorokra, csak azok belső szorzatára. A belső szorzat általános fogalma: szimmetrikus bilineáris forma, melyből származó kvadratikus forma pozitív definit. Így (esetleg újabb transzformáció után, amelyet φ -be „beépítünk”), jutunk az alábbi képlethez

$$\langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle = \sum_i \mu_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{y}), \quad \mu_i > 0.$$

Tehát az ilyen alakú belső szorzatokat érdemes tekintenünk. Lényegében ilyen alakú kifejezéshez jutunk, ha a belső szorzat helyett magfüggvényt használunk. Ez az alább ismertetendő Mercer-féle tételből következik.

Legyen $K: [a, b] \times [a, b] \rightarrow \mathbb{R}$ szimmetrikus függvény, azaz $K(x, y) = K(y, x)$. Tegyük fel, hogy K négyzetesen integrálható $[a, b]^2$ -en. Defináljuk a K -hoz tartozó integrál operátort

$$(A_K f)(x) = \int_a^b K(x, y) f(y) dy$$

szerint. Egy $\varphi: [a, b] \rightarrow \mathbb{R}$ négyzetesen integrálható (nem-nulla) függvényt az A_K operátor sajátfüggvényének nevezzük, a μ számot pedig sajátértékének, ha $A_K \varphi = \mu \varphi$.

Schmidt tétele szerint tetszőleges szimmetrikus, négyzetesen integrálható $K(x, y)$ függvény a

$$K(x, y) = \sum_{i=1}^{\infty} \mu_i \varphi_i(x) \varphi_i(y) \tag{5.37}$$

sorba fejthető, ahol a fenti sor L^2 -ben konvergencia, a $\{\varphi_i(x)\}$ függvények a $K(x, y)$ magból származtatott A_K operátor ortonormált sajátfüggvényei, $\{\mu_i\}$ -k pedig a megfelelő sajátértékek. Azonban a fenti sor nem mindig egyenletesen konvergens, még folytonos magfüggvény esetén sem. A pozitív szemidefinit esetben azonban igen. K -t pozitív szemidefinitnek nevezzük, ha minden $\mu_i \neq 0$ sajátérték pozitív.

5.4. Tétel (Mercer tétele, lásd Riesz–Szőkefalvi [41]). *Legyen $K: [a, b] \times [a, b] \rightarrow \mathbb{R}$ folytonos, szimmetrikus, pozitív szemidefinit függvény, akkor az (5.37) sorfejtés egyenletesen konvergens.*

A fenti tételnek vannak általánosabb változatai is. Például az $[a, b]$ intervallum helyettesíthető egy kompakt metrikus térrel, amelyet ellátnunk egy véges mértékkel. Haykin [18] könyvében az $[a, b]$ intervallum helyett egy $[\mathbf{a}, \mathbf{b}] \in \mathbb{R}^d$ téglalapot tekint. Számunkra az az alak alkalmas. Valójában azonban Mercer tétele számunkra nem bizonyításra, hanem csupán elvi alátámasztásra szolgál.

Tehát ha olyan $K(x, y)$ magfüggvényt használunk, amely a Mercer-tétel feltételeit kielégíti, akkor lényegében egy tulajdonságtérben dolgozunk. Általában az alábbi magfüggvényeket használjuk.

1. Polinomiális:

$$K(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + 1)^p.$$

Ez egy d -változós, p -edfokú polinom. p értékét nekünk kell megadnunk.

2. Radial Basis Function:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp \left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2 \right).$$

Ez lényegében a Radial Basis Function hálózatot (RBF) valósítja meg.

Viszont az SVM automatikusan határozza meg a tartó vektorok számát, magukat a tartó vektorokat és a (kimeneti rétegbeni) α_i együtthatókat. A σ paramétert magunk választhatjuk meg. Ezzel szemben a hagyományos RBF heurisztikus módszerrel határozza meg a középpontokat (azaz a tartó vektorokat).

3. Kétrétegű perceptron tangens hiperbolikus transzfer függvényvel:

$$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \langle \mathbf{x}, \mathbf{x}_i \rangle + \beta_1).$$

Ez a magfüggvény csak bizonyos β_0, β_1 értékekkel teljesíti a Mercer-tétel feltételeit. Ez az SVM az MLP hálózatok sigmoid függvényét (a

tangens hiperbolikus függvényt) használja, lényegében kétrétegű perceptront valósít meg. Viszont az SVM automatikusan határozza meg a rejtett szint neuronjainak a számát (azaz a tartó vektorok számát), a hozzá tartozó súlyokat (\mathbf{x}_i lesz az i -edik neuron súlyvektora) és a kimeneti szint α_i súlyait. Ezzel szemben a hagyományos MLP heurisztikus módszerrel határozza meg a rejtett rétegbeni neuronok számát.

Megjegyezzük, hogy a képfeldolgozásban gyakran használják a hisztogram-metszet magfüggvényt.

Az osztályozás végrehajtása

A fent említett transzformáció miatt a ténylegesen megfigyelt

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$$

értékek helyett a

$$(\varphi(\mathbf{x}_1), y_1), \dots, (\varphi(\mathbf{x}_N), y_N)$$

pontokat kell szeparálnunk. Továbbá, tudva hogy az optimális hipersík

$$\langle \psi_0, \mathbf{x} \rangle + b_0 = 0$$

alakú, ahol

$$\psi_0 = \sum_{i=1}^N y_i \alpha_i^0 \mathbf{x}_i,$$

sőt az optimalizálási feladatban nem maguk a vektorok, csak azok belső szorzatai szerepelnek, minden belső szorzatot magfüggvénnyel helyettesítünk.

Tehát mindenütt a $\langle \cdot, \cdot \rangle$ belső szorzat helyett a $K(\cdot, \cdot)$ magfüggvényt írva az alábbiakat kapjuk. Az elválasztó felület egyenlete:

$$f(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{i=1}^N y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b = 0. \quad (5.38)$$

Az SVM által megvalósított ezen megoldást hálózatként is ábrázolhatjuk, amit az 5.5. ábrán mutatunk be.

Az (5.38) egyenletben az α_i együtthatók és a b értékét kell meghatároznunk. Az előző szakaszok számításai alapján ezekhez a

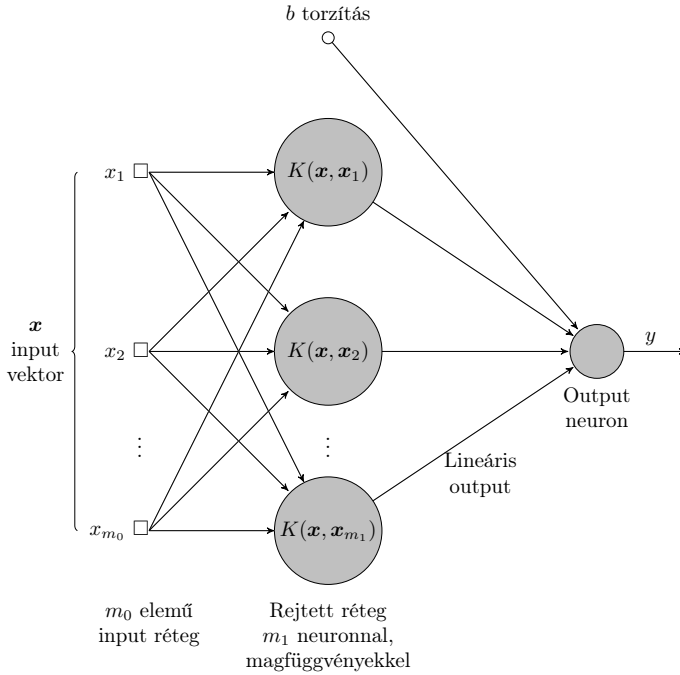
$$W(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (5.39)$$

kvadratikus célfüggvény

$$\sum_{i=1}^N y_i \alpha_i = 0, \quad (5.40)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N, \quad (5.41)$$

lineáris feltételek melletti maximalizálásával jutunk. A kapott α_i^0 értékek egy része nullával egyenlő, a nem nullához tartozó \mathbf{x}_i vektorok a tartó vektorok.



5.5. ábra. Az SVM által megvalósított hálózat

Az SVM használata: ha a fentieket kiszámoltuk, akkor egy tetszőleges \mathbf{z} pontot (nem tanító pontot) a

$$d(\mathbf{z}, \boldsymbol{\alpha}) = \text{sgn} \left(\sum_{\mathbf{x}_i \text{ tartó vektor}} y_i \alpha_i^0 K(\mathbf{z}, \mathbf{x}_i) + b_0 \right)$$

döntésfüggvény szerint osztályozunk.

Több osztályba sorolás

Ha több osztály létezik, akkor a feladatot visszavezetjük két osztályra.

5.2.4. Az SVM tanítása

A tanítás jelen esetben egy függvény feltételes maximumának meghatározása. Valójában az MLP esetén is egy függvény (a hiba függvény) szélsőértékét kerestük, azonban ott az error-backpropagation végrehajtása során egyre „jobb” hálózatot kaptunk. Tehát az MLP esetén a tanítás egy (látványos) folyamat, míg az SVM esetén inkább csak egy numerikus procedúra.

Az (5.39)–(5.41) kvadratikus programozási feladat megoldására többféle módszer is ismeretes. Alkalmazható a gradiens módszer, a konjugált gradiens módszer, a Newton-módszer, a primál-duál módszer (lásd [10]). Számos programcsomag is rendelkezésre áll. Azonban a feladat fő nehézsége, hogy a gyakorlatban nagyon nagy méretű adathalmazzal kell dolgozni. Éppen a feladat mérete miatt adódnak jelentős nehézségek a numerikus számításoknál.

Tehát (5.39)–(5.41) szerint egy N -dimenziós α vektort kell meghatározunk. Itt N a tanító halmaz mérete, azaz α minden koordinátája egy tanító pontnak felel meg. A végeredményként kapott vektor sok koordinátája egyenlő lesz nullával (csak azok nem, amelyekhez tartozó vektorok tartó vektorok). Osszuk a tanító halmazt két diszjunkt részhalmazzra B -re és D -re ($B \cap D = \emptyset$, $B \cup D = \{1, \dots, N\}$). A D -hez tartozó pontok passzívak lesznek, ezekhez kezdeti értéként 0-t rendelünk. Az optimalizálást a B halmazon hajtjuk végre valamilyen ismert módszerrel (B az aktív halmaz, vagy munkahalmaz). Ha teljesülnek a Kuhn–Tucker-feltételek, akkor elértük az optimumot.

Tehát elemezzük a Kuhn–Tucker-feltételeket. Ezek (5.34) és (5.35), azaz

$$\alpha_i[y_i(\langle \mathbf{x}_i, \boldsymbol{\psi} \rangle + b) - 1 + \xi_i] = 0, \quad i = 1, \dots, N, \quad (5.42)$$

$$\beta_i \xi_i = 0, \quad i = 1, \dots, N, \quad (5.43)$$

ahol $\boldsymbol{\psi} = \sum_{i=1}^N y_i \alpha_i \mathbf{x}_i$. Itt azonban sem a β_i -ket, sem a ξ_i -ket nem számoljuk ki. Tehát ezeket valahogyan ki kell küszöbölni.

Felhasználjuk, hogy (5.30) alapján $\beta_i = C - \alpha_i$. Kihasználjuk azt is, hogy (5.41) miatt $0 \leq \alpha_i \leq C$. Továbbá azt is ki kell használnunk, hogy ξ_i -k alkalmas nem-negatív értékek. Végül a Kuhn–Tucker-feltételek következő alakjához jutunk:

$$y_i f(\mathbf{x}_i) \geq 1, \quad \text{ha} \quad \alpha_i = 0, \quad (5.44)$$

$$y_i f(\mathbf{x}_i) = 1, \quad \text{ha} \quad 0 < \alpha_i < C, \quad (5.45)$$

$$y_i f(\mathbf{x}_i) \leq 1, \quad \text{ha} \quad \alpha_i = C, \quad (5.46)$$

ahol $f(\mathbf{x}) = \sum_{j=1}^N y_j \alpha_j \langle \mathbf{x}_j, \mathbf{x} \rangle + b = \langle \boldsymbol{\psi}, \mathbf{x} \rangle + b$. Ezekben a kifejezésekben már minden összetevő numerikusan meghatározott.

(5.44) igazolása. $\alpha_i = 0$ esetén $\beta_i = C$, azaz $\xi_i = 0$. Tehát (5.44) ekvivalens az (5.26) feltétellel.

(5.45) igazolása. $0 < \alpha_i < C$ esetén $\beta_i = C - \alpha_i > 0$, amiből $\xi_i = 0$. Ezért (5.42) ekvivalens $y_i(\langle \mathbf{x}_i, \boldsymbol{\psi} \rangle + b) = 1$ -gyel.

(5.46) igazolása. $\alpha_i = C$ esetén $\beta_i = 0$, azaz $\xi_i \geq 0$. Így (5.42) ekvivalens az (5.46) egyenlőtlenséggel.

Térjünk most vissza a kiinduló feladathoz. Tehát kiválasztottuk a B munkahalmazt. B -n optimalizáltunk. Így B -n teljesülni fognak a Kuhn–Tucker-feltételek. A kapott megoldás a teljes tanító halmazon is optimális, ha a D -beli pontokra is teljesülnek a Kuhn–Tucker-feltételek. Ha létezik olyan D -beli pont, amelyre nem teljesülnek a Kuhn–Tucker-feltételek, akkor néhány ilyen pontot bevonunk B -be, viszont a 0 együtthatós B -beli elemeket kihagyjuk. Így egy új munkahalmazt képezünk. Ebben a halmazban az előző megoldás nem optimális, hisz nem teljesülnek a Kuhn–Tucker-feltételek. Tehát optimalizáljunk itt! Ezzel szigorúan növelhető a W függvény értéke (azaz a ciklizálást is elkerüljük). Addig folytatjuk a munkahalmaz változtatását, ott optimalizálást, a Kuhn–Tucker-feltételek ellenőrzését, amíg az optimumhoz nem érünk. Ezt a heurisztikát szokták *chunking* módszernek is nevezni (azaz a mintából egy „nagy darabot”, nevezetesen B -t, vágunk ki optimalizálás céljából).

Sequential Minimal Optimization (SMO)

Ebben a részben a Sequential Minimal Optimization (SMO) eljárást ismertetjük Cristianini és Shawe-Taylor [10], valamint Platt [37] művekre támaszkodva. A Sequential Minimal Optimization kifejezést soros minimalizálásnak is fordíthatjuk.

Az SMO módszer egy-egy lépésben az $\boldsymbol{\alpha}$ vektor két koordinátáját tekinti csak változónak, és azok szerint optimalizál. Válasszunk két olyan α_i értéket, melyek nem teljesítik a Kuhn–Tucker-feltételeket. Az egyszerűség kedvéért legyenek ezek α_1 és α_2 .

Mivel a

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (5.47)$$

feltételnek teljesülnie kell az előző és a mostani lépésben is, és csak az első két koordináta változik, ezért

$$y_1 \alpha_1^{\text{új}} + y_2 \alpha_2^{\text{új}} = y_1 \alpha_1^{\text{rég}} + y_2 \alpha_2^{\text{rég}}. \quad (5.48)$$

Ha $y_1 = y_2$, akkor a fenti egyenletet szorozva y_1 -gyel (és kihasználva, hogy y_i csak -1 vagy 1 lehet), kapjuk

$$\alpha_1^{\text{új}} + \alpha_2^{\text{új}} = \alpha_1^{\text{rég}} + \alpha_2^{\text{rég}}.$$

Innen fejezzük ki $\alpha_1^{\text{új}}$ -at, használjuk ki, hogy $0 \leq \alpha_i \leq C$ mindig szükséges, kapjuk

$$0 \leq \alpha_1^{\text{rég}} + \alpha_2^{\text{rég}} - \alpha_2^{\text{új}} \leq C.$$

Ebből

$$\alpha_1^{\text{rég}} + \alpha_2^{\text{rég}} - C \leq \alpha_2^{\text{új}} \leq \alpha_1^{\text{rég}} + \alpha_2^{\text{rég}}.$$

Végül még egyszer használva a minden lépésben szükséges $0 \leq \alpha_i \leq C$ relációt, kapjuk az alábbi

$$\max\{0, \alpha_1^{\text{rég}} + \alpha_2^{\text{rég}} - C\} \leq \alpha_2^{\text{új}} \leq \min\{C, \alpha_1^{\text{rég}} + \alpha_2^{\text{rég}}\}. \quad (5.49)$$

Ha most $y_1 \neq y_2$, akkor az (5.48) egyenletet szorozva y_1 -gyel (és kihasználva, hogy y_i csak -1 vagy 1 lehet), kapjuk

$$\alpha_1^{\text{új}} - \alpha_2^{\text{új}} = \alpha_1^{\text{rég}} - \alpha_2^{\text{rég}}.$$

Ebből most

$$0 \leq \alpha_1^{\text{rég}} - \alpha_2^{\text{rég}} + \alpha_2^{\text{új}} \leq C$$

adódik. Végül az $y_1 \neq y_2$ esetben

$$\max\{0, -\alpha_1^{\text{rég}} + \alpha_2^{\text{rég}}\} \leq \alpha_2^{\text{új}} \leq \min\{C, C - \alpha_1^{\text{rég}} + \alpha_2^{\text{rég}}\}. \quad (5.50)$$

Az (5.49) és az (5.50) egyenlőtlenségekbeni határokat az algoritmus során vágásra fogjuk használni.

Az előző számítások azt mutatják, hogy ha érvényes az (5.48) egyenlőség, azaz

$$\alpha_1^{\text{új}} = \alpha_1^{\text{rég}} + y_1 y_2 (\alpha_2^{\text{rég}} - \alpha_2^{\text{új}}), \quad (5.51)$$

továbbá $\alpha_2^{\text{új}}$ értékét az (5.49), illetve az (5.50) határok között választjuk, akkor $\alpha_1^{\text{új}}$ értéke is $[0, C]$ -be fog esni. Ezt fel fogjuk használni α_1 aktualizálásakor.

Emlékeztetünk, hogy a maximalizálandó célfüggvény

$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j).$$

Ez két változó α koordináta (azaz α_1 és α_2) esetén szintén $W(\alpha)$ -vel jelölve:

$$W(\alpha) = \alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2 K_{11} - \frac{1}{2}\alpha_2^2 K_{22} - s\alpha_1\alpha_2 K_{12} - y_1\alpha_1 v_1 - y_2\alpha_2 v_2 + W_0,$$

ahol

$$s = y_1 y_2, \quad K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j), \quad v_i = \sum_{j=3}^N \alpha_j y_j K_{ij},$$

továbbá W_0 tartalmazza az α_1 -től és α_2 -től nem függő $W(\alpha)$ -beli tagokat. (Kihasználtuk $K(\cdot, \cdot)$ szimmetrikus voltát is.) Vezessük még be az

$$\alpha_1 + s\alpha_2 = \gamma$$

jelölést. Az (5.51) egyenlőség miatt γ konstans marad α_1 és α_2 aktualizálásakor. Ezzel

$$\begin{aligned} W(\alpha) = & \gamma - s\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}(\gamma - s\alpha_2)^2 - \frac{1}{2}\alpha_2^2 K_{22} - \\ & - s(\gamma - s\alpha_2)\alpha_2 K_{12} - y_1(\gamma - s\alpha_2)v_1 - y_2\alpha_2 v_2 + W_0. \end{aligned}$$

A maximum szükséges feltétele az első derivált nulla volta.

$$0 = \frac{\partial W(\alpha)}{\partial \alpha_2} = 1 - s + s(\gamma - s\alpha_2)K_{11} - \alpha_2 K_{22} + s v_1 y_1 - y_2 v_2 - s\gamma K_{12} + 2\alpha_2 K_{12}.$$

Ebből $s y_1 = y_2$ felhasználásával kapjuk

$$\alpha_2^{\text{új}} = \frac{s(K_{11} - K_{12})\gamma + y_2(v_1 - v_2) + 1 - s}{K_{11} + K_{22} - 2K_{12}}.$$

Itt a peremfeltétel miatt γ -t fixen kell tartani a maximum keresés közben. Viszont az alábbi algebrai átalakításban természetesen γ definíciójára támaszkodunk. Mivel az elválasztó felület egyenletében szereplő függvény

$$f(\mathbf{z}) = \sum_i y_i \alpha_i K(\mathbf{z}, \mathbf{x}_i) + b,$$

így egyszerű számolással

$$\begin{aligned} \alpha_2^{\text{új}} = & \frac{y_2(f(\mathbf{x}_1) - \alpha_1 y_1 K_{11} - \alpha_2 y_2 K_{12} - f(\mathbf{x}_2) + \alpha_1 y_1 K_{12} + \alpha_2 y_2 K_{22})}{K_{11} + K_{22} - 2K_{12}} + \\ & + \frac{y_1 y_2 (K_{11} - K_{12})(\alpha_1 + y_1 y_2 \alpha_2) + 1 - y_1 y_2}{K_{11} + K_{22} - 2K_{12}} = \end{aligned}$$

$$= \frac{y_2(f(\mathbf{x}_1) - f(\mathbf{x}_2)) - \alpha_1 y_1 y_2 K_{11} - \alpha_2 K_{12} + \alpha_1 y_1 y_2 K_{12} + \alpha_2 K_{22}}{K_{11} + K_{22} - 2K_{12}} + \\ + \frac{y_1 y_2 (K_{11} - K_{12}) \alpha_1 + (K_{11} - K_{12}) \alpha_2 + 1 - y_1 y_2}{K_{11} + K_{22} - 2K_{12}},$$

ahol az rövidség kedvéért a „régí” jelzöt elhagytuk. α_2 együttthatóit összegyűjtve végül

$$\alpha_2^{\text{új}} = \alpha_2^{\text{régí}} + \frac{y_2(f_{\text{régí}}(\mathbf{x}_1) - f_{\text{régí}}(\mathbf{x}_2) - y_1 + y_2)}{K_{11} + K_{22} - 2K_{12}}$$

adódik. Természetesen arra nincs garancia, hogy a megkapott $\alpha_2^{\text{új}}$ és $\alpha_1^{\text{új}}$ értékek benne vannak az előírt $[0, C]$ intervallumban, ebben az esetben vágni kell, azaz 0-val vagy C -vel egyenlővé kell tenni. A vágás konkrét menetét korábban elemeztük. Továbbá minden egyes lépésben használni kell az addig érvényben lévő f döntésfüggvényt, ezért mindig meg kell határozni b aktuális értékét is.

Egy lehetséges SMO algoritmus tehát a következő.

1. Kiválasztjuk α két koordinátáját, például α_1 -et és α_2 -t, melyek nem teljesítik a Kuhn–Tucker-feltételt.

2. Legyen

$$L = \max\{0, \alpha_2^{\text{régí}} - \alpha_1^{\text{régí}}\}, \quad H = \min\{C, C + \alpha_2^{\text{régí}} - \alpha_1^{\text{régí}}\}, \quad \text{ha } y_1 \neq y_2,$$

$$L = \max\{0, \alpha_1^{\text{régí}} + \alpha_2^{\text{régí}} - C\}, \quad H = \min\{C, \alpha_1^{\text{régí}} + \alpha_2^{\text{régí}}\}, \quad \text{ha } y_1 = y_2.$$

3. Legyen

$$\alpha_2^{\text{új}} = \alpha_2^{\text{régí}} + \frac{y_2(f_{\text{régí}}(\mathbf{x}_1) - f_{\text{régí}}(\mathbf{x}_2) - y_1 + y_2)}{K_{11} + K_{22} - 2K_{12}}. \quad (5.52)$$

4. A vágás:

$$\alpha_2^{\text{új}} = \begin{cases} H, & \text{ha } \alpha_2^{\text{új}} \geq H, \\ \alpha_2^{\text{új}}, & \text{ha } L < \alpha_2^{\text{új}} < H, \\ L, & \text{ha } \alpha_2^{\text{új}} \leq L. \end{cases}$$

5. Legyen

$$\alpha_1^{\text{új}} = \alpha_1^{\text{régí}} + y_1 y_2 (\alpha_2^{\text{régí}} - \alpha_2^{\text{új}}). \quad (5.53)$$

6. Minden lépés után újra kell számolni b_0 értékét olyan módon, hogy a Kuhn–Tucker-feltételek teljesüljenek ([37]).

Ha α_1 nincs a vágási határon, akkor legyen:

$$b^{\text{új}} = -f_{\text{rég}}(\mathbf{x}_1) + y_1 - y_1(\alpha_1^{\text{új}} - \alpha_1^{\text{rég}})K_{11} - y_2(\alpha_2^{\text{új}} - \alpha_2^{\text{rég}})K_{12} + b^{\text{rég}}. \quad (5.54)$$

Ha α_2 nincs a vágási határon, akkor legyen:

$$b^{\text{új}} = -f_{\text{rég}}(\mathbf{x}_2) + y_2 - y_1(\alpha_1^{\text{új}} - \alpha_1^{\text{rég}})K_{21} - y_2(\alpha_2^{\text{új}} - \alpha_2^{\text{rég}})K_{22} + b^{\text{rég}}. \quad (5.55)$$

Ha sem α_1 , sem α_2 nincs a vágási határon akkor a két fenti $b^{\text{új}}$ érték egybeesik. Ha mind α_1 , mind α_2 a vágási határon van, akkor legyen b új értéke a fenti két $b^{\text{új}}$ számtani közepe.

7. Kiválasztjuk α két újabb koordinátáját, melyek nem teljesítik a Kuhn–Tucker-feltételeket. A fenti algoritmust végrehajtjuk ezekre.
8. Az eljárást addig folytatjuk, ameddig α összes koordinátájára nem teljesülnek a feltételek.

Léteznek heurisztikák arra vonatkozóan, hogy α melyik két koordinátáját vonjuk be a folyamatba. Az SMO algoritmus előnyeiről lásd Platt [37].

5.3. SVM regressziószámításra

5.3.1. Veszteségfüggvények

A matematika klasszikus területein elterjedt a négyzetes távolság használata. Ez jelenik meg többek között az L_2 és az l_2 terek esetén, a numerikus matematika legkisebb négyzetes módszerében, a valószínűségi számításban pedig a szórásnégyzet definíciójában. Ennek oka a négyzet függvény jó tulajdonságaiiban rejlik. A statisztikában a klasszikus regresszióanalízis is ezt használja. Ennek alakja

$$(y - f(\mathbf{x}, \alpha))^2,$$

ahol az \mathbf{x} inputhoz y output tartozik. Azonban a robusztus statisztikai módszerek más veszteségfüggvényeken alapulnak. Az SVM regresszió esetén négyzetes veszteségfüggvény helyett a robusztus statisztikában használt veszteségfüggvényeket alkalmazzuk. Ezek közül tekintsük az ε -inszenzitív veszteségfüggvényt. Azaz az ε -nál kisebb értékű eltérésekre nem vagyunk „érzékenyek”. Az ε -nál nagyobb mértékű eltéréseket pedig nem négyzetesen,

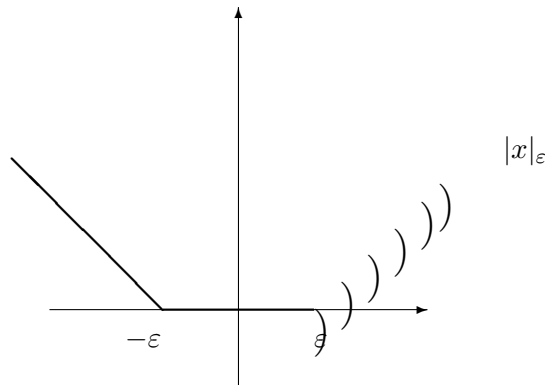
hanem lineárisan, pontosabban szólva az eltéréstől ε -nal kisebb mértékben vesszük figyelembe. Azaz

$$L_\varepsilon(y - f(\mathbf{x}, \boldsymbol{\alpha})) = |y - f(\mathbf{x}, \boldsymbol{\alpha})|_\varepsilon$$

a lineáris ε -inszenzitív veszteségfüggvény, ahol

$$|y - f(\mathbf{x}, \boldsymbol{\alpha})|_\varepsilon = \begin{cases} 0, & \text{ha } |f(\mathbf{x}, \boldsymbol{\alpha}) - y| \leq \varepsilon, \\ |f(\mathbf{x}, \boldsymbol{\alpha}) - y| - \varepsilon, & \text{ha } |f(\mathbf{x}, \boldsymbol{\alpha}) - y| > \varepsilon, \end{cases}$$

lásd 5.6. ábra.



5.6. ábra. Az ε -inszenzitív veszteségfüggvény

Ebben a részben csak ezt a veszteségfüggvényt fogjuk tekinteni. Léteznek és használnak még más típusú veszteségfüggvényeket is.

5.3.2. A lineáris regresszió

Induljunk ki a legegyszerűbb feladatból, a lineáris regresszióból. Ez egyváltozós esetben egyenes illesztését jelenti a mérési pontokra. d -dimenziós esetben pedig az alábbi alakú függvény illesztését:

$$f(\mathbf{x}, \boldsymbol{\psi}) = \sum_{i=1}^d \psi_i x^i + b = \langle \boldsymbol{\psi}, \mathbf{x} \rangle + b,$$

ahol $\mathbf{x} = (x^1, \dots, x^d)^\top \in \mathbb{R}^d$, $\boldsymbol{\psi} = (\psi_1, \dots, \psi_d)^\top \in \mathbb{R}^d$.

Legyenek $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ a megfigyelések (adatok, tanító pontok), ahol $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, \dots, N$. Ekkor a tapasztalati rizikó:

$$R_{\text{emp}}(\boldsymbol{\psi}, b) = \frac{1}{N} \sum_{i=1}^N L_{\varepsilon_i}(y_i - \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle - b).$$

Ezt kellene minimalizálni.

Ennek megoldására Vapnik [49] a következőt javasolta. Vezessük be a nem-negatív $\{\xi_i\}_{i=1}^N$ és $\{\xi_i^*\}_{i=1}^N$ segédváltozókat. Ezek segítségével mérjük az eltéréseket pozitív, illetve negatív irányban, azaz tekintsük az alábbi feltételeket.

$$\begin{aligned} y_i - \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle - b &\leq \varepsilon_i + \xi_i, & i = 1, \dots, N, \\ \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b - y_i &\leq \varepsilon_i + \xi_i^*, & i = 1, \dots, N, \\ \xi_i &\geq 0, & i = 1, \dots, N, \\ \xi_i^* &\geq 0, & i = 1, \dots, N. \end{aligned} \tag{5.56}$$

Belátható, hogy az $R_{\text{emp}}(\boldsymbol{\psi}, b)$ függvény $\boldsymbol{\psi}$ és b szerinti minimuma ugyanott lesz, mint a $\sum_{i=1}^N (\xi_i + \xi_i^*)$ függvénynek az (5.56) feltételek mellett a ξ_i , ξ_i^* , $\boldsymbol{\psi}$ és b változók szerint vett minimuma.

A fenti probléma kezelhetővé tételére Vapnik a következőt javasolta. Minimalizáljuk az (5.56) feltételek esetén a

$$\Phi(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = C \left(\sum_{i=1}^N (\xi_i + \xi_i^*) \right) + \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle$$

függvényt.

Ez az ismert konvex optimalizálási probléma, amire alkalmazhatjuk a Kuhn–Tucker-tételt. Vezessük be az α_i , α_i^* , β_i , β_i^* nem-negatív multiplikátorokat. Ezekkel a Lagrange-függvény:

$$\begin{aligned} L(\boldsymbol{\psi}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*) = \\ = C \sum_i (\xi_i + \xi_i^*) + \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle + \sum_i \alpha_i (y_i - \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle - b - \varepsilon_i - \xi_i) + \\ + \sum_i \alpha_i^* (-y_i + \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b - \varepsilon_i - \xi_i^*) - \sum_i \beta_i \xi_i - \sum_i \beta_i^* \xi_i^*. \end{aligned}$$

Ennek a függvénynek kell a nyeregpontját meghatározni, azaz $\boldsymbol{\psi}$, b , ξ_i , ξ_i^* szerint minimalizálni, míg α_i , α_i^* , β_i , β_i^* szerint maximalizálni. A minimum szükséges feltétele a deriváltak eltűnése.

$$\mathbf{0} = \frac{\partial L}{\partial \boldsymbol{\psi}} = \boldsymbol{\psi} - \sum_i (\alpha_i \mathbf{x}_i - \alpha_i^* \mathbf{x}_i), \tag{5.57}$$

$$0 = \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i, \tag{5.58}$$

$$0 = \frac{\partial L}{\partial \xi_i^*} = C - \alpha_i^* - \beta_i^*, \quad (5.59)$$

$$0 = \frac{\partial L}{\partial b} = \sum_i (-\alpha_i + \alpha_i^*). \quad (5.60)$$

Alkalmazva az (5.57) egyenlőséget: $\psi = \sum_i (\alpha_i \mathbf{x}_i - \alpha_i^* \mathbf{x}_i)$. Látjuk, hogy (5.58)–(5.60) miatt L képletében sok tag kiesik, kapjuk

$$L(\psi, b, \xi, \xi^*, \alpha, \alpha^*, \beta, \beta^*) = \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) - \sum_{i=1}^N \varepsilon_i (\alpha_i + \alpha_i^*) - \frac{1}{2} \langle \psi, \psi \rangle.$$

Ebből pedig, a kieső változókat nem jelölve, és L helyett W -t írva

$$\begin{aligned} W(\alpha, \alpha^*) &= \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) - \sum_{i=1}^N \varepsilon_i (\alpha_i + \alpha_i^*) - \\ &\quad - \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \end{aligned} \quad (5.61)$$

a maximalizálandó függvény. Tehát maximalizálandó az (5.61) képletbeli W kvadratikus függvény a

$$\begin{aligned} \sum_{i=1}^N (\alpha_i - \alpha_i^*) &= 0 \\ 0 \leq \alpha_i &\leq C, \quad i = 1, \dots, N, \\ 0 \leq \alpha_i^* &\leq C, \quad i = 1, \dots, N, \end{aligned} \quad (5.62)$$

feltételek mellett. Ezeket a feltételeket $\alpha_i, \alpha_i^*, \beta_i, \beta_i^*$ nem-negativitásából az (5.58)–(5.60) egyenlőségek figyelembe vételével kaptuk. Ezen kvadratikus programozási feladat megoldásával kapjuk az optimális α^0 és α^{*0} értékeket. Ezekkel ψ fenti kifejezéséből:

$$\psi^0 = \sum_{i=1}^N (\alpha_i^0 - \alpha_i^{*0}) \mathbf{x}_i.$$

Az optimális b értéket $R_{\text{emp}}(\psi^0, b)$ b -szerinti minimalizálásával kapjuk. Végül a regressziós hipersík a következő

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i^0 - \alpha_i^{*0}) \langle \mathbf{x}, \mathbf{x}_i \rangle + b^0.$$

5.3.3. Nem-lineáris függvények közelítése

Ha nem lineáris függvényt akarunk közelíteni, akkor a feladatot nem-lineáris transzformációval megpróbáljuk visszavezetni lineárisra. Viszont, hasonlóan az osztályozáshoz, most is csak a szereplő vektorok belső szorzata lép fel, nem maguk a vektorok. Tehát transzformáció és belső szorzat helyett egyből használhatunk magfüggvényeket. Ezzel megoldhatjuk a nem-lineáris regresszió (függvényközelítés) feladatát.

Legyen tehát K a magfüggvény. Ekkor (5.61) alapján a célfüggvény

$$W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = \sum_{i=1}^N y_i(\alpha_i - \alpha_i^*) - \sum_{i=1}^N \varepsilon_i(\alpha_i + \alpha_i^*) - \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j).$$

Ezt kell maximalizálni az (5.62) feltételek mellett. A maximumhelyet jelölje $\alpha_i^0, \alpha_i^{*0}, i = 1, 2, \dots, N$. Az illesztett függvény alakja pedig

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i^0 - \alpha_i^{*0}) K(\mathbf{x}, \mathbf{x}_i) + b^0$$

lesz.

Megjegyezzük, hogy az SMO eljárás alkalmas változata működik SVM regresszióra is.

Magfüggvények előállítása

Függvény illesztése (azaz regressziószámítás) esetén alkalmazhatjuk azokat a magfüggvényeket, amelyeket az osztályozás esetén már megismertünk. Mivel ezek között van az RBF esetén használt, valamint a szigmoid függvényt tartalmazó, így megállapíthatjuk, hogy az SVM segítségével megoldhatunk olyan feladatokat, amelyekre korábban az RBF-et és az MLP-t alkalmaztuk. Hamarosan látni fogjuk, hogy az SVM segítségével megoldható számos, a matematikai analízis, a numerikus matematika, a számítógépes grafika és a statisztika tárgykörébe eső feladat is.

Spline interpolációt előállító magfüggvények

A spline a numerikus matematikában gyakran használt közelítés. A spline szakaszonként polinom, folytonos (bizonyos esetekben a deriváltjai is folytonosak valamilyen rendig) és tipikusan átmegy a kijelölt pontokon.

Legyen

$$a = t_0 < t_1 < t_2 < \dots < t_m < t_{m+1} = b$$

az $[a, b]$ intervallum egy beosztása. Az f függvényt d -edfokú spline-nak nevezzük, ha f az $[a, b]$ intervallumon $d - 1$ -szer folytonosan differenciálható, és minden $[t_i, t_{i+1}]$ intervallumon (legfeljebb) d -edfokú polinom. Legyen

$$(x - t_i)_+^d = \begin{cases} (x - t_i)^d, & \text{ha } x > t_i, \\ 0, & \text{egyébként.} \end{cases}$$

Világos, hogy az $(x - t_i)_+^d$ függvény d -edfokú spline. Egy legfeljebb d -edfokú polinom is d -edfokú spline. Másrészt spline függvények lineáris kombinációja is spline. Tehát

$$f(x) = \sum_{r=0}^d a_r x^r + \sum_{i=1}^m b_i (x - t_i)_+^d \quad (5.63)$$

d -edfokú spline.

A d -edfokú spline approximációt tehát (5.63) alakban keressük. Az x számhoz rendeljük az alábbi $(m + d + 1)$ -dimenziós vektort

$$x \mapsto \mathbf{u} = (1, x, \dots, x^d, (x - t_1)_+^d, \dots, (x - t_m)_+^d).$$

Tehát az f spline függvény az

$$f(x) = \langle \mathbf{a}, \mathbf{u} \rangle$$

belső szorzat alakban áll elő, ahol $\mathbf{a} = (a_0, \dots, a_d, b_1, \dots, b_m)$. Tehát az alábbi magfüggvény spline interpolációt állít elő:

$$K(x, x_t) = \langle \mathbf{u}, \mathbf{u}_t \rangle = \sum_{r=0}^d x^r x_t^r + \sum_{i=1}^m (x - t_i)_+^d (x_t - t_i)_+^d.$$

Megjegyezzük, hogy Vapnik [49] ekvidisztáns alappontok esetén ajánlja a spline approximációt:

$$\{t_1, \dots, t_m\}, \quad t_i = \frac{ia}{m}, \quad i = 1, \dots, m,$$

a $[0, a]$ beosztása.

B-spline interpolációt előállító magfüggvények

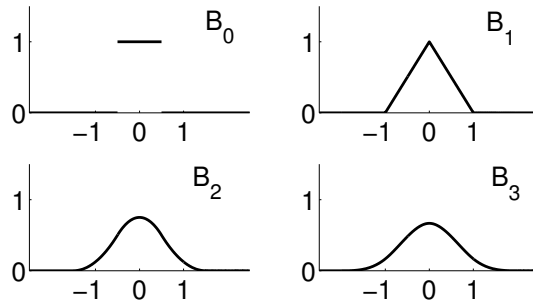
A számítógépes grafikában közismert, görbék előállítására használt módszer a B-spline interpoláció. A

$$B_0(u) = \begin{cases} 1, & \text{ha } |u| \leq 0.5, \\ 0, & \text{ha } |u| > 0.5 \end{cases}$$

függvényt B_0 spline-nak (vagy 0-adrendű spline-nak) nevezzük. A magasabb rendű B-spline függvényeket rekurzíven definiáljuk. A d -edrendű B_d spline függvény a B_{d-1} spline és a B_0 spline konvolúciójaként adott:

$$B_d(u) = \int_{-\infty}^{\infty} B_{d-1}(u-t)B_0(t) dt.$$

Ezek alapján látszik, hogy a B_d függvény $d-1$ számú, független, a $[-0.5, 0.5]$ intervallumon egyenletes eloszlású valószínűségi változó összegének sűrűségfüggvénye. Az első néhány B_d függvény az 5.7. ábrán látható.



5.7. ábra. A B_d spline függvény, $d = 0, 1, 2, 3$

A B_d függvényre $d \rightarrow \infty$ esetén ismeretes az alábbi közelítés (mely már kis d -k esetén is elég jó):

$$B_d(u) \approx \sqrt{\frac{6}{\pi(d+1)}} \exp \left\{ -\frac{6u^2}{d+1} \right\}. \quad (5.64)$$

Ez a formula a centrális határeloszlás-tételből adódik, a közelítő függvény egy alkalmas normális sűrűségfüggvény.

5.5. Tétel.

$$B_d(u) = \sum_{r=0}^{d+1} \frac{(-1)^r}{d!} \binom{d+1}{r} \left(u + \frac{d+1}{2} - r \right)_+^d. \quad (5.65)$$

B_d spline függvényekkel a függvényközelítés

$$f(x, \beta) = \sum_{i=1}^n \beta_i B_d(x - t_i)$$

alakú, ahol t_i , $i = 1, \dots, n$, a sorfejtés csomópontjai. Ez alapján a B-spline interpolációhoz alkalmas magfüggvény:

$$K(x, x_i) = \sum_{k=1}^n B_d(x - t_k) B_d(x_i - t_k).$$

Fourier-sort előállító magfüggvények

A matematikai analízisben jól ismert a függvények Fourier-sorba fejtése, azaz a szinusz és koszinusz függvényekkel való kifejezése. Rendeljük hozzá az x valós számhoz az alábbi $(2n + 1)$ -dimenziós \mathbf{u} vektort:

$$\mathbf{u} = \left(\frac{1}{\sqrt{2}}, \sin x, \dots, \sin nx, \cos x, \dots, \cos nx \right).$$

Rögzített $x \in \mathbb{R}$ esetén a Fourier-sor (egy szelete) tekinthető $(2n + 1)$ -dimenziós térbeli alábbi belső szorzatnak:

$$f(x) = \langle \mathbf{a}, \mathbf{u} \rangle = \frac{a_0}{\sqrt{2}} + \sum_{k=1}^n (a_k \sin kx + b_k \cos kx).$$

Ebben a térben a belső szorzat:

$$\begin{aligned} K_n(x, x_i) &= \frac{1}{2} + \sum_{k=1}^n (\sin kx \sin kx_i + \cos kx \cos kx_i) = \\ &= \frac{1}{2} + \sum_{k=1}^n \cos k(x - x_i) = \\ &= \frac{\sin\left(\frac{2n+1}{2}(x - x_i)\right)}{\sin\left(\frac{(x-x_i)}{2}\right)}. \end{aligned}$$

Itt a trigonometrikus sorok elméletében használt Dirichlet-féle magfüggvényhez jutottunk. Ezzel a magfüggvénnyel az SVM az alábbi alakú függvényközelítést adja:

$$f(x, \boldsymbol{\beta}) = \sum_{i=1}^N \beta_i K_n(x, x_i).$$

Összefoglaljuk az eddig ismertetett magfüggvényeket.

Közelítés típusa	Alkalmazott magfüggvény
Lineáris regresszió	$\langle \mathbf{x}, \mathbf{x}_i \rangle$
Polinomiális	$K(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + 1)^p$
RBF	$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma^2} \ \mathbf{x} - \mathbf{x}_i\ ^2\right)$
Kétrétegű perceptron	$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \langle \mathbf{x}, \mathbf{x}_i \rangle + \beta_1)$
Fourier-sor	$\frac{\sin\left(\frac{2n+1}{2}(x-x_i)\right)}{\sin\left(\frac{(x-x_i)}{2}\right)}$
spline interpoláció	$\sum_{r=0}^d x^r x_i^r + \sum_{j=1}^m (x - t_j)_+^d (x_i - t_j)_+^d$
B-spline interpoláció	$\sum_{k=1}^n B_d(x - t_k) B_d(x_i - t_k)$

5.1. táblázat. Függvényközelítésre alkalmazott magfüggvények

5.3.4. Többváltozós függvények közelítése

Ismeretes, hogy egyváltozós magfüggvények szorzata többváltozós magfüggvény lesz. Legyen tehát a közelítendő függvény d -változós. Ekkor

$$K(\mathbf{x}, \mathbf{y}) = \prod_{k=1}^d K_k(x^k, y^k)$$

alkalmas magfüggvény, ahol $\mathbf{x} = (x^1, \dots, x^d)$, $\mathbf{y} = (y^1, \dots, y^d)$, K_k pedig egyváltozós magfüggvény minden k -ra.

5.4. Numerikus példák

5.6. Példa. Két halmaz szétválasztása SVM segítségével megvalósítható. Legyen az egyik halmaz 100 elemű minta kétdimenziós, $(0, 0)^\top$ várható érték vektorú és

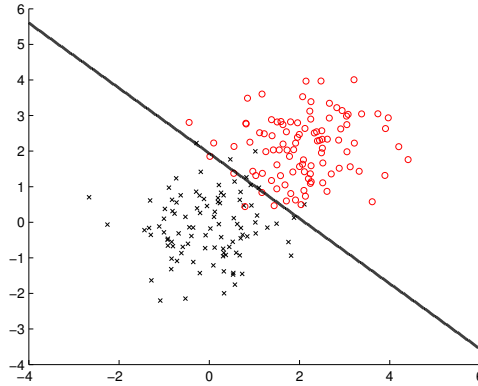
$$\begin{pmatrix} 0.8 & 0 \\ 0 & 0.8 \end{pmatrix}$$

szórás-mátrixú normális eloszlásból. A másik halmaz pedig ugyanilyen minta, azzal a különbséggel, hogy ott a várható érték vektor $(2, 2)^\top$.

Az ábrákon \times jelöli az első, \circ pedig a második halmaz elemeit. A megkonstruált elválasztó görbék különböznek.

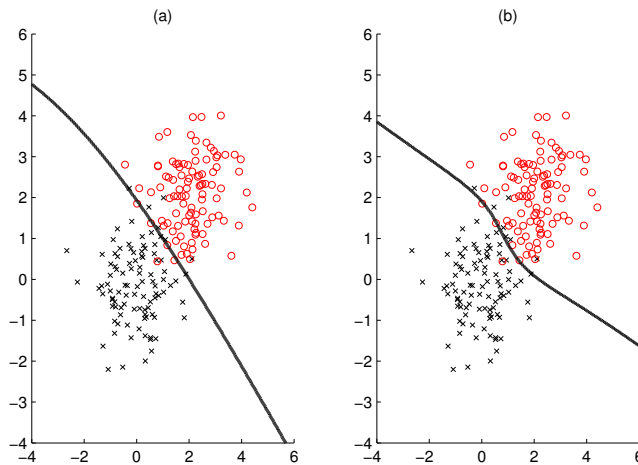
Az 5.8. ábrán lineáris magfüggvényű és legkisebb négyzetes módszerrel megalkotott SVM szeparálása szerepel. Mivel az aktuális problémában az

elméletileg optimális elválasztás hipersíkkal valósítható meg, így az SVM meg is találta az optimális hipersíkot.



5.8. ábra. Két normális eloszlás szeparálása lineáris SVM-mel

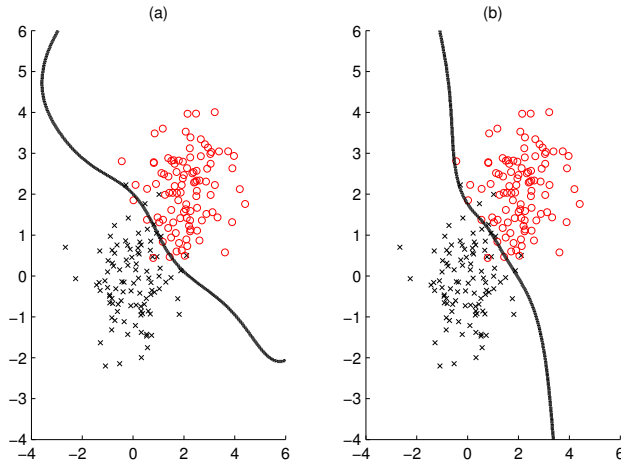
Az 5.9. ábra (a) részén kvadratikus magfüggvényű és kvadratikus programozással betanított SVM szeparálása szerepel. Az 5.9. ábra (b) részén pedig harmadfokú (polinom) magfüggvényű és SMO módszerrel betanított SVM szeparálása szerepel.



5.9. ábra. Két normális eloszlás szeparálása (a): másodfokú, ill. (b): harmadfokú SVM-mel

Az 5.10. ábra (a) részén Gauss-féle radiális bázis függvény magfüggvényű és kvadratikus programozással betanított SVM szeparálása szerepel.

Az 5.10. ábra (b) részén pedig MLP magfüggvényű (pontosabban szólva tangens hiperbolikus függvényt használó) és SMO módszerrel betanított SVM szeparálása szerepel.



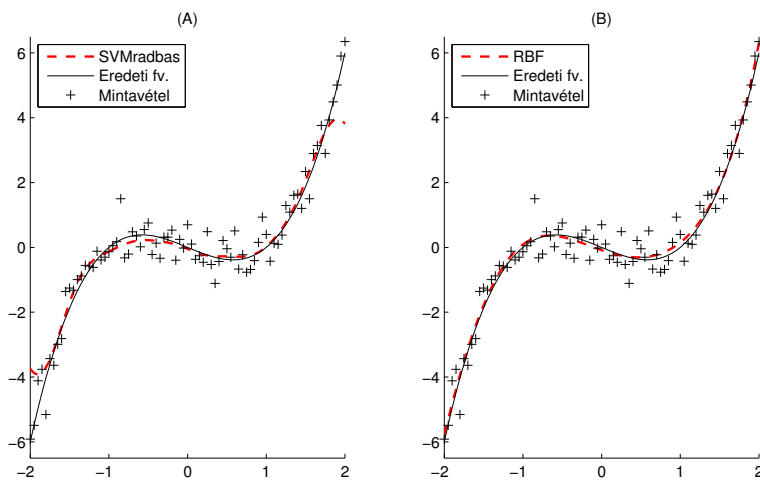
5.10. ábra. Két normális eloszlás szeparálása (a): RBF, ill. (b): MLP SVM-mel

Megállapíthatjuk, hogy mind a négy utóbbi szeparálás jó a „kritikus részen”, azaz ott, ahol a két minta egymásba átnyúlik. Viszont ott, ahova aktuálisan nem esnek mintapontok, már a különböző magfüggvények a saját speciális alakjuknak megfelelően folytatják a szétválasztást, és nyilvánvalóan eltérnek az optimális szeparáló görbétől.

5.7. Példa. Az $f(x) = x^3 - x$ függvényt a $[-2, 2]$ intervallumon figyeljük meg, a megfigyelések -2 és 2 között 81 db ekvidisztáns alappontban történnek, de a függvényértékeket csak hibával terheltén tudjuk mérni. A mérési hiba 0 várható értékű, 0.5 szórású (független) normális eloszlású. Viszont közelíteni a mérési hibák nélküli függvényt kell!

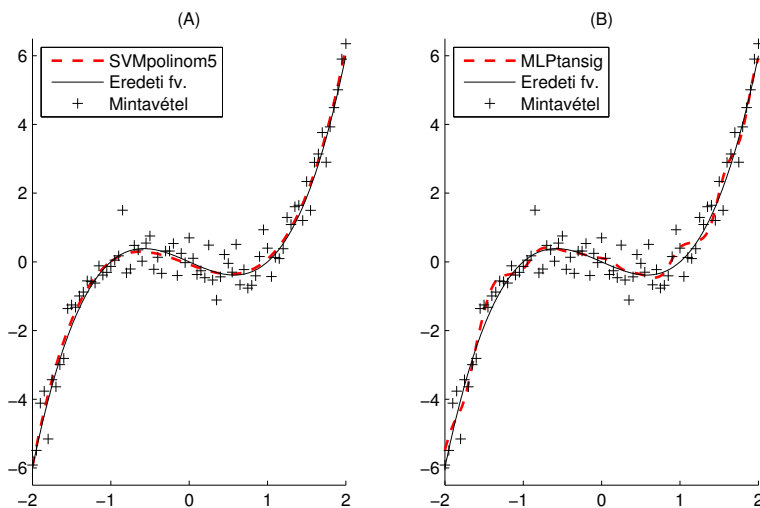
Az 5.11. ábra (A) részén látható az SVM által adott közelítés. Gauss-féle (azaz RBF) magfüggvényt használtunk.

Az 5.11. ábra (B) részén a hagyományos RBF által adott közelítést ábráztuk. Az RBF-ben Gauss-féle magfüggvényt használtunk.



5.11. ábra. Az $f(x) = x^3 - x$ függvény közelítése mérési hibák esetén. (A) SVM Gauss-magfüggvénnyel; (B) RBF Gauss-magfüggvénnyel

Az 5.12. ábra (A) részén látható olyan SVM által adott közelítés, amelyben ötödfokú polinom a magfüggvény.



5.12. ábra. Az $f(x) = x^3 - x$ függvény közelítése mérési hibák esetén. (A) SVM ötödfokú polinom magfüggvénnyel; (B) egy rejtett rétegű MLP

Az 5.12. ábra (B) részén hagyományos MLP által adott közelítés szerepel. Itt egy rejtett réteg van 12 neuronnal tangens hiperbolikus transzfer

függvénnyel. A kimeneti rétegben egyetlen neuron található lineáris transzfer függvénnyel. A tanítás konjugált gradiens módszerrel történt.

5.8. Példa (Zajszűrés SVM-regresszióval). A képkötő berendezések leképezési folyamata nem tökéletes, minden esetben hibával terhelt. A képkötés során jelentkező hibákat forrásuk szerint számos csoportba sorolhatjuk.

- *Geometriai hibák.* Fizikai tulajdonságai miatt egy lencserendszer leképezése radiálisan változik. A lencse optikai tengelyétől távol eső objektumok képe nem a fókusz síkban áll elő, így élettenség és a geometriai struktúrák torzulása jelentkezhet (például egyenes struktúrák meggörbülhetnek).
- *Szisztematikus hibákról* beszélünk, ha egy képhiba a leképező rendszerrel készült minden képen ugyanott jelentkezik. Előfordulhat például, ha a leképező rendszer valamely lencséje megsérül (például karcolás) vagy a CCD-chip valamely fényérzékelő eleme meghibásodik. Utóbbi esetben a képek egy adott (i, j) indexű pixele minden esetben fekete.
- *Kvantálási hibák.* A látható fény elektromágneses hullám, melynek lehetséges intenzitásai folytonosnak tekinthetők. A digitális képeken csak diszkrét intenzitásokat, színeket ábrázolhatunk, így a mért intenzitásértékeket kvantálni kell. A képtérrel érkező különböző erősségű fényimpulzusokat jellemzően 256 diszkrét értékre képezzük le, ami jelentős információvesztést jelent. Ha képet készítünk egy olyan képtérrel, amely sötét és világos régiókat is tartalmaz, a sötét és világos régiók részletgazdagsága kisebb, mint abban az esetben, amikor csak a sötét vagy csak a világos régióról készítünk képet.
- *Statisztikus hibák.* A képet feldolgozó félvezető elektronika működéséből a termikus zaj nem küszöbölhető ki. A termikus zaj elsősorban a gyenge elektromos impulzusok erősítésénél fejti ki hatását, így erősítőzajnak is nevezzük. Eredményeként, ha egy homogén színű és megvilágítású felületről képet készítünk, a kép intenzitásértékei különböznek.

Egy képkötő rendszer hibáinak utófeldolgozással történő eltávolítása a digitális képfeldolgozás alapvető feladatának tekinthető. Ahhoz, hogy a hibákat, zajokat megfelelően el tudjuk távolítani, kellő mennyiségű információval kell rendelkezünk a zaj jellegéről. Az egyik lehetséges megközelítésben a képkötő rendszert fekete doboznak tekintjük, és feltételezzük, hogy rendelkezünk a képkötő rendszerrel készült zajjal terhelt képekkel, valamint a képek tartalmának egzakt leírásával, vagy olyan képekkel, amelyek

ugyanazt a szcénát ábrázolják, azonban elfogadhatóan kevés hibával operáló képpalkotó rendszerrel készültek. Az ilyen képek készítésére, s a képek alapján a megfelelő korrekciós módszerek kidolgozására kalibrációként is szokás hivatkozni.

Az etalonnak tekinthető és zajjal terhelt képek alapján a Support Vector Regresszió (SVR) eszközeivel egyszerűen dolgozhatunk ki a hibákat korrigáló eljárást.

Az egyszerűség kedvéért szűrkeskálás intenzitásképekkel dolgozunk és feltételezzük, hogy a kép intenzitásai a $[0, 1]$ tartományba vannak normálva, azaz egy S sorból és O oszlopból álló K képet $K \in [0, 1]^{S \times O}$ módon ábrázolunk. Jelen szakaszban csak a statisztikus hibák javítására alkalmas modellt készítünk. Megjegyezzük azonban, hogy alkalmas módosításokkal a módszer alkalmas kvantálási, szisztematikus és geometriai hibák korrigálására is.

Tegyük fel, hogy rendelkezünk N darab etalon képpel $(K^i, i = 1, \dots, N)$ és rendre ugyanazon szcénát ábrázoló zajjal terhelt képpel $(\hat{K}^i, i = 1, \dots, N)$. Utóbbiakat a vizsgált képpalkotó rendszerrel készült képeknek tekintjük.

A zajszűrést megvalósító modellt előállításának, és zajjal terhelt képek szűrésének lépéseit az alábbi pontokban foglaljuk össze.

1. Feltételezzük, hogy a zajjal terhelt képeken egy (i, j) koordinátájú pixel valódi intenzitása meghatározható, de legalábbis jól becsülhető annak lokális környezete alapján.
2. Az SVR alkalmazásához tanító halmazt készítünk úgy, hogy a rendelkezésünkre álló zajjal terhelt képek minden pixeléhez egy tanító vektort rendelünk.
3. Minden tanító vektor $v \in [0, 1]^{25} \times [0, 1]$ formában áll elő. Legyen v a \hat{K} képen az (i, j) indexű képponthez rendelt tanító vektor. Ekkor v első 25 koordinátája az (i, j) indexű képpont lokális, 5×5 méretű környezetében található intenzitásokat tartalmazza. A célváltozó, azaz a 26-odik koordináta az etalon K kép (i, j) indexű képpontjának az intenzitása.
4. Elkészítjük a tanító adatbázis alapján a regressziós modellt.
5. Egy új, zajjal terhelt kép szűrése során sorra vesszük annak minden elemét, és az (i, j) indexű pixel feldolgozásakor a pixel lokális 5×5 méretű környezetéből előállított 25 elemű vektor helyén kiértékeljük a regressziós függvényt. A szűrt kép (i, j) pixele az így kapott értéket veszi fel.

A módszert szavakban úgy foglalhatjuk össze, hogy a regressziós függvény „megtanulja”, hogy egy zajos pixel és annak zajos környezete függvényében a pixel milyen eredeti értékkel rendelkezhetett.

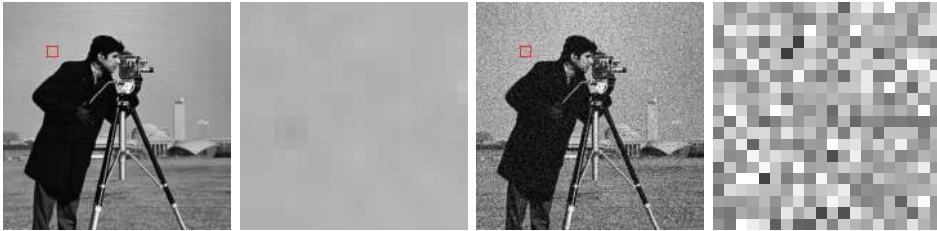
A módszer működését a következő példával demonstráljuk. Négy etalon képpel dolgozunk, melyekhez zajjal terhelt képeket állítunk elő. A hozzáadott zaj két komponensből tevődik össze. Az első komponens erősítőzaj, amely fizikai okokra visszavezethetőleg Gauss-eloszlású additív zajként modellezhető. Másrészt megjelenik egy nem-lineáris „salt-and-pepper” zaj, amely valamilyen valószínűséggel egy pixel értékét 0-ra vagy 1-re állítja. A zajos képek $\hat{K}_{i,j}$ pixele így a következő értéket veszi fel:

$$\hat{K}_{i,j} = SAP(K_{i,j} + \epsilon),$$

ahol ϵ egy $\mathcal{G}(0, 0.15)$ eloszlású véletlen szám, míg a SAP függvényt az alábbi módon definiáljuk:

$$SAP(x) = \begin{cases} 0, & \text{ha } u < 0.15, \\ 1, & \text{ha } u > 0.85, \\ x, & \text{egyébként.} \end{cases}$$

A fenti definícióban u egy $\mathcal{U}(0, 1)$ eloszlású véletlen szám ($\mathcal{U}(0, 1)$ a $(0, 1)$ intervallumon egyenletes eloszlást jelöli). Megjegyezzük, hogy a példában a képekhez adott zaj igen erős zajnak tekinthető (5.13. ábra).



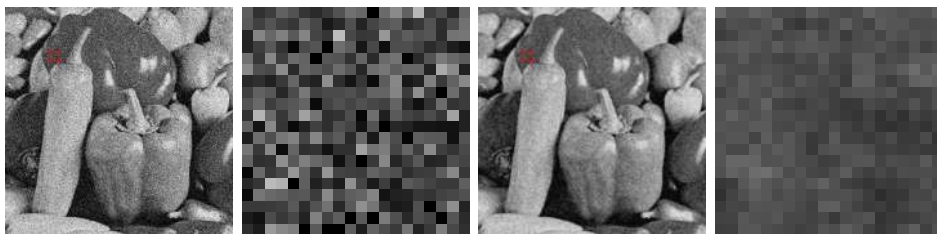
(a) Egy etalon kép (b) Az etalon kép egy részlete (c) A zajjal terhelt kép (d) A zajjal terhelt kép egy részlete

5.13. ábra. Egy etalon kép és annak zajjal terhelt változata

A nem-lineáris SVR-modell megalkotása során ki kell választanunk a használni kívánt magfüggvényt. Választásunk a Gauss-magfüggvényre esett, melynek szórás paraméterét 1-re állítjuk. Kiemeljük, hogy a Gauss-magfüggvény választása nincs kapcsolatban azzal, hogy a zajos képeken megjelenő

erősítőzajt Gauss-eloszlású véletlen számmal modellezzük. A tapasztalat azt mutatja, hogy a Gauss-magfüggvény jól használható tetszőleges SVR problémák megoldására.

A modell előállítását követően vizsgáljuk meg kvalitatívan, hogy működik-e a megoldásunk. Egy újabb képet tekintünk, melyet a korábbiakhoz azonos zajjal terhelünk. A zajszűrés során a kép minden pixelének lokális környezetéből egy 25 elemű vektort állítunk elő, s a korábban kapott regressziós függvény kiértékelésével meghatározzuk a pixelhez rendelő, immár zajmentesnek tekintett értéket. Illusztrációként a 5.14. ábrán látható egy zajos tesztkép, illetve a szűrés eredménye. Ahogy az ábrán látható, a szűrés valóban működik, a zaj mértéke jelentősen csökkent.

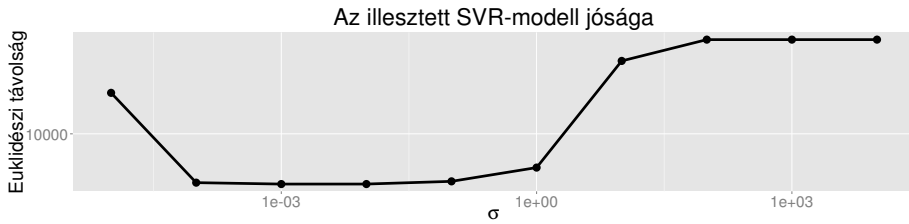


(a) A zajjal terhelt tesztkép (b) A zajjal terhelt tesztkép egy részlete (c) A szűrés eredményeként előállt kép (d) A szűrt kép egy részlete

5.14. ábra. A zajjal terhelt kép és a szűrt kép

A Gauss-magfüggvény szórás paraméterét $\sigma = 1$ módon választottuk meg a regressziós függvény illesztése során. Ez a választás azonban esetleges. Általánosságban elmondható a kernel-módszerekről, s így az SVR-ről is, hogy nem tudhatjuk, mely magfüggvény milyen paraméterekkel hozza a legjobb eredményeket. Ahhoz, hogy elfogadható beállításokat találjunk, több különböző magfüggvényt és azok különböző paraméterezéseit is ki kell próbálnunk. Ezt a folyamatot *modellillesztésnek* nevezzük. Ahhoz, hogy az egyes magfüggvények és paraméterezéseik között kvantitatívan különbséget tehesünk, a szűrt kép jóságát valamilyen mérőszámmal kell jellemeznünk. Esetünkben ez a mérőszám kézenfekvő módon a szűrt kép és a megfelelő etalon kép euklidészi távolsága lehet.

A modellillesztés során elvégezzük Gauss-magfüggvénnyel a regressziós függvény illesztését, $\sigma = 10^k$, $k \in \{-5, -4, \dots, 5\}$ paraméterekkel. A modellek jellemzésére kiszámítjuk a szűrt kép és az etalonkép távolságát, s az eredményeket grafikonon ábrázoljuk (5.15. ábra).



5.15. ábra. Az illesztett SVR-modell jósága. Az olvashatóság kedvéért a tengelyeken logaritmikus skálát használtunk

Ahogy az ábráról leolvasható, az előző szakaszban tett $\sigma = 1$ választásunk nem a lehető legjobb: $\sigma = 0.001$ választással az etalon kép és a szűrt kép euklidészi távolsága kisebb. Másként fogalmazva a zaj eltávolítása hatékonyabb, ha a Gauss-magfüggvény szórás paraméterét 0.001-re állítjuk.

Megjegyezzük, hogy egy valós alkalmazásban több, lehetőleg különböző struktúrákat és mintázatokat tartalmazó tanító képet és több teszt képet kell használnunk az SVR-modell illesztése során. A geometriai és szisztematikus hibák pozíciófüggők, így azok kezeléséhez a 25 elemű sajáttságvektort ki kell egészítenünk a sajáttságvektorhoz tartozó pixel koordinátaival. Kvantálási és színhibák kezeléséhez a pixelek környezetét leíró vektorokat színinformációval egészíthetjük ki.

5.5. Feladatok

1. Igazoljuk az SMO algoritmusban a b aktualizálására vonatkozó képletek alkalmas voltát. Azaz, hogy minden lépés után az újra számolt b_0 értékkel a Kuhn–Tucker-feltételek teljesüljenek.
2. (a) A négyzetes ε -inszenzitív veszteségfüggvényt az alábbi módon definiáljuk.

$$L(y - f(\mathbf{x}, \boldsymbol{\alpha})) = |y - f(\mathbf{x}, \boldsymbol{\alpha})|_{\varepsilon}^2.$$

Ábrázoljuk az $|x|_{\varepsilon}^2$ függvényt!

(b) A Huber-féle veszteségfüggvény az alábbi. Legyen $c > 0$ fix,

$$L(|y - f(\mathbf{x}, \boldsymbol{\alpha})|) = \begin{cases} c|f(\mathbf{x}, \boldsymbol{\alpha}) - y| - \frac{c^2}{2}, & \text{ha } |f(\mathbf{x}, \boldsymbol{\alpha}) - y| > c, \\ \frac{1}{2}|f(\mathbf{x}, \boldsymbol{\alpha}) - y|^2, & \text{ha } |f(\mathbf{x}, \boldsymbol{\alpha}) - y| \leq c. \end{cases}$$

Ábrázoljuk az $L(|x|)$ függvényt!

3. Hasonlítsuk össze a lineáris ε -inszenzitív veszteségfüggvényt, a négyzetes ε -inszenzitív veszteségfüggvényt és a Huber-féle veszteségfüggvényt! Ábrázoljuk ezeket közös koordináta-rendszerben!
4. Oldjuk meg a lineáris függvényközelítés feladatát a négyzetes ε -inszenzitív veszteségfüggvény esetén! Azaz írjuk fel a minimalizálandó veszteségfüggvényt, alkalmazzuk a Kuhn–Tucker-tételt a kapott feltételes minimum problémára, ebből vezessük le a kvadratikus programozási feladatot!
5. Oldjuk meg a lineáris függvényközelítés feladatát a Huber-féle veszteségfüggvény esetén! Azaz írjuk fel a minimalizálandó veszteségfüggvényt, alkalmazzuk a Kuhn–Tucker-tételt a kapott feltételes minimum problémára, ebből vezessük le a kvadratikus programozási feladatot!
6. (a) Lássuk be, hogy egy függvény legfeljebb egyféle módon írható fel (5.63) alakban!
(b) Lássuk be, hogy minden d -edfokú spline felírható (5.63) alakban!
7. Adott pontokra illesszünk elsőfokú spline-t! Ábrázoljuk a kapott eredményt!
8. Igazoljuk az (5.64) közelítést! Ábrázoljuk a B_d spline függvényt és a közelítését!
9. Igazoljuk az 5.5. Tételt!
10. Vizsgáljuk az SVM-mel való szeparálást! Állítsuk valamilyen módon elő a tanító pontok A és a B halmazát, és szeparáljuk SVM-mel! A két halmaz lehet 2- vagy 3-dimenziós, véletlen vagy determinisztikus módon megadott, akár egymásba átnyúló is. Alkalmazzunk különböző magfüggvényű SVM-eket! Ábrázoljuk az eredményt!
11. Hasonlítsuk össze az MLP, az RBF és az SVM szeparálását numerikus példákra!
12. Közelítsük az alábbi függvényeket SVM segítségével!

$$f(x) = x^2, \quad f(x) = \log x, \quad f(x) = \sin x, \quad f(x) = (\sin x)/x,$$

$$f(x, y) = x^2 + y^2, \quad f(x, y) = x^2/y.$$

Válasszunk alkalmas tartományt, ahol fenti függvényeket közelíteni akarjuk! Generáljunk különböző számú és sűrűségű tanító pontokat! Vizsgáljuk külön azt az esetet is, amikor a függvényértéket csak valamilyen véletlen hibával tudjuk megfigyelni (azaz a tanító pontok ilyenek, de a közelítendő függvény a hiba nélküli)! Ábrázoljuk az eredményt!

- 13.** Hasonlítsuk össze az MLP, az RBF és az SVM hálózatok függvény approximálását numerikus példákon!

6. fejezet

Appendix

6.1. Néhány matematikai fogalom

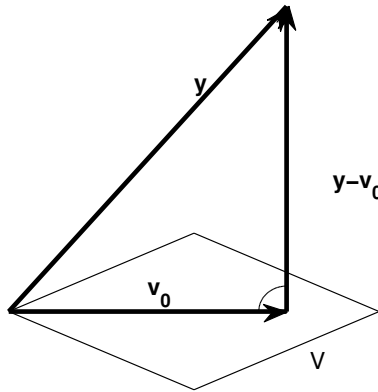
6.1.1. Vektorok és mátrixok

Transzponálás. Az n -dimenziós euklideszi tér (\mathbb{R}^n) vektorait oszlopvektoroknak tekintjük, a $^\top$ segítségével jelölt transzponáltjaik tehát sorvektorok: $\mathbf{y}^\top = (y_1, y_2, \dots, y_n)$.

Belső szorzat és diadikus szorzat. Legyen $\mathbf{x} = (x_1, x_2, \dots, x_n)^\top$ és $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$ két \mathbb{R}^n -beli vektor. Az $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}$ skalár a két vektor belső szorzata (skaláris szorzata), míg az $\mathbf{x}\mathbf{y}^\top$ $n \times n$ méretű mátrix a két vektornak a diadikus szorzata:

$$\mathbf{x}^\top \mathbf{y} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \sum_{i=1}^n x_i y_i,$$
$$\mathbf{x}\mathbf{y}^\top = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \dots & y_n \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \dots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \dots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n y_1 & x_n y_2 & \dots & x_n y_n \end{pmatrix}.$$

Merőleges (ortogonális) vetítés. Legyen \mathbf{y} az n -dimenziós euklideszi tér egy vektora, V pedig egy altér. Ekkor egyértelműen létezik egy $\mathbf{v}_0 \in V$, amely esetén $\mathbf{y} - \mathbf{v}_0$ merőleges V -re (azaz merőleges V minden elemére). \mathbf{v}_0 az \mathbf{y} vektor merőleges vetülete V -re, $\mathbf{y} - \mathbf{v}_0$ pedig a merőleges (ortogonális) komplementere (lásd 6.1 ábrát). \mathbf{v}_0 van \mathbf{y} -hoz a legközelebb a V altérből:



6.1. ábra. Az \mathbf{y} vektor merőleges vetítése a V altérre

$$\min_{\mathbf{v} \in V} \sum_{i=1}^n (y_i - v_i)^2 = \min_{\mathbf{v} \in V} \|\mathbf{y} - \mathbf{v}\|^2 = \|\mathbf{y} - \mathbf{v}_0\|^2.$$

Ez a legkisebb négyzetek elvének az alapja.

Sajátérték, sajátvektor. Legyen A $n \times n$ -es mátrix, $\mathbf{v} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$. Ha

$$A\mathbf{v} = \lambda\mathbf{v}$$

teljesül, és $\mathbf{v} \neq \mathbf{0}$, akkor \mathbf{v} -t az A sajátvektorának, λ -t pedig sajátértékének nevezzük.

Szimmetrikus mátrixok spektrálfelbontása. Az A valós elemű szimmetrikus mátrix sajátértékei valósak, a különböző sajátértékekhez tartozó sajátvektorok ortogonálisak. Van a térnek az A sajátvektoraiból álló ortonormált bázisa. Ennek alapján az A mátrix spektrálfelbontása:

$$A = V\Lambda V^\top, \quad (6.1)$$

ahol a V ortogonális mátrix oszlopai az A ortonormált sajátvektorai, a Λ diagonális mátrix főátlójában pedig az A sajátértékei állnak.

6.1.2. Differenciálszámítás

A Taylor-formula. Legyen f az x és x_0 közötti nyílt intervallumban n -szer differenciálható, az x és x_0 közötti zárt intervallumban legyen $f^{(n-1)}$ folytonos. Ekkor

$$f(x) = f(x_0) + \frac{(x - x_0)}{1!} f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \cdots +$$

$$+ \frac{(x - x_0)^{n-1}}{(n-1)!} f^{(n-1)}(x_0) + \frac{(x - x_0)^n}{n!} f^{(n)}(z),$$

ahol z az x és az x_0 pontok közötti nyílt intervallumban fekvő valamely pont.

Az e^x függvényre a Taylor-formula.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{n-1}}{(n-1)!} + \frac{x^n}{n!} e^{\vartheta x}, \quad \text{ahol } \vartheta \in (0, 1).$$

Az e^x Taylor-sora.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Az $\ln x$ függvényre a Taylor-formula.

$$\ln(1+x) = \frac{x}{1} - \frac{x^2}{2} + \frac{x^3}{3} - \cdots + (-1)^n \frac{x^{n-1}}{n-1} + (-1)^{n+1} \frac{x^n}{n(1+\vartheta x)^n},$$

ahol $\vartheta \in (0, 1)$.

Kétváltozós függvény szélsőértékei. Ha az $f(x, y)$ függvénynek az (x_0, y_0) pontban szélsőértéke van (és léteznek a parciális deriváltjai), akkor

$$\frac{\partial f(x_0, y_0)}{\partial x} = 0, \quad \frac{\partial f(x_0, y_0)}{\partial y} = 0. \quad (6.2)$$

Legyen továbbá

$$\Delta = \frac{\partial^2 f(x_0, y_0)}{\partial x^2} \frac{\partial^2 f(x_0, y_0)}{\partial y^2} - \left[\frac{\partial^2 f(x_0, y_0)}{\partial x \partial y} \right]^2$$

(és legyenek f első és második parciális deriváltjai az (x_0, y_0) egy környezetében folytonosak). Teljesüljön (6.2). Ekkor

a) $\Delta > 0$ esetén az f függvénynek az (x_0, y_0) pontban szélsőértéke van, mégpedig

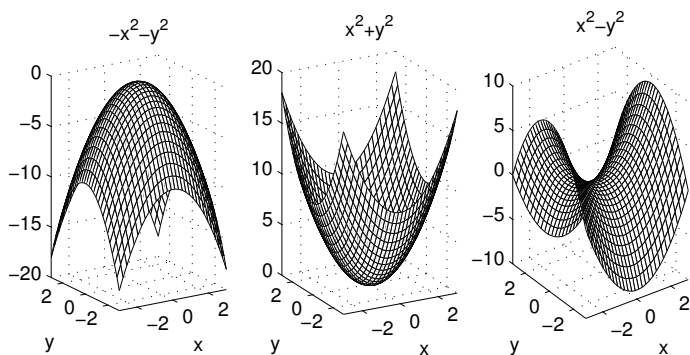
(i) szigorú maximuma, ha $\frac{\partial^2 f(x_0, y_0)}{\partial x^2} < 0$,

(ii) szigorú minimuma, ha $\frac{\partial^2 f(x_0, y_0)}{\partial x^2} > 0$;

b) $\Delta < 0$ esetén az f függvénynek az (x_0, y_0) pontban nincs szélsőértéke;

c) $\Delta = 0$ esetén pedig előfordulhat, hogy az (x_0, y_0) pontban van szélsőérték, de az is, hogy nincs szélsőérték.

Az a) rész (i) esetére példa az $f(x, y) = -x^2 - y^2$ lefelé néző paraboloid, melynek az $(x_0, y_0) = (0, 0)$ pontban maximuma van; az (ii) esetre példa az $f(x, y) = x^2 + y^2$ felfelé néző paraboloid, melynek az $(x_0, y_0) = (0, 0)$ pontban minimuma van; míg a b) részre példa az $f(x, y) = x^2 - y^2$ nyeregfelület, melynek az $(x_0, y_0) = (0, 0)$ pontban nincsen sem maximuma, sem minimuma, lásd a 6.2 ábrát.



6.2. ábra. Paraboloidok és nyeregfelület

6.2. Mátrixok általánosított inverze és kvadratus formák minimuma

Ebben a szakaszban egy speciális, de fontos optimalizációs problémát tárgyalunk. A [33] könyvre támaszkodunk.

6.1. Definíció. Az A $p \times n$ -es mátrix *általánosított inverzének* azt az A^- $n \times p$ -s mátrixot nevezzük, melyre

$$AA^-A = A. \quad (6.3)$$

6.2. Megjegyzés. Ha A invertálható, akkor A^- egyértelműen létezik, és $A^- = A^{-1}$.

Ugyanis ekkor $AA^{-1}A = A$ nyilvánvalóan teljesül. Másrészt az $AA^-A = A$ egyenletet jobbról és balról is megszorozva A^{-1} -gyel, $A^- = A^{-1}$ -et kapjuk.

Természetesen az általánosított inverz nem minden mátrixra egyértelmű. Például a zérus mátrixnak bármely (alkalmas méretű) mátrix általánosított inverze. Később definiáljuk az ún. pszeudoinverzet, amely kielégíti az általánosított inverz definícióját (és még extra feltételeket). Annak fogjuk igazolni a létezését és egyértelműségét.

6.3. Állítás. Ha A szimmetrikus $n \times n$ -es mátrix, akkor létezik általánosított inverze.

Bizonyítás. Legyenek $\lambda_1, \dots, \lambda_k$ az A nem zérus sajátértékei, $\mathbf{u}_1, \dots, \mathbf{u}_n$ pedig az A sajátvektorainak ortonormált rendszere. Legyen U az $\mathbf{u}_1, \dots, \mathbf{u}_n$ -et

tartalmazó ortonormált mátrix,

$$\Lambda = \begin{pmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_k & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix}, \quad \Lambda^{-} = \begin{pmatrix} \lambda_1^{-1} & & & & \\ & \ddots & & & \\ & & \lambda_k^{-1} & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix}$$

$n \times n$ -es diagonális mátrixok. Ismeretes (és könnyen igazolható), hogy

$$A = U\Lambda U^{\top}. \quad (6.4)$$

Belátjuk, hogy

$$A^{-} = U\Lambda^{-}U^{\top} \quad (6.5)$$

teljesíti a feltételt:

$$AA^{-}A = (U\Lambda U^{\top})(U\Lambda^{-}U^{\top})(U\Lambda U^{\top}) = U\Lambda U^{\top} = A. \quad \square$$

6.4. *Megjegyzés.* Erre a speciális A^{-} -ra $A^{-}AA^{-} = A^{-}$ és $(A^{-})^{\top} = A^{-}$ is teljesül.

6.5. Állítás. $A(A^{\top}A)^{-}A^{\top}$ ortogonális projekció az A oszlopvektorai által generált \mathcal{F} altérre.

Bizonyítás. Azt kell belátni, hogy $A(A^{\top}A)^{-}A^{\top}$ az \mathcal{F} altérre merőleges vektorokat $\mathbf{0}$ -ba, az \mathcal{F} -belieket pedig önmagukba viszi.

Ha $\mathbf{x} \perp \mathcal{F}$, akkor $A(A^{\top}A)^{-}A^{\top}\mathbf{x} = \mathbf{0}$.

Ahhoz, hogy $A(A^{\top}A)^{-}A^{\top}$ az \mathcal{F} -beli vektorokat önmagukba viszi, elég belátni, hogy $[A(A^{\top}A)^{-}A^{\top}]A = A$. Ez viszont igaz, mert tetszőleges \mathbf{v} vektor $\mathbf{v} = \mathbf{v}_1 + A\mathbf{v}_2$ alakba írható, ahol $\mathbf{v}_1 \perp \mathcal{F}$. Ezért

$$\begin{aligned} \mathbf{v}^{\top}A(A^{\top}A)^{-}A^{\top}A &= \mathbf{v}_1^{\top}A(A^{\top}A)^{-}A^{\top}A + \mathbf{v}_2^{\top}(A^{\top}A)(A^{\top}A)^{-}(A^{\top}A) = \\ &= \mathbf{0} + \mathbf{v}_2^{\top}(A^{\top}A) = \mathbf{v}^{\top}A. \end{aligned} \quad \square$$

Az $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer megoldása helyett gyakran érdemes a

$$\mathbf{b} = A\mathbf{x} + \varepsilon \quad (6.6)$$

lineáris modellben az \mathbf{x} legkisebb négyzetes becslését keresni. Ez a

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|^2 \quad (6.7)$$

minimum problémát jelenti. Ennek mindig van megoldása, és amennyiben a minimum értéke 0, úgy egyben a kiinduló $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer megoldását kapjuk. (Megjegyezzük, hogy a numerikus programcsomagok gyakran eleve a fenti általános értelemben adják meg a lineáris egyenletrendszerek megoldását.)

A (6.7) feladat azzal ekvivalens, hogy keressük a A oszlopvektorai azon $A\mathbf{x}$ lineáris kombinációját, mely \mathbf{b} -hez legközelebb van. Ez éppen \mathbf{b} merőleges vetülete a A oszlopai által generált \mathcal{F} altérre. Az ortogonális komplementer ekkor $\mathbf{b} - A\mathbf{x}$. Az, hogy ez merőleges \mathcal{F} -re, ekvivalens $A^\top(\mathbf{b} - A\mathbf{x}) = \mathbf{0}$ -val. Innen az

$$A^\top A\mathbf{x} = A^\top \mathbf{b} \quad (6.8)$$

normálegyenletet kapjuk. Ennek \mathbf{x} megoldásai egybeesnek a (6.7) minimum feladat megoldásaival. Mivel a merőleges vetület létezik, így a normálegyenletnek mindig van megoldása (de lehet, hogy több is van). Viszont a merőleges vetület egyértelmű, így az \mathbf{x} megoldásra az $A\mathbf{x}$ vektor már egyértelmű.

Az általánosított inverzzel is kifejezhetjük a megoldást:

$$\mathbf{x} = (A^\top A)^{-1} A^\top \mathbf{b}. \quad (6.9)$$

Valóban, ekkor $A\mathbf{x} = A(A^\top A)^{-1} A^\top \mathbf{b}$, azaz a 6.5. Állítás miatt $A\mathbf{x}$ éppen \mathbf{b} merőleges vetülete az A oszlopai által generált \mathcal{F} altérre.

A $\|\mathbf{b} - A\mathbf{x}\|^2$ minimumhelyét

$$\mathbf{x} = A^+ \mathbf{b}$$

alakban is ki szokták fejezni, ahol A^+ az A mátrix pszeudoinverze. Belátjuk, hogy ez ekvivalens az előzőekkel.

A^+ meghatározása az alábbi ismereteket igényli. Jelölje δ_{ij} a Dirac-deltát: $\delta_{ij} = 1$, ha $i = j$, $\delta_{ij} = 0$, ha $i \neq j$.

6.6. Definíció. Az A $p \times n$ típusú mátrix *sajátbázisának* nevezzük az \mathbb{R}^n tér $\mathbf{u}_1, \dots, \mathbf{u}_n$ ortonormált bázisát, ha

$$\mathbf{u}_i^\top A^\top A \mathbf{u}_j = \sigma_i^2 \delta_{ij}, \quad 1 \leq i, j \leq n. \quad (6.10)$$

A σ_i számokat az A *szinguláris értékeinek* nevezzük. Ezek sorrendjéről felteesszük hogy $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$.

Sajátbázis létezik, nem egyértelmű, de a szinguláris értékek egyértelműek.

6.7. Tétel. *Tetszőleges A mátrixnak létezik sajátbázisa.*

Bizonyítás. 1. Az első módszer: visszavezetjük a szimmetrikus mátrixok spektrálfelbontására. Valóban, az $A^\top A$ szimmetrikus, pozitív szemidefinit mátrixnak létezik n darab nem negatív sajátértéke (ezek legyenek $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_n^2 \geq 0$), valamint létezik a fenti sajátértékekhez tartozóan, n darab ortonormált sajátvektora (ezek legyenek $\mathbf{u}_1, \dots, \mathbf{u}_n$). Így (6.10) teljesül.

Másrészt, ha (6.10) teljesül, akkor az \mathbf{u}_i vektorok és a σ_i^2 számok nem lehetnek mások, mint $A^\top A$ sajátvektorai és sajátértékei. Azaz sajátbázis létezik, nem egyértelmű, de a szinguláris értékek egyértelműek.

2. A második módszer: a szimmetrikus mátrixok spektrálfelbontására nem támaszkodva. Legyen \mathbf{u}_1 az \mathbb{R}^n tér olyan egységvektora, mely megoldása a $\max_{\|\mathbf{u}\|=1} \|\mathbf{A}\mathbf{u}\|$ maximum problémának. Ilyen létezik, hiszen kompakt halmazon folytonos függvény felveszi a maximumát. Ha ez a maximum érték nulla, akkor $A = 0$, így tegyük fel, hogy $\mathbf{A}\mathbf{u}_1 \neq \mathbf{0}$.

Legyen most \mathbf{u} az \mathbf{u}_1 -re merőleges egységvektor. Ekkor az \mathbf{u}_1 választása miatt az

$$\begin{aligned} f(\alpha) &= \|A(\mathbf{u}_1 \cos \alpha + \mathbf{u} \sin \alpha)\|^2 = \\ &= \|\mathbf{A}\mathbf{u}_1\|^2 \cos^2 \alpha + 2\langle \mathbf{A}\mathbf{u}_1, \mathbf{A}\mathbf{u} \rangle \cos \alpha \sin \alpha + \|\mathbf{A}\mathbf{u}\|^2 \sin^2 \alpha \end{aligned}$$

függvénynek az $\alpha = 0$ helyen maximuma van. Így deriváltja 0, azaz $0 = f'(0) = 2\langle \mathbf{A}\mathbf{u}_1, \mathbf{A}\mathbf{u} \rangle$. Tehát az A mátrix az \mathbf{u}_1 -re merőleges alteret az $\mathbf{A}\mathbf{u}_1$ -re merőleges altérbe képezi. Azaz az \mathbf{u}_1 -re merőleges altérből (ami már eggyel kevesebb dimenziós) választhatjuk a következő vektorokat éppen úgy, ahogyan \mathbf{u}_1 -et választottuk \mathbb{R}^n -ből. Indukcióval kész vagyunk. \square

Az $\mathbf{A}\mathbf{u}_i$ vektorok egymásra merőlegesek. Azon indexekre, melyekre teljesül, hogy $\sigma_i \neq 0$, a $\mathbf{v}_i = \mathbf{A}\mathbf{u}_i/\sigma_i$ vektorok ortonormáltak. Egészítsük ki ezeket az \mathbb{R}^p tér ortonormált bázisává: $\mathbf{v}_1, \dots, \mathbf{v}_p$.

6.8. Definíció. Az $\mathbf{u}_1, \dots, \mathbf{u}_n$, és a $\mathbf{v}_1, \dots, \mathbf{v}_p$ bázisokat együtt az A sajátbázis párjának nevezzük.

Ezekre teljesül, hogy

$$\mathbf{v}_j^\top \mathbf{A}\mathbf{u}_i = \sigma_i \delta_{ij}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq p. \quad (6.11)$$

6.9. Állítás. Szinguláris felbontási tétel. Egy $p \times n$ -es, valós A mátrixhoz léteznek

$$U = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\} \quad \text{és} \quad V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$$

ortogonális mátrixok és $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_l)$ diagonális mátrix (ahol $l = \min\{p, n\}$ és $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_l \geq 0$) úgy, hogy

$$A = V\Sigma U^\top.$$

V , Σ és U rendre $p \times p$, $p \times n$ és $n \times n$ típusú mátrixok.

Bizonyítás. Az A szinguláris felbontása ekvivalens $V^\top AU = \Sigma$ -val, ez pedig pontosan (6.11) mátrixos alakja. \square

6.10. Definíció. Az A $p \times n$ -es mátrix *pszeudoinverzének* (Moore–Penrose-inverzének) azt az A^+ $n \times p$ típusú mátrixot nevezzük, melyre teljesülnek az alábbiak

$$AA^+A = A, \quad (AA^+)^\top = AA^+, \quad (6.12)$$

$$A^+AA^+ = A^+, \quad (A^+A)^\top = A^+A. \quad (6.13)$$

A pszeudoinverz nyilván teljesíti az általánosított inverz (6.3) definícióját.

6.11. Tétel. *Tetszőleges A valós elemű mátrixnak létezik pszeudoinverze, és az egyértelmű.*

Bizonyítás. Legyen $A = V\Sigma U^\top$ a fenti szinguláris felbontás. Legyen Σ^+ $n \times p$ méretű diagonális mátrix:

$$\Sigma^+ = \text{diag} \left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_l} \right)$$

($\frac{1}{0} = 0$ értelmezéssel). Könnyen ellenőrizhető, hogy Σ^+ a Σ mátrix pszeudoinverze. Legyen

$$A^+ = U\Sigma^+V^\top.$$

Ez teljesíti a definícióban szereplő feltételeket. Például

$$AA^+A = (V\Sigma U^\top)(U\Sigma^+V^\top)(V\Sigma U^\top) = V\Sigma U^\top = A.$$

Illetve

$$(AA^+)^\top = (V\Sigma U^\top U\Sigma^+V^\top)^\top = V(\Sigma^+\Sigma)^\top V^\top = V\Sigma\Sigma^+V^\top = AA^+.$$

Az egyértelműség igazolásához tegyük fel, hogy a B $n \times p$ típusú mátrixra is teljesülnek a (6.12)–(6.13) feltételek. A feltételek miatt BA és A^+A szimmetrikus. Tekintsük a BA mátrixot:

$$BA = B(AA^+A) = (BA)(A^+A).$$

Transzponáltat véve:

$$BA = (BA)^\top = (A^+A)(BA) = A^+(ABA) = A^+A.$$

Hasonlóan

$$AB = (AA^+A)B = (AA^+)(AB).$$

Transzponáltat véve, és kihasználva, hogy AB és AA^+ szimmetrikus:

$$AB = (AB)^\top = (AB)(AA^+) = (ABA)A^+ = AA^+.$$

Az ezekből adódó $BA = A^+A$ és $AB = AA^+$ összefüggéseket használva:

$$A^+ = A^+AA^+ = (A^+A)A^+ = (BA)A^+ = B(AA^+) = B(AB) = B.$$

Tehát egyrészt konstrukciót adtunk a pszeudoinverzre, másrészt beláttuk, hogy a definiáló tulajdonságait csak egyetlen mátrix elégíti ki. \square

Most vessük össze a $\|\mathbf{b} - A\mathbf{x}\|^2$ minimumhelyére szolgáló közismert $\mathbf{x} = A^+\mathbf{b}$ képletet a szintén minimumhelyet szolgáltató $\mathbf{x} = (A^\top A)^- A^\top \mathbf{b}$ képlettel. Ez utóbbi a vetítéssel szemléletesen igazolható, ráadásul a szimmetrikus $A^\top A$ mátrix $(A^\top A)^-$ általánosított inverze könnyen származtatható.

Most belátjuk, hogy $(A^\top A)^- A^\top = A^+$. Valóban, $A = V\Sigma U^\top$ alapján

$$\begin{aligned} (A^\top A)^- A^\top &= (U\Sigma^\top V^\top V\Sigma U^\top)^- U\Sigma^\top V^\top = (U\Sigma^\top \Sigma U^\top)^- U\Sigma^\top V^\top = \\ &= (U(\Sigma^\top \Sigma)^- U^\top) U\Sigma^\top V^\top = U(\Sigma^\top \Sigma)^- \Sigma^\top V^\top = U\Sigma^+ V^\top = A^+. \end{aligned}$$

Azaz $(A^\top A)^- A^\top = A^+$. Tehát beláttuk az alábbiakat.

6.12. Állítás. *A $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|^2$ minimum probléma megoldásai egybeesnek az $A^\top A\mathbf{x} = A^\top \mathbf{b}$ normálegyenlet megoldásaival.*

Mind az $\mathbf{x} = A^+\mathbf{b}$ képlet, mind az $\mathbf{x} = (A^\top A)^- A^\top \mathbf{b}$ képlet a minimum probléma ugyanazon megoldását adja (amennyiben szimmetrikus mátrix általánosított inverzét a fenti módon adjuk meg).

Most rátérünk általános típusú kvadratikuss forma minimumának vizsgálatára. E célból bevezetjük pozitív szemidefinit mátrixok négyzetgyökét. Legyen $A = U\Lambda U^\top$ az előbbi, (6.4) képletben szereplő mátrix, legyen A pozitív szemidefinit. Ekkor a sajátértékei nem negatívak. Ezért az alábbi mátrix jól definiált:

$$\Lambda^{1/2} = \begin{pmatrix} \sqrt{\lambda_1} & & & & \\ & \ddots & & & \\ & & \sqrt{\lambda_k} & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix}.$$

6.13. Definíció. Az $A^{1/2} = U\Lambda^{1/2}U^\top$ mátrixot nevezzük az A mátrix négyzetgyökének.

Ekkor $A^{1/2}$ szimmetrikus, pozitív szemidefinit és teljesül az $A = A^{1/2}A^{1/2}$ egyenlőség.

Csak megjegyezzük, hogy az $A^* = U\Lambda^{1/2}$ mátrix is teljesít négyzetgyök-szerű tulajdonságot. Nevezetesen $A = A^*(A^*)^\top$. De A^* nem szimmetrikus.

Most a fenti $\|A\mathbf{x} - \mathbf{b}\|^2 = \mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{x}^\top A^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b}$ speciális kvadratikus forma helyett elemezzük az

$$f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} - 2\mathbf{x}^\top \mathbf{b} + c$$

általános kvadratikus forma minimumának létezését.

A legegyszerűbb hozzáállás: a minimumhelyen az első deriváltak nullának kell lennie. Azaz az \mathbf{x} minimumhely kielégíti az $A\mathbf{x} - \mathbf{b} = \mathbf{0}$ egyenletet.

A pontosabb elemzés az alábbi.

6.14. Állítás. Ha A szimmetrikus, pozitív szemidefinit, és \mathbf{b} benne van az A képterében, akkor az $f(\mathbf{x})$ minimumhelye az $A\mathbf{x} = \mathbf{b}$ tetszőleges megoldása, továbbá minden ilyen megoldásnál azonos az f függvény értéke.

Ha \mathbf{b} nincs benne A képterében, vagy ha A nem pozitív szemidefinit, akkor az f függvény alulról nem korlátos.

Bizonyítás. Ha \mathbf{b} nincs benne A képterében, akkor nincs minimum (pontosabban: az f függvény alulról nem korlátos). Legyen ugyanis $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$, ahol \mathbf{b}_1 benne van A képterében, \mathbf{b}_2 pedig arra merőleges, $\mathbf{b}_2 \neq \mathbf{0}$. Legyen $\mathbf{x} = \alpha\mathbf{b}_2$. Ekkor

$$f(\alpha\mathbf{b}_2) = \alpha^2\mathbf{b}_2^\top A\mathbf{b}_2 - 2\alpha\mathbf{b}_2^\top \mathbf{b} + c = 0 - 2\alpha\mathbf{b}_2^\top \mathbf{b}_2 + c = -2\alpha\|\mathbf{b}_2\|^2 + c \rightarrow -\infty,$$

amint $\alpha \rightarrow \infty$.

Ha A nem pozitív szemidefinit, akkor az f függvény alulról nem korlátos. Legyen ugyanis \mathbf{v} olyan vektor, melyre $\mathbf{v}^\top A\mathbf{v} < 0$. Ekkor

$$f(\alpha\mathbf{v}) = \alpha^2(\mathbf{v}^\top A\mathbf{v}) - 2\alpha(\mathbf{v}^\top \mathbf{b}) + c \rightarrow -\infty,$$

midőn $\alpha \rightarrow \infty$ (ugyanis az α -ban másodfokú tag együtthatója negatív).

Most belátjuk, hogy az $f(\mathbf{x})$ minimumhelye az $A\mathbf{x} = \mathbf{b}$ megoldása, ha A szimmetrikus, pozitív szemidefinit és \mathbf{b} benne van az A képterében. Legyen ekkor ugyanis $\mathbf{b} = A\mathbf{a}$, amivel

$$f(\mathbf{x}) = \left(\mathbf{x}^\top A^{1/2} - \mathbf{a}^\top A^{1/2}\right) \left(A^{1/2}\mathbf{x} - A^{1/2}\mathbf{a}\right) - \mathbf{a}^\top A\mathbf{a} + c =$$

$$= \left\| A^{1/2} \mathbf{x} - A^{1/2} \mathbf{a} \right\|^2 - \mathbf{a}^\top A \mathbf{a} + c.$$

Ez akkor minimális, ha $A^{1/2} \mathbf{x} - A^{1/2} \mathbf{a} = \mathbf{0}$, azaz $\mathbf{x} - \mathbf{a}$ benne van $A^{1/2}$ nullterében. De $A^{1/2}$ nulltere megegyezik A nullterével. Tehát $A\mathbf{x} = A\mathbf{a} = \mathbf{b}$. De ennek az egyenletnek lehetnek különböző megoldásai. Szerencsére, az f függvény értéke ezeken a helyeken azonos:

$$f(\mathbf{x}) = -\mathbf{a}^\top A \mathbf{a} + c = -\mathbf{a}^\top A A^- A \mathbf{a} + c = -\mathbf{b}^\top A^- \mathbf{b} + c.$$

Tehát beláttuk, hogy ha A szimmetrikus, pozitív szemidefinit és \mathbf{b} benne van az A képterében, akkor az $f(\mathbf{x})$ minimumhelye az $A\mathbf{x} = \mathbf{b}$ tetszőleges megoldása, és minden ilyen megoldásnál azonos az f függvény értéke. \square

6.3. Optimalizációs technikák

Először csupán egy szóhasználati módot tisztázunk. Az elektromérnöki gyakorlatban tipikus feladat a jelfeldolgozás (signal processing). Ebben fontos eszköz a visszacsatolós szűrő (adaptive filter). Mivel a visszacsatolós szűrő felfogható mint egy rekurziós eljárás, így ebben a műben ritkán használjuk ezt a kifejezést.

Tegyük fel, hogy a hálózat a \mathbf{w} súlyvektor esetén az \mathbf{x} bemenetre az \mathbf{y} kimenetet produkálja. Azonban az igazi \mathbf{w}^* súlyvektor esetén az igazi kimenet \mathbf{d} lenne (\mathbf{x} , \mathbf{y} és \mathbf{d} alkalmas méretű vektorok). Legyen $\mathcal{E}(\mathbf{w})$ az \mathbf{y} és a \mathbf{d} eltérésének egy mérőszáma.

Például a leggyakrabban használt veszteségfüggvény az

$$\mathcal{E}(\mathbf{w}) = \|\mathbf{d} - \mathbf{y}(\mathbf{w})\|^2 \geq 0$$

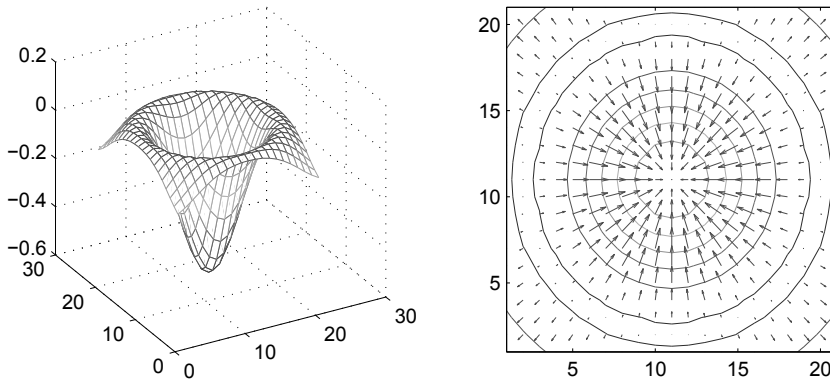
négyzetes hiba. Általában a \mathbf{w}^* igazi súlyvektorra $\mathcal{E}(\mathbf{w}^*) = \|\mathbf{d} - \mathbf{y}(\mathbf{w}^*)\|^2 = 0$. (Itt $\|\cdot\|$ a szokásos euklideszi norma.)

Tehát amennyiben általános értelemben a \mathbf{w} súlyvektor „jóságát” az $\mathcal{E}(\mathbf{w})$ függvény méri, akkor ésszerű feltenni, hogy az igazi \mathbf{w}^* súly esetén $\mathcal{E}(\mathbf{w}^*) \leq \mathcal{E}(\mathbf{w})$ minden \mathbf{w} esetén. Tehát az \mathcal{E} (többváltozós) függvény globális minimumhelyét kell meghatározni, ott lesz az igazi súlyvektor.

6.3.1. A gradiens módszer

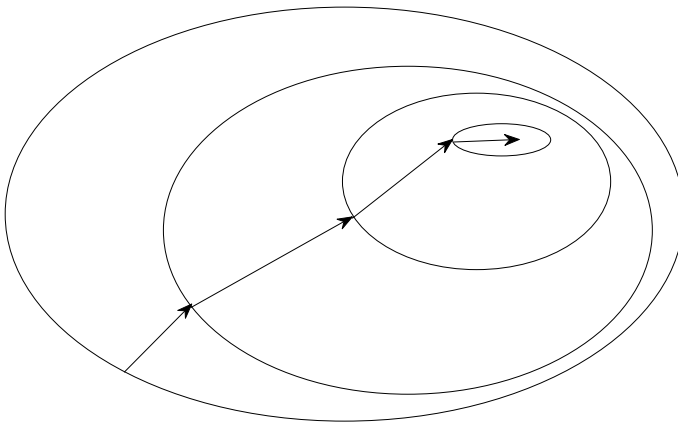
A gradiens módszer (delta rule, method of steepest descent) lényege, hogy mindig a legnagyobb csökkenés irányába haladunk. Kétváltozós függvény esetén ez úgy képzelhető el, hogy a hegyről a legmeredekebb úton ereszkedünk le a völgybe. A térképen megszokott szintvonalakkal ez jól ábrázolható.

A 6.3. ábra első részén egy kétváltozós függvény képe, a második részén pedig a szintvonalak és a negatív gradiensek láthatóak. A csökkenés iránya jól megfigyelhető.



6.3. ábra. Kétváltozós függvény képe, ill. szintvonalak és a negatív gradiensek

A 6.4 ábrán egy felület szintvonalai és a gradiens módszer szerinti csökkenés irányai láthatóak. Megfigyelhető, hogy a lépések a „völgy aljára” vezetnek.



6.4. ábra. A gradiens módszer szerinti lépések

Hogyan kell tehát lépni a $\mathbf{w}(n)$ közelítésről a $\mathbf{w}(n+1)$ közelítésre? Legyen $\Delta \mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n)$ a \mathbf{w} vektor $(n+1)$ -edik és n -edik értékének a differenciája. Az \mathcal{E} függvény gradiens vektora:

$$\mathbf{g}(\mathbf{w}) = \nabla \mathcal{E}(\mathbf{w}) = \left(\frac{\partial \mathcal{E}(\mathbf{w})}{\partial w_0}, \dots, \frac{\partial \mathcal{E}(\mathbf{w})}{\partial w_m} \right)^\top,$$

ahol $\mathbf{w} = (w_0, w_1, \dots, w_m)^\top$. A Taylor-formula szerint lokálisan jó (elsőfokú) közelítés:

$$\mathcal{E}(\mathbf{w}(n+1)) \simeq \mathcal{E}(\mathbf{w}(n)) + \mathbf{g}(\mathbf{w}(n))^\top \Delta \mathbf{w}(n).$$

A Cauchy-egyenlőtlenség miatt

$$-\|\mathbf{g}(\mathbf{w}(n))\| \|\Delta \mathbf{w}(n)\| \leq \mathbf{g}(\mathbf{w}(n))^\top \Delta \mathbf{w}(n) \leq \|\mathbf{g}(\mathbf{w}(n))\| \|\Delta \mathbf{w}(n)\|,$$

és itt az alsó határ akkor éretik el, ha $\Delta \mathbf{w}(n) = -\eta \mathbf{g}(\mathbf{w}(n))$, ahol $\eta > 0$. Tehát a gradiens módszer szerint a súlyvektor új értéke:

$$\boxed{\mathbf{w}(n+1) = \mathbf{w}(n) - \eta \mathbf{g}(\mathbf{w}(n))}. \quad (6.14)$$

Itt $\eta > 0$ konstans, amit tanulási paraméternek nevezünk. Tehát

$$\Delta \mathbf{w}(n) = -\eta \mathbf{g}(\mathbf{w}(n))$$

a korrekciós szabály leírása. η értékét pedig úgy határozzák meg, hogy olyan nagy legyen, ameddig az \mathcal{E} függvény értéke csökken. Ez egy egyenes mentén történő keresés, ami számítógépen megoldható.

Kicsit más magyarázattal is szokták indokolni a gradiens módszert. Vizsgáljuk meg, hogy a módszer kielégíti a $\mathcal{E}(\mathbf{w}(n+1)) < \mathcal{E}(\mathbf{w}(n))$ feltételt (azaz lépésenként csökkenti a hibát). Induljunk ki megint a $\mathbf{w}(n)$ körüli Taylor-sorból. $\Delta \mathbf{w}(n) = -\eta \mathbf{g}(\mathbf{w}(n))$, azaz a gradiens módszer szerinti lépés esetén

$$\mathcal{E}(\mathbf{w}(n+1)) \simeq \mathcal{E}(\mathbf{w}(n)) - \eta \mathbf{g}(\mathbf{w}(n))^\top \mathbf{g}(\mathbf{w}(n)) = \mathcal{E}(\mathbf{w}(n)) - \underbrace{\eta \|\mathbf{g}(\mathbf{w}(n))\|^2}_{>0}.$$

Tehát az \mathcal{E} függvény értéke folyamatosan csökkenni fog.

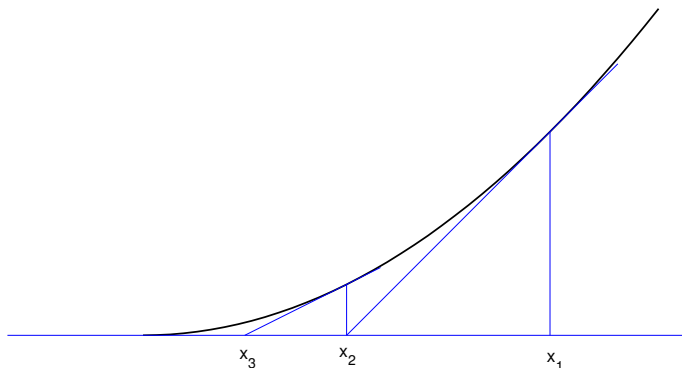
Vizsont η nagysága az alábbi módon befolyásolja a konvergenciát.

- Ha η kicsi, akkor az algoritmus lefojtott (csillapított, overdamped), és a trajektóriája sima.
- Ha η nagy, akkor az algoritmus nem eléggé csillapított (underdamped), és a trajektóriája cikcakkban halad.
- Ha η meghalad egy kritikus értéket, akkor az algoritmus instabillá válik, tehát nem konvergál.

A gradiens módszer tulajdonságainak elemzése megtalálható Fletcher [14] könyve 2. fejezetében. Az egyenes mentén történő keresésnél a pontos minimum helyett közelítő minimumhelyet ajánl. Habár a gradiens módszerre érvényes a stacionárius ponthoz való konvergencia (l. [14], Theorem 2.5.1), mégsem javasolja az általános használatát: a gyakorlatban megbízhatatlan.

6.3.2. A Newton-módszer

A Newton-módszer legismertebb esete az $f(x) = 0$ (nem-lineáris, egyismeretlenes) egyenlet numerikus megoldása. A megoldás során a gyök x_k közelítésétől úgy jutunk el az x_{k+1} közelítéshez, hogy a függvény görbéjéhez az x_k pontban érintőt húzunk, ahol ez metszi az x -tengelyt, ott lesz x_{k+1} . Ez a 6.5. ábrán látható.



6.5. ábra. A Newton-módszer szerinti lépések

Mivel az érintő egyenlete

$$y - f(x_k) = f'(x_k)(x - x_k),$$

így $y = 0$ helyettesítéssel kapjuk, hogy

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Az általánosított Newton-módszer az $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ többváltozós egyenletrendszer megoldására szolgál, ahol $\mathbf{x} = (x_1, \dots, x_m)^\top$ és $\mathbf{f} = (f_1, \dots, f_m)^\top$. Az egyváltozós esethez hasonlóan a gyökök közelítő sorozat képzési szabálya:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J^{-1}(\mathbf{x}^{(k)})\mathbf{f}(\mathbf{x}^{(k)}).$$

Itt $\mathbf{x}^{(k)}$ a k -adik közelítés, J pedig az \mathbf{f} függvény Jacobi-mátrixa, azaz $J(\mathbf{x})$ j -edik sorának k -adik eleme: $\frac{\partial f_j(\mathbf{x})}{\partial x_k}$.

Az aktuális feladatunk azonban \mathcal{E} globális minimumhelyének meghatározása. A minimum szükséges feltétele: a parciális deriváltak eltűnése. Tehát kapunk egy egyenletrendszert, aminek a megoldását kereshetjük az egyenletrendszer megoldására szolgáló fenti általánosított Newton-módszerrel.

Ugyanezen végeredményhez, azaz a (6.17) képlethez fogunk eljutni az alábbi levezetéssel is.

A Newton-algoritmust megkaphatjuk a hibafüggvény másodfokú közelítéséből. A másodrendű Taylor-polinom az aktuális $\mathbf{w}(n)$ körül:

$$\begin{aligned}\Delta\mathcal{E}(\mathbf{w}(n)) &= \mathcal{E}(\mathbf{w}(n+1)) - \mathcal{E}(\mathbf{w}(n)) \simeq \\ &\simeq \mathbf{g}(\mathbf{w}(n))^\top \Delta\mathbf{w}(n) + \frac{1}{2} \Delta\mathbf{w}(n)^\top H(\mathbf{w}(n)) \Delta\mathbf{w}(n),\end{aligned}\quad (6.15)$$

ahol $H(\mathbf{w})$ a Hesse-mátrix

$$H = H(\mathbf{w}) = \nabla^2 \mathcal{E}(\mathbf{w}) = \left(\frac{\partial^2 \mathcal{E}(\mathbf{w})}{\partial w_i \partial w_j} \right).$$

$\mathcal{E}(\mathbf{w})$ kétszer folytonosan differenciálhatóságát feltételezzük. $\mathbf{w}(n+1)$ -et $\Delta\mathbf{w}(n)$ szerinti minimalizálással kapjuk. A fenti (6.15) kifejezést deriválva adódik a minimum szükséges feltétele:

$$\mathbf{g}(\mathbf{w}(n)) + H(\mathbf{w}(n)) \Delta\mathbf{w}(n) = 0. \quad (6.16)$$

Ezen egyenlet megoldása:

$$\Delta\mathbf{w}(n) = - \left[H(\mathbf{w}(n)) \right]^{-1} \mathbf{g}(\mathbf{w}(n)).$$

Tehát a Newton-módszerrel képezett, a minimumot közelítő sorozat

$$\boxed{\mathbf{w}(n+1) = \mathbf{w}(n) - \left[H(\mathbf{w}(n)) \right]^{-1} \mathbf{g}(\mathbf{w}(n))}, \quad (6.17)$$

ahol H^{-1} a H inverze. Mátrix invertálás helyett az alábbi ajánlott. Legyen $\boldsymbol{\delta}(n)$ a

$$H(\mathbf{w}(n)) \boldsymbol{\delta} = -\mathbf{g}(\mathbf{w}(n))$$

megoldása. Legyen $\mathbf{w}(n+1) = \mathbf{w}(n) + \boldsymbol{\delta}(n)$. A Newton-módszert érdemes a fenti $\boldsymbol{\delta}(n)$ irányban egyenes mentén történő kereséssel alkalmazni.

A (6.15) és (6.16) formulákból látható, hogy a Newton-módszer szerinti lépés esetén

$$\Delta\mathcal{E}(\mathbf{w}(n)) \simeq -\frac{1}{2} \Delta\mathbf{w}(n)^\top H(\mathbf{w}(n)) \Delta\mathbf{w}(n).$$

Tehát ahhoz, hogy a célfüggvény értéke mindig csökkenjen, $H(\mathbf{w}(n))$ pozitív definit volta szükséges.

Általában igaz, hogy a Newton-módszer gyorsan konvergál, és nem halad „cikkcakkban”, mint a gradiens módszer. Bizonyos feltételek esetén a Newton-módszer másodrendben konvergál (egy lokális minimumhoz), l. [14], Theorem 3.1.1. A módszer alkalmazhatóságának feltétele, hogy $H(\mathbf{w}(n))$ -nek pozitív definitnek kell lennie bármely n -re. Ha nem pozitív definit, akkor módosítani kell az eljáráson. Egy lehetséges módosítás, hogy az aktuális Hesse-mátrixhoz hozzáadjuk az egység mátrix egy (kis) pozitív számszorosát. Azonban az egyik legfontosabb hátrány, hogy ki kell számolni a második deriváltakat (azaz a Hesse-mátrixot), ami általában nehézségekbe ütközik.

Most adunk az eljárás igazolására egy olyan érvelést, amelyből kiderül, hogy pozitív definit $H(\mathbf{w}(n))$ esetén működik a módszer. A (6.15) képletből (nyilvánvaló rövidített jelöléssel) kapjuk:

$$\begin{aligned}\mathcal{E}(\mathbf{w}(n+1)) &\simeq c + \mathbf{g}^\top \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top H \boldsymbol{\delta} = \\ &= \frac{1}{2} \left[\boldsymbol{\delta}^\top \sqrt{H} + \mathbf{g}^\top \sqrt{H^{-1}} \right] \left[\sqrt{H} \boldsymbol{\delta} + \sqrt{H^{-1}} \mathbf{g} \right] - \frac{1}{2} \mathbf{g}^\top H^{-1} \mathbf{g} + c.\end{aligned}$$

Itt, ha H pozitív definit, akkor neki és inverzének is létezik négyzetgyöke, amelyet \sqrt{H} , illetve $\sqrt{H^{-1}}$ jelöl. Az előző kifejezés első tagja lényegében egy vektor hosszának a négyzete, amely akkor minimális, ha a vektor nulla, azaz ha $\boldsymbol{\delta} = -H^{-1} \mathbf{g}$. Tehát most is megkaptuk a Newton-módszer előző (6.17) képletét.

6.3.3. Kvázi Newton-módszerek

A Newton-módszer hátrányait bizonyos módosításokkal küszöbölhetjük ki. Ebben a részben Fletcher [14] könyve alapján tekintjük át a kvázi Newton-módszereket. Induljunk ki a hagyományos Newton-módszerből. Tehát az \mathcal{E} többváltozós célfüggvény minimumát keressük. A rövidség kedvéért jelölje \mathbf{g}_k a gradiens k -adik értékét, H_k a Hesse-mátrix k -adik értékét, \mathbf{w}_k a minimumhely k -adik közelítését. Ekkor a Newton-módszer az alábbi. Oldjuk meg a

$$H_k \boldsymbol{\delta} = -\mathbf{g}_k$$

egyenletet, a megoldást jelölje $\boldsymbol{\delta}_k$, a következő közelítés pedig legyen

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\delta}_k.$$

Mivel ezzel a módszerrel nem garantált a célfüggvény csökkenése, így egy egyenes mentén történő keresést alkalmazunk. Legyen

$$H_k \mathbf{s} = -\mathbf{g}_k$$

megoldása \mathbf{s}_k , és az aktuális \mathbf{w}_k pontból \mathbf{s}_k irányában keressük meg az \mathcal{E} célfüggvény minimumhelyét. Ez a pont lesz a minimumhely következő közelítése: \mathbf{w}_{k+1} .

A Newton-módszer fő nehézsége, hogy minden lépésben ki kell számolni a Hesse-mátrixot (azaz a második deriváltakat), és az nem is lesz mindig pozitív definit. Ezeket kiküszöbölhetjük, ha a H_k^{-1} helyett annak egy szimmetrikus pozitív definit G_k közelítését használjuk. Legyen G_1 tetszőleges pozitív definit mátrix, leggyakrabban az egységmátrixot választják. A k -adik iterációs lépés az alábbi:

- legyen $\mathbf{s}_k = -G_k \mathbf{g}_k$;
- alkalmazzunk \mathbf{s}_k irányában egyenes mentén történő keresést az \mathcal{E} célfüggvény minimumhelyének megtalálására, a minimumhely lesz az új közelítés: $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{s}_k$;
- aktualizáljuk a G_k mátrixot: G_{k+1} .

Ez a *kvázi Newton-módszer* általános alakja.

A kvázi Newton-módszer az alábbi előnyökkel rendelkezik: csak elsőrendű deriváltak kellenek, a G_k mátrix pozitív definit, iterációs lépésenként $O(m^2)$ szorzás kell (m -változós esetben). Ezzel szemben a Newton-módszerhez másodrendű deriváltak kellenek, a H_k mátrix lehet indefinit, iterációs lépésenként $O(m^3)$ szorzás kell.

A kvázi Newton-módszer nehézsége a G_k mátrix aktualizálása. Jelölje $\boldsymbol{\delta}_k = \mathbf{w}_{k+1} - \mathbf{w}_k$ a változó differenciáját, $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ pedig a gradiens differenciáját. Ekkor a Taylor-formulából

$$\boldsymbol{\gamma}_k = H_k \boldsymbol{\delta}_k + o(\|\boldsymbol{\delta}_k\|).$$

Mivel G_k a H_k^{-1} közelítése, így a fenti egyenlőségből realiztikus feltétel

$$G_{k+1} \boldsymbol{\gamma}_k = \boldsymbol{\delta}_k. \quad (6.18)$$

Ezt nevezzük *kvázi Newton-feltételnek*. Több, ezt teljesítő módszer ismeretes. Most néhány ezt teljesítő, de egyszerű formulát ismertetünk.

Legyen

$$G_{k+1} = G_k + a \mathbf{u} \mathbf{u}^\top.$$

Azaz egy egyrangú mátrixszal (diáddal) korrigáltunk. Mivel a (6.18) feltételnek teljesülnie kell, így

$$G_k \boldsymbol{\gamma}_k + a \mathbf{u} \mathbf{u}^\top \boldsymbol{\gamma}_k = \boldsymbol{\delta}_k.$$

Innen látszik, hogy az \mathbf{u} vektor szükségképpen a $\boldsymbol{\delta}_k - G_k \boldsymbol{\gamma}_k$ vektor skalárszorosa. Tehát ha $\mathbf{u} = \boldsymbol{\delta}_k - G_k \boldsymbol{\gamma}_k$ választással élünk, akkor $\mathbf{a} \mathbf{u}^\top \boldsymbol{\gamma}_k = 1$, amiből már a kifejezhető. Ezzel meg is kaptuk az *elsőrendű formulát*:

$$G_{k+1} = G_k + \frac{(\boldsymbol{\delta}_k - G_k \boldsymbol{\gamma}_k)(\boldsymbol{\delta}_k - G_k \boldsymbol{\gamma}_k)^\top}{(\boldsymbol{\delta}_k - G_k \boldsymbol{\gamma}_k)^\top \boldsymbol{\gamma}_k}.$$

Ez a formula Broyden, Davidon, Fiacco, McCormick, Murtagh, Sargent és Wolfe nevéhez köthető. Ez a formula jól viselkedik kvadratikusan célfüggvény esetén, de pl. nem minden esetben tartja meg a G_k mátrix pozitív definit voltát (l. [14]).

Ettől rugalmasabb a másodrendű formula, amelynek alakja

$$G_{k+1} = G_k + \mathbf{a} \mathbf{u} \mathbf{u}^\top + \mathbf{b} \mathbf{v} \mathbf{v}^\top.$$

Azaz kettő darab egyrangú mátrixszal (diáddal) korrigáltunk. Mivel a (6.18) feltételnek teljesülnie kell, így

$$G_k \boldsymbol{\gamma}_k + \mathbf{a} \mathbf{u} \mathbf{u}^\top \boldsymbol{\gamma}_k + \mathbf{b} \mathbf{v} \mathbf{v}^\top \boldsymbol{\gamma}_k = \boldsymbol{\delta}_k.$$

Itt már \mathbf{u} és \mathbf{v} nincs egyértelműen meghatározva. Azonban $\mathbf{u} = \boldsymbol{\delta}_k$ és $\mathbf{v} = -G_k \boldsymbol{\gamma}_k$ nyilvánvaló választás, amely esetén $\mathbf{a} \mathbf{u}^\top \boldsymbol{\gamma}_k = 1$ és $\mathbf{b} \mathbf{v}^\top \boldsymbol{\gamma}_k = -1$ már meghatározza a -t és b -t. Az adódó *másodrendű formula*:

$$G_{k+1} = G_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^\top}{\boldsymbol{\delta}_k^\top \boldsymbol{\gamma}_k} - \frac{G_k \boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^\top G_k}{\boldsymbol{\gamma}_k^\top G_k \boldsymbol{\gamma}_k}. \quad (6.19)$$

Ez a *Davidon–Fletcher–Powell-formula*, [14]. Ez a formula számos jó tulajdonsággal rendelkezik: megőrzi G_k pozitív definit voltát, szuperlineárisan konvergál. További jó tulajdonságai vannak kvadratikusan és szigorúan konvex célfüggvények esetén (l. [14]).

Ezen formulát átírhatjuk az alábbi alakba. Jelölje B_k a G_k mátrix inverzét, azaz a Hesse-mátrix közelítését. Ekkor

$$B_{k+1} = B_k + \left(1 + \frac{\boldsymbol{\delta}_k^\top B_k \boldsymbol{\delta}_k}{\boldsymbol{\gamma}_k^\top \boldsymbol{\delta}_k} \right) \frac{\boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^\top}{\boldsymbol{\gamma}_k^\top \boldsymbol{\delta}_k} - \frac{\boldsymbol{\gamma}_k \boldsymbol{\delta}_k^\top B_k + B_k \boldsymbol{\delta}_k \boldsymbol{\gamma}_k^\top}{\boldsymbol{\gamma}_k^\top \boldsymbol{\delta}_k}. \quad (6.20)$$

Ezt a Woodbury-egyenlőség felhasználásával kaphatjuk meg.

Azonban napjainkban a Davidon–Fletcher–Powell-formula már nem preferált, vannak hatékonyabb formulák is. Alkalmazott viszont a *Broyden–Fletcher–Goldfarb–Shanno-formula*, amelynek alakja:

$$G_{k+1} = G_k + \left(1 + \frac{\boldsymbol{\gamma}_k^\top G_k \boldsymbol{\gamma}_k}{\boldsymbol{\delta}_k^\top \boldsymbol{\gamma}_k} \right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^\top}{\boldsymbol{\delta}_k^\top \boldsymbol{\gamma}_k} - \frac{\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^\top G_k + G_k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^\top}{\boldsymbol{\delta}_k^\top \boldsymbol{\gamma}_k}. \quad (6.21)$$

Ezen formula motivációja az alábbi. Jelölje ismét B_k a G_k mátrix inverzét, azaz a Hesse-mátrix közelítését. Ekkor a Woodbury-egyenlőség felhasználásával az alábbi adódik.

$$B_{k+1} = B_k + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k} - \frac{B_k \delta_k \delta_k^\top B_k}{\delta_k^\top B_k \delta_k}. \quad (6.22)$$

Láthatjuk, hogy (6.22) a (6.19) egyenlőség duálisa, azaz abból a $B \leftrightarrow G$ és $\gamma \leftrightarrow \delta$ cserével adódik. Ugyancsak duálisa (6.20) a (6.21) egyenlőségnek.

További formulák kaphatóak az alábbi, a ϕ valós paramétertől függő családban:

$$G_k^\phi = (1 - \phi) G_k^{\text{DFP}} + \phi G_k^{\text{BFGS}},$$

ahol G_k^{DFP} a Davidon–Fletcher–Powell-formulából, G_k^{BFGS} pedig a Broyden–Fletcher–Goldfarb–Shanno-formulából adódik. Ez az ún. *Broyden-család*.

Végül tekintsük át a leggyakrabban használt Broyden–Fletcher–Goldfarb–Shanno-formula algoritmusát. Tehát az \mathcal{E} többváltozós célfüggvény minimumát keressük. Jelölje \mathbf{g}_k a gradiens k -edik közelítését, B_k a Hesse-mátrix k -edik közelítését, \mathbf{w}_k a minimumhely k -edik közelítését. Legyen \mathbf{w}_0 a kezdeti közelítés, B_0 a közelítő Hesse-mátrix kezdeti értéke (általában az egységmátrix). Határozzuk meg az \mathcal{E} célfüggvény \mathbf{g}_0 gradiensét a \mathbf{w}_0 pontban.

Az iterációs lépések:

- Oldjuk meg a $B_k \mathbf{s} = -\mathbf{g}_k$ egyenletet, a megoldást jelölje \mathbf{s}_k .
- Egyenes mentén történő keresését alkalmazzunk. Az aktuális \mathbf{w}_k pontból \mathbf{s}_k irányában keressük meg az \mathcal{E} célfüggvény minimumhelyét. Ez a pont lesz a minimumhely következő közelítése: \mathbf{w}_{k+1} .
- Számítsuk ki az \mathcal{E} célfüggvény \mathbf{g}_{k+1} gradiensét a \mathbf{w}_{k+1} pontban.
- Legyen $\delta_k = \mathbf{w}_{k+1} - \mathbf{w}_k$ a változó differenciája, $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ pedig a gradiens differenciája.
- Legyen

$$B_{k+1} = B_k + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k} - \frac{B_k \delta_k \delta_k^\top B_k}{\delta_k^\top B_k \delta_k}$$

a közelítő Hesse-mátrix aktualizálása.

Az MLP esetén a gradienst az error-backpropagation eljárással számítjuk ki. A BFGS-formula a gyakorlatban jól működik. Számos jó tulajdonságát igazolták, konvergenciáját bizonyították (l. [14]).

6.3.4. Levenberg–Marquardt-módszerek

A Levenberg–Marquardt-módszereket szokták a korlátozott lépésű módszerek, illetve a megbízhatósági tartomány módszerek között tárgyalni. Azaz úgy tekintjük, hogy egy bizonyos tartományon a másodfokú Taylor-közelítés elég jó, és ezen a tartományon belül keressük a közelítő polinom minimumát. Ebben a részben Fletcher [14], valamint Nocedal és Wright [34] könyve alapján tekintjük át a Levenberg–Marquardt-módszereket. Kezdjük a *korlátozott lépésű módszerekkel* (restricted step method). Tehát az \mathcal{E} többváltozós célfüggvény minimumát keressük. A rövidség kedvéért jelölje \mathbf{g}_k a gradiens k -adik értékét, H_k a Hesse-mátrix k -adik értékét, \mathbf{w}_k a minimumhely k -adik közelítését. Tekintsük az

$$\mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}) \sim \mathcal{E}(\mathbf{w}_k) + \mathbf{g}_k^\top \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top H_k \boldsymbol{\delta} = q_k(\boldsymbol{\delta}) \quad (6.23)$$

Taylor-közelítést a \mathbf{w}_k pont körül. Tételezzük fel, hogy a fenti közelítés kielégítő a \mathbf{w}_k pont egy bizonyos környezetében (ez a tartomány a *megbízhatósági tartomány*, azaz trust-region). Ebben keressük a q_k másodfokú közelítés minimumát. Két szempont között kell egyensúlyoznunk: a környezet legyen minél tágabb, de a választott környezetben (6.23) legyen elég jó közelítés.

A prototípus algoritmus a következő (LM0). Tegyük fel, hogy már $k - 1$ lépést megtettünk.

- (i) Adott \mathbf{w}_k és $h_k > 0$, számítsuk ki az \mathcal{E} célfüggvény \mathbf{w}_k -beli gradiensét és Hesse-mátrixát: \mathbf{g}_k , H_k ;

- (ii) oldjuk meg a

$$\min_{\|\boldsymbol{\delta}\| \leq h_k} q_k(\boldsymbol{\delta})$$

minimum problémát, a megoldás legyen $\boldsymbol{\delta}_k$;

- (iii) számítsuk ki az $\mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}_k)$ függvényértéket és a (6.23) közelítés jóságát mérő

$$r_k = \frac{\mathcal{E}(\mathbf{w}_k) - \mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}_k)}{q_k(\mathbf{0}) - q_k(\boldsymbol{\delta}_k)}$$

hányadost;

- (iv) ha $r_k < 0.25$, akkor legyen $h_{k+1} = \|\boldsymbol{\delta}_k\|/4$,
ha $r_k > 0.75$ és $\|\boldsymbol{\delta}_k\| = h_k$, akkor legyen $h_{k+1} = 2h_k$,
egyébként legyen $h_{k+1} = h_k$;
- (v) ha $r_k \leq 0$, akkor legyen $\mathbf{w}_{k+1} = \mathbf{w}_k$, egyébként legyen $\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\delta}_k$.

Az algoritmus nem túlságosan érzékeny a benne szereplő konstansok megváltoztatására. Az algoritmus általános feltételek mellett konvergál (l. [14]).

A fentiekben a $\|\delta_k\|$ norma tetszőleges volt. Amennyiben a szokásos euklideszi normát használjuk, akkor további jó tulajdonságok is adódnak. Ezek az alábbi tételen alapulnak (l. [14], Theorem 5.2.1).

6.15. Tétel. δ_k a

$$\min_{\delta^\top \delta \leq h_k^2} \left(\mathcal{E}(\mathbf{w}_k) + \mathbf{g}_k^\top \delta + \frac{1}{2} \delta^\top H_k \delta \right) \quad (6.24)$$

minimum problémának akkor és csak akkor globális megoldása, ha létezik olyan $\nu \geq 0$ szám, melyre

$$\begin{aligned} (H_k + \nu I) \delta_k &= -\mathbf{g}_k, \\ \nu(\delta_k^\top \delta_k - h_k^2) &= 0, \end{aligned}$$

$H_k + \nu I$ pedig pozitív szemidefinit (itt I egységmátrix). Ha még $H_k + \nu I$ pozitív definit volta is fennáll, akkor δ_k egyetlen megoldása a (6.24) minimum problémának.

Ez alapján a tétel alapján a korábbi algoritmus a következőképp módosul. (Ezt az algoritmust jelölje (LM1).)

- (i) Adott \mathbf{w}_k és $\nu_k > 0$, számítsuk ki az \mathcal{E} célfüggvény \mathbf{w}_k -beli gradiensét és Hesse-mátrixát: \mathbf{g}_k, H_k ;
- (ii) állapítsuk meg, hogy $H_k + \nu_k I$ pozitív definit-e: ha nem, akkor legyen $\nu_k = 4\nu_k$, és ismételjük ezt addig, amíg pozitív definit lesz a mátrixunk;
- (iii) oldjuk meg a

$$(H_k + \nu_k I) \delta = -\mathbf{g}_k \quad (6.25)$$

egyenletet, a megoldás legyen δ_k ;

- (iv) számítsuk ki az $\mathcal{E}(\mathbf{w}_k + \delta_k)$ függvényértéket és a (6.23) közelítés jóságát mérő

$$r_k = \frac{\mathcal{E}(\mathbf{w}_k) - \mathcal{E}(\mathbf{w}_k + \delta_k)}{q_k(\mathbf{0}) - q_k(\delta_k)}$$

hányadost;

- (v) ha $r_k < 0.25$, akkor legyen $\nu_{k+1} = 4\nu_k$,
 ha $r_k > 0.75$, akkor legyen $\nu_{k+1} = \nu_k/2$,
 egyébként legyen $\nu_{k+1} = \nu_k$;

(vi) ha $r_k \leq 0$, akkor legyen $\mathbf{w}_{k+1} = \mathbf{w}_k$, egyébként legyen $\mathbf{w}_{k+1} = \mathbf{w}_k + \delta_k$.

A ν_1 kezdeti érték lehet tetszőleges pozitív szám.

Abban az esetben, amikor a négyzetes hibafüggvényt kell minimalizálni, a Levenberg–Marquardt-módszernek az alábbi speciális változatát szokták alkalmazni. (Ez tekinthető az eredeti Levenberg–Marquardt-módszernek.) Legyen most tehát

$$\mathcal{E}(\mathbf{w}) = \frac{1}{2} \|\mathbf{e}(\mathbf{w})\|^2 = \frac{1}{2} \sum_{i=1}^p e_i^2(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^p (d_i - y_i(\mathbf{w}))^2$$

a szokásos négyzetes hibafüggvény. Itt $\mathbf{e}(\mathbf{w}) = (e_1(\mathbf{w}), \dots, e_p(\mathbf{w}))^\top$ a p -dimenziós hiba vektor, $\mathbf{w} = (w_1, \dots, w_m)^\top$ pedig m -dimenziós változó. $J(\mathbf{w})$ jelölje a $p \times m$ -es Jacobi-mátrixot:

$$J(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_1(\mathbf{w})}{\partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_p(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_p(\mathbf{w})}{\partial w_m} \end{bmatrix}.$$

Ezekkel az \mathcal{E} gradiense:

$$\mathbf{g}(\mathbf{w}) = (J(\mathbf{w}))^\top \mathbf{e}(\mathbf{w}),$$

Hesse-mátrixa pedig:

$$H(\mathbf{w}) = (J(\mathbf{w}))^\top J(\mathbf{w}) + \sum_{j=1}^p e_j(\mathbf{w}) \nabla^2 e_j(\mathbf{w}),$$

ahol

$$\nabla^2 e_j(\mathbf{w}) = \left(\frac{\partial^2 e_j(\mathbf{w})}{\partial w_k \partial w_l} \right)_{k,l=1}^m$$

az e_j többváltozós függvény második deriváltjaiból álló Hesse-mátrix. Azonban a második deriváltak kiszámítása általában nehézségekbe ütközik. Viszont ezt megspórolhatjuk úgy, hogy a fenti képletben a második tagot elhagyjuk. Ez valójában nem okoz túlságosan nagy problémát, hisz a minimumhely közelében az $e_j(\mathbf{w})$ hibák kicsik. Tehát a fenti (LM1) algoritmust úgy kell módosítani, hogy a (6.25) egyenletben H_k helyett $J_k^\top J_k$ -t, \mathbf{g}_k helyett pedig $J_k^\top \mathbf{e}_k$ -t írunk. Itt $J_k = J(\mathbf{w}_k)$, $\mathbf{e}_k = \mathbf{e}(\mathbf{w}_k)$ a Jacobi-mátrix, illetve a hiba vektor értéke a k -adik közelítés esetén. Természetesen ekkor a közelítő Taylor-polinomot is az alábbiak szerint módosítani kell:

$$q_k(\delta) = \mathcal{E}(\mathbf{w}_k) + \mathbf{e}_k^\top J_k \delta + \frac{1}{2} \delta^\top J_k^\top J_k \delta.$$

Tehát az így kapott algoritmus a következő.

(i) Adott \mathbf{w}_k és $\nu_k > 0$, számítsuk ki az \mathbf{e} hiba vektor \mathbf{w}_k -beli Jacobi-mátrixát: J_k ;

(ii) oldjuk meg a

$$(J_k^\top J_k + \nu_k I) \boldsymbol{\delta} = -J_k^\top \mathbf{e}_k \quad (6.26)$$

egyenletet, a megoldás legyen $\boldsymbol{\delta}_k$;

(iii) számítsuk ki az $\mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}_k)$ függvényértéket és a közelítés jóságát mérő

$$r_k = \frac{\mathcal{E}(\mathbf{w}_k) - \mathcal{E}(\mathbf{w}_k + \boldsymbol{\delta}_k)}{q_k(\mathbf{0}) - q_k(\boldsymbol{\delta}_k)}$$

hányadost;

(v) ha $r_k < 0.25$, akkor legyen $\nu_{k+1} = 4\nu_k$,
 ha $r_k > 0.75$, akkor legyen $\nu_{k+1} = \nu_k/2$,
 egyébként legyen $\nu_{k+1} = \nu_k$;

(vi) ha $r_k \leq 0$, akkor legyen $\mathbf{w}_{k+1} = \mathbf{w}_k$, egyébként legyen $\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\delta}_k$.

A fenti algoritmusnak számos változata van, azoknak pedig számos implementációja. Ezekről, valamint a Levenberg–Marquardt-algoritmus jó tulajdonságairól lásd a Fletcher [14], továbbá Nocedal és Wright [34] műveket. A Levenberg–Marquardt-algoritmus a gyakorlatban bevált módszer.

6.3.5. A lineáris modell

A statisztikában széles körben használatos lineáris modellre vonatkozó néhány tény hasznos lesz az alábbiakban. A lineáris modell definíciója a statisztikában használatos jelölésekkel

$$\boxed{\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}}, \quad (6.27)$$

ahol \mathbf{Y} az n -dimenziós megfigyelés vektor, X a magyarázó változók $n \times p$ méretű, nem véletlen, megfigyelt mátrixa, $\boldsymbol{\beta}$ a p -dimenziós ismeretlen paraméter, $\boldsymbol{\varepsilon}$ pedig nem megfigyelhető n -dimenziós véletlen vektor (hiba).

Általában $n \gg p$, ezt szükség esetén fel fogjuk tenni. A gyakorlatban p a magyarázó változók száma, n pedig a megfigyelt objektumok száma, tehát $n \gg p$ ésszerű feltétel.

A $\boldsymbol{\beta}$ paraméter vektort a legkisebb négyzetek módszerével becsüljük. Ha a várható érték $\mathbb{E}\boldsymbol{\varepsilon} = \mathbf{0}$ és a szórásmátrix $\text{var}(\boldsymbol{\varepsilon}) = \sigma^2 I$ (σ^2 ismeretlen paraméter), akkor homoszkedasztikus esetről beszélünk. Ekkor a közönséges

legkisebb négyzetes becslést (OLS=Ordinary Least Squares) alkalmazzuk β -ra: ez lesz $\hat{\beta}$.

Legyen tehát $\hat{\beta}$ az $\|Y - X\beta\|^2$ -et minimalizáló vektor. (Itt $\|\cdot\|$ a norma \mathbb{R}^n -ben.) Jelölje P_F az X oszlopai által generált F altérre való merőleges vetítést.

6.16. Tétel. $\hat{\beta}$ akkor és csak akkor legkisebb négyzetes becslés, ha

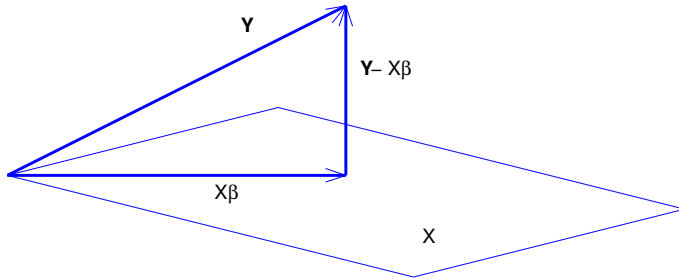
$$X\hat{\beta} = P_F Y.$$

$\hat{\beta}$ akkor és csak akkor legkisebb négyzetes becslés, ha $\hat{\beta}$ az

$$\boxed{X^T X \beta = X^T Y}$$

normálegyenlet megoldása.

Bizonyítás. Tekintsük a 6.6. ábrát. Y -hoz az X oszlopai által generált altér



6.6. ábra. Vektor és altér távolsága

mely $X\beta$ eleme lesz legközelebb? Éppen az Y vetülete, azaz $P_F Y$. Így $X\hat{\beta} = P_F Y$.

$\|Y - X\beta\|^2$ mikor a legkisebb? Ha $Y - X\beta$ éppen az Y ortogonális komplementere az F altérre vonatkozóan. Azaz $Y - X\beta$ merőleges X minden oszlopára, tehát

$$X^T Y - X^T X \beta = 0,$$

vagyis

$$X^T X \hat{\beta} = X^T Y.$$

□

6.3.6. A Gauss–Newton-módszer

A Gauss–Newton-módszer általános leírása megtalálható Nocedal és Wright [34] 10. fejezetében. Itt csak egy rövid bevezetést adunk. Ez a módszer abban

az esetben alkalmazható, amikor a négyzetes hibafüggvényt kell minimalizálni, azaz

$$\|e(\mathbf{w})\|^2 = \sum_{i=1}^p e_i^2(\mathbf{w}) = \sum_{i=1}^p (d_i - y_i(\mathbf{w}))^2.$$

Itt $e(\mathbf{w}) = (e_1(\mathbf{w}), \dots, e_p(\mathbf{w}))^\top$ a hiba vektor, \mathbf{w} pedig m -dimenziós vektor. Az $\mathbf{f}(\mathbf{x}) \simeq \mathbf{f}(\mathbf{x}_0) + \mathbf{f}'(\mathbf{x})(\mathbf{x} - \mathbf{x}_0)$ alakú Taylor-formulát alapul véve felírhatjuk az

$$e(\mathbf{w}) \simeq e(\mathbf{w}(n)) + \underbrace{\left(\frac{\partial e(\mathbf{w}(n))}{\partial \mathbf{w}} \right)}_{J(n)} (\mathbf{w} - \mathbf{w}(n))$$

lineáris közelítést. Itt $J(n)$ a $p \times m$ -es Jacobi-mátrix:

$$J(n) = \begin{bmatrix} \frac{\partial e_1(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_1(\mathbf{w})}{\partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_p(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_p(\mathbf{w})}{\partial w_m} \end{bmatrix}.$$

A fenti egyenlet átrendezésével

$$\underbrace{e(\mathbf{w}(n))}_{\mathbf{Y}} = \underbrace{-J(n)}_X \underbrace{(\mathbf{w} - \mathbf{w}(n))}_{\boldsymbol{\beta}} + \underbrace{e(\mathbf{w})}_{\boldsymbol{\varepsilon}}.$$

Ez természetesen éppen az $\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ lineáris modell, amelynél $\boldsymbol{\beta}$ becslése a $\|\boldsymbol{\varepsilon}\|^2 = \|e(\mathbf{w})\|^2$ minimalizálásával a

$$\hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \mathbf{Y}$$

képlet alapján adódik. Innen pedig már könnyen felírható az igazi \mathbf{w} súly approximáló sorozata:

$$\boxed{\mathbf{w}(n+1) = \mathbf{w}(n) - (J(n)^\top J(n))^{-1} J(n)^\top e(\mathbf{w}(n))}. \quad (6.28)$$

A gyakorlatban egyenes mentén történő keresést kell végrehajtani $\mathbf{s}(n)$ irányban, ahol $\mathbf{s}(n)$ a $J(n)^\top J(n)\mathbf{s} = -J(n)^\top e(\mathbf{w}(n))$ megoldása.

További részleteket lásd Fletcher [14] 6. fejezet.

6.3.7. Lineáris legkisebb négyzetes módszer

Ez a legegyszerűbb szituáció, azaz amikor a kimenet lineáris és a hiba négyzetes. A hibavektor:

$$\mathbf{e}(n) = \mathbf{d}(n) - X(n)\mathbf{w}(n),$$

ahol $\mathbf{d}(n)$ az alábbi $n \times 1$ -es vektor

$$\mathbf{d}(n) = [d(1), d(2), \dots, d(n)]^\top,$$

$X(n)$ pedig az alábbi $n \times m$ -es mátrix

$$X(n) = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)]^\top.$$

Azaz $X(n)$ az első n inputból, $\mathbf{d}(n)$ pedig a megfelelő outputokból áll. A fentiek alapján:

$$\mathbf{d}(n) = X(n)\mathbf{w}(n) + \mathbf{e}(n).$$

Ez éppen a lineáris modell $\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ alakú alapegyenlete. Ekkor azonban tudjuk, hogy a legkisebb négyzetes becslés $\hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \mathbf{Y}$. Ezek alapján

$$\boxed{\mathbf{w}(n+1) = \left(X(n)^\top X(n) \right)^{-1} X(n)^\top \mathbf{d}(n)}. \quad (6.29)$$

Tehát ekkor egyetlen lépésben megkapjuk a minimumot (rögzített számú tanító pont esetén).

Most megmutatjuk, hogy amikor a tanító pontok száma a végtelenhez tart, bizonyos esetekben a lineáris legkisebb négyzetes szűrő határértéke a Wiener-szűrő. Tegyük fel, hogy az $(X(n), \mathbf{d}(n))$ $n \times (m+1)$ -es mátrix sorvektorai egymástól független azonos eloszlású valószínűségi vektor változók (vagy általánosabban egy stacionárius és ergodikus sorozatot alkotnak).

Viszont $X(n)^\top X(n)$ az alábbi diádösszegre bontható

$$\frac{1}{n} X(n)^\top X(n) = \frac{1}{n} \sum_{i=1}^n \mathbf{x}(i) \mathbf{x}(i)^\top \longrightarrow \mathbb{E} \mathbf{x}(1) \mathbf{x}(1)^\top = R_{\mathbf{x}\mathbf{x}}. \quad (6.30)$$

Ez a konvergencia a nagy számok törvénye (illetve az ergodikus tétel) miatt teljesül. Itt $R_{\mathbf{x}\mathbf{x}}$ az $\mathbf{x}(1)$ valószínűségi vektor változó variancia mátrixa (amennyiben nulla a várható érték). Továbbá

$$\frac{1}{n} X(n)^\top \mathbf{d}(n) = \frac{1}{n} \sum_{i=1}^n \mathbf{x}(i) d(i) \longrightarrow \mathbb{E} \mathbf{x}(1) d(1) = \mathbf{r}_{\mathbf{x}d} \quad (6.31)$$

szintén a nagy számok törvénye (illetve az ergodikus tétel) miatt. Itt $\mathbf{r}_{\mathbf{x}d}$ az $\mathbf{x}(1)$ valószínűségi vektor változó és $d(1)$ kovariancia mátrixa (amennyiben nulla a várható érték).

Tehát

$$\lim_{n \rightarrow \infty} \mathbf{w}(n) = R_{\mathbf{x}\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}d}.$$

Ezen utóbbi határérték pedig éppen a Wiener-szűrő.

6.3.8. A Least-Mean-Square (LMS) módszer

Az előző módszer (a lineáris legkisebb négyzetek módszere) esetén az $n + 1$ -edik közelítéshez az első n megfigyelés (adat, tanító pont) által adott információt használtuk. Most viszont nem az n -edik megfigyelésig terjedő összes (azaz n db) eltérés alapján, hanem csupán az n -edik (azaz 1 db) eltérés alapján aktualizáljuk az előző becslést. Tegyük fel, hogy a kimenet lineáris. Ekkor az n -edik eltérés:

$$e(n) = d(n) - \mathbf{x}(n)^\top \mathbf{w}(n),$$

a legkisebb négyzetes hiba pedig:

$$\mathcal{E}(\mathbf{w}(n)) = \frac{1}{2} e(n)^2.$$

Innen

$$\frac{\partial e(n)}{\partial \mathbf{w}(n)} = -\mathbf{x}(n),$$

amiből

$$\frac{\partial \mathcal{E}(\mathbf{w}(n))}{\partial \mathbf{w}(n)} = -e(n) \mathbf{x}(n).$$

Tehát a gradiens vektor becslése:

$$\mathbf{g}(n) = -e(n) \mathbf{x}(n).$$

Felhasználva a gradiens módszer (6.14) képletét azt kapjuk, hogy

$$\boxed{\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta e(n) \mathbf{x}(n)}. \quad (6.32)$$

Ez tehát az LMS (Least-Mean-Square) algoritmus, más néven a Widrow-Hoff-algoritmus. Ennek részletes elemzését lásd Haykin [18] 3. fejezet.

6.3.9. A konjugált gradiens módszer

A gradiens módszer viszonylag egyszerűen megvalósítható, de lassú és megbízhatatlan lehet. A gyakorlatban a gradiens módszert ritkán használják az eredeti formában. Ennek módosítása, a konjugált gradiens módszer gyakran alkalmazott, ezért részletesen elemezzük.

Induljunk ki az átlagos hiba $\mathbf{w}(n)$ körüli Taylor-sorából:

$$\begin{aligned}\bar{\mathcal{E}}(\mathbf{w}(n) + \Delta\mathbf{w}(n)) &= \\ &= \bar{\mathcal{E}}(\mathbf{w}(n)) + \mathbf{g}(\mathbf{w}(n))^\top \Delta\mathbf{w}(n) + \frac{1}{2} \Delta\mathbf{w}(n)^\top H(\mathbf{w}(n)) \Delta\mathbf{w}(n) + \dots,\end{aligned}$$

ahol $\mathbf{g}(\mathbf{w})$ a gradiens vektor, $H(\mathbf{w})$ pedig a Hesse-mátrix. A gradiens módszerénél a súlyvektor megváltozása:

$$\mathbf{w}(n+1) - \mathbf{w}(n) = \Delta\mathbf{w}(n) = -\eta \mathbf{g}(\mathbf{w}(n)).$$

Most először általános értelemben kifejtjük a konjugált gradiens módszert, majd alkalmazzuk a többretegű perceptronra. Ismeretes, hogy hogyan tudjuk minimalizálni az alábbi kvadratikus formát. Legyen

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top A \mathbf{x} - \mathbf{b}^\top \mathbf{x} + c,$$

ahol \mathbf{x} k -dimenziós vektor, A pedig $k \times k$ -s pozitív definit szimmetrikus mátrix. A $\mathbf{0} = \mathbf{f}'(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ feltétel alapján a minimum $\mathbf{x} = A^{-1}\mathbf{b}$ -nél lesz. Most tehát az $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszert kell megoldanunk. Legyenek az $\mathbf{x}_0, \mathbf{x}_1, \dots$ vektorok az \mathbf{x} megoldás közelítései. A közelítés hibája:

$$\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}.$$

A közelítésből adódó *reziduál* definíciója: $\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i$. Erre

$$\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i = A\mathbf{x} - A\mathbf{x}_i = -A\mathbf{e}_i = \underbrace{-\mathbf{f}'(\mathbf{x}_i)}_{\text{negatív gradiens}}, \quad (6.33)$$

ahol az $\mathbf{f}'(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ képletet használtuk. Adott \mathbf{x}_i és \mathbf{r}_i esetén határozzuk meg \mathbf{x}_{i+1} -et

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{r}_i \quad (6.34)$$

szerint. Az α_i skalár értéket pedig határozzuk meg úgy, hogy addig lépünk \mathbf{r}_i irányába, amíg az f függvény értéke csökken. Így a minimum feltétele az α_i szerinti deriváltra

$$0 = \frac{df(\mathbf{x}_i + \alpha_i \mathbf{r}_i)}{d\alpha_i} = \mathbf{f}'(\mathbf{x}_i + \alpha_i \mathbf{r}_i)^\top \mathbf{r}_i =$$

$$= (A\mathbf{x}_i + \alpha_i A\mathbf{r}_i - \mathbf{b})^\top \mathbf{r}_i = -\mathbf{r}_i^\top \mathbf{r}_i + \alpha_i \mathbf{r}_i^\top A\mathbf{r}_i,$$

ahol a (6.33) egyenlőséget alkalmaztuk. Ebből

$$\alpha_i = \frac{\mathbf{r}_i^\top \mathbf{r}_i}{\mathbf{r}_i^\top A\mathbf{r}_i}. \quad (6.35)$$

A reziduál definíciója és (6.34) miatt az $i + 1$ -edik reziduál:

$$\mathbf{r}_{i+1} = \mathbf{b} - A\mathbf{x}_{i+1} = \mathbf{r}_i + A\mathbf{x}_i - A\mathbf{x}_i - \alpha_i A\mathbf{r}_i = \mathbf{r}_i - \alpha_i A\mathbf{r}_i.$$

Vizsgáljuk meg, hogy milyen viszonyban vannak egymással a reziduálok. A (6.35) összefüggés és az előző egyenlőség miatt

$$\mathbf{r}_i^\top \mathbf{r}_{i+1} = \mathbf{r}_i^\top \mathbf{r}_i - \alpha_i \mathbf{r}_i^\top A\mathbf{r}_i = 0,$$

azaz

$$\mathbf{r}_i \perp \mathbf{r}_{i+1}.$$

Tehát a következő lépés az előzőre merőlegesen történik. Az eddigieket összefoglalva

$$\boxed{\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{r}_i, \quad \mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i, \quad \alpha_i = \frac{\mathbf{r}_i^\top \mathbf{r}_i}{\mathbf{r}_i^\top A\mathbf{r}_i}.$$

A konjugált gradiens módszerhez szükség van *konjugált irányok* fogalmára. Defináljunk egy szimmetrikus bilineáris formát a következő módon

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top A\mathbf{y}, \quad (6.36)$$

ahol A szimmetrikus, pozitív definit mátrix. Tehát $\langle \cdot, \cdot \rangle$ egy belső szorzat.

Azt mondjuk, hogy az \mathbf{x} és az \mathbf{y} vektorok ($\mathbf{x} \neq \mathbf{y}$) az A szimmetrikus, pozitív definit mátrixra nézve konjugáltak, ha

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top A\mathbf{y} = 0.$$

Más szóval az \mathbf{x} és az \mathbf{y} vektorok A -ortogonálisak, amit így jelölünk:

$$\mathbf{x} \perp_A \mathbf{y}.$$

6.17. Állítás. *Tetszőleges A szimmetrikus, pozitív definit mátrix nullától különböző konjugált vektorai lineárisan függetlenek.*

Bizonyítás. Legyenek $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_l$ nem nulla vektorok az A mátrix konjugált vektorai. Indirekt tegyük fel, hogy nem lineárisan függetlenek, azaz valamelyiküket, pl. \mathbf{x}_0 -t ki tudjuk fejezni a többi vektor lineáris kombinációjaként:

$$\mathbf{x}_0 = \sum_{j=1}^l \alpha_j \mathbf{x}_j.$$

Alkalmazzuk a belső szorzást $A\mathbf{x}_0$ -lal, így kapjuk:

$$\mathbf{x}_0^\top A\mathbf{x}_0 = \sum_{j=1}^k \alpha_j \mathbf{x}_j^\top A\mathbf{x}_0 = 0$$

az A -ortogonalitás miatt. Ez azonban lehetetlen, mivel A pozitív definit, \mathbf{x}_0 pedig nem nulla vektor. Így a konjugált vektorok lineárisan függetlenek. \square

A Gram–Schmidt-féle ortogonalizáció alapján mindig ki tudunk alakítani ortonormált bázist egy lineárisan független vektor rendszerből. Tehát létezik az A konjugált vektoraiból álló $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}$ bázis (k -dimenziós térben dolgozunk).

Most ezen vektorok mentén keressük az $A\mathbf{x} = \mathbf{b}$ egyenlet \mathbf{x} megoldását. Haladjunk egymás után a \mathbf{d}_i bázis vektorok irányában úgy, hogy ezekben az irányokban később már ne kelljen lépni. Tehát az egymás utáni közelítések

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i. \quad (6.37)$$

Legyen $\mathbf{x} - \mathbf{x}_{i+1} \perp_A \mathbf{d}_i$, és így a maradéknak már nem lesz \mathbf{d}_i irányú komponense. Ehhez hogyan válasszuk α_i -t?

$$\mathbf{x} - \mathbf{x}_{i+1} = \underbrace{\mathbf{x} - \mathbf{x}_i}_{-\mathbf{e}_i} - \alpha_i \mathbf{d}_i \perp_A \mathbf{d}_i,$$

azaz

$$-\mathbf{e}_i^\top A\mathbf{d}_i - \alpha_i \mathbf{d}_i^\top A\mathbf{d}_i = 0,$$

amiből (6.33) felhasználásával:

$$\alpha_i = \frac{\mathbf{d}_i^\top \mathbf{r}_i}{\mathbf{d}_i^\top A\mathbf{d}_i}. \quad (6.38)$$

Most a konjugált gradiens módszer algoritmusát fogjuk levezetni. A konjugált irányokat és a gradiens módszert kell egyszerre alkalmazni.

A célunk tehát az $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer \mathbf{x} megoldásának meghatározása. Az $\mathbf{x}_0, \mathbf{x}_1, \dots$ vektorokkal közelítjük \mathbf{x} -et.

Az $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}$ vektor a közelítés hibája, $\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i = -A\mathbf{e}_i$ pedig a reziduál.

A reziduálokból kell meghatároznunk a $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}$ A -konjugált bázist, valamint az \mathbf{x} megoldást. Ezt az alábbi módon valósítjuk meg.

- Legyen \mathbf{x}_0 adott kezdeti közelítés. Ekkor $\mathbf{r}_0 = -A\mathbf{e}_0 = -A(\mathbf{x}_0 - \mathbf{x}) = -A\mathbf{x}_0 + \mathbf{b}$ számolható.
- Ha $\mathbf{r}_0 = \mathbf{0}$, akkor $\mathbf{x} = \mathbf{x}_0$ és az algoritmus ezzel véget ér.

Ha $\mathbf{r}_0 \neq \mathbf{0}$, akkor legyen $\mathbf{d}_0 = \mathbf{r}_0$.

Indukcióval haladunk. Tegyük fel, hogy eljutottunk az i -edik lépésig. Ez azt jelenti, hogy megvan az $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_i$ közelítése \mathbf{x} -nek, megvan az $\mathbf{r}_0, \dots, \mathbf{r}_{i-1}$ reziduálok ($\mathbf{r}_j = \mathbf{b} - A\mathbf{x}_j = A(\mathbf{x} - \mathbf{x}_j) = -A\mathbf{e}_j$, $j = 0, 1, \dots, i-1$), valamint megvan az $\mathbf{r}_0, \dots, \mathbf{r}_{i-1}$ -ből A -ortogonalizálással képzett $\mathbf{d}_0, \dots, \mathbf{d}_{i-1}$ vektor rendszer is. Az \mathbf{x}_j vektorokat pedig úgy képeztük, hogy mindig hozzávettük \mathbf{x}_{j-1} -hez az aktuális \mathbf{d}_{j-1} irányú komponenst. Azaz az \mathbf{e}_j hiba már nem tartalmaz $\mathbf{d}_0, \dots, \mathbf{d}_{j-1}$ irányú komponenst, pontosabban szólva, ezekre A -ortogonális. Ebből tehát

$$\mathbf{r}_j^\top \mathbf{d}_l = -\mathbf{e}_j^\top A \mathbf{d}_l = 0, \quad \text{ha } 0 \leq l \leq j-1. \quad (6.39)$$

Azaz $\mathbf{r}_j \perp \mathbf{d}_0, \dots, \mathbf{d}_{j-1}$. Mivel a $\mathbf{d}_0, \dots, \mathbf{d}_{j-1}$ és az $\mathbf{r}_0, \dots, \mathbf{r}_{j-1}$ vektorrendszerek által generált alterek azonosak, ebből következik:

$$\mathbf{r}_j \perp \mathbf{r}_0, \dots, \mathbf{r}_{j-1}. \quad (6.40)$$

- Azaz, ha már a $\mathbf{d}_0, \dots, \mathbf{d}_{i-1}$ vektorok megvannak, akkor \mathbf{x}_i is megvan (6.37) alapján, továbbá $\mathbf{r}_i = -A\mathbf{e}_i = -A\mathbf{x}_i + \mathbf{b}$ is kiszámolható. Ha $\mathbf{r}_i = \mathbf{0}$, akkor $\mathbf{x} = \mathbf{x}_i$, és az algoritmus ezzel véget ér. Ha $\mathbf{r}_i \neq \mathbf{0}$, akkor a következő indukciós lépést tesszük. (6.39) alapján tudjuk, hogy \mathbf{r}_i független a $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{i-1}$ vektoroktól, tehát csak A -ortogonalizálnunk kell ezekre. Keressük \mathbf{d}_i -t a

$$\mathbf{d}_i = \mathbf{r}_i + \sum_{l=0}^{i-1} \beta_{il} \mathbf{d}_l \quad (6.41)$$

alakban. Alkalmazzuk erre az egyenletre a belső szorzást $\mathbf{d}_j^\top A$ -val a $j < i$ esetben. Azt kapjuk, hogy

$$0 = \mathbf{d}_j^\top A \mathbf{d}_i = \mathbf{d}_j^\top A \mathbf{r}_i + \beta_{ij} \mathbf{d}_j^\top A \mathbf{d}_j,$$

mivel a $\mathbf{d}_0, \dots, \mathbf{d}_{i-1}$ vektorok A -konjugáltak. Innen

$$\beta_{ij} = -\frac{\mathbf{d}_j^\top A \mathbf{r}_i}{\mathbf{d}_j^\top A \mathbf{d}_j}, \quad j = 0, \dots, i-1. \quad (6.42)$$

Most a $j+1$ -edik reziduál az $\mathbf{e}_{j+1} = \mathbf{x}_{j+1} - \mathbf{x} = \mathbf{x}_j + \alpha_j \mathbf{d}_j - \mathbf{x} = \alpha_j \mathbf{d}_j + \mathbf{e}_j$ egyenlőség miatt:

$$\mathbf{r}_{j+1} = -A\mathbf{e}_{j+1} = -A(\mathbf{e}_j + \alpha_j \mathbf{d}_j) = \mathbf{r}_j - \alpha_j A \mathbf{d}_j.$$

Ezt szorozva \mathbf{r}_i^\top -tal:

$$\mathbf{r}_i^\top \mathbf{r}_{j+1} = \mathbf{r}_i^\top \mathbf{r}_j - \alpha_j \mathbf{r}_i^\top A \mathbf{d}_j,$$

ahonnan

$$\mathbf{r}_i^\top A \mathbf{d}_j = \frac{1}{\alpha_j} \left(\mathbf{r}_i^\top \mathbf{r}_j - \mathbf{r}_i^\top \mathbf{r}_{j+1} \right). \quad (6.43)$$

A (6.40) képlet alapján \mathbf{r}_i a kisebb indexű \mathbf{r} -ekre merőleges. Tehát (6.42) és (6.43) alapján $\beta_{ij} = 0$, ha $j = 0, 1, \dots, i-2$.

Továbbá $j = i-1$ esetén a (6.42) és a (6.43) egyenletekből

$$\beta_{ij} = \frac{1}{\alpha_j} \frac{\mathbf{r}_i^\top \mathbf{r}_i}{\mathbf{d}_j^\top A \mathbf{d}_j}.$$

De a (6.38) egyenlet miatt $\alpha_j = \mathbf{d}_j^\top \mathbf{r}_j / \mathbf{d}_j^\top A \mathbf{d}_j$. Végül a fentiekből $j = i-1$ esetén kapjuk, hogy $\beta_{i,i-1} = \mathbf{r}_i^\top \mathbf{r}_i / \mathbf{d}_{i-1}^\top A \mathbf{d}_{i-1}$, amiből (6.41) felhasználásával, valamint abból, hogy az \mathbf{r} -ek ortogonálisak, és a \mathbf{d} -kel azonos alteret feszítenek ki, adódik:

$$\boxed{\beta_{i,i-1} = \frac{\mathbf{r}_i^\top \mathbf{r}_i}{\mathbf{r}_{i-1}^\top \mathbf{r}_{i-1}}}. \quad (6.44)$$

A (6.44) képletet *Fletcher-Reeves-formulának* nevezik.

- Ezt az eljárást addig folytatjuk, amíg a reziduál $\mathbf{0}$ nem lesz, azaz amíg meg nem találjuk a minimumot.

A konjugált gradiens módszer alkalmazása többrétegű perceptronra

Az átlagos hiba $\mathbf{w}(n)$ körüli Taylor-sora:

$$\bar{\mathcal{E}}(\mathbf{w}(n) + \Delta \mathbf{w}(n)) =$$

$$= \bar{\mathcal{E}}(\mathbf{w}(n)) + \mathbf{g}(\mathbf{w}(n))^\top \Delta \mathbf{w}(n) + \frac{1}{2} \Delta \mathbf{w}(n)^\top H(\mathbf{w}(n)) \Delta \mathbf{w}(n) + \dots,$$

ahol $\mathbf{g}(\mathbf{w})$ a gradiens vektor, $H(\mathbf{w})$ pedig a Hesse-mátrix. A gradiens módszer esetén a súly korrekciója a negatív gradiens irányába történik:

$$\Delta \mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n) = -\eta \mathbf{g}(\mathbf{w}(n)).$$

Viszont a konjugált gradiens módszer alkalmazható, hiszen a fenti Taylor-sor alapján közelítőleg egy

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top A \mathbf{x} - \mathbf{b}^\top \mathbf{x} + c$$

alakú kvadratikus formát kell minimalizálni. Tehát úgy fogunk eljárni, hogy a gradienst a korábban megismert error back-propagation eljárással kiszámítjuk, de azt a konjugált gradiens módszer alapján úgy módosítjuk, mintha a közelítő kvadratikus forma minimumát keresnénk.

Jelölje tehát $\mathbf{s}(0), \mathbf{s}(1), \dots, \mathbf{s}(n-1)$ azokat az irányokat, amikben már léptünk. A rekurzív eljárás az alábbi.

1. Inicializálás. Legyen $\mathbf{w}(0)$ a kezdeti súlyvektor. Alkalmazzuk ebben a pontban a back-propagation eljárást a $\mathbf{g}(0)$ gradiens kiszámítására ugyanúgy, ahogyan korábban megismertük. Legyen

$$\mathbf{s}(0) = \mathbf{r}(0) = -\mathbf{g}(0)$$

a negatív gradiens, tehát az az irány, amerre $\mathbf{w}(0)$ -at változtatjuk.

2. Lépés $n-1$ -ről n -re. Ha az $\mathbf{s}(0), \dots, \mathbf{s}(n-1)$ irányok már megvannak, akkor megvan a $\mathbf{w}(n-1)$ súlyvektor is. Keressük az új súlyvektort

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \eta(n-1) \mathbf{s}(n-1)$$

alakban. Úgy határozzuk meg $\eta(n-1)$ -et, hogy $\bar{\mathcal{E}}(\mathbf{w}(n))$ minimális legyen. (Ez ún. direkt keresés, azaz egy egyenes mentén való keresés, ami numerikusan megoldható.)

Ezután a $\mathbf{w}(n)$ pontban back-propagation algoritmussal keressük meg $\bar{\mathcal{E}}$ gradiensét, azaz $\mathbf{g}(n)$ -et. Legyen $\mathbf{r}(n) = -\mathbf{g}(n)$. Ha $\|\mathbf{r}(n)\| \leq \delta \|\mathbf{r}(0)\|$, ahol $\delta > 0$ előre adott kis szám, akkor megállunk. Ellenkező esetben folytatjuk. Viszont a következő lépés nem a $-\mathbf{g}(n)$ negatív gradiens irányába történik, hanem a konjugált gradiens módszer szerint az

$$\mathbf{s}(n) = \mathbf{r}(n) + \beta(n) \mathbf{s}(n-1)$$

irányba, ahol (a *Fletcher–Reeves-formula* szerint)

$$\beta(n) = \frac{\mathbf{r}(n)^\top \mathbf{r}(n)}{\mathbf{r}(n-1)^\top \mathbf{r}(n-1)}.$$

Ezután n -et 1-gyel megnöveljük, és visszatérünk a **2.** lépés elejéhez.

Megjegyezzük, hogy a Fletcher–Reeves-formula helyett használható a *Polak–Ribière-formula* is, ami szerint

$$\beta(n) = \frac{\mathbf{r}(n)^\top (\mathbf{r}(n) - \mathbf{r}(n-1))}{\mathbf{r}(n-1)^\top \mathbf{r}(n-1)}.$$

6.4. Feltételes szélsőérték problémák

Feltételes szélsőérték problémák megoldásakor igen hasznos segédeszköz a *Kuhn–Tucker-féle tétel*, más néven a *Karush–Kuhn–Tucker-féle tétel*, lásd [23] és [26]. A tétel számos változata közül itt a Boyd és Vandenberghe [2] könyvében lévő alakot ismertetjük (lásd még [1]).

Legyenek $f_k(\mathbf{x})$, $k = 0, \dots, m$, d -változós, valós értékű függvények. Feltesszük, hogy ezen függvények értelmezési tartományainak D metszete nem üres.

Az elsődleges optimalizálási probléma (primal problem) az alábbi:

$$\begin{aligned} &\text{minimalizáljuk az } f_0(\mathbf{x}) \text{ függvényt} \\ &\text{az } f_k(\mathbf{x}) \leq 0, \quad k = 1, \dots, m, \text{ feltételek mellett.} \end{aligned}$$

Jelölje p^* az optimális értéket, azaz a feltételek teljesülése esetén $f_0(\mathbf{x})$ infimumát. Legyen

$$L = L(\mathbf{x}, \boldsymbol{\lambda}) = f_0(\mathbf{x}) + \sum_{k=1}^m \lambda_k f_k(\mathbf{x}),$$

a *Lagrange-függvény*, ahol $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_m)$ a *Lagrange-féle multiplikátorok*. Legyen

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\lambda})$$

a Lagrange-féle *duális függvény*. Könnyű látni, hogy g mindig konkáv. Illetve, hogy tetszőleges $\boldsymbol{\lambda} \geq \mathbf{0}$ esetén $g(\boldsymbol{\lambda}) \leq p^*$. Ebből az egyenlőtlenségből származó legjobb alsó becslést szeretnénk elérni. Ez vezet az alábbi optimalizálási feladatra.

A duális optimalizálási probléma (Lagrange dual problem) az alábbi:

Maximalizáljuk a $g(\boldsymbol{\lambda})$ függvényt
a $\lambda_i \geq 0, \quad i = 1, \dots, m,$ feltételek mellett.

Jelölje d^* az optimális értéket, azaz a feltételek teljesülése esetén $g(\boldsymbol{\lambda})$ szuprémumát. Az előzőek alapján $d^* \leq p^*$. A $p^* - d^* \geq 0$ mennyiséget *optimális dualitási résnek* nevezzük (optimal duality gap).

A következő két tételben tegyük fel, hogy az $f_k(\mathbf{x})$, $k = 0, \dots, m$, függvények differenciálhatóak. Ekkor ezen függvények értelmezési tartományai nyíltak, azaz D nyílt halmaz.

6.18. Tétel. *Legyen \mathbf{x}^* az elsődleges optimalizálási probléma optimális megoldása, $\boldsymbol{\lambda}^* = (\lambda_1^*, \dots, \lambda_m^*)$ pedig a duális optimalizálási probléma optimális megoldása, és tegyük fel, hogy a dualitási rés nulla. Ekkor teljesülnek a Kuhn–Tucker-feltételek, azaz*

$$\begin{aligned} f_k(\mathbf{x}^*) &\leq 0, & k &= 1, \dots, m, \\ \lambda_k^* &\geq 0, & k &= 1, \dots, m, \\ \lambda_k^* f_k(\mathbf{x}^*) &= 0, & k &= 1, \dots, m, \\ \frac{df_0(\mathbf{x}^*)}{d\mathbf{x}} + \sum_{k=1}^m \lambda_k^* \frac{df_k(\mathbf{x}^*)}{d\mathbf{x}} &= \mathbf{0}. \end{aligned}$$

Bizonyítás. A négy feltétel közül az első kettő nyilvánvaló. Az egyes mennyiségek definíciójából:

$$g(\boldsymbol{\lambda}) \leq L(\mathbf{x}, \boldsymbol{\lambda}) \leq f_0(\mathbf{x})$$

minden \mathbf{x} és $\boldsymbol{\lambda} \geq \mathbf{0}$ esetén. Így

$$d^* = g(\boldsymbol{\lambda}^*) \leq L(\mathbf{x}, \boldsymbol{\lambda}^*) \leq f_0(\mathbf{x}).$$

Innen

$$d^* = g(\boldsymbol{\lambda}^*) \leq L(\mathbf{x}^*, \boldsymbol{\lambda}^*) \leq f_0(\mathbf{x}^*) = p^*.$$

Mivel a dualitási rés nulla, így ebben az egyenlőtlenségben végig egyenlőség áll. Így a tételbeli négy feltétel közül a harmadik közvetlenül adódik. Viszont az is látszik, hogy \mathbf{x}^* az $L(\mathbf{x}, \boldsymbol{\lambda}^*)$ függvény minimumhelye, azaz ott a derivált nulla, így adódik a negyedik feltétel. \square

A fenti tétel tehát szükséges feltételt mond ki. Ezek a feltételek konvex függvények esetén elégségesek is.

6.19. Tétel. *Legyenek az $f_k(\mathbf{x})$, $k = 0, \dots, m$, függvények konvexek. Tegyük fel, hogy az $\tilde{\mathbf{x}}$ és a $\tilde{\lambda}_1, \dots, \tilde{\lambda}_m$ pontok teljesítik a Kuhn–Tucker-feltételeket, azaz*

$$f_k(\tilde{\mathbf{x}}) \leq 0, \quad k = 1, \dots, m, \quad (6.45)$$

$$\tilde{\lambda}_k \geq 0, \quad k = 1, \dots, m, \quad (6.46)$$

$$\tilde{\lambda}_k f_k(\tilde{\mathbf{x}}) = 0, \quad k = 1, \dots, m, \quad (6.47)$$

$$\frac{df_0(\tilde{\mathbf{x}})}{d\mathbf{x}} + \sum_{k=1}^m \tilde{\lambda}_k \frac{df_k(\tilde{\mathbf{x}})}{d\mathbf{x}} = \mathbf{0}. \quad (6.48)$$

Ekkor $\tilde{\mathbf{x}}$ az elsődleges optimalizálási probléma optimális megoldása, $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_m)$ pedig a duális optimalizálási probléma optimális megoldása, és a dualitási rés nulla.

Bizonyítás. Az $L(\mathbf{x}, \tilde{\boldsymbol{\lambda}})$ függvény \mathbf{x} -ben konvex. Mivel a deriváltja $\tilde{\mathbf{x}}$ -ben eltűnik az utolsó Kuhn–Tucker-feltétel miatt, így ott minimuma van. Tehát a harmadik Kuhn–Tucker-feltételt felhasználva

$$g(\tilde{\boldsymbol{\lambda}}) = L(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}) = f_0(\tilde{\mathbf{x}}).$$

Azaz $\tilde{\mathbf{x}}$ és $\tilde{\boldsymbol{\lambda}}$ nulla dualitási rést ad, így ezek primál, illetve duál optimális megoldások. \square

Tegyük fel, hogy az $f_k(\mathbf{x})$, $k = 0, \dots, m$, függvények konvexek és differenciálhatóak. Ekkor a (6.45)–(6.48) Kuhn–Tucker-feltételek szükségesek és elegendők az optimalitáshoz, azaz ahhoz, hogy $\tilde{\mathbf{x}}$ az elsődleges optimalizálási probléma optimális megoldása, $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_m)$ pedig a duális optimalizálási probléma optimális megoldása legyen nulla dualitási réssel.

A Slater-feltétel azt jelenti, hogy létezik $\underline{\mathbf{x}}$ pont a D halmaz belsejében úgy, hogy $f_i(\underline{\mathbf{x}}) < 0$, $i = 1, \dots, m$ (tehát szigorú egyenlőtlenség teljesül).

Ismeretes, hogy amikor az f_0, f_1, \dots, f_m függvények konvexek, akkor a Slater-feltétel garantálja, hogy a dualitási rés 0 legyen. Tehát, ha olyan esetben alkalmazzuk a 6.19. Tételt, amikor a Slater-feltétel teljesül, akkor garantált a (6.45)–(6.48) feltételek teljesülése valamely $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}})$ pontra. Tehát csak ezen $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}})$ pont (numerikus) meghatározása a feladat.

Az optimális megoldás keresése megegyezik az $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}})$ nyeregpont keresésével:

$$\min_{\mathbf{x}} L(\mathbf{x}, \tilde{\boldsymbol{\lambda}}) = L(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}) = \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\tilde{\mathbf{x}}, \boldsymbol{\lambda}).$$

Pontosabban szólva $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}})$ akkor és csak akkor nyeregpontja L -nek, ha $\tilde{\mathbf{x}}$ primál, $\tilde{\boldsymbol{\lambda}}$ pedig duális optimális pont 0 dualitási réssel. A részleteket lásd Boyd és Vandenberghe [2] könyvében.

6.5. Feladatok

1. Igazoljuk, hogy az $A^\top A$ mátrix szorzat felbomlik diádösszegre!
2. Igazoljuk a Woodbury-egyenlőség általános alakját! Legyenek A , B , C_1 , C_2 és D alkalmas méretű mátrixok. Ha

$$A = B + C_1 D C_2^\top,$$

akkor

$$A^{-1} = B^{-1} - B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} C_2^\top B^{-1}.$$

Ennek speciális esete a Sherman–Morrison-formula, amikor $D = 1$.

3. A Woodbury-egyenlőség felhasználásával vezessük le a (6.21) egyenlőségből a (6.22) egyenlőséget! Hasonló módon vezessük le a (6.19) egyenlőségből a (6.20) egyenlőséget!
4. Igazoljuk, hogy a Gauss-Newton-módszer megkapható a Newton-módszerből, ha a Hesse-mátrixra a $H \approx J^\top J$ közelítést alkalmazzuk, a gradiens vektor pedig $\mathbf{g} = J^\top \mathbf{e}$.
5. Igazoljuk a (6.30) és a (6.31) képletekben szereplő konvergenciákat!
6. Írjunk programot a gradiens, a konjugált gradiens, a kvázi Newton-(BFGS) és a Levenberg–Marquardt-módszerre!
7. Ábrázoljuk az

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

ún. Rosenbrock-függvényt! Állapítsuk meg, hogy ennek abszolút minimuma van az $(x_1, x_2) = (1, 1)$ pontban!

8. Alkalmazzuk a minimalizáló eljárásokat kétváltozós függvényekre! Kezdjük „egyszerű” alakú függvényekkel, majd folytassuk a Rosenbrock-függvénnyel! Ábrázoljuk a minimumot közelítő sorozatot a síkon!

7. fejezet

Feladatok megoldása, útmutatások

2.7. Feladatok megoldása

1. Szakaszonként lineáris aktivációs függvény:

$$\varphi(x) = \begin{cases} 0, & \text{ha } x < -1, \\ x, & \text{ha } -1 \leq x < 1, \\ 1, & \text{ha } x \geq 1. \end{cases}$$

Arcus tangens függvény:

$$\varphi(x) = \frac{2}{\pi} \arctan(x).$$

Mindkét függvény teljesíti a $\lim_{x \rightarrow -\infty} \varphi(x) = -1$ feltételt. A Cauchy-eloszlás eloszlásfüggvénye:

$$\varphi(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2}.$$

Ez a függvény a $\lim_{x \rightarrow -\infty} \varphi(x) = 0$ feltételt teljesíti. Minden eloszlásfüggvényből (azaz valószínűségi változó eloszlásfüggvényéből) készíthető aktivációs függvény; ezek teljesítik a $\lim_{x \rightarrow -\infty} \varphi(x) = 0$ feltételt, de átalakíthatóak $\lim_{x \rightarrow -\infty} \varphi(x) = -1$ feltételt teljesítő változattá is. Természetesen a gyakorlatban csak néhány konkrét aktivációs függvényt használnak.

2. Használjuk a perceptron betanításának leírását, azaz az alábbi.

- Ha $\mathbf{w}(n)^\top \mathbf{x}(n) > 0$ és $\mathbf{x}(n) \in A_1$, vagy pedig ha $\mathbf{w}(n)^\top \mathbf{x}(n) \leq 0$ és $\mathbf{x}(n) \in A_2$, akkor

$$\mathbf{w}(n+1) = \mathbf{w}(n).$$

- Ha $\mathbf{w}(n)^\top \mathbf{x}(n) \leq 0$, de $\mathbf{x}(n) \in A_1$, akkor:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{x}(n).$$

- Ha $\mathbf{w}(n)^\top \mathbf{x}(n) > 0$, de $\mathbf{x}(n) \in A_2$, akkor:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)\mathbf{x}(n).$$

3. Illesszünk $\mathbf{x}^\top \mathbf{w} + b$ alakú (lineáris) függvényt az alábbi mért adatokra. Tegyük fel, hogy $(\mathbf{x}(1), d(1)), \dots, (\mathbf{x}(N), d(N))$ a megfigyeléseink, ahol $d(i) = f(\mathbf{x}(i))$ a függvényérték. Ekkor a Least-Mean-Square algoritmus (más néven Widrow–Hoff algoritmus) az alábbi (lásd [10]).

Legyen $\mathbf{w} := \mathbf{0}$; $b := 0$; $\eta := 0$.

repeat

 for $i = 1$ to N

$$(\mathbf{w}^\top, b) := (\mathbf{w}^\top, b) + \eta(d(i) - (\mathbf{w}^\top \mathbf{x}(i) + b))(\mathbf{x}(i)^\top, 1)$$

 end for

until konvergencia kritérium teljesül

return \mathbf{w}, b

4. XOR esetén az igaz halmaz $\{(1, 0), (0, 1)\}$, a hamis $\{(1, 1), (0, 0)\}$. Ez a két halmaz nem szeparálható lineárisan a síkon.

5. $\mathbf{x} \in A_1$ esetén $\mathbf{x}^\top \mathbf{w}_{\text{opt}} + b_{\text{opt}} > 0$, azaz

$$-b_{\text{opt}} < \mathbf{x}^\top \mathbf{w}_{\text{opt}} \leq \|\mathbf{x}\| \|\mathbf{w}_{\text{opt}}\| \leq R,$$

ahol használtuk a Cauchy-egyenlőtlenséget, valamint az $\|\mathbf{x}\| \leq R$ és a $\|\mathbf{w}_{\text{opt}}\| = 1$ feltételeket. Hasonlóan, $\mathbf{x} \in A_2$ választással, $b_{\text{opt}} \leq R$ adódik.

3.6. Feladatok megoldása

1. A tangens hiperbolikus függvény:

$$\varphi(x) = \frac{2}{1 + \exp(-2x)} - 1 = \tanh(x).$$

Ezért

$$\varphi'(x) = \frac{4 \exp(-2x)}{(1 + \exp(-2x))^2} = (1 + \varphi(x))(1 - \varphi(x)).$$

Innen

$$\varphi'_j(v_j(n)) = (1 + \varphi_j(v_j(n)))(1 - \varphi_j(v_j(n))) = (1 + y_j(n))(1 - y_j(n)).$$

A $\delta_j(n)$ lokális gradiens a kimeneti rétegben:

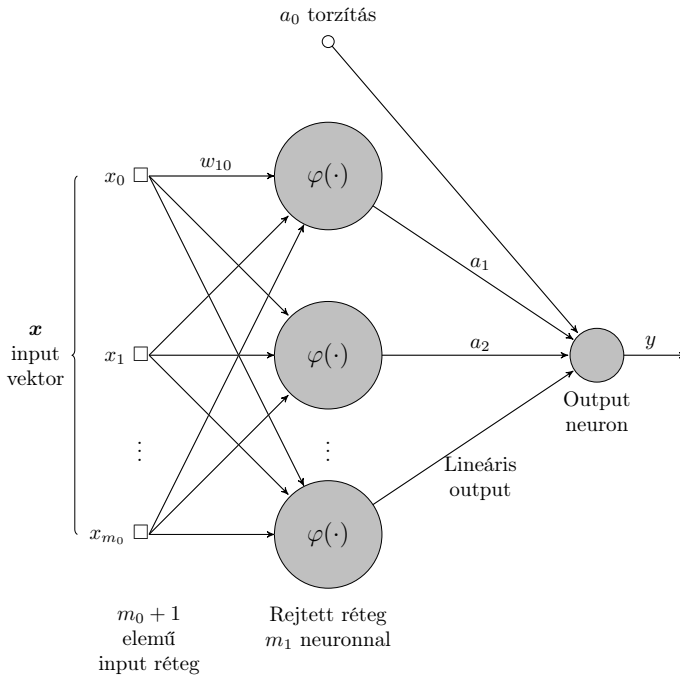
$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n) \varphi'_j(v_j(n)).$$

A lokális gradiens rejtett rétegben:

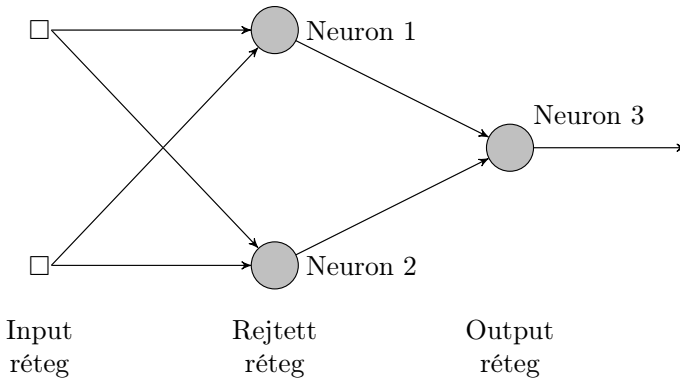
$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_{k \in \{j \text{ utáni réteg}\}} \delta_k(n) w_{kj}(n).$$

Tehát ebben az esetben a lokális gradiens számolásánál a j neuron y_j outputja kell csupán (és a már korábban kiszámolt mennyiségek).

2. A tételbeli approximáló függvényt egy rejtett rétegű perceptronnal lehet megvalósítani (lásd 7.1. ábra). Az input x_1, \dots, x_{m_0} , míg $x_0 = 1$ a rejtett rétegbeni neuronokban a torzítás kezelésére szolgál. w_{ij} az i -edik neuronhoz tartozó súly, amelyek közül w_{i0} a torzítás. Minden rejtett neuronban φ az aktivációs függvény. Az output rétegben egy neuron van, lineáris aktivációs függvénnyel, a_i súlyokkal (amelyek közül a_0 a torzítás).
3. A közelítendő f függvényt kell először a Stone-Weierstrass-tétel segítségével approximálni. A részleteket lásd Luc Devroye, László Györfi, Gábor Lugosi [28], Theorem 30.4.
4. Az XOR-problémát megoldó hálózat vázlata a 7.2. ábrán látható. Az input kétdimenziós. Az input rétegben a beérkező (x_1, x_2) jel $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$ lehet. Egy rejtett réteg van, két neuronnal. Az output rétegben egy neuron van. Mindhárom neuron transzfer függvénye a küszöb függvény: negatív számokhoz 0-t, nem-negatívokhoz 1-et rendel.



7.1. ábra. A függvényt approximáló MLP



7.2. ábra. Az XOR problémát megoldó MLP

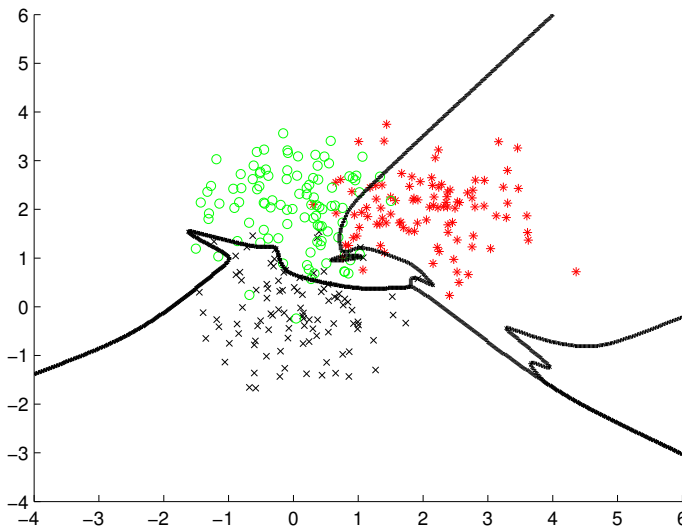
A rejtett rétegben minden súly legyen 1, azaz $w_{11} = w_{12} = w_{21} = w_{22} = 1$. Neuron 1 torzítása legyen $b_1 = -3/2$. Ekkor Neuron 1 outputja akkor 1, ha $x_2 \geq -x_1 + 3/2$. Neuron 2 torzítása legyen $b_2 = -1/2$. Ekkor Neuron 2 outputja akkor 1, ha $x_2 \geq -x_1 + 1/2$.

Az output neuron (azaz Neuron 3) súlyai legyenek: $w_{31} = -2$, $w_{32} = 1$, torzítása pedig: $b_3 = -1/2$. Ez a hálózat helyesen osztályozza a négy pontot. Valójában ez a hálózat a sík pontjait is osztályozza, egy sávhoz rendeli az 1 értéket, a többi síkbeli ponthoz 0-át. Lásd Haykin [18].

5. Használjunk programcsomagot! A Matlab és az R erre alkalmasak. Járjunk el hasonlóan, mint a kidolgozott példákban!
6. Használjunk programcsomagot! A Matlab és az R erre alkalmasak. Járjunk el hasonlóan, mint a kidolgozott példában!
7. Három halmaz szétválasztása MLP segítségével megvalósítható. Legyen mindhárom halmaz 100 elemű minta kétdimenziós,

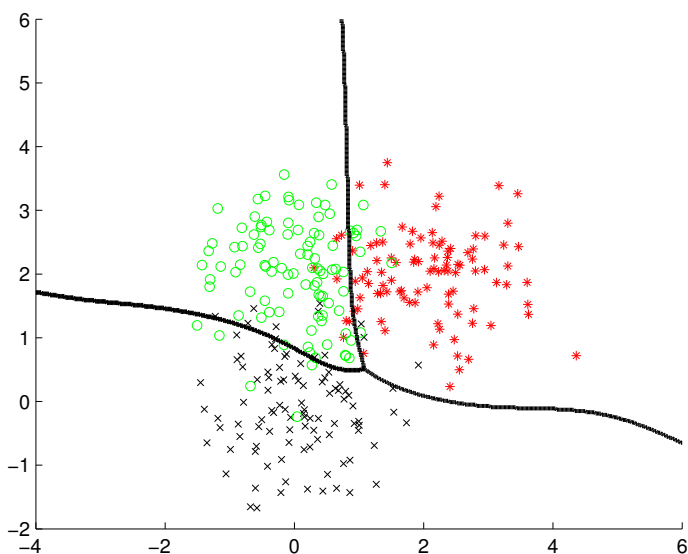
$$\begin{pmatrix} 0.6 & 0 \\ 0 & 0.6 \end{pmatrix}$$

szórásmátrixú normális eloszlásból. De az első halmazban $(0,0)^\top$ a várható érték vektor, a második halmazban pedig $(2,2)^\top$ a várható érték vektor, míg a harmadikban $(0,2)^\top$. A 7.3. és 7.4. ábrákon \times jelöli az



7.3. ábra. Három normális eloszlás szeparálása MLP-vel: túlillesztett hálózat, rosszul általánosít

első, $*$ pedig a második halmaz elemeit, míg \circ a harmadikét. A megkonstruált elválasztó görbék jelentősen különböznek. A 7.3. ábrán 3 rejtett rétegű MLP szeparálása látható. Az első rejtett rétegben 4 neuron

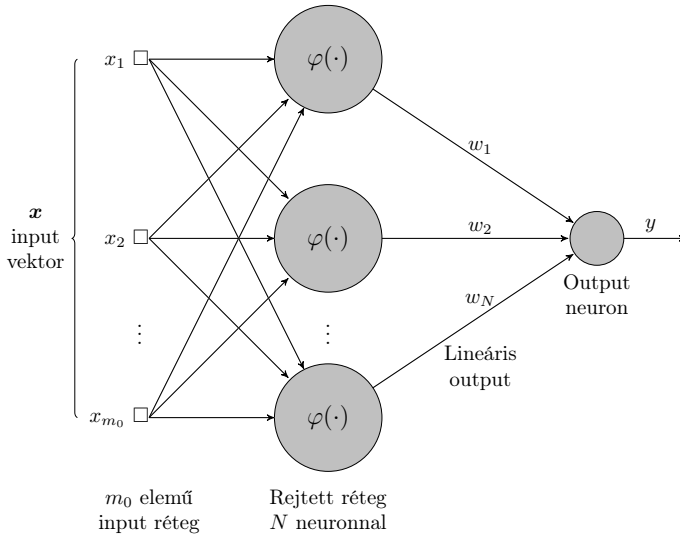


7.4. ábra. Három normális eloszlás szeparálása MLP-vel: megfelelő számú neuron, jól általánosít

van logisztikus, a másodikban 8 neuron tangens hiperbolikus, a harmadikban pedig 4 neuron tangens hiperbolikus transzfer függvényvel. A kimeneti rétegben 3 neuron van logisztikus transzfer függvényvel. A tanítás Levenberg-Marquardt back-propagation eljárással történt. Az elválasztó görbe a tanító pontokat viszonylag helyesen szeparálja, azonban rosszul általánosít, nagyon „bonyolult” az elválasztás. Ennek oka a bonyolult hálózat és a túlillesztés. A 7.4. ábrán lévő szeparálás azért sokkal jobb, mert ugyan az aktuális tanító pontokat nem tökéletesen osztja háromfelé, azonban jól általánosít. Ezt az eredményt két rejtett réteggel, azokban 5, ill. 3 neuronnal értük el. Az első rétegben logisztikus, a másodikban tangens hiperbolikus transzfer függvényt használunk. A kimeneti rétegben 3 neuron található logisztikus transzfer függvényvel. A tanítás Levenberg-Marquardt back-propagation eljárással történt.

4.7. Feladatok megoldása

1. Az $F(\mathbf{x}) = \sum_{j=1}^N w_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|)$ alakú függvény, azaz az RBF-interpoláció a 7.5. ábrán látható hálózattal valósítható meg.



7.5. ábra. Az RBF-interpolációt megvalósító hálózat

2. A két feladat és megoldása Kabanikhin [22] cikkéből származik.

(a) Be kell látnunk, hogy az alábbi egy rosszul felállított probléma. Legyen $q(x) = f'(x)$ és tegyük fel, hogy $f(x)$ -et csak egy kis perturbációval ismerjük: $f_n(x) = f(x) + \sin(nx)/\sqrt{n}$.

Ekkor $f_n \rightarrow f$ teljesül supremum normában, hiszen $f_n(x) - f(x) = \sin(nx)/\sqrt{n}$ nullához tart egyenletesen, ha $n \rightarrow \infty$. Azonban az f_n deriváltja $q_n(x) = q(x) + \sqrt{n} \cos(nx)$, aminek az eltérése q -tól $\|q_n(\cdot) - q(\cdot)\| = \sqrt{n} \|\cos(n\cdot)\| \rightarrow \infty$, ha $n \rightarrow \infty$. Itt $\|\cdot\|$ a supremum normát jelöli.

(b) Be kell látnunk, hogy rosszul felállított probléma a Laplace-féle differenciálegyenletre vonatkozó Cauchy-probléma. Legyen $u(x, y)$ a következő feladat megoldása:

$$\begin{aligned}\nabla^2 u &= 0, \\ u(0, y) &= f(y), \\ \frac{\partial u(0, y)}{\partial x} &= 0.\end{aligned}$$

Itt $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ a Laplace-operátor. Legyen $f(y) = \frac{1}{n} \sin(ny)$. Belátjuk, hogy a fenti probléma megoldása

$$u(x, y) = \frac{1}{n} \sin(ny) \frac{e^{nx} + e^{-nx}}{2}.$$

Ugyanis

$$\begin{aligned}\frac{\partial u(x, y)}{\partial x} &= \frac{1}{n} \sin(ny) \frac{ne^{nx} - ne^{-nx}}{2}, \\ \frac{\partial^2 u(x, y)}{\partial x^2} &= \frac{1}{n} \sin(ny) \frac{n^2 e^{nx} + n^2 e^{-nx}}{2}, \\ \frac{\partial u(x, y)}{\partial y} &= \frac{n}{n} \cos(ny) \frac{e^{nx} + e^{-nx}}{2}, \\ \frac{\partial^2 u(x, y)}{\partial y^2} &= -\frac{n^2}{n} \sin(ny) \frac{e^{nx} + e^{-nx}}{2}.\end{aligned}$$

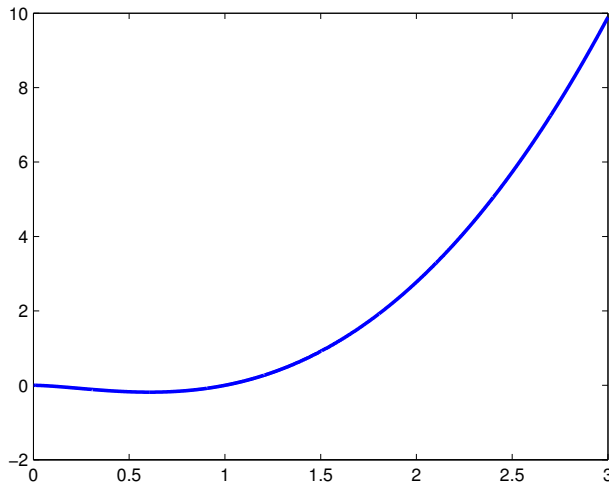
Így belátható, hogy az adott u függvény megoldása a fenti problémának. Minden $x > 0$ esetén, n -et elég nagyoknak választhatjuk ahhoz, hogy f tetszőlegesen kicsi, míg u tetszőlegesen nagy legyen.

3. $\text{tr } AB = \sum_i \sum_j a_{ij} b_{ji} = \text{tr } BA.$

4. A

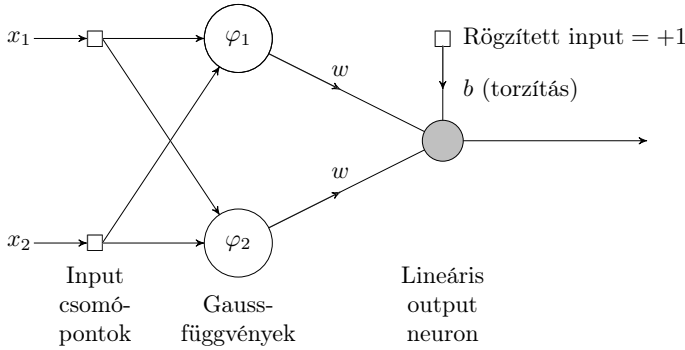
$$\varphi(r) = \left(\frac{r}{\sigma}\right)^2 \log\left(\frac{r}{\sigma}\right)$$

thin-plate-spline függvény grafikonja $\sigma = 1$ paraméter esetén a 7.6. ábrán látható.



7.6. ábra. A thin-plate-spline függvény

5. A megoldás Haykin [18] alapján. Az XOR-problémát megoldó RBF há-



7.7. ábra. Az XOR problémát megoldó RBF

lázat vázlata a 7.7. ábrán látható. Az input kétdimenziós. Az input rétegben a beérkező (x_1, x_2) jel $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$ lehet. Egy rejtett réteg van, két neuronnal. Az output rétegben egy neuron van.

Tehát az XOR esetén az \mathbf{x}_j, d_j tanító pontok:

i	1	2	3	4
\mathbf{x}_i	(1, 1)	(0, 1)	(0, 0)	(1, 0)
d_i	0	1	0	1

Legyen a bázis függvény Gauss-féle:

$$G(\|\mathbf{x} - \mathbf{t}_i\|) = \exp(-\|\mathbf{x} - \mathbf{t}_i\|^2),$$

ahol a középpontok: $\mathbf{t}_1 = (1, 1)^\top$, $\mathbf{t}_2 = (0, 0)^\top$. Olyan

$$y(\mathbf{x}) = \sum_{i=1}^2 w_i G(\|\mathbf{x} - \mathbf{t}_i\|) + b$$

alakú függvényt kell konstruálni, ami a tanító pontok esetén a megfelelő outputot adja, azaz

$$y(\mathbf{x}_j) = d_j, \quad j = 1, 2, 3, 4.$$

Mivel a $G(\|\mathbf{x}_j - \mathbf{t}_i\|)$ értékek kiszámolhatóak, így az alábbiakat kapjuk

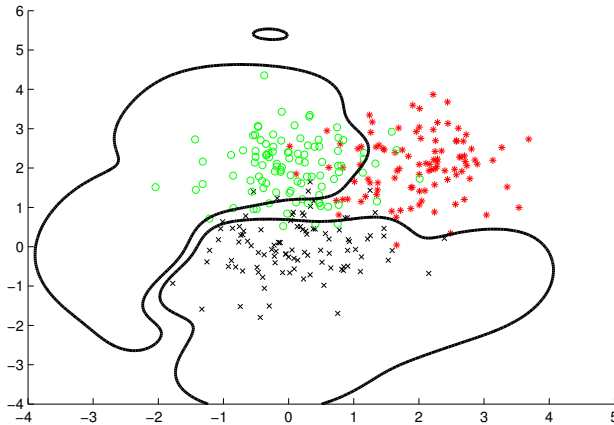
$$\begin{aligned} w + 0.1353w + b &= 0 \\ 0.3678w + 0.3678w + b &= 1 \end{aligned}$$

$$\begin{aligned} 0.1353w + \quad \quad \quad w + b &= 0 \\ 0.3678w + 0.3678w + b &= 1. \end{aligned}$$

Ez egy túlhatározott egyenletrendszer, ennek a legkisebb négyzetek elve szerinti legjobb megoldása: $w = -2.5018$, $b = 2.8404$. Tehát a hálózat:

$$y(\mathbf{x}) = -2.5018 \sum_{i=1}^2 G(\|\mathbf{x} - \mathbf{t}_i\|) + 2.8404.$$

6. Használjunk programcsomagot! Járjunk el hasonlóan, mint a kidolgozott példákban!
7. Használjunk programcsomagot! Járjunk el hasonlóan, mint a kidolgozott példákban!
8. Használjunk programcsomagot! Járjunk el hasonlóan, mint a kidolgozott példákban!
- 9.

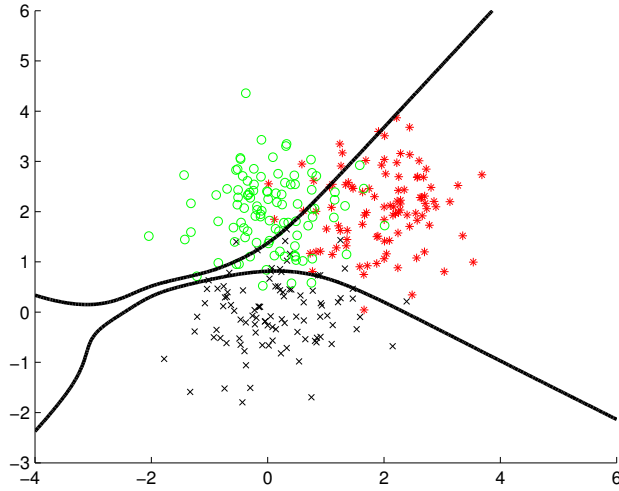


7.8. ábra. Három normális eloszlás szeparálása RBF-fel

Három halmaz szétválasztása RBF-fel. Legyen mindhárom halmaz 100 elemű minta kétdimenziós,

$$\begin{pmatrix} 0.6 & 0 \\ 0 & 0.6 \end{pmatrix}$$

szórásmátrixú normális eloszlásból. Az első halmazban $(0, 0)^\top$ a várható érték vektor, a második halmazban pedig $(2, 2)^\top$ a várható érték vektor, míg a harmadikban $(0, 2)^\top$.



7.9. ábra. Három normális eloszlás szeparálása MLP-vel

A 7.8. és 7.9. ábrákon \times jelöli az első, $*$ pedig a második halmaz elemeit, míg \circ a harmadikét. A megkonstruált elválasztó görbék jelentősen különböznek. A 7.8. ábrán az RBF szeparálása látható. Gauss-féle bázis függvény használunk az RBF-ben. Az elválasztó görbe a tanító pontokat viszonylag helyesen szeparálja. A 7.9. ábrán lévő szeparálás ugyanarra az adathalmazra vonatkozik, mint az előző, csak összehasonlítás képpen MLP-vel végeztük. Három rejtett réteg, azokban 3, 9, ill. 3 neuron van, mindben tangens hiperbolikus transzfer függvényt használunk. A kimeneti rétegben 1 neuron található lineáris transzfer függvénnyel. A tanítás Levenberg-Marquardt back-propagation eljárással történt.

5.5. Feladatok megoldása

- Ha α_1 nincs a vágási határon, akkor $\alpha_1 \neq 0$ és $\alpha_1 \neq C$. Így a Kuhn–Tucker-feltétel (5.45) miatt $f(\mathbf{x}_1) = y_1$, azaz $f_{\text{új}}(\mathbf{x}_1) = y_1$ kell legyen. Mivel az $f(\mathbf{x}_1) = \sum_i y_i \alpha_i K(\mathbf{x}_1, \mathbf{x}_i) + b$ képletben most csak az α_1 -hez és α_2 -höz tartozó tagok és b változhatnak, ezért a fixen maradó részek egyenlőségét felírva adódik (5.54), azaz

$$b^{\text{új}} = -f_{\text{régi}}(\mathbf{x}_1) + y_1 - y_1(\alpha_1^{\text{új}} - \alpha_1^{\text{régi}})K_{11} - y_2(\alpha_2^{\text{új}} - \alpha_2^{\text{régi}})K_{12} + b^{\text{régi}}. \quad (7.1)$$

Ha α_2 nincs a vágási határon, akkor az előző gondolatot α_2 -re alkalmazva a Kuhn–Tucker-feltétel miatt $f_{\text{új}}(\mathbf{x}_2) = y_2$ kell legyen, amiből adódik

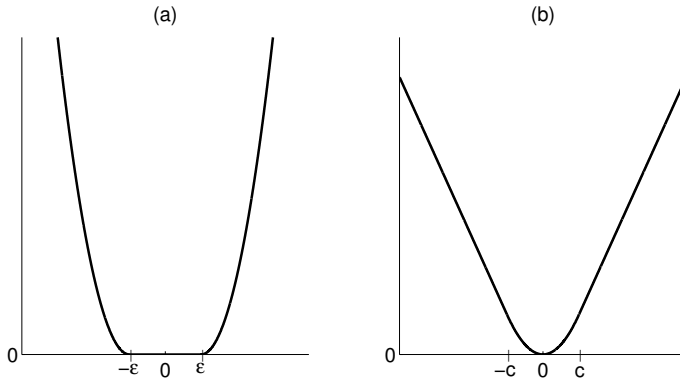
(5.55), azaz

$$b^{\text{új}} = -f_{\text{rég}}(\mathbf{x}_2) + y_2 - y_1(\alpha_1^{\text{új}} - \alpha_1^{\text{rég}})K_{21} - y_2(\alpha_2^{\text{új}} - \alpha_2^{\text{rég}})K_{22} + b^{\text{rég}}. \quad (7.2)$$

Ha sem α_1 , sem α_2 nincs a vágási határon akkor a két fenti $b^{\text{új}}$ érték egybeesik. Ugyanis ekkor (5.52) vágás nélkül és (5.53) alkalmazható. Ezeket beírva a (7.1) és a (7.2) képletekbe, kapjuk, hogy $b^{\text{új}}$ két értéke egybeesik.

Ha mind α_1 , mind α_2 a vágási határon van, akkor legyen b új értéke a fenti két $b^{\text{új}}$ számtani közepe.

2. A 7.10. ábra (a) részén a négyzetes ε -inszenzitív veszteségfüggvény látható, míg a (b) részén a Huber-féle veszteségfüggvény. Az ε és a c értékét egyformának választottuk.

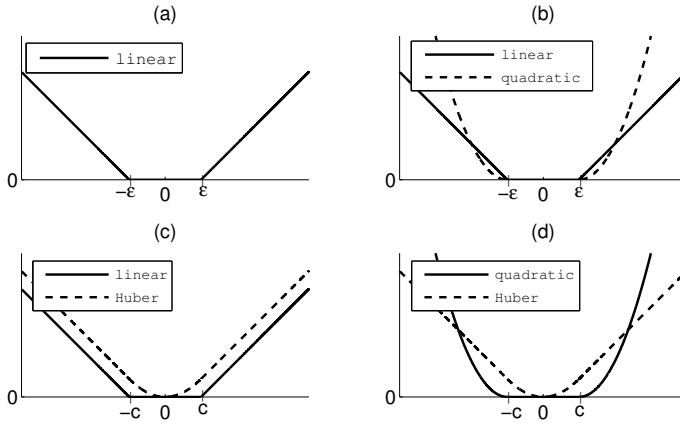


7.10. ábra. (a) négyzetes ε -inszenzitív veszteségfüggvény, (b) Huber-féle veszteségfüggvény

3. A 7.11. ábra (a) részén a lineáris ε -inszenzitív veszteségfüggvény látható; a (b) részén a lineáris és a négyzetes ε -inszenzitív veszteségfüggvény; a (c) részén a lineáris ε -inszenzitív és a Huber-féle veszteségfüggvény; a (d) részén pedig a négyzetes ε -inszenzitív és a Huber-féle veszteségfüggvény. Az ε és a c értékét egyformának választottuk.

4. Négyzetes ε -inszenzitív veszteségfüggvény. Minimalizáljuk a

$$\begin{aligned} y_i - \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle - b &\leq \varepsilon_i + \xi_i^*, & i = 1, \dots, N, \\ \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b - y_i &\leq \varepsilon_i + \xi_i, & i = 1, \dots, N, \\ \xi_i &\geq 0, & i = 1, \dots, N, \\ \xi_i^* &\geq 0, & i = 1, \dots, N. \end{aligned}$$



7.11. ábra. A lineáris és a négyzetes ϵ -inszenzitiv veszteségfüggvények, valamint a Huber-féle veszteségfüggvény

feltételek esetén a

$$\Phi(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{C}{2} \left(\sum_{i=1}^N (\xi_i^2 + (\xi_i^*)^2) \right) + \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle$$

célfüggvényt (lásd [49]).

Ez az ismert konvex optimalizálási probléma, amire alkalmazhatjuk a Kuhn–Tucker-tételt. Vezessük be az α_i , α_i^* , β_i , β_i^* nem-negatív multiplikátorokat. Ezekkel írjuk fel a Lagrange-függvényt. Ezt $\boldsymbol{\psi}$, b , ξ_i , ξ_i^* szerint minimalizálni kell. Az így kapott kvadratikusan programozási feladat az alábbi. Maximalizálandó a

$$\begin{aligned} W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) &= \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) - \sum_{i=1}^N \varepsilon_i (\alpha_i + \alpha_i^*) - \\ &- \frac{1}{2} \left(\sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{C} \sum_{i=1}^N (\alpha_i^*)^2 + \frac{1}{C} \sum_{i=1}^N (\alpha_i)^2 \right) \end{aligned}$$

függvény a

$$\sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0$$

$$\begin{aligned} 0 &\leq \alpha_i, \quad i = 1, \dots, N, \\ 0 &\leq \alpha_i^*, \quad i = 1, \dots, N, \end{aligned}$$

feltételek mellett.

Ennek megoldásával végül a regressziós hipersík a következő

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i^{*0} - \alpha_i^0) \langle \mathbf{x}, \mathbf{x}_i \rangle + b^0.$$

5. Huber-féle veszteségfüggvény. Legyen

$$F(x) = \begin{cases} c|x| - \frac{c^2}{2}, & \text{ha } |x| > c, \\ \frac{1}{2}x^2, & \text{ha } |x| \leq c. \end{cases}$$

Minimalizáljuk a

$$\Phi(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = C \sum_{i=1}^N (F(\xi_i) + F(\xi_i^*)) + \frac{1}{2} \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle$$

célfüggvényt az

$$\begin{aligned} y_i - \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle - b &\leq \xi_i^*, & i = 1, \dots, N, \\ \langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b - y_i &\leq \xi_i, & i = 1, \dots, N, \\ \xi_i &\geq 0, & i = 1, \dots, N, \\ \xi_i^* &\geq 0, & i = 1, \dots, N. \end{aligned}$$

feltételek esetén ([49]).

Ez az ismert konvex optimalizálási probléma, amire alkalmazhatjuk a Kuhn–Tucker-tételt. Vezessük be a nem-negatív multiplikátorokat. Ezekkel írjuk fel a Lagrange-függvényt. Ezt $\boldsymbol{\psi}$, b , ξ_i , ξ_i^* szerint minimalizálni kell. Az így kapott kvadratikus programozási feladat az alábbi.

Maximalizálandó a

$$\begin{aligned} W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) &= \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) - \\ &- \frac{1}{2} \left(\sum_{i,j=1}^N (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{c}{C} \sum_{i=1}^N (\alpha_i^*)^2 + \frac{c}{C} \sum_{i=1}^N (\alpha_i)^2 \right) \end{aligned}$$

függvény a

$$\begin{aligned} \sum_{i=1}^N (\alpha_i - \alpha_i^*) &= 0 \\ 0 &\leq \alpha_i^* \leq C, & i = 1, \dots, N, \\ 0 &\leq \alpha_i \leq C, & i = 1, \dots, N, \end{aligned}$$

feltételek mellett.

Ennek megoldásával végül a regressziós hipersík a következő

$$\langle \mathbf{w}_0, \mathbf{x} \rangle + b^0 = \sum_{i=1}^N (\alpha_i^{*0} - \alpha_i^0) \langle \mathbf{x}, \mathbf{x}_i \rangle + b^0.$$

6. (a) Haladjunk részintervallumonként! (b) Mutassuk meg, hogy $d = 1$ esetén érvényes az állítás. Egy d -edfokú spline $d - 1$ -edik deriváltja elsőfokú spline. A részleteket lásd [27].
7. Az adott pontokat összekötő töröttvonalat kapunk.
8. A B_0 függvény a $[-0.5, 0.5]$ intervallumon egyenletes eloszlású valószínűségi változó sűrűségfüggvénye. Ennek 0 a várható értéke és $1/\sqrt{12}$ a szórása. Így a konvolúciós képlet miatt a B_d függvény $d + 1$ számú, független, a $[-0.5, 0.5]$ intervallumon egyenletes eloszlású valószínűségi változó összegének sűrűségfüggvénye. Ezen összeget jelölje ξ_{d+1} . Ennek standardizáltja $\xi_{d+1}/\sqrt{(d+1)/12}$. Jelölje f_d ezen standardizált sűrűségfüggvényét. Ekkor a központi határeloszlás-tétel lokális alakja (lásd [13]) miatt

$$\lim_{d \rightarrow \infty} f_d(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

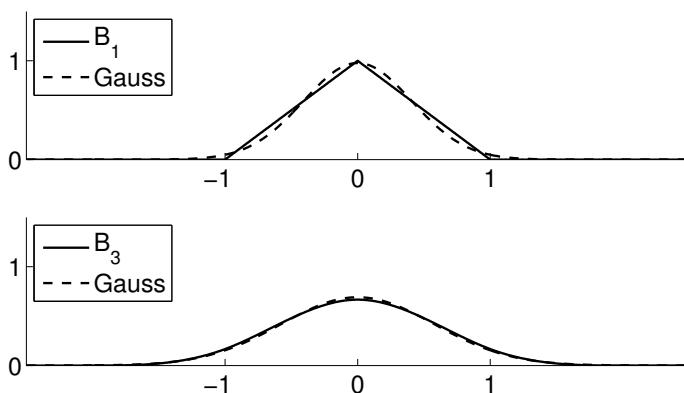
és a konvergencia x -ben egyenletes. A limesz a standard normális sűrűségfüggvény. Azt tudjuk, hogy ξ_{d+1} sűrűségfüggvénye a B_d függvény. Másrészt viszont ξ_{d+1} sűrűségfüggvénye $\sqrt{\frac{12}{d+1}} f_d\left(\sqrt{\frac{12}{d+1}} x\right)$. Ezekből és az előbbi limesz relációból adódik, hogy

$$\lim_{d \rightarrow \infty} B_d(u) = \sqrt{\frac{6}{\pi(d+1)}} \exp\left\{-\frac{6u^2}{d+1}\right\},$$

azaz (5.64). B_1 és B_3 grafikonja és a közelítő normális sűrűségfüggvény a 7.12. ábrán látható.

9. Haladjunk indukcióval! $d = 0$ -ra és $d = 1$ -re (5.65) közvetlenül ellenőrizhető. Tegyük fel (5.65) igaz voltát d -re. Számítsuk ki a B_{d+1} -et a $B_{d+1}(u) = \int_{-\infty}^{\infty} B_d(u-t)B_0(t)dt$ konvolúciós képlettel úgy, hogy B_0 a definíció szerinti, míg B_d az (5.65) szerinti, azaz

$$B_d(u) = \sum_{r=0}^{d+1} \frac{(-1)^r}{d!} \binom{d+1}{r} \left(u + \frac{d+1}{2} - r\right)_+^d. \quad (7.3)$$



7.12. ábra. A B_d spline függvény és a közelítő Gauss-féle sűrűségfüggvény, $d = 1, 3$

Kiderül, hogy a konvolúciós képletben

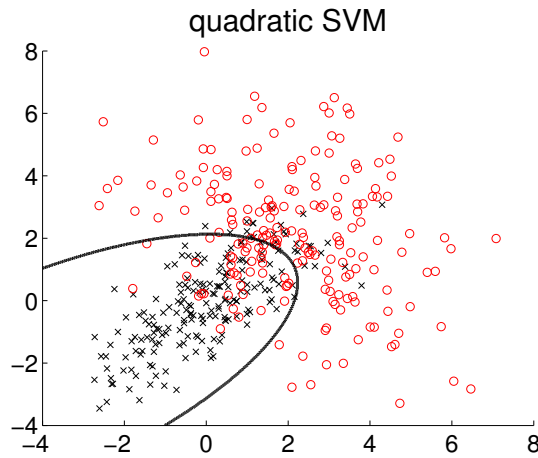
$$\begin{aligned} \int_{-\infty}^{\infty} \left(u - t + \frac{d+1}{2} - r \right)_+^d B_0(t) dt = \\ = \frac{1}{d+1} \left(u + \frac{d+2}{2} - r \right)_+^{d+1} - \frac{1}{d+1} \left(u + \frac{d+2}{2} - (r+1) \right)_+^{d+1}. \end{aligned}$$

A kapott tagokat összegezve és a $\binom{d+1}{r} + \binom{d+1}{r-1} = \binom{d+2}{r}$ képletet alkalmazva adódik (5.65) $d+1$ -re.

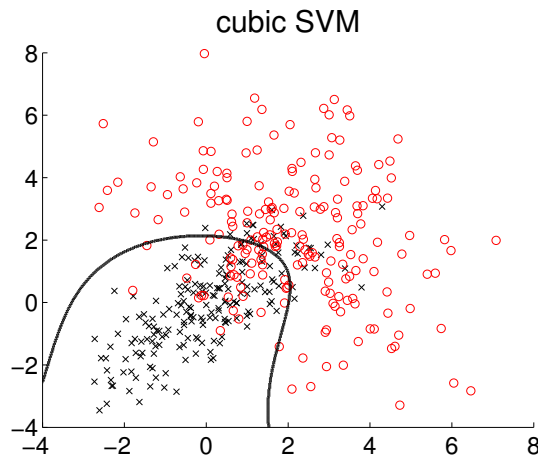
10. Két halmaz szétválasztása SVM segítségével. Legyen

$$D_1 = \begin{pmatrix} 2 & 1.5 \\ 1.5 & 2 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 3.5 & -1 \\ -1 & 3.5 \end{pmatrix}.$$

Legyen az egyik halmaz 200 elemű minta kétdimenziós, $(0, 0)^\top$ várható érték vektorú és D_1 szórásmátrixú normális eloszlásból. A másik halmaz pedig szintén 200 elemű minta kétdimenziós, $(2, 2)^\top$ várható érték vektorú és D_2 szórásmátrixú normális eloszlásból. Ez a két halmaz nem szeparálható lineárisan, így a lineáris magfüggvényű SVM mindig helytelenül szeparálta őket. Az ábrákon \times jelöli az első, \circ pedig a második halmaz elemeit. A megkonstruált elválasztó görbék különböznek. A 7.13. ábrán kvadratikus magfüggvényű és SMO-val betanított SVM szeparálása szerepel. A 7.14. ábrán pedig harmadfokú (polinom) magfüggvényű és SMO módszerrel betanított SVM szeparálása.



7.13. ábra. Két normális eloszlás szeparálása kvadrati-
kus SVM-mel

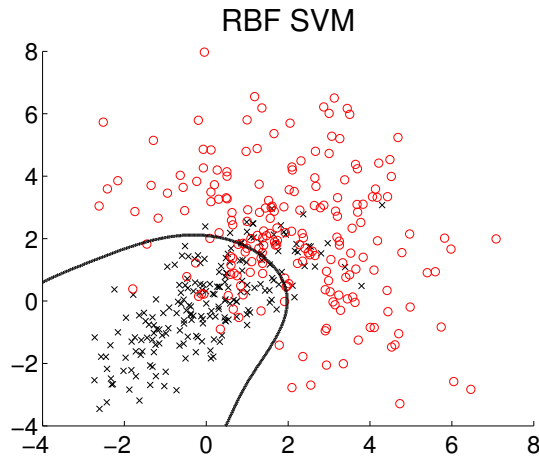


7.14. ábra. Két normális eloszlás szeparálása harmad-
fokú SVM-mel

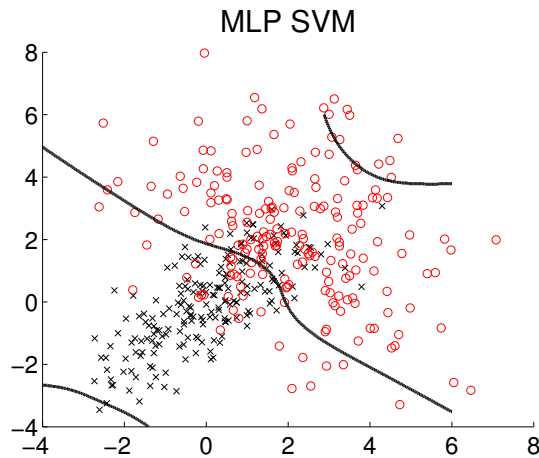
A 7.15. ábrán Gauss-féle radiális bázis függvény magfüggvényű és SMO-val betanított SVM szeparálása látható. A 7.16. ábrán MLP magfüggvényű (pontosabban szövege tangens hiperbolikus függvényt használó) és SMO módszerrel betanított SVM szeparálása szerepel.

Számos futás tapasztalata: a kvadratikus és a harmadfokú polinom, valamint az RBF magfüggvény esetén az SVM ezt a szeparálást szinte mindig

jól oldja meg, MLP magfüggvény esetén viszont a megoldás nem kielégítő (sőt gyakran igen rossz).

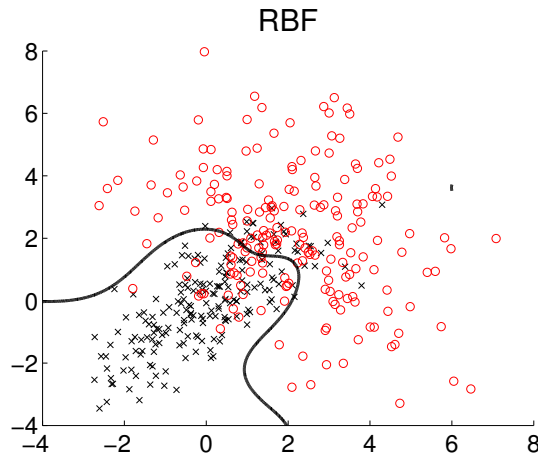


7.15. ábra. Két normális eloszlás szeparálása RBF SVM-mel

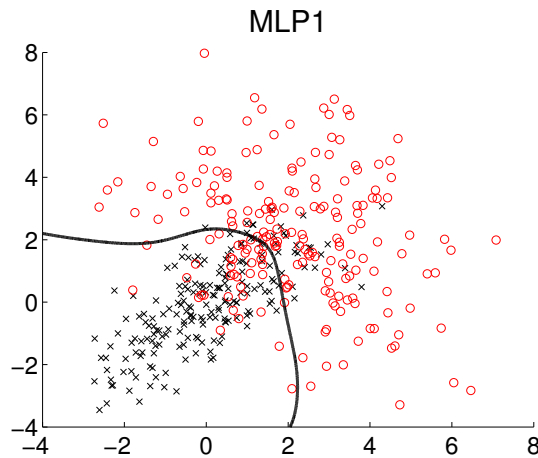


7.16. ábra. Két normális eloszlás szeparálása MLP SVM-mel

11. Az előző feladatban generált két mintát hagyományos RBF-fel és MLP-vel is szeparáltuk. A 7.17. ábrán Gauss-féle magfüggvényű RBF szeparálása látható. A 7.18. ábrán az MLP szeparálása szerepel. Az alkalmazott többrétegű perceptronban 2 rejtett réteg volt, rétegenként 12,



7.17. ábra. Két normális eloszlás szeparálása RBF-fel



7.18. ábra. Két normális eloszlás szeparálása MLP-vel

ill. 8 neuronnal, tangens hiperbolikus transzfer függvénnyel. A betanítás Levenberg–Marquardt-módszerrel történt.

Számos futás tapasztalata: az RBF ezt a szeparálást szinte mindig jól oldja meg. Az MLP az esetek többségében jól szeparál, de néha nem tudja megoldani a feladatot. Más felépítésű MLP-vel is hasonló eredményre jutottunk.

12. Használjunk programcsomagot! Járjunk el hasonlóan, mint a kidolgozott példákban!

13. Használjunk programcsomagot! Járjunk el hasonlóan, mint a kidolgozott példákban!

6.5. Feladatok megoldása

1. Az $A^\top A$ mátrix i, k -adik eleme: $\sum_j a_{ji} a_{jk}$, ahol a_{jk} az A mátrix jk -adik eleme. Most jelölje A sorvektorait (de oszlopként írva) $\mathbf{a}_1, \dots, \mathbf{a}_n$. Ekkor ezek diádösszegének i, k -adik eleme:

$$\left(\sum_j \mathbf{a}_j \mathbf{a}_j^\top \right)_{ik} = \sum_j a_{ji} a_{jk}.$$

Tehát igazoljuk, hogy az $A^\top A$ mátrix szorzat felbomlik A saját sorvektorainak diádösszegére!

2. Legyenek A, B, C_1, C_2 és D alkalmas méretű mátrixok. Legyen

$$A = B + C_1 D C_2^\top.$$

Be kell látni, hogy

$$A^{-1} = B^{-1} - B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} C_2^\top B^{-1}.$$

Szorozzuk össze a fenti két egyenlőség jobb oldalait (és jelöljük a szorzatot X -szel):

$$\begin{aligned} X &= \left[B^{-1} - B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} C_2^\top B^{-1} \right] \left[B + C_1 D C_2^\top \right] = \\ &= I - \underbrace{B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} C_2^\top + B^{-1} C_1 D C_2^\top}_{B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} D^{-1} D C_2^\top} - \\ &\quad - B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} C_2^\top B^{-1} C_1 D C_2^\top. \end{aligned}$$

A második tagot azért bővítettük $D^{-1} D$ -vel, hogy össze tudjuk vonni a negyedik taggal:

$$X = I - B^{-1} C_1 (D^{-1} + C_2^\top B^{-1} C_1)^{-1} (D^{-1} + C_2^\top B^{-1} C_1) D C_2^\top + B^{-1} C_1 D C_2^\top.$$

A második tagban két tényező egymás inverze, így

$$X = I - B^{-1} C_1 D C_2^\top + B^{-1} C_1 D C_2^\top = I.$$

Ezzel igazoltuk a Woodbury-egyenlőség általános alakját. Ennek speciális esete a Sherman-Morrison-formula, amikor $D = 1$.

3. Közvetlen számolás.

4. Négyzetes hibafüggvény esetén a gradiens vektor $\mathbf{g} = J^\top \mathbf{e}$. A négyzetes hibafüggvény esetén a Hesse-mátrixra a $H \approx J^\top J$ közelítést alkalmazhatjuk. Ezeket a Newton-módszer

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \left[H(\mathbf{w}(n)) \right]^{-1} \mathbf{g}(\mathbf{w}(n))$$

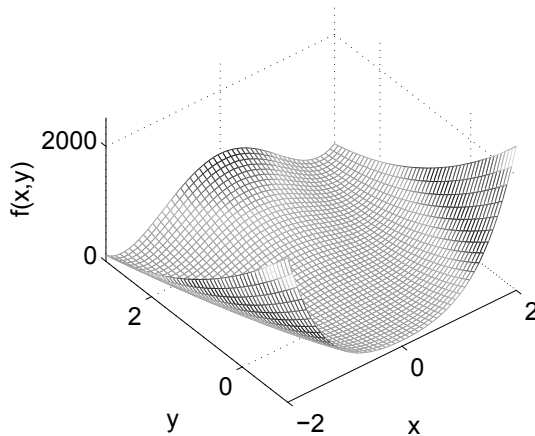
képletébe beírva kapjuk a Gauss-Newton-módszer képletét:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - (J(n)^\top J(n))^{-1} J(n)^\top \mathbf{e}(\mathbf{w}(n)).$$

5. A (6.30) és a (6.31) képletekben szereplő konvergenciákat a teljesen független esetben megkaphatjuk a nagy számok erős törvényéből. Az ugyanis azt állítja, hogy független azonos eloszlású valószínűségi változók szám-tani közepe 1 valószínűséggel konvergál a közös várható értékhez.

6. Lásd [14, 16, 40].

7. Az $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ Rosenbrock-függvénynek abszolút minimuma van az $(x_1, x_2) = (1, 1)$ pontban. Hiszen egyrészt ebben a pontban a függvény értéke 0, aminél kisebb értéket a függvény nem vesz fel. Másrészt, ha valahol a függvény érték 0, akkor ott $(1 - x_1)^2 = 0$, amiből $x_1 = 1$. De ha a függvényérték 0, akkor ott $(x_2 - x_1^2)^2 = 0$, amiből $x_2 = x_1^2 = 1$. Így $(x_1, x_2) = (1, 1)$ egyedüli abszolút minimumhely. A Rosenbrock-függvény grafikonja a 7.19. ábrán látható.



7.19. ábra. A Rosenbrock-függvény grafikonja

8. Alkalmazzunk saját programot, vagy programcsomagot!

Irodalomjegyzék

- [1] Mokhtar S. Bazaraa, Hanif D. Sherali, C. M. Shetty. *Nonlinear programming. Theory and algorithms. Third edition.* Wiley-Interscience, John Wiley & Sons, Hoboken, NJ, 2006.
- [2] Stephen Boyd, Lieven Vandenberghe. *Convex optimization.* Cambridge University Press, Cambridge, 2004.
- [3] D. S. Broomhead, David Lowe. Multi-variable functional interpolation and adaptive networks. *Complex Systems 2*, pp. 321–355.
- [4] M. D. Buhmann. *Radial basis functions: theory and implementations.* Cambridge University Press, Cambridge, 2003.
- [5] Chih-Chung Chang, Chih-Jen Lin. *LIBSVM - A Library for Support Vector Machines.*
<http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>.
- [6] Chih-Chung Chang, Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, Vol.2, No.2, Article 27*, 2011.
- [7] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Electron. Comput. 14*, pp. 326–334, 1965.
- [8] Peter Craven, Grace Wahba. Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math. 31, no. 4*, pp. 377–403, 1978/79.
- [9] Michael J. Crawley. *The R Book.* John Wiley & Sons, 2007.

- [10] Nello Cristianini, John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Methods*. Cambridge University Press, Cambridge, 2000.
- [11] L. Devroye, L. Györfi, G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.
- [12] Fazekas István. *Valószínűesszámitás*. Debreceni Egyetemi Kiadó, Debrecen, 2009.
- [13] Fazekas István (szerk.). *Bevezetés a matematikai statisztikába*. Kossuth Egyetemi Kiadó, Debrecen, 1997.
- [14] R. Fletcher. *Practical methods of optimization, Second edition*. A Wiley-Interscience Publication, John Wiley & Sons Ltd., Chichester, 1987.
- [15] Stoyan Gisbert (szerk.). *MATLAB - Frissített kiadás*. Typotex, Budapest, 2005.
- [16] Martin T. Hagan, Howard B. Demuth, Mark H. Beale. *Neural Network Design*. Boston, MA: PWS Publishing, 1996.
- [17] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning: Prediction, Inference and Data Mining*. Second Edition, Springer Verlag, 2009.
- [18] Simon Haykin. *Neural networks. A comprehensive foundation*. Prentice Hall, London, 1999.
- [19] D. O. Hebb. *The Organization of Behavior: A neuropsychological theory*. New York: Wiley, 1949.
- [20] Horváth Gábor (szerk.). *Neurális hálózatok és műszaki alkalmazásai*. Műegyetemi Kiadó, Budapest, 1998.
- [21] Horváth Roland. *A neurális hálók alapjai és alkalmazásai*. Diplomamunka. Témavezető: Fazekas István. Debreceni Egyetem, 2002.
- [22] S. I. Kabanikhin. Definitions and examples of inverse and ill-posed problems. *J. Inverse Ill-Posed Probl.* 16, no. 4, pp. 317–357, 2008.
- [23] William Karush. *Minima of functions of several variables with inequalities as side conditions*. Thesis (M.Sc.), The University of Chicago, 1939.

- [24] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43 (1), pp. 59–69, 1982.
- [25] Kovács György. *A Support Vector Regresszió elmélete, implementációjának elemzése és alkalmazása*. Diplomamunka. Témavezető: Fazekas István. Debreceni Egyetem, 2011.
- [26] H. W. Kuhn, A. W. Tucker. Nonlinear programming. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950, University of California Press, Berkeley and Los Angeles, 1951*, pp. 481–492.
- [27] Kátai Imre. *Numerikus analízis*. Tankönyvkiadó, Budapest, 1992.
- [28] Devroye Luc, László Györfi, Gábor Lugosi. *A probabilistic theory of pattern recognition*. Springer, New York, 1996.
- [29] *MATLAB The Language of Technical Computing*. MathWorks, <http://www.mathworks.com/products/matlab/>.
- [30] W. McCulloch, W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol 5, issue 4, pp. 115–133, 1943.
- [31] Charles A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.* 2, no. 1, pp. 11–22, 1986.
- [32] Marvin Minsky, Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, Cambridge MA, 1972 (2nd edition with corrections, first edition 1969).
- [33] Móri F. Tamás, Székely J. Gábor. *Többváltozós statisztikai analízis*. Műszaki Könyvkiadó, Budapest, 1986.
- [34] Jorge Nocedal, Stephen J. Wright. *Numerical Optimization*. Springer, 2006.
- [35] A. B. Novikoff. On convergence proofs on perceptrons. *Proceedings of Symposium on the Mathematical Theory of Automata*, vol 12, pp. 615–622, 1962.
- [36] J. Park, I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation* 3, pp. 246–257, 1991.

- [37] John C. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Microsoft Research, Technical Report MSR-TR-98-14, <http://research.microsoft.com/en-us/um/people/jplatt/smoTR.pdf>, 1998.
- [38] M. J. D. Powell. Radial basis function approximations to polynomials. *Numerical analysis 1987 (Dundee, 1987)*, *Pitman Res. Notes Math. Ser.*, 170, Longman Sci. Tech., Harlow, pp. 223–241, 1988.
- [39] B. L. S. Prakasa Rao. *Nonparametric functional estimation*. Probability and Mathematical Statistics. Academic Press, Inc., Harcourt Brace Jovanovich, Publishers, New York, 1983.
- [40] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*. Cambridge University Press, <http://www.nr.com/>, New York, 2007.
- [41] Riesz Frigyes, Szőkefalvi-Nagy Béla. *Funkcionálanalízis*. Tankönyvkiadó, Budapest, 1998.
- [42] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [43] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. Learning representations by back-propagating errors. *Nature* 323, pp. 533–536, 09 October 1986.
- [44] Szép Gábor. *A neurális hálóak egy osztálya: az SVM*. Diplomamunka. Témavezető: Fazekas István. Debreceni Egyetem, 2002.
- [45] *The R Project for Statistical Computing*. <http://www.r-project.org/>.
- [46] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. (*English. Russian original*) *Sov. Math., Dokl.* 5, pp. 1035–1038, 1963. (translation from *Dokl. Akad. Nauk SSSR* 151, 501–504).
- [47] A. N. Tikhonov, V. Ya. Arsenin. *Metody resheniya nekorrektnykh zadach*. (Russian) [Methods for the solution of ill-posed problems] Izdat. „Nauka”, Moscow, 1974.

-
- [48] V. Vapnik, A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2), pp. 264–280, 1971.
 - [49] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons Inc., New York, 1998.
 - [50] B. Widrow, M. E. Jr. Hoff. Adaptive switching circuits. *1960 IRE WES-CON Convention Record, Part 4*, New York: IRE, pp. 96–104, 1960.

Tárgymutató

Jelek

δ 69

ε -inszenzitív veszteségfüggvény 115

A, Á

activation function 16

Adaline 29

adaptive filter 145

aktivációs függvény 16, 18, 38

általánosít

jól 41

általánosított inverz 138

B

B-spline 120

back-propagation 40

belső szorzat 135

bemenet 15

bemeteti réteg 32

BFGS-formula 44

bias 16

Broyden–Fletcher–Goldfarb–
Shanno-formula 44, 152

büntető függvény 46, 68

C

Cauchy-egyenlőtlenség 24

chunking 111

cross-validation 81

D

Davidon–Fletcher–Powell-formula
152

delta rule 34

delta szabály

általánosított 39

diadikus szorzat 135

Dirac-delta disztribúció 69

dualitási rés 169

E, É

előrecsatolt hálózat 33

eltolásinvariáns 73

epoch 23

error back-propagation 33

Euler–Lagrange-egyenlet 70

F

feature space 60, 106

feedforward network 33

feltételes szélsőérték 168

Fletcher–Reeves-formula 166, 168

forгатásinvariáns 73

function

activation 16

hard limit 19

Heaviside 19

threshold 19

transfer 16

függvény

- aktivációs 16
- küszöb 19
- logisztikus 18, 38
- szakaszonként lineáris 19
- szigmoid 18
- tangens hiperbolikus 20, 38
- transzfer 16

G

- Gauss–Newton-módszer 158
- gradiens módszer 34, 83, 145
- Green-függvény 71
- Green-mátrix 72

H

- hálózat
 - metszése 46
- Hesse-mátrix 47
- hiba 18
- hiba visszaáramoltatása 33

I, í

- idegsejt 15
- ill-posed problem 66
- input 15
- interpoláció 63
- interpolációs mátrix 64

J

- Jacobi-mátrix 45, 156, 159
- jelfeldolgozás 145
- jól felállított inverz probléma 66

K

- k -közép módszer 82
- Karush–Kuhn–Tucker-tétel 168
- kimenet 16
- kimeteti réteg 32
- klaszterezés 82

konjugált gradiens 42, 162, 166

- korrekciós pont 23
- közelítő simítás 47
- Kuhn–Tucker-feltételek 169
- Kuhn–Tucker-tétel 168
- kvázi Newton-feltétel 151
- kvázi Newton-módszerek 44, 150

L

- Lagrange-féle multiplikátor 168
- Lagrange-függvény 168
- lágymargó 104
- layer 32
 - hidden 32
 - input 32
 - output 32
- learning rate 23
- Least-Mean-Square 29, 161
- legkisebb négyzetek módszere 136, 157
- legkisebb négyzetes becslés 158
- legközelebbi társ 83
- Levenberg–Marquardt-eljárás 44, 154
- lineáris modell 157
- lineáris szeparálás 25
- LMS 29, 161
- lokális gradiens 35

M

- magfüggvény 59, 84
- Mercer tétele 107
- MLP 31
- momentum konstans 39
- Moore–Penrose-inverz 142
- Multi Layer Perceptron 31

N

- Nadaraya–Watson-féle becslés 86
- Network

Radial Basis Function 59
network pruning 46
neuron 15
Newton-módszer 148
normálegyenlet 29, 140, 158

Ny

nyeregpon t 170

O, Ó

OLS 158
optimális dualitási rés 169
optimális hipersík 95
optimalizáció 145
ordinary least squares 158
ortogonális projekció 139
ortogonális vetítés 135
osztályozás 40
output 16

Ö, Ő

összegző csomópont 16

P

Parzen–Rosenblatt-féle becslés 85
perceptron 15
 konvergencia tétel 25
 tanítása 21
 többrétegű 31
Polak–Ribière-formula 43, 168
pseudoinverz 82, 140, 142

R

radial basis function 74
Radial Basis Function Network 59
radiális bázis függvény 59, 63, 74
RBF 59, 63
regresszió
 lineáris 28
regularizáció 46

regularizációs paraméter 68
regularizációs tag 68
rejtett réteg 32
réteg
 bemeneti 32
 kimeteti 32
 rejtett 32
ridge regresszió 29
Rosenblatt–Novikoff-tétel 26
Rosenbrock-függvény 171, 193
rosszul felállított probléma 66

S

sajátbázis 140
sajátbázis pár 141
sajátérték 136
sajátvektor 136
sávszélesség 85
Sequential Minimal Optimization
 111
Sherman–Morrison-formula 171
signal processing 145
soft margin 104
spline 120
statistical learning theory 93
statisztikus tanulási elmélet 93
súly 15
súly eliminálása 47
súly pusztítása 47
summing junction 16
support vector 101
Support Vector Machine 93
SVM 93

Sz

szélsőérték
 feltételes 168
 kétváltozós függvényé 137
szigmoid 18
szimmetrikus mátrix

spektrálfelbontása 136
szinguláris érték 140
szinguláris felbontási tétel 141

T

tanítás 18
 batch 39
 köteget 39
 soros 39
 szekvenciális 39
tanító pont 17
tanulás 18
tanulási paraméter 23, 39
tartó vektor 101
tartó vektor gépek 93
Taylor
 -formula 136
 -sor 137
thin-plate-spline 91
torzítás 16
többrétegű perceptron 31
training 18
training point 17
transfer function 16

transzfer függvény 16
tulajdonságok tere 60, 106

Ty

Tyihonov 46
 -funkcionál 68
Tyihonov-féle regularizáció 66

V

Vapnik–Chervonenkis-dimenzió 63
vektor 10, 15
veszteségfüggvény
 ε -inszenzitív 115
 Huber-féle 131
 négyzetes ε -inszenzitív 131
visszacsatolósos szűrő 145

W

weight 15
 synaptic 15
Widrow–Hoff-algoritmus 29
Wiener-szűrő 160
Woodbury-egyenlőség 50, 171